REPORT 47

ARGESIM

ASIM SST 2024

Tagungsband Langbeiträge

ASIM SST 2024

27. ASIM Symposium Simulationstechnik

Universität der Bundeswehr München

der Bundeswehr Universität München

4.9. – 6.9.2024

Herausgegeben von Oliver Rose und Tobias Uhlig

ISBN ebook 978-3-903347-65-6
ARGESIM Report 47
www.argesim.org

DOI 10.11128/arep.47
ASIM Mitteilung 190
www.asim-gi.org

ASIM SST 2024

Tagungsband Langbeiträge

27. ASIM Symposium Simulationstechnik

4.9.-6.9.2024

Universität der Bundeswehr München

Herausgegeben von
Oliver Rose, Tobias Uhlig, Univ. der Bundeswehr München

ISBN ebook ISBN 978-3-903347-65-6 DOI 10.11128/arep.47
ARGESIM Report 47 ASIM Mitteilung 190
ARGESIM Verlag, Wien, 2024
www.asim-gi.org ***www.argesim.org***

Bibliographic Data:

Publisher: ARGESIM Publisher, Vienna
Title: Tagungsband Langbeiträge ASIM SST 2024
Subtitle 27. ASIM Symposium Simulationstechnik, 4.9.-6.9.2024, Universität der Bundeswehr München
Author(s):
Editor(s): Oliver Rose, Tobias Uhlig, Universität der Bundeswehr München
Series: ARGESIM Reports
Series Editors: Felix Breitenecker, Thorsten Pawletta, ASIM
Volume: ARGESIM Report no. 47
ISBN ebook: 978-3-903347-65-6, ARGESIM Verlag
DOI: 10.11128/arep.47
ASIM ID: ASIM Mitteilung 190
Publication Date: September 1, 2024
Number of Pages: vi +234 pages
Cover: © Adobe Stock

Copyright Information / Regulations ARGESIM

ARGESIM is a non-profit scientific society generally aiming for dissemination of information on system simulation - from research via development to applications of system simulation. ARGESIM's primary publication is the journal SNE – Simulation Notes Europe with open access to all contributions; generally, the authors retain the copyright of their contribution.

SNE contributions. This copyright regulation holds also for *ARGESIM Reports* and *ARGESIM Advances in Simulation* publishing conference volumes for ASIM, MATHMOD, and EUROSIM (in consideration of copyright regulations for related conference publications) and monographs on system simulation (mainly PhD theses).

About ARGESIM

ARGESIM is a non-profit society generally aiming for dissemination of information on system simulation from research via development to applications of system simulation. ARGESIM is closely co-operating with EUROSIM, the Federation of European Simulation Societies, and with ASIM, the German Simulation Society. ARGESIM is an 'outsourced' activity from the Mathematical Modelling and Simulation Group of TU Wien, there is also close co-operation with TU Wien (organisationally and personally). *ARGESIM Publisher* organizes publishing activities, with ISBN root 978-3-901608-xx-y and DOI root 10.11128/xx…x.

ARGESIM's activities are:

- Publication of the scientific journal SNE - Simulation Notes Europe (Membership Journal of EUROSIM, the Federation of European Simulation Societies) → *www.sne-journal.org*
- Organisation and Publication of the ARGESIM Benchmarks for Modelling Approaches and Simulation Implementations → *www.argesim.org/benchmarks/*
- Publication of the series ARGESIM Reports (for monographs in system simulation, and proceedings of simulation conferences and workshops) → *www.argesim.org/publications/*
- Publication of the special series FBS Simulation - Advances in Simulation / Fortschrittsberichte Simulation (monographs in co-operation with ASIM, the German Simulation Society
- Organisation of the Conference Series MATHMOD Vienna (triennial, in co-operation with EUROSIM, ASIM, and TU Wien) → *www.mathmod.at*
- Administration and support of ASIM (German Simulation Society → *www.asim-gi.org*) and of EUROSIM (Federation of European Simulation Societies → *www.eurosim.info*)

ARGESIM – Arbeitsgemeinschaft Simulation News – Working Committee Simulation News – SNE Publication
Mommsengasse 19/8, 1040 Vienna, Austria; Tel +43-1-58801-10111, -10115; Fax +43-1-58801-910111
Email: office@argesim.org, office@sne-journal.org; WWW: www.argesim.org, www.sne-journal.org
Incorporated Austrian Society ZVR No 213056164 – EU VAT ID No ATU 72054279
Bank Account: ARGESIM, IBAN AT07 2011 1828 9115 0800, BIC GIBAATWWXXX, ERSTE BANK VIENNA

VORWORT

ASIM lädt zum 27. ASIM Symposium Simulationstechnik, zu einer bereits klassischen ASIM-Tagung. Traditionell findet das **ASIM Symposium Simulationstechnik (SST)** alle zwei Jahre im Herbst statt, alternierend Tagung **ASIM SPL, der** internationale Fachtagung *Simulation in Produktion und Logistik*. Mit **ASIM SST 2024** ist die Tagung nach Verschiebungen und virtuellen Durchführungen in der Corona-Zeit wieder in den planmäßigen ASIM-Tagungskalender zurückgekehrt.

ASIM SST 2024 bietet traditionellerweise zwei Arten von Beiträgen, die alle einem Peer Review Prozess unterzogen wurden und sich in der Publikationsform unterscheiden:

- **ASIM SST 2024 Langbeiträge** (6-10 Seiten, deutsch oder englisch)
  mit individueller DOI für jeden Beitrag e-Tagungsband:
  Tagungsband Langbeiträge ASIM SST 2024, ARGESIM Report 47, ASIM Mitteilung 190, ISBN ebook 978-3-903347-65-6 ARGESIM Verlag
- **ASIM SST 2024 Kurzbeiträge** (2-5 Seiten, deutsch oder englisch), mit Tagungsband-DOI:
  Tagungsband Kurzbeiträge ASIM SST 2024, ARGESIM Report 46, ASIM Mitteilung 189, ISBN ebook 978-3-903347-64-9 ARGESIM Verlag

ASIM SST 2024 wird veranstaltet an der Universität der Bundeswehr München am Campus Neubiberg, Veranstalter sind Oliver Rose und Tobias Uhlig vom Institut für Technische Informatik, unterstützt vom lokalen Organisationskomitee

Das Programmkomitee bestand aus den ASIM-Mitgliedern Felix Breitenecker, Kurt Chudej, Walter Commerell, Olaf Hagendorf, Lukas Hollenstein, Andreas Körner, Christoph Laroque, Xiaobo Liu-Henke, Daniel Lückerath, Thorsten Pawletta, Niki Popper, Oliver Rose, Michael Striebel, Tobias Uhlig, Oliver Ullrich, Sigrid Wenzel, und Jochen Wittmann. Gemeinsam mit dem SNE Review Board wird es Tagungsbeiträge zur Publikation in ASIM's Scientific Journal Simulation Notes Europe vorschlagen.

Zum **ASIM SST 2024** haben sich etwa 60 Fachleute zusammengefunden, zwei Drittel davon ASIM-Mitglieder. *Dieser Tagungsband, Langbeiträge ASIM SST 2024,* publiziert die 29 angenommenen Langbeiträge – mit individueller DOI für jeden Beitrag, und Volume-DOI für den Band. Wie bei jeder ASIM-Tagung ist die Möglichkeit für Postconference Publications in ASIM's Scientific Journal *Simulation Notes Europe* **SNE** vorgesehen

Für Hauptvorträge konnten wir zwei Simulationsfachleute gewinnen.
Hans Ehm, Head of Supply Chain Innovation der Infineon Technologies AG und
Felix Breitenecker, langjähriger ASIM-Sprecher.

ASIM SST 2024 bot auch interessantes Begleitprogramm: Mittwoch, 4.9.2024 gab es einen kleinen Empfang in der Münchener Innenstadt mit anschließenden Themen-Stadtführungen, und Donnerstag, 5.9.2024 eine Besichtigung des Flugsimulators mit anschließendem Abendbankett im Offizierscasino.

Wir Organisatoren möchten allen herzlich danken, die das **ASIM SST 2024** zu einem Erfolg machen: den Teilnehmerinnen und Teilnehmern, den Autoren der Beiträge, dem Programmkomitee für das Reviewing, dem ARGESIM Verlag für die Publikationen, und unseren Tagungsmitarbeitern.

Oliver Rose, Tobias Uhlig, September 2024

Inhaltsverzeichnis – List of Content

27. ASIM Symposium Simulationstechnik - Langbeiträge

Langbeiträge (L) sind zu finden im
ARGESIM Report 47 Tagungsband Langbeiträge ASIM SST 2024
ISBN ebook 978-3-903347-65-6, DOI 10.11128/arep.47, ARGESIM Report 47, ASIM Mitteilung 190

Kurzbeiträge (K) und Abstracts der Hauptvorträge (H) sind zu finden im
ARGESIM Report 46 Tagungsband Kurzbeiträge ASIM SST 2024
ISBN ebook 978-3-903347-64-9, DOI 10.11128/arep.46, ARGESIM Report 46, ASIM Mitteilung 189

AUTORENINDEX – INDEX OF AUTHORS

27. ASIM Symposium Simulationstechnik - Langbeiträge

Langbeiträge (L) sind zu finden im
ARGESIM Report 47 Tagungsband Langbeiträge ASIM SST 2024
ISBN ebook 978-3-903347-65-6, DOI 10.11128/arep.47, ARGESIM Report 47, ASIM Mitteilung 190

Kurzbeiträge (K) und Abstracts der Hauptvorträge (H) sind zu finden im
ARGESIM Report 46 Tagungsband Kurzbeiträge ASIM SST 2024
ISBN ebook 978-3-903347-64-9, DOI 10.11128/arep.46, ARGESIM Report 46, ASIM Mitteilung 189

Mathematical Analysis of a Compartment Model

Aristeidis Niotis[1], Kurt Chudej[1,2*]

[1]Lehrstuhl für Wissenschaftliches Rechnen, Universität Bayreuth,
95440 Bayreuth, Germany

[2]Forschungszentrum Modellierung und Simulation (MODUS), Universität Bayreuth, 95440 Bayreuth, Germany;
kurt.chudej@uni-bayreuth.de

Abstract. We give a theoretical investigation of a Covid model. This model was successfully used in the context of MPC control to keep the Covid pandemic manageable [1]. We compute especially a next generation basic reproduction number, a Lyapunov function and reveal some peculiarities if demography is skipped. Finally, we look at the model with two age groups.

1 Introduction

Epidemics and infectious diseases are often described using so-called compartment models in order to be able to make the best possible, realistic predictions about the course of the disease in larger population groups. A population is divided into different groups (*"compartments"*) and the flow (inflows and outflows) between these compartments is examined more closely. A classic compartment model is the SIR model according to Kermack and McKendrick.

2 ODE model

In [1] a detailed Covid model is developed and successfully used in the context of model predictive control (MPC) in order to mitigate the COVID-19 outbreak. The model was used with weekly updates of the parameters. Optimal mass-testing and age-dependent social distancing policies were determined [1].

Here a theoretical investigation of the mathematical properties of an autonomous version of this model is performed. At the beginning the model is simplified to one age class and enhanced with demography. Therefore the following autonomous mathematical compartment model is investigated in this research:

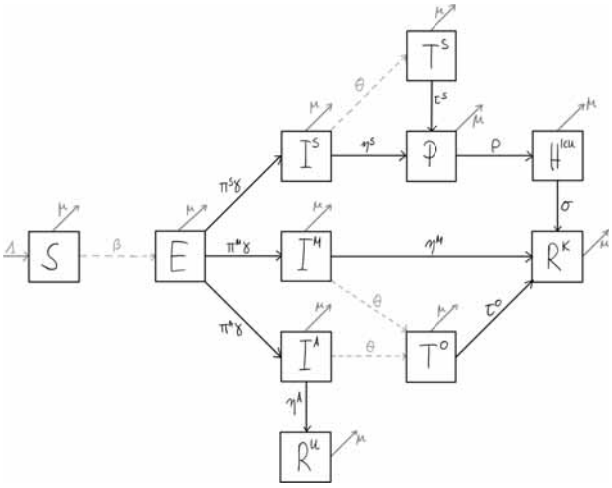


Figure 1: SEIPTHR model with demography

2.1 ODE system

The associated ODE system with *mass action incidence* is given by

$$\dot{S} = \Lambda - \beta S[I^S + I^M + I^A + T^S + T^O] - \mu S \quad (1a)$$
$$\dot{E} = \beta S[I^S + I^M + I^A + T^S + T^O] - (\gamma + \mu)E \quad (1b)$$
$$\dot{I}^S = \pi^S \gamma E - (\eta^S + \theta + \mu)I^S \quad (1c)$$
$$\dot{I}^M = \pi^M \gamma E - (\eta^M + \theta + \mu)I^M \quad (1d)$$
$$\dot{I}^A = \pi^A \gamma E - (\eta^A + \theta + \mu)I^A \quad (1e)$$
$$\dot{T}^S = \theta I^S - (\tau^S + \mu)T^S \quad (1f)$$
$$\dot{T}^O = \theta[I^M + I^A] - (\tau^O + \mu)T^O \quad (1g)$$
$$\dot{P} = \eta^S I^S + \tau^S T^S - (\rho + \mu)P \quad (1h)$$
$$\dot{H}^{ICU} = \rho P - (\sigma + \mu)H^{ICU} \quad (1i)$$
$$\dot{R}^K = \eta^M I^M + \tau^O T^O + \sigma H^{ICU} - \mu R^K \quad (1j)$$
$$\dot{R}^U = \eta^A I^A - \mu R^U \quad (1k)$$

together with non-negative initial values

$$S(0)=S_0, E(0)=E_0, I^S(0)=I_0^S, I^M(0)=I_0^M,$$
$$I^A(0)=I_0^A, T^S(0)=T_0^S, T^O(0)=T_0^O, P(0)=P_0,$$
$$H^{ICU}(0)=H_0^{ICU}, R^K(0)=R_0^K, R^U(0)=R_0^U. \quad (11)$$

2.2 Model description

The following compartments are introduced in [1]:

- $S(t), E(t)$: Susceptible persons (S) accumulate after infection with the pathogen in a compartment E (exposed), but are not yet infectious themselves.
- $I(t) = I^S(t) + I^M(t) + I^A(t)$: The infectious compartment I is divided into three classes depending on the course of infection. A distinction is made between severe cases I^S, mild cases I^M and asymptomatic cases I^A.
- $T^S(t), T^O(t)$: Infectious persons have the opportunity to be tested, in which we in turn differentiate according to the course of the disease.
- $P(t)$: Seriously ill people will either go directly to a physician (P) and go into quarantine or only after receiving a positive test result.
- $H^{ICU}(t)$: After isolation in P, severely ill persons are transferred to an intensive care unit H^{ICU}.
- $R(t) = R^K(t) + R^U(t)$: The compartment of recovered individuals R is divided into two classes depending on the course of infection: individuals who have actually been identified as infected are collected in R^K (known), the other ones recover in a natural way without having previously been identified as diseased (R^U - unknown).

In order to finally understand the interaction between the individual compartments, we need an overview of the parameters modelled in Table 1.
Set $\pi^S + \pi^M + \pi^A = 1$ and $U := S + E + I^S + I^M + I^A + R^U$. All constants in Table 1 are non-negative.

Parameters	Description
$\beta > 0$	infection rate
$\gamma > 0$	average incubation time γ^{-1} in days
$\pi^S > 0$	Proportion of severely ill patients
$\pi^M > 0$	Proportion of mildly ill patients
$\pi^A > 0$	Proportion of asymptomatic patients
$\theta \geq 0$	Test rate (Tests spread in U per day)
$\eta^S > 0$	Recovery rate for severe course
$\eta^M > 0$	Recovery rate for mild course
$\eta^A > 0$	Recovery rate for asymptomatic course
$\tau^S > 0$	Rate at which tested persons recover (severe)
$\tau^O > 0$	Rate at which tested persons recover (others)
$\rho > 0$	average duration of isolation ρ^{-1}
$\sigma > 0$	average length of stay in ICU σ^{-1}

Table 1: Model parameters

3 Mathematical analysis

We decompose all compartments into the vector $x = (E, I^S, I^M, I^A, T^S, T^O)$ of infected compartments and the remaining compartments $y = (S, P, H^{ICU}, R^K, R^U)$. All compartments are denoted by $z = (x, y)$.

3.1 Positive invariance and existence of solutions

Starting from our IVP (1), it is important from a real point of view that the biological relevance of the solutions is ensured. This requirement on the mathematical model is fulfilled by the *positive invariance* shown below:

Theorem: The non-negative orthant $\mathbb{R}^{11}_{\geq 0}$ is a positive invariant set.
Proof: The r.h.s. of the ODE is quasipositive. Apply Thm 4.2.2 in Prüss, Wilke [2], pp. 83–84.

The total population $N(t) := \sum_{i=1}^{11} z_i(t)$ fulfills the initial value problem

$$\dot{N} = \Lambda - \mu N,\ N(0) = N_0,$$

where $N_0 := S_0 + E_0 + I_0^S + I_0^M + I_0^A + T_0^S + T_0^O + P_0 + H_0^{ICU} + R_0^K + R_0^U$.
Its solution $N(t) = \Lambda/\mu + e^{-\mu t}(N_0 - \Lambda/\mu)$ converges monotonically with $t \to \infty$ to $N_\infty := \Lambda/\mu$.
Therefore $N(t) \leq K := \max(N_0, N_\infty)$.

This proves, that the polytop $\Omega := \{z \in \mathbb{R}^{11}_{\geq 0} | N(t) \leq K\}$

is a positive invariant set of the IVP (1).
The next step is to ensure that there is indeed a solution of the IVP:

Theorem: The initial value problem (1) has a unique solution for $t \in [0,\infty[$.
Proof: On the compact set Ω the IVP is globally Lipschitz continuous, due to the polynomial r.h.s. of the ODE. Apply Picard-Lindelöf.

Without loss of generality we set $\Lambda = N_0 \cdot \mu = \mu$ for $N_0 = 1$, this implies $N_\infty = 1$ and $K = 1$. This choice means that the total population at all times is constant equal to 1 and therefore fractions of the population are considered.

3.2 Next-Generation-Approach for $\mathscr{R}_0$

Our next goal is to calculate a next-generation basic reproduction number according to [3]. For that we need the existence of a disease-free equilibrium of our ODE, which can be easily seen:
If $\mu > 0$, then there exists a unique disease-free equilibrium $\mathscr{E}_{DFE}$ with compartiment $S = 1 := N$ and all other compartments equal to zero.

Using the decomposition of the compartments the initial value problem

$$\dot{z} = h(z),\ z(0) = z_0$$

is rewritten as

$$\begin{aligned} \dot{x} &= f(x,y) = \mathscr{F}(x,y) - \mathscr{V}(x,y),\ x(0) = x_0, \\ \dot{y} &= g(x,y),\ y(0) = y_0 \end{aligned}$$

with

$$\begin{aligned} \mathscr{F}(x,y) &= \begin{pmatrix} \beta S[I^S + I^M + I^A + T^S + T^O] \\ \mathbf{0} \end{pmatrix} \\ \mathscr{V}(x,y) &= \begin{pmatrix} (\gamma+\mu)E \\ (\eta^S+\theta+\mu)I^S - \pi^S\gamma E \\ (\eta^M+\theta+\mu)I^M - \pi^M\gamma E \\ (\eta^A+\theta+\mu)I^A - \pi^A\gamma E \\ (\tau^S+\mu)T^S - \theta I^S \\ (\tau^O+\mu)T^O - \theta[I^M + I^A] \end{pmatrix}. \end{aligned}$$

Let $F = \frac{\partial \mathscr{F}}{\partial x}$ and $V = \frac{\partial \mathscr{V}}{\partial x}$. We will postpone the evaluation of the Jacobian matrices at the coordinates of the DFE to a later time.

The Jacobian matrices are given in this application by

$$\begin{aligned} F &= \frac{\partial \mathscr{F}}{\partial x} = \left(\begin{array}{cccccc} 0 & \beta S & \beta S & \beta S & \beta S & \beta S \\ \hline & & & \mathbf{0} & & \end{array}\right) \\ V &= \frac{\partial \mathscr{V}}{\partial x} = \begin{pmatrix} r_1 & 0 & 0 & 0 & 0 & 0 \\ -\pi^S\gamma & r_2 & 0 & 0 & 0 & 0 \\ -\pi^M\gamma & 0 & r_3 & 0 & 0 & 0 \\ -\pi^A\gamma & 0 & 0 & r_4 & 0 & 0 \\ 0 & -\theta & 0 & 0 & r_5 & 0 \\ 0 & 0 & -\theta & -\theta & 0 & r_6 \end{pmatrix} \end{aligned}$$

with $r_1 = \gamma+\mu,\ r_2 = \eta^S+\theta+\mu, r_3 = \eta^M+\theta+\mu,$
$r_4 = \eta^A+\theta+\mu, r_5 = \tau^S+\mu, r_6 = \tau^O+\mu.$

For the next-generation matrix we need

$$V^{-1} = \begin{pmatrix} r_1^{-1} & 0 & 0 & 0 & 0 & 0 \\ m_1 & r_2^{-1} & 0 & 0 & 0 & 0 \\ m_2 & 0 & r_3^{-1} & 0 & 0 & 0 \\ m_3 & 0 & 0 & r_4^{-1} & 0 & 0 \\ m_4 & m_6 & 0 & 0 & r_5^{-1} & 0 \\ m_5 & 0 & m_7 & m_8 & 0 & r_6^{-1} \end{pmatrix}$$

where $m_1 = \frac{\gamma}{\gamma+\mu}\frac{\pi^S}{\eta^S+\theta+\mu}, m_2 = \frac{\gamma}{\gamma+\mu}\frac{\pi^M}{\eta^M+\theta+\mu},$
$m_3 = \frac{\gamma}{\gamma+\mu}\frac{\pi^A}{\eta^A+\theta+\mu}, m_4 = \frac{\gamma}{\gamma+\mu}\frac{\pi^S}{\eta^S+\theta+\mu}\frac{\theta}{\tau^S+\mu},$
$m_5 = \frac{\gamma}{\gamma+\mu}\left(\frac{\pi^M}{\eta^M+\theta+\mu}\frac{\theta}{\tau^O+\mu} + \frac{\pi^A}{\eta^A+\theta+\mu}\frac{\theta}{\tau^O+\mu}\right),$
$m_6 = \frac{\theta}{(\tau^S+\mu)(\eta^S+\theta+\mu)}, m_7 = \frac{\theta}{(\tau^O+\mu)(\eta^M+\theta+\mu)},$
$m_8 = \frac{\theta}{(\tau^O+\mu)(\eta^A+\theta+\mu)}.$

The next-generation-matrix is computed as

$$K = FV^{-1} = \beta S \begin{pmatrix} K_1 & 0 & 0 & 0 & 0 & 0 \\ * & & & \mathbf{0} & & \end{pmatrix}^T$$

with

$$\begin{aligned} K_1 = \frac{\gamma}{\gamma+\mu} \cdot \Big(& \frac{\pi^S}{\eta^S+\theta+\mu} + \frac{\pi^M}{\eta^M+\theta+\mu} \\ & + \frac{\pi^A}{\eta^A+\theta+\mu} + \frac{\pi^S}{\eta^S+\theta+\mu}\frac{\theta}{\tau^S+\mu} + \\ & + \frac{\pi^M}{\eta^M+\theta+\mu}\frac{\theta}{\tau^O+\mu} + \frac{\pi^A}{\eta^A+\theta+\mu}\frac{\theta}{\tau^O+\mu}\Big). \end{aligned}$$

The spectral radius of K is given by $\rho(K) = \beta S \cdot K_1$, since K is a triangle matrix.
Finally the next generation basic reproduction number

is given by $\mathscr{R}_0 = \rho(K)|_{DFE} = \beta N \cdot K_1 \overset{N=1}{=} \beta \cdot K_1$.

3.3 Stability

In order to analyze the stability of our DFE $(0, y_0)$, it's helpful to write the partitioned linearized ODE as

$$\dot{x} = J_1|_{DFE} \cdot x + J_2|_{DFE} \cdot (y - y_0)$$
$$\dot{y} = J_3|_{DFE} \cdot x + J_4|_{DFE} \cdot (y - y_0).$$

The relevant Jacobian matrices are given by $J_1 = \frac{\partial f}{\partial x}$ and $J_4 = \frac{\partial g}{\partial y}$, an easy computation shows $J_2|_{DFE} = \frac{\partial f}{\partial y}|_{DFE} = 0$.

Note that $F \geq 0$ (componentwise) and V is a regular M-matrix, since V^{-1} exists and $V^{-1} \geq 0$ (componentwise). Therefore $-J_1|_{DFE} = (V - F)|_{DFE}$ is a regular splitting due to Varga [4] p. 95, Def. 3.28.

Additionally we know [3, 4]:
If $\mathscr{R}_0 < 1$, then $J_1|_{DFE}$ has only eigenvalues with negative real part.
If $\mathscr{R}_0 > 1$, then $J_1|_{DFE}$ has at least one eigenvalue with positive real part.
The matrix $J_4|_{DFE}$ has eigenvalues with negative real part, a simple calculation gives the eigenvalues $-\mu, -(\rho + \mu)$ and $-(\sigma + \mu)$.

Applyication of the linearization theorem gives the result a). A more complicated argument is required in order to prove b).

Theorem: The considered Covid model has the following properties:

a) If $\mathscr{R}_0 < 1$, then the DFE is locally asymptotically stable.

b) If $\mathscr{R}_0 > 1$, then the DFE is instable.

Following an idea of Shuai/van den Driessche [5] we compute a linear convex Lyapunov function for the case with $\mathscr{R}_0 < 1$ in order to analyze the global stability of our DFE.
According to [5], we have to analyze first when $\tilde{f}(x,y) = (F - V)|_{DFE} \cdot x - \mathscr{F}(x,y) + \mathscr{V}(x,y) \geq 0$ on a positive invariant set. For our model only the first component of $\tilde{f}$ is not equal to zero, more precisely $\tilde{f}_1(x,y) = \beta \cdot (N - S) \cdot [I^S + I^M + I^A + T^S + T^O]$. Thus our positive invariant set is $\bar{\Omega} = \{z \in \mathbb{R}^{11}_{\geq 0} | N(t) \leq N_\infty\} \subset \Omega$.

Next we need a left eigenvector ω^T of the non-negative matrix $V^{-1}F$ to the eigenvalue $\mathscr{R}_0 \geq 0$.
This yields $\omega^T = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$, where the E-component takes the value 0. Thus, the linear convex Lyapunov function on $\bar{\Omega}$ is given by

$$Q(x) = \omega^T V^{-1} x = \begin{pmatrix} K_1 & q_1 & q_2 & q_3 & q_4 & q_5 \end{pmatrix} \cdot x,$$

with $q_1 = \frac{1}{\eta^S + \theta + \mu} + \frac{\theta}{(\tau^S + \mu)(\eta^S + \theta + \mu)}, q_2 = \frac{1}{\eta^M + \theta + \mu} + \frac{\theta}{(\tau^O + \mu)(\eta^M + \theta + \mu)}, q_3 = \frac{1}{\eta^A + \theta + \mu} + \frac{\theta}{(\tau^O + \mu)(\eta^A + \theta + \mu)}, q_4 = \frac{1}{\tau^S + \mu}, q_5 = \frac{1}{\tau^O + \mu}$.
One can show, that the DFE is the largest invariant set in the set given by $Q(x) = 0$. Then apply LaSalle [6].

Theorem: The considered Covid model has the following properties:
If $\mathscr{R}_0 < 1$, then the DFE is globally asymptotically stable in $\bar{\Omega}$.

3.4 Numerical solutions

Finally, we will consider numerical solutions for the ODE system. Two scenarios are investigated: with testing and without testing.
Consider the following IVP:

$$E_0 = 0.01, \;\; S_0 = 1 - E_0 = 0.99, \;\; I_0^S = ... = R_0^U = 0$$

The following parameter selection for the age group of 15- to 60-year-olds was taken from [1]:

Parameter	Value
β	0.63
γ	0.19
π^S	$\frac{0.31}{100}$
π^M	$\frac{22.01}{100}$
π^A	$\frac{77.68}{100}$
η^S	0.25
η^M	0.25
η^A	0.17
τ^S	0.75
τ^O	0.92
ρ^{-1}	10.98
σ^{-1}	10.5

Table 2: Parameters for the simulation

We set $\mu = \frac{1}{365 \cdot 20}$. Unfortunaly, problems occur with the numerical solutions, because the solution curves leave the positive invariant set Ω for a very large t value (see Figure 2).

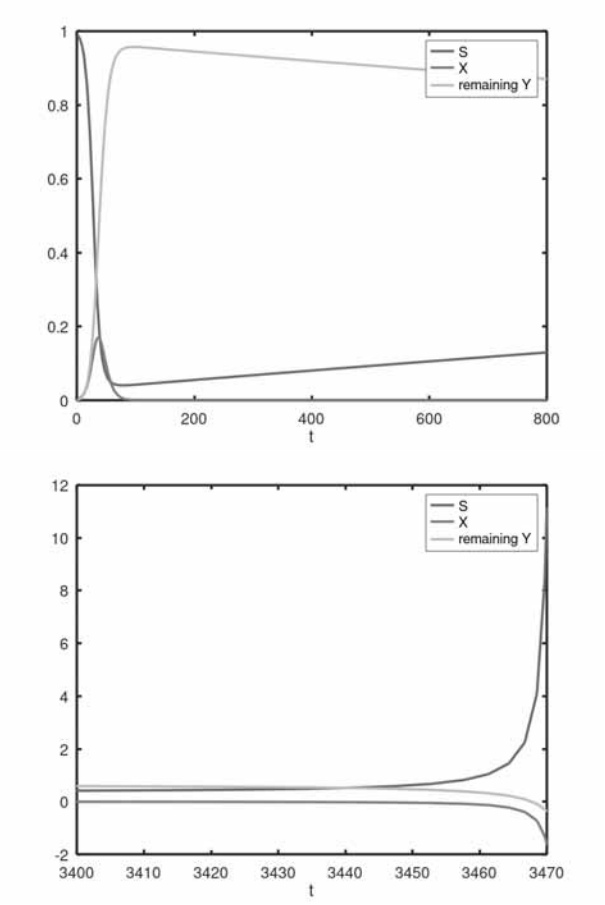


Figure 2: Simulation with demography, without testing

An implicit stiff integrator can fix this problem (see Figure 3).

For the scenario without testing $\mathscr{R}_0 \approx 3.44$ holds.
Finally, consider test measures with $\theta = 0.4$. The numeric solutions can bee seen in Figure 4.
For this scenario we obtain a basic reproduction number of $\mathscr{R}_0 \approx 1.54$.

4 Model without demography

We investigate now our model without demography, we set $\mu = \Lambda = 0$.
In contrast to the first model, this time there are pe-

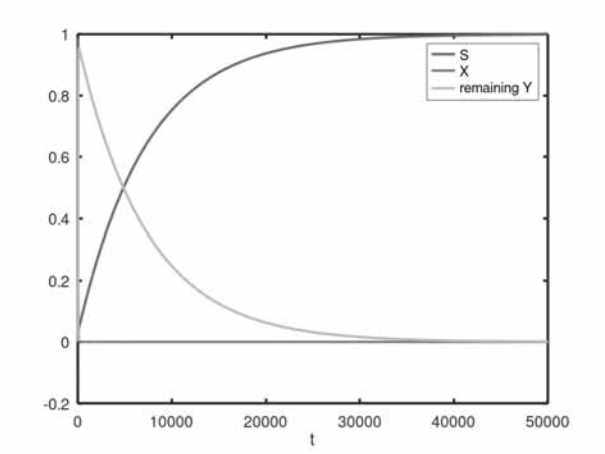


Figure 3: Simulation with demography, without testing, using an implicit stiff integrator

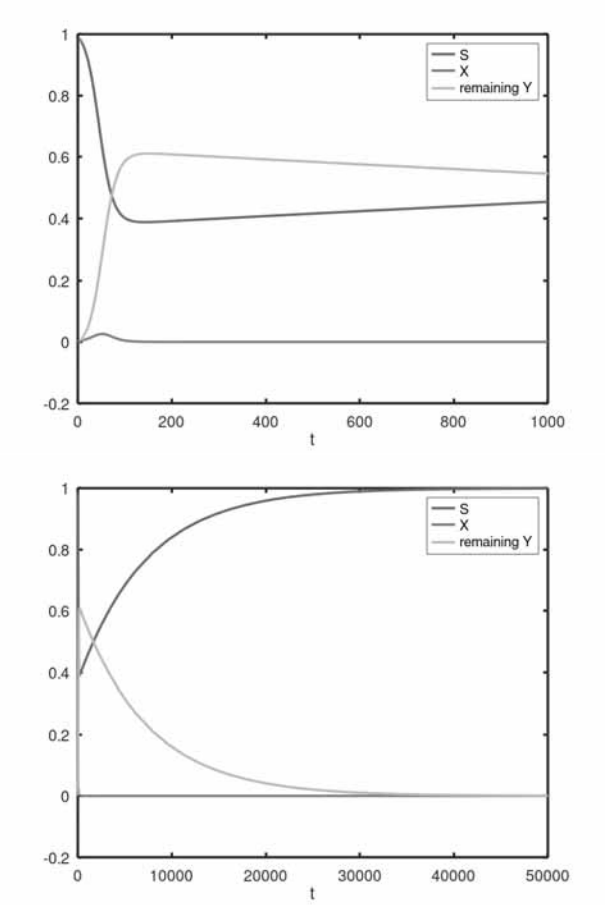


Figure 4: Simulation with demography, with testing, $\theta = 0.4$, using an implicit stiff integrator

culiarities in the calculations for $\mathscr{R}_0$. Firstly, by analogy with the previous considerations, it can be shown that the non-negative orthant $\mathbb{R}^{11}_{\geq 0}$ and the set $\tilde{\Omega} := \{z \in \mathbb{R}^{11}_{\geq 0} \mid \sum_{i=1}^{11} z_i = 1\}$ are positive invariant for our second ODE.
Next, we need to ensure the existence of a DFE: the difference to the first model is that this equilibrium point is *no longer unique*. Simple calculations together with the positive invariance provide that all disease-free-equlibria fulfill the equation $\tilde{S} + \tilde{R^K} + \tilde{R^U} = 1$.
Therefore, there will be no unique basic reproduction number. The calculation of $\mathscr{R}_0$ is similar to that of the first model: We get the same formulas as before, only with $\Lambda = \mu = 0$.

We obtain the next-generation-matrix

$$K = FV^{-1} = \beta S \begin{pmatrix} K_2 & 0 & 0 & 0 & 0 & 0 \\ * & & & \mathbf{0} & & \end{pmatrix}^T$$

with

$$K_2 = \frac{\pi^S}{\eta^S + \theta} + \frac{\pi^M}{\eta^M + \theta} + \frac{\pi^A}{\eta^A + \theta} + \frac{\pi^S}{\eta^S + \theta}\frac{\theta}{\tau^S} + \frac{\pi^M}{\eta^M + \theta}\frac{\theta}{\tau^O} + \frac{\pi^A}{\eta^A + \theta}\frac{\theta}{\tau^O}.$$

The spectral radius of K is given by $\rho(K) = \beta S \cdot K_2$.

Finally, the next generation basic reproduction number is given by $\mathscr{R}_0 = \rho(K)|_{DFE} = \beta \tilde{S} \cdot K_2$.

Depending on the initial values of the unknown DFE-components, we calculate the limit value of S and choose it as $\tilde{S}$.

Another observation is that for $\tilde{S} = 1 = N$ and $\mu \to 0$ the basic reproduction numbers of both models coincide, which can be seen approximately for $\mu = \frac{1}{365 \cdot 20}$ in the following table:

β	$\mathscr{R}_0^{\text{with}}$	$\mathscr{R}_0^{\text{without}}$
0.1	0.545411144346408	0.546221176470588
0.4	2.18164457738563	2.18488470588235
0.8	4.36328915477126	4.36976941176471

Table 3: Basic reproduction numbers with/without demography, without testing

In contrast to the inclusion of demography in the first model, the linearization theorem cannot be used to investigate the local asymptotic stability of the DFE, since the corresponding matrix $J_4|_{DFE}$ has the eigenvalue 0. Instead, we can again calculate a linear convex Lyapunov function for the case with $\mathscr{R}_0 < 1$ of our DFE.
Again we have to analyze when the function $\tilde{f}(x,y) = (F-V)|_{DFE} \cdot x - \mathscr{F}(x,y) + \mathscr{V}(x,y) \geq 0$. For our second model only the first component of $\tilde{f}$ is not equal to zero, more precisely $\tilde{f}_1(x,y) = \beta \cdot (\tilde{S} - S) \cdot [I^S + I^M + I^A + T^S + T^O]$. Thus, we can consider a Lyapunov function on our positive invariant set $\tilde{\Omega}$ only for $S_0 \leq \tilde{S} \leq N_\infty = 1$, since the S-component is monotonically decreasing.

Once again we get $\omega^T = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$, where the E-component takes the value 0. Thus, the linear convex Lyapunov function is given by

$$Q(x) = \omega^T V^{-1} x = \begin{pmatrix} K_2 & q_1 & q_2 & q_3 & q_4 & q_5 \end{pmatrix} \cdot x,$$

where q_1, q_2, q_3, q_4, q_5 are given by the same formulas as before with $\mu = 0$.

Finally, we will consider again a numerical solution with the same parameter selection as in the previous simulations. We implement no testing ($\theta = 0$) and set $\tilde{S} = \lim_{t \to \infty} S(t) \approx 0.0358718761871864$. The numeric solutions can bee seen in Figure 5:

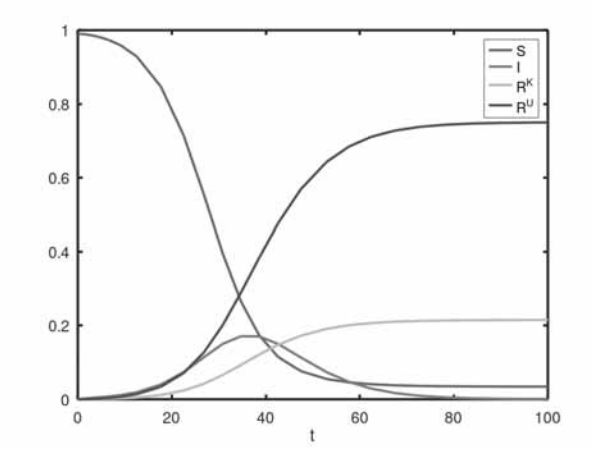


Figure 5: Simulation without demography, without testing

For this scenario $\mathscr{R}_0 \approx 0.123$ holds.

5 Model without demography for two age-groups

Finally, we examine the second model again, which we now expand to include another age group. The respective new infections now no longer depend only on the infected persons of the own age group, but also on those in the other age group, since contacts between population groups are taken into account.

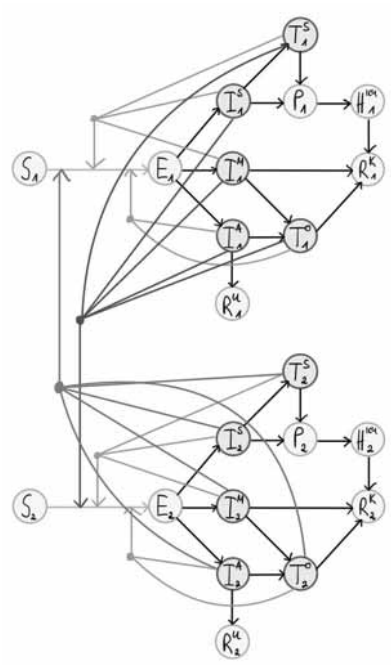


Figure 6: Model with two age groups, without demography

In Figure 6 we see the mutual influence of the two age groups: The blue and purple arrows describe the influence on the new infections of the other group (orange). The grey arrows indicate the previous influence on the new infections of the own age group.
This extension of the model also changes the differential equation system, which doubles due to the additional compartments of the other age group. The ODE structure is the same for both groups:
We obtain for $i = 1,2$ the ODE system

$$\dot{S}_i = -\sum_{j=1}^{2} \beta_{ij} S_i [I_j^S + I_j^M + I_j^A + T_j^S + T_j^O] \tag{2a}$$

$$\dot{E}_i = \sum_{j=1}^{2} \beta_{ij} S_i [I_j^S + I_j^M + I_j^A + T_j^S + T_j^O] - \gamma E_i \tag{2b}$$

$$\dot{I}_i^S = \pi_i^S \gamma E_i - (\eta^S + \theta_i) I_i^S \tag{2c}$$

$$\dot{I}_i^M = \pi_i^M \gamma E_i - (\eta^M + \theta_i) I_i^M \tag{2d}$$

$$\dot{I}_i^A = \pi_i^A \gamma E_i - (\eta^A + \theta_i) I_i^A \tag{2e}$$

$$\dot{T}_i^S = \theta_i I_i^S - \tau^S T_i^S \tag{2f}$$

$$\dot{T}_i^O = \theta_i [I_i^M + I_i^A] - \tau^O T_i^O \tag{2g}$$

$$\dot{P}_i = \eta^S I_i^S + \tau^S T_i^S - \rho P_i \tag{2h}$$

$$\dot{H}_i^{ICU} = \rho P_i - \sigma H_i^{ICU} \tag{2i}$$

$$\dot{R}_i^K = \eta^M I_i^M + \tau^O T_i^O + \sigma H_i^{ICU} \tag{2j}$$

$$\dot{R}_i^U = \eta^A I_i^A \tag{2k}$$

together with non-negative initial values.
The calculation of the basic reproduction number now becomes a bit more complicated, the structure of the matrices changes a little compared to the second model. As before, due to the missing demography, we have the peculiarity of the non-unique DFE, so again we have to select an arbitrary equilibrium point with $\tilde{S}_1 + \tilde{S}_2 + \tilde{R}_1^K + \tilde{R}_2^K + \tilde{R}_1^U + \tilde{R}_2^U = 1$. In the following calculations, the corresponding placeholders are used again without presenting a detailed calculation option for the missing components as before.
We decompose the x-ODE using

$$\mathscr{F}(x,y) = \begin{pmatrix} \sum_{j=1}^{2} \beta_{1j} S_1 [I_j^S + I_j^M + I_j^A + T_j^S + T_j^O] \\ \sum_{j=1}^{2} \beta_{2j} S_2 [I_j^S + I_j^M + I_j^A + T_j^S + T_j^O] \\ \hline \mathbf{0} \end{pmatrix}$$

and

$$\mathscr{V}(x,y) = \begin{pmatrix} \gamma E_1 \\ \gamma E_2 \\ (\eta^S + \theta_1) I_1^S - \pi_1^S \gamma E_1 \\ (\eta^S + \theta_2) I_2^S - \pi_2^S \gamma E_2 \\ (\eta^M + \theta_1) I_1^M - \pi_1^M \gamma E_1 \\ (\eta^M + \theta_2) I_2^M - \pi_2^M \gamma E_2 \\ (\eta^A + \theta_1) I_1^A - \pi_1^A \gamma E_1 \\ (\eta^A + \theta_2) I_2^A - \pi_2^A \gamma E_2 \\ \tau^S T_1^S - \theta_1 I_1^S \\ \tau^S T_2^S - \theta_2 I_2^S \\ \tau^O T_1^O - \theta_1 (I_1^M + I_1^A) \\ \tau^O T_2^O - \theta_2 (I_2^M + I_2^A) \end{pmatrix}$$

Both functions have values in $\mathbb{R}^{12}$ and their entry structure is identical to that of the base model.
The big Jacobian matrices are not included, since noth-

ing fundamentally changed in the structure.

The next-generation matrix does not require all entries for the spectral radius, it has the following structure:

$$K = FV^{-1} = \left(\begin{array}{cc|c} \mathscr{R}_{11}^{(S_1,\pi_1)} & \mathscr{R}_{12}^{(S_2,\pi_1)} & \\ \mathscr{R}_{12}^{(S_1,\pi_2)} & \mathscr{R}_{22}^{(S_2,\pi_2)} & \mathbf{0} \\ * & * & \end{array}\right),$$

where

$$\mathscr{R}_{ij}^{(S_k,\pi_l)} = \beta_{ij}\tilde{S}_k \cdot \left(\frac{\pi_l^S}{\eta^S+\theta_l} + \frac{\pi_l^M}{\eta^M+\theta_l} + \frac{\pi_l^A}{\eta^A+\theta_l} + \frac{\pi_l^S}{\eta^S+\theta_l}\frac{\theta_l}{\tau^S} + \frac{\pi_l^M}{\eta^M+\theta_l}\frac{\theta_l}{\tau^O} + \frac{\pi_l^A}{\eta^A+\theta_l}\frac{\theta_l}{\tau^O} \right).$$

Next, K is a 12×12 matrix. For the spectral radius one needs the eigenvalues of K, which in this case must be calculated using the characteristic polynomial. The Laplace expansion for determinants provides:

$$\det(K-\lambda I) = \lambda^{10} \cdot \left[\lambda^2 - \lambda\left(\mathscr{R}_{11}^{(S_1,\pi_1)} + \mathscr{R}_{22}^{(S_2,\pi_2)}\right) + \left(\mathscr{R}_{11}^{(S_1,\pi_1)}\mathscr{R}_{22}^{(S_2,\pi_2)} - \mathscr{R}_{12}^{(S_2,\pi_1)}\mathscr{R}_{12}^{(S_1,\pi_2)}\right)\right]$$

Thus, K has ten times the eigenvalue 0 and the two eigenvalues

$$\frac{\mathscr{R}_{11}^{(S_1,\pi_1)} + \mathscr{R}_{22}^{(S_2,\pi_2)} \pm \sqrt{A}}{2},$$

where

$$A = \left(\mathscr{R}_{11}^{(S_1,\pi_1)} + \mathscr{R}_{22}^{(S_2,\pi_2)}\right)^2 - 4\left(\mathscr{R}_{11}^{(S_1,\pi 1)}\mathscr{R}_{22}^{(S_2,\pi_2)} - \mathscr{R}_{12}^{(S_2,\pi_1)}\mathscr{R}_{12}^{(S_1,\pi_2)}\right).$$

Therefore the basic reproduction number is defined by

$$\mathscr{R}_0 := \frac{\mathscr{R}_{11}^{(S_1,\pi_1)} + \mathscr{R}_{22}^{(S_2,\pi_2)} + \sqrt{A}}{2}.$$

If we start again from just one age group, then the following applies:

$$R := \mathscr{R}_{11}^{(S_1,\pi_1)} = \mathscr{R}_{22}^{(S_2,\pi_2)}, \quad 0 = \mathscr{R}_{12}^{(S_2,\pi_1)} = \mathscr{R}_{12}^{(S_1,\pi_2)}$$

and thus the basic reproduction number simplifies to

$$\mathscr{R}_0 = R,$$

which corresponds to the original $\mathscr{R}_0$ of the second model.
Finally, we see that for several age groups our $\mathscr{R}_0$ depends on the contacts within and between the individual groups, which would have made intuitive sense even without the calculations.

6 Conclusion

Enhancing the compartment model with demography simplifies the computation of the basic reproduction number. In this setting the disease free equilibria is unique. In the model predictive control setting of the original approach in [1] demography can be skipped, the solutions of the initial-value problem are only needed for one week, and demography changes the solutions only slightly. The advice is to skip demography in the model predictive control application if the time horizon is short but to use demography for the computation of the basic reproduction number and the Lyapunov function.

References

[1] Grundel S, Heyder S, Hotz T, Ritschel TKS, Sauerteig P, Worthmann K. How Much Testing and Social Distancing is Required to Control COVID-19? Some Insight Based on an Age-Differentiated Compartmental Model. *SIAM Journal on Control and Optimization*. 2022; 60(2): S145–S169. doi: 10.1137/20M1377783.

[2] Prüss JW, Wilke M. *Gewöhnliche Differentialgleichungen und dynamische Systeme*. 2nd Edition. Cham: Birkhäuser; 2019.

[3] van den Driessche P, Watmough J. Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission. *Mathematical Biosciences*. 2002; 180(1-2): 29–48. doi: 10.1016/s0025-5564(02)00108-6.

[4] Varga RS. *Matrix Iterative Analysis*. 2nd Edition. Heidelberg: Springer; 2000.

[5] Shuai Z, van den Driessche P. Global Stability of Infectious Disease Models Using Lyapunov Functions. *SIAM Journal on Applied Mathematics*. 2013; 73(4): 1513–1532. doi: 10.1137/120876642.

[6] LaSalle J. Some Extensions of Liapunov's Second Method. *IRE Transactions on Circuit Theory.* 1960; 7(4):520–527.

Hopfbifurkation und Grenzzyklus bei einem Modell einer mückenübertragenen Krankheit mit temporärer Kreuzimmunität

Kurt Chudej[1,2*], Jann Rohlfs[1]

[1]Lehrstuhl für Wissenschaftliches Rechnen, Universität Bayreuth,
95440 Bayreuth, Germany; *kurt.chudej@uni-bayreuth.de*

[2]Forschungszentrum Modellierung und Simulation (MODUS), Universität Bayreuth, 95440 Bayreuth, Germany

Abstract. Die neue detailierte Untersuchung eines mathematischen zwei Serotyp Kompartimentmodells einer mückenübertragenen Krankheit mit temporärer Kreuzimmunität zeigt eine Hopf-Bifurkation mit einem Grenzzyklus der Lösung, abhängig von der Dauer der temporären Kreuzimmunität.

Einleitung

Aktuell breiten sich in Europa Tigermücken aus, eine invasive Mückenart die das Potential hat, Dengue-Fieber und Chikungunyafieber weiterzugeben. Dengue Fieber tritt in vier Serotypen auf. Die Verteilung der Serotypen ist regional verschieden. Bei Dengue Fieber ist die Zweitinfektion mit einem neuen Serotyp potentiell der gefährlichere Fall, der schwerwiegende gesundheitliche Probleme mit Todesfolge auslösen kann [1]. Deshalb ist die Möglichkeit der Vorhersage der Größe und Art der Zweitinfektionen von besonderem Interesse.

Wir beschränken uns auf die detailierte Untersuchung eines mathematischen Kompartimentmodells und klammern die Diskussion der praktischen Übertragbarkeit auf die Realität aus.

Viele Dengue Fieber Modelle, wie auch hier, benutzen eine Modellierung der Mücken, die auch aquatische Mücken einschließt [5, 12, 2, 7, 8]. Zwei-Serotyp Dengue Fieber Modelle ohne explizites Kompartiment für die temporäre Kreuzimmunität finden sich z.B. in [6, 2, 15, 8, 14]. Zwei-Serotyp Dengue Fieber Modelle mit explizitem Kompartiment für die temporäre Kreuzimmunität sind seltener untersucht, siehe [7, 8, 9].

1 Kompartimentmodell

Wir betrachten ein Kompartimentmodell (Abbildung 1) für eine mückenübertragene Krankheit mit zwei Serotypen und temporärer Kreuzimmunität. Alle Gewichte an den Pfeilen in Abbildung 1 haben die Dimension 1/Tag und beschreiben die Übergangsraten bzw. Zuwachsraten zwischen den Kompartimenten.

Im Inneren der Abbildung 1 befindet sich das Mückenmodell mit den aquatischen Mücken A_m mit Geburten, den gesunden erwachsenen Mücken S_m und den mit Serotyp i der Krankheit infizierten und infizierenden erwachsenen Mücken I_m^i. Die Infektion der vorher gesunden erwachsenen Mücke geschieht durch den Stich eines mit Serotyp i der Krankheit infizierenden Menschen. Infizierte Mücken genesen nicht mehr.

Wir beschreiben jetzt den oberen Ast des menschlichen Kompartimentmodells: gesunde infizierbare Menschen S mit Geburten, durch einen Stich einer infizierenden Mücke mit Serotyp 1 infizierte und infizierende Menschen I^1, wieder gesundete Menschen in R^1. Die Menschen in den folgenden Kompartimenten R^1, S^2, I^{12}, R des Astes sind immun gegen die früher (im Kompartiment I^1) durchgemachte Erkrankung mit dem Serotyp 1 der Krankheit. Das entscheidende neue Kompartiment R^1 sichert die temporäre Kreuzimmunität der Menschen gegen den anderen Serotyp 2 der Krankheit. Die mittlere Aufenthaltsdauer im Kompartiment R^1 dauert ζ^{-1} Tage. Danach wandern die Menschen in das Kompartiment S^2, sie sind jetzt empfänglich für eine Infektion mit Serotyp 2 der Krankheit durch einen Stich einer mit Serotyp 2 der Krankheit infizierenden Mücke. Im Kompartiment I^{12} befinden sich die infizierten und infizierenden Menschen mit Zweitinfektion vom Serotyp 2. Im Kompartiment R sind die von der Zweitinfektion genesenen Menschen. Analog für den unteren Ast im menschlichen Kompartimentmodell, mit Vertauschen der beiden Serotypen.

Jeder Mensch kann also nur einmal an jedem Serotyp erkranken. Zudem gibt es eine temporäre Kreuzim-

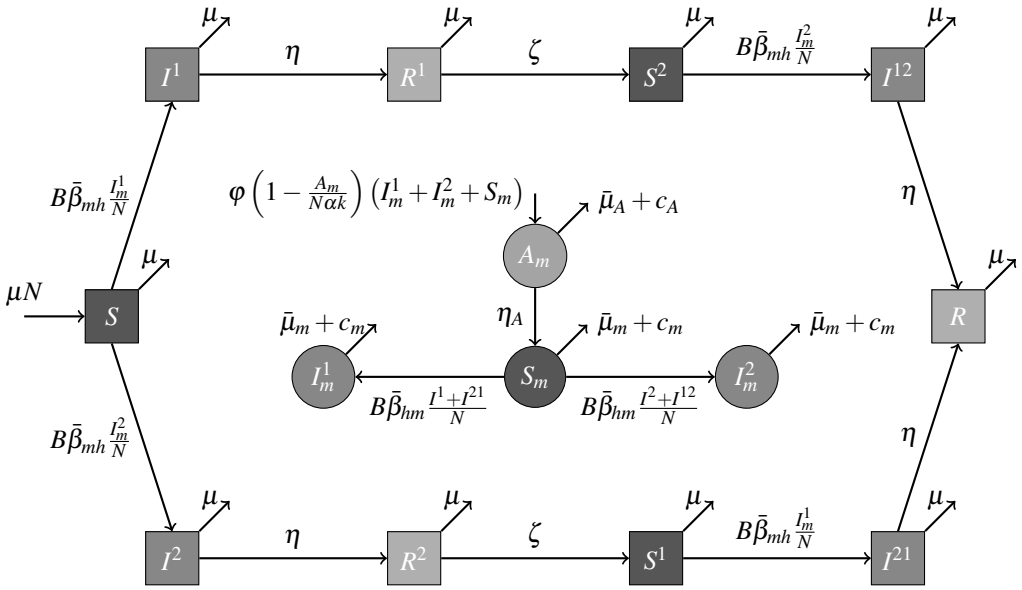


Abbildung 1: Kompartimentmodell der mückenübertragenen Krankheit mit zwei Serotypen und temporärer Kreuzimmunität

munität gegenüber dem anderen Serotyp der Krankheit.

Die Modellparameter sind in der Tabelle 1 erklärt. Damit passt das mathematische Modell prinzipiell zu Dengue-Fieber. Dieses Modell findet sich bereits in [7] zusammen mit einer Bestimmung der positiv invarianten Mengen und der Koordinaten des trivialen und krankheitsfreien Gleichgewichtspunktes sowie der Koordinaten der beiden halbendemischen Gleichgewichtspunkte. Wir ergänzen jetzt mit der genauen Anzahl der Gleichgewichtspunkte. Zudem zeigen wir, dass abhängig von der Dauer der temporären Kreuzimmunität $1/\zeta$ für $\mathscr{R}_0 > 1$ (Basisreproduktionszahl) mittels Hopf-Bifurkation des endemischen Gleichgewichtspunktes eine Grenzzykluslösung möglich ist.

Aus der Abbildung 1 ergibt sich ein System von nichtlinearen Differentialgleichungen. Sei N eine positive Konstante. Sei $\mu_A := \bar{\mu}_A + c_A$ und $\mu_m := \bar{\mu}_m + c_m$. Sei $\beta_{mh} := B\bar{\beta}_{mh}$ und $\beta_{hm} := B\bar{\beta}_{hm}$. Dabei gilt für die menschliche Population

$$\begin{aligned}
\dot{S} &= \mu N - \left(\beta_{mh}\tfrac{I_m^1+I_m^2}{N}) + \mu\right) S \quad , \quad S(0) = S_0 \\
\dot{I}^i &= \beta_{mh}\tfrac{I_m^i}{N} S - (\eta+\mu) I^i \quad , \quad I^i(0) = I_0^i \\
\dot{R}^i &= \eta I^i - (\zeta+\mu) R^i \quad , \quad R^i(0) = R_0^i \\
\dot{S}^i &= \zeta R^j - \left(\beta_{mh}\tfrac{I_m^i}{N} + \mu\right) S^i \quad , \quad S^i(0) = S_0^i \\
\dot{I}^{ij} &= \beta_{mh}\tfrac{I_m^j}{N} S^j - (\eta+\mu) I^{ij} \quad , \quad I^{ij}(0) = I_0^{ij} \\
\dot{R} &= \eta\left(I^{12}+I^{21}\right) - \mu R \quad , \quad R(0) = R_0
\end{aligned}$$

und für die Mückenpopulation (Index m für Mücke)

$$\begin{aligned}
&\dot{A}_m = \varphi\left(1-\tfrac{A_m}{\alpha k N}\right)\left(S_m + I_m^1 + I_m^2\right) - (\eta_A+\mu_A) A_m, \\
&\dot{S}_m = \eta_A A_m - \left(\beta_{hm}\tfrac{I^1+I^2+I^{12}+I^{21}}{N} + \mu_m\right) S_m, \\
&\dot{I}_m^i = \beta_{hm}\tfrac{I^i+I^{ji}}{N} S_m - \mu_m I_m^i, \\
&A_m(0) = A_{m,0}, S_m(0) = S_{m,0}, I_m^i(0) = I_{m,0}^i
\end{aligned}$$

mit $i, j \in \{1,2\}$, $i \neq j$. Wir betrachten nur Anfangswerte, die $N = S_0 + \ldots + R_0$ für die menschlichen Kompartimente erfüllen. Die menschliche Gesamtpopulation ist also konstant gleich N. Wir definieren den Mückenparameter $\mathscr{M} := \eta_A \varphi - \mu_m(\eta_A + \mu_A)$ und die Größe $\mathscr{R}_0 := \sqrt{\frac{\alpha k \beta_{hm} \beta_{mh} \cdot \mathscr{M}}{\varphi \mu_m^2 (\eta+\mu)}}$ (Basisreproduktionszahl).

Die Parameter $A_m^* := N\mathscr{M}\alpha k/(\varphi\eta_A)$, $N_m := A_m^* \eta_A/\mu_m$ spielen im Folgenden auch eine wichtige Rolle. Wir definieren mit $m := k\eta_A/\mu_m$ den Polyeder

$$\begin{aligned}
\Omega = \{&(S, I^1, I^2, R^1, R^2, S^1, S^2, I^{12}, I^{21}, R, A_m, S_m, I_m^1, I_m^2) \\
&\in \mathbb{R}_{\geq 0}^{14} \mid S + I^{12} + I^{21} + R + \textstyle\sum(I^i + R^i + S^i) \leq N, \\
&A_m \leq kN, S_m + I_m^1 + I_m^2 \leq mN\}
\end{aligned}$$

und erhalten:

Satz 1 [7]**:** Für das Anfangswertproblem sind die Mengen $\mathbb{R}_{\geq 0}^{14}$ und Ω positiv invariant.

Satz 2: Aus [7] sind bereits der triviale Gleichgewichtspunkt E_t ($S = N$, alle übrigen Komponenten null) und der krankheitsfreie Gleichgewichtspunkt

Parameter	Bedeutung	Wertebereich	Einheit
N	Konstante Anzahl der Menschen	$0 < N$	-
B	Stechrate Mücke	$0 \leq B$	Stiche/Tag
$\bar{\beta}_{mh}$	Ansteckungswahrscheinlichkeit Mücke→Mensch	$0 \leq \bar{\beta}_{mh} \leq 1$	1/Stich
$\bar{\beta}_{hm}$	Ansteckungswahrscheinlichkeit Mensch→Mücke	$0 \leq \bar{\beta}_{hm} \leq 1$	1/Stich
$1/\mu$	Lebenserwartung Mensch	$0 \leq 1/\mu$	Tage
$1/\bar{\mu}_m$	nat. Lebenserwartung Mücke	$0 \leq 1/\bar{\mu}_m$	Tage
$1/\eta$	infektiöse Periode (Mensch)	$0 \leq 1/\eta$	Tage
$1/\zeta$	Dauer der temp. Kreuzimmunität	$0 \leq 1/\zeta$	Tage
$1/\eta_A$	Dauer der aquatische Phase	$0 \leq 1/\eta_A$	Tage
k	(max.) Anzahl der Larven pro Mensch	$0 \leq k$	-
φ	Anzahl der Eier pro Brutplatz	$0 \leq \varphi$	-
α	mechanische Kontrolle	$0 < \alpha_0 \leq \alpha \leq 1$	-
c_A	Larvizid	$0 \leq c_A \leq 1$	1/Tag
c_m	Adultizid	$0 \leq c_m \leq 1$	1/Tag

Tabelle 1: Konstante Modellparameter

E_{DFE} ($S = N$, $A_m = A_m^*$, $S_m = N_m$, alle übrigen Komponenten null) sowie die beiden halbendemischen Gleichgewichtspunkte E_{HE1} und E_{HE2} bekannt. Sei mit $(S^*, I^*, R^*, A_m^*, S_m^*, I_m^*)$ der endemische Gleichgewichtspunkt des zugehörigen Ein-Serotyp $SIR - A_m S_m I_m$ Modells aus [12] (vgl. auch [2, 7]) bezeichnet. Dann lauten die halbendemischen Gleichgewichtspunkte mit $\bar{R}^* + \bar{S}^* = R^*$

$$E_{HE1} = (S^*, I^*, 0, \bar{R}^*, 0, 0, \bar{S}^*, 0, 0, 0, A_m^*, S_m^*, I_m^*, 0),$$
$$E_{HE2} = (S^*, 0, I^*, 0, \bar{R}^*, \bar{S}^*, 0, 0, 0, 0, A_m^*, S_m^*, 0, I_m^*).$$

Satz 3 (Rohlfs 2023, [3]): Das Differentialgleichungssystem mit symmetrischen Parametern besitzt im $\mathbb{R}^{14}_{\geq 0}$ *höchstens* die 6 Gleichgewichtspunkte E_t, E_{DFE}, E_{HE1}, E_{HE2}, E_{EE} und E_-.

Beweis: Systematisches Auflösung des polynomialen Gleichungssystems mit [13], siehe [3].

Satz 4 (Rohlfs 2023, [3]): Für $\mathscr{M} < 0$ existiert in $\mathbb{R}^{14}_{\geq 0}$ nur der GGP E_t. Für $\mathscr{M} = 0$ gilt $E_t = E_{DFE}$.

Beweis: Für $\mathscr{M} < 0$ gilt, dass für die Gleichgewichtspunkte $E_{DFE}, E_{HE1}, E_{HE2}, E_{EE}, E_-$ die Komponente $A_m < 0$ ist.

Satz 5 (Folger [8]): Für $\mathscr{M} > 0$ gilt für Anfangswerte mit $A_m(0) > 0, S_m(0) + I_m^1(0) + I_m^2(0) > 0$ und in Ω: $A_m \overset{t\to\infty}{\to} A_m^*$ und $S_m + I_m^1 + I_m^2 \overset{t\to\infty}{\to} N_m$. D.h. die aquatische Mückenpopulation und die erwachsene Gesamtmückenpopulation konvergiert gegen eine stationäre Lösung.

Generalannahme: Wir beschränken uns daher im Folgenden auf den interessanten Fall einer nichtaussterbenden Mückenpopulation. Sei im folgenden stets $\mathscr{M} > 0$.

Satz 6 (Rohlfs 2023, [3]): Die Next-Generation-Basisreproduktionszahl aus dem Original-Algorithmus nach van den Driessche und Watmough [4] ist durch die Größe $\mathscr{R}_0$ gegeben und stimmt mit der Berechnung mittels der Variante des Algorithmus aus [7] überein. Für $\mathscr{R}_0 < 1$ ist E_{DFE} lokal asymptotisch stabil.

Satz 7 (Rohlfs 2023, [3]): Für $\mathscr{R}_0 < 1$ liegen E_{HE1} und E_{HE2} nicht in Ω. Der Gleichgewichtspunkt E_- liegt nie in Ω. Der endemische Gleichgewichtspunkt E_{EE} liegt für $\mathscr{R}_0 > 1$ im Inneren von $\mathbb{R}^{14}_{\geq 0}$.

Die Stabilität des endemischen Gleichgewichtspunktes E_{EE} hängt (mindestens) vom Parameter ζ ab, wie eine neue numerische Simulation zeigt.

2 Numerische Simulation, Hopf-Bifurkation, Grenzzyklus

Für die numerischen Simulationen werden neue skalierte Variablen eingeführt: $S_h = \frac{S}{N}$, $I_h^1 = \frac{I^1}{N}$, $I_h^2 = \frac{I^2}{N} \ldots$, $R_h = \frac{R}{N}$, $\bar{A}_m = \frac{A_m}{kN}$, $\bar{S}_m = \frac{S_m}{\frac{k\eta_A}{\mu_m} N}$, $\ldots$, $\bar{I}_m^2 = \frac{I_m^2}{\frac{k\eta_A}{\mu_m} N}$. Alle Komponenten beschreiben jetzt Bruchteile der Gesamt-

bevölkerung der Menschen bzw. der Mücken und liegen im Intervall $[0,1]$. Die numerischen Lösungen wurden mit SciPy [13] mit dem expliziten Löser DOP853 der Ordnung 8 erstellt. Folgende Parameter werden benutzt, siehe [7]:

$$\begin{array}{lll} N = 386000 & \eta = \frac{1}{3} & \mu_A = 0.18 \\ B = 0.22 & \zeta = \frac{1}{365} & k = 3 \\ \tilde{\beta}_{mh} = 0.78 & \eta_A = 0.12 & \alpha = 1 \\ \tilde{\beta}_{hm} = 0.66 & \mu_m = \frac{1}{23} & c_A = 0 \\ \mu = \frac{1}{80 \cdot 365} & \varphi = 8 & c_m = 0 \end{array} .$$

Für diese Wahl der Parameter berechnet sich die Basisreproduktionszahl zu $\mathscr{R}_0 \approx 3.7$. Als Anfangswert nahe E_{DFE} wählen wir zunächst den symmetrischen Wert $z_0 = (0.9999, 5 \cdot 10^{-5}, 5 \cdot 10^{-5}, 0, 0, 0, 0, 0, 0, 0; 1, 1, 0, 0)$. Da $\mathscr{R}_0 > 1$ ist, ist E_{DFE} instabil, wir können also einen epidemischen Ausbruch erwarten. Abbildung 2 zeigt die Entwicklung über ein Jahr. Man erkennt für die infizierten Kompartimente den typischen „Krankheitsberg“ mit initial exponentiellem Anstieg der Erstinfektionen. Das Maximum ist nach 35 Tagen erreicht, wobei ca. 20.5% der Menschen infiziert sind (beide Serotypen zu gleichen Teilen). Die Kurven der Zweitinfektionen verläuft deutlich flacher, dies ist ein Effekt der temporären Kreuzimmunität. Die durchschnittliche Dauer der Kreuzimmunität in Tagen ist durch $\frac{1}{\zeta}$ gegeben. Je größer ζ ist, desto früher und schneller sind Zweitinfektionen möglich und desto höher ist der Krankheitsberg der Zweitinfektionen. Der Verlauf der Erstinfektionen wird durch ζ nicht nennenswert beeinflusst. Abbildung 3 zeigt den Verlauf für verschiedene Werte von ζ. In Abbildung 4 ist der Verlauf über einen Zeitraum von zehn bzw. 100 Jahren dargestellt. Wir sehen, dass sich die Infektionszahlen auf ein niedriges Niveau einpendeln. Die entsprechenden Komponenten von E_{EE} sind in der Abbildung mit $\star$ markiert. Die Lösung konvergiert anscheinend gegen das endemische Gleichgewicht E_{EE}. Es stellt sich die Frage ob das endemische Gleichgewicht E_{EE} lokal asymptotisch stabil ist. Wir werden sehen, dass dies nicht immer so ist.

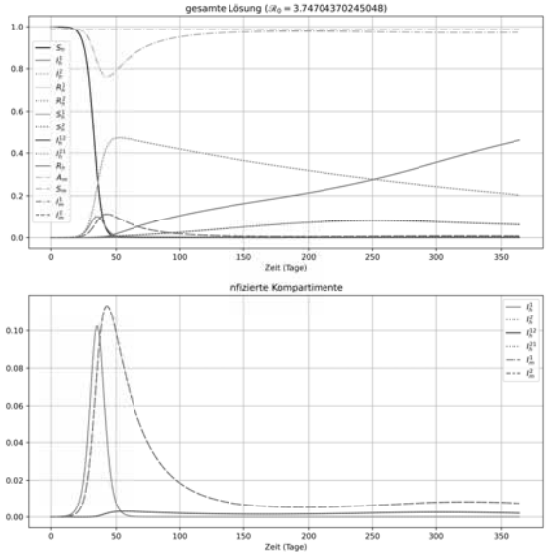


Abbildung 2: Verlauf über ein Jahr

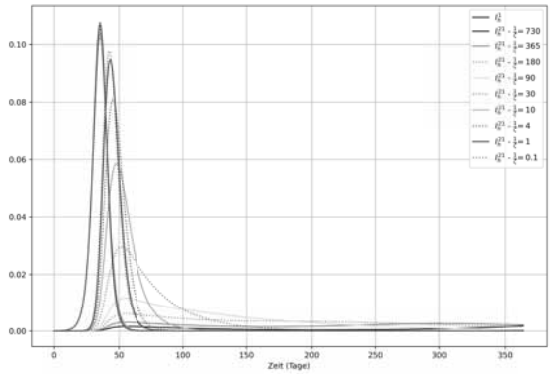


Abbildung 3: Effekt der Kreuzimmunität

Im Anfangswert z_0 sind die Werte von $I_h^1 = I^1/N$ und $I_h^2 = I^2/N$ identisch gewählt. Wegen der Symmetrie der beiden Krankheitsäste mit gleichen Parametern und symmetrischen Anfangswerten in z_0 ist die Dynamik der beiden Serotypen daher vollkommen identisch. Für asymmetrische Anfangsbedingungen kann die Dynamik anders sein. Wir wählen jetzt einen anderen asymmetrischen Anfangswert $z_1 = (0.9999, 5.1e-5, 4.9e-5, 0, 0, 0, 0, 0, 0, 0; 1, 1, 0, 0)$ und lösen das Anfangswertproblem für diesen noch einmal. Das Langzeitverhalten bei Start in z_1 (100 Jahre) zeigt Abbildung 5. Nach dem Krankheitsberg nähert sich die Lösung zunächst dem endemischen Gleichgewicht E_{EE} an, beginnt dann aber, immer stärker zu oszillieren. Dabei ist zu beachten, dass sich zuerst die Erst- und Zweitinfektionen annähern, die Kurven sich aber danach nach Serotypen aufspalten. Es ist dann in periodischen Abständen immer einer der beiden Serotypen dominant. Das endemische Gleichgewicht scheint also nicht global sta-

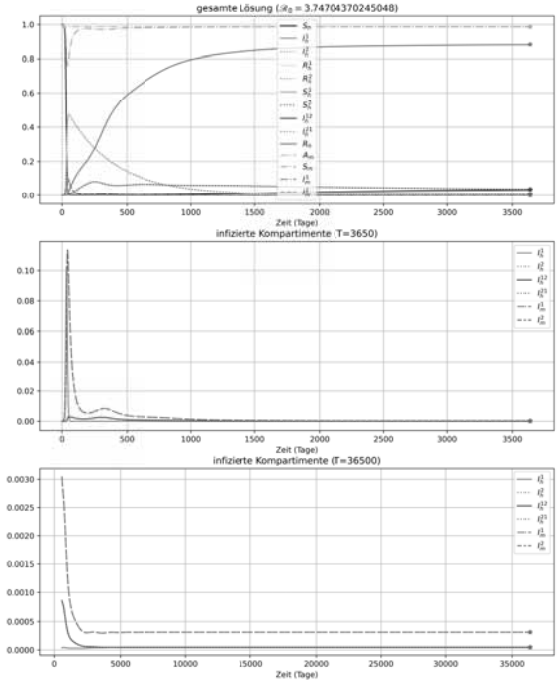


Abbildung 4: Langzeitverlauf. Die $\star$-Marker rechts sind die entsprechenden Werte des endemischen Gleichgewichts E_{EE}.

bil zu sein. Wir untersuchen die lokale Stabilität und wählen Anfangswerte nahe E_{EE}:

$$\begin{aligned} z_2 &\approx E_{EE}(\zeta = \frac{1}{4 \cdot 365}) + w \\ z_3 &\approx E_{EE}(\zeta = \frac{12}{365}) + w \\ w &= (0, -10^{-7}, 10^{-7}, 0, \ldots, 0) \end{aligned}$$

In Abbildung 6 sind die resultierenden Lösungen zu sehen. Hier wurden die Erst-und Zweitinfektionen (pro Serotyp) summiert und einmal als Zeitreihe (links) und einmal als Phasenplot (rechts) aufgetragen. Die Lösungen zu den beiden Anfangswerten z_2 und z_3 unterscheiden sich deutlich. Für $\zeta = \frac{1}{4 \cdot 365}$ (oben) beginnt die Lösung wieder zu oszillieren, d.h. E_{EE} scheint in diesem Falle instabil zu sein. Nach sehr langer Zeit ($T > 146000$) scheint sich das System auf einen periodischen Orbit einzuschwingen. Der periodische Orbit ist im Phasenplot (oben rechts) in orange dargestellt. Die Periode ist etwa 2116 Tage (5.8 Jahre).

Für $\zeta = \frac{12}{365}$ (unten) nähert sich die Lösung dem endemischen Gleichgewicht, in diesem Fall ist E_{EE} anscheinend stabil.

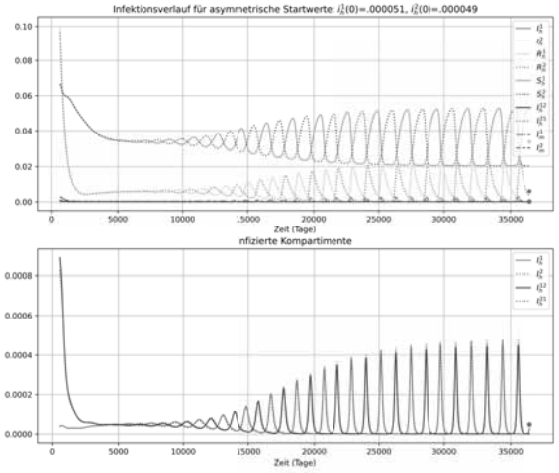


Abbildung 5: Verlauf über 100 Jahre, der Anfang der Lösung ($t < 637$) ist aus Skalierungsgründen nicht geplottet.

Wir berechnen jetzt numerisch mit SymPy [13] die Eigenwerte der Jacobimatrix in $E_{EE}(\zeta = \frac{1}{4 \cdot 365})$ bzw. $E_{EE}(\zeta = \frac{1}{4 \cdot 365})$ und erhalten 8 reelle negative Eigenwerte, *ein Paar konjugiert komplexer Eigenwerte mit positivem Realteil* und zwei Paare konjugiert komplexer Eigenwerte mit negativen Realteil bzw. 8 reelle negative Eigenwerte und drei Paare konjugiert komplexer Eigenwerte mit negativen Realteil.

Das Ergebnis bestätigt die numerischen Resultate in Abbildung 6. Aus dem Satz über die linearisierte Stabilität folgt, dass $E_{EE}(\zeta = \frac{12}{365})$ lokal asymptotisch stabil ist, da alle Eigenwerte negativen Realteil besitzen. Für die Linearisierung in $E_{EE}(\zeta = \frac{1}{4 \cdot 365})$ finden wir ein Paar konjugiert komplexer Eigenwerte mit positivem Realteil, das endemische Gleichgewicht ist in diesem Fall instabil.

Da die Jacobimatrix stetig von ζ und der Realteil eines Eigenwerts stetig von der Matrix abhängt, können wir mit dem Zwischenwertsatz folgern, dass der Realteil des Eigenwerts zwischen $\frac{1}{4 \cdot 365}$ und $\frac{12}{365}$ eine Nullstelle ζ^* besitzt. Dort ändert sich die Stabilität des endemischen Gleichgewichts E_{EE} , d.h. das qualitative Verhalten des Systems ändert sich in ζ^*. Ein Parameterwert mit dieser Eigenschaft heißt Bifurkationspunkt.

Wir wollen jetzt die Bifurkationspunkte für ζ numerisch bestimmen. Auf einem äquidistanten Gitter

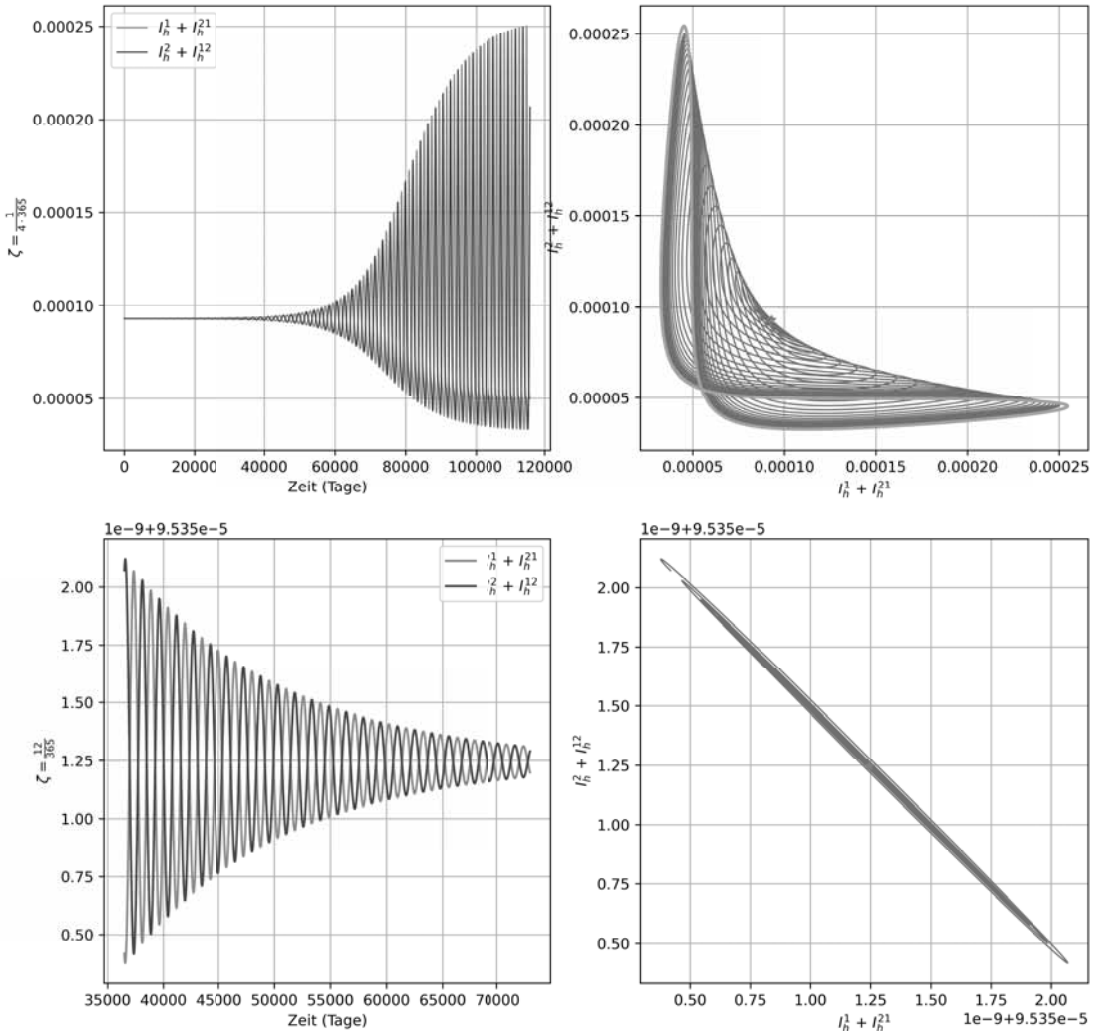


Abbildung 6: Lösungsverhalten nahe E_{EE} für verschiedene Werte von ζ; oben: instabil, unten: stabil.

auf dem Intervall $\left[\frac{0.01}{365}, \frac{100}{365}\right]$ mit 1000 Gitterpunkten wurde jeweils die Jacobimatrix für ζ ausgewertet, die Eigenwerte numerisch bestimmt und der „Pfad“ des kritischen Eigenwertpaares verfolgt. Der Verlauf des Realteils der Werte ist in Abbildung 7 über ζ aufgetragen. Der Realteil überquert zweimal die imaginäre Achse, d.h. das endemische Gleichgewicht E_{EE} wechselt zweimal die Stabilität (stabil→instabil→stabil). Die beiden Bifurkationspunkte wurden mit „manueller Intervallhalbierung“ bestimmt zu:

$$\zeta_1 \approx 0.0003929509374 \approx 0.14342709214/365$$
$$\zeta_2 \approx 0.0208272049771 \approx 7.60192981667/365.$$

Die bisherigen Berechnungen legen die Vermutung nahe, dass hier in beiden Bifurkationspunkten eine Andronov-Hopf-Bifurkation vorliegt. Zum Nachweis beziehen wir uns auf die Darstellung in [11]. Wie oben beschrieben sind die Voraussetzungen erfüllt:

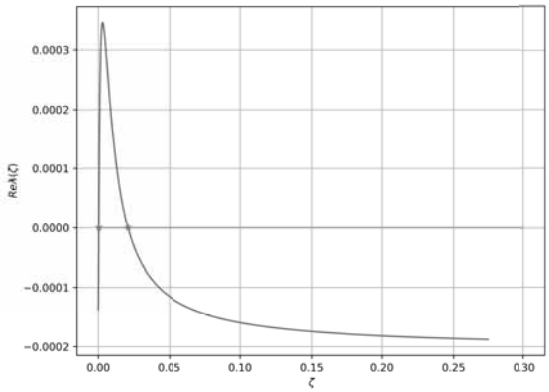


Abbildung 7: Realteil des interessanten Eigenwertpaars in E_{EE} für $\zeta \in [\frac{0.01}{365}, \frac{100}{365}]$.

- Es gibt eine Familie von Gleichgewichten $E_{EE}(\zeta)$ nahe ζ_i.
- Die Jacobimatrix $F'(E_{EE}(\zeta),\zeta)$ hat ein einfaches Paar konjugiert komplexer Eigenwerte, das in ζ_i rein imaginär ist.

Nachzuweisen sind noch die Bedingungen für die Nichtdegeneriertheit [10]:

1. $(\mathrm{Re}\lambda)'(\zeta_i) \neq 0$. Dies können wir in Abbildung 7 ablesen, es gilt $(\mathrm{Re}\lambda)'(\zeta_1) > 0$ bzw. $(\mathrm{Re}\lambda)'(\zeta_2) < 0$ (angenommen, dass die Funktion glatt ist).
2. Der (erste) Lyapunov-Koeffizient $l_1(\zeta_i)$ muss ungleich null sein.

Für die Lyapunov-Koeffizienten benutzen wir die folgende Formel [11]:

$$l_1(\zeta_i) = \tfrac{1}{2\omega_0}\mathrm{Re}[\langle p, C(q,q,\bar{q})\rangle - \\ -2\langle p, B(q, A_0^{-1}B(q,\bar{q}))\rangle + \\ +\langle p, B(\bar{q}, (2i\omega_0 I_n - A_0)^{-1}B(q,q))\rangle].$$

Dabei ist ω_0 der Imaginärteil des Eigenwertpaares $\lambda(\zeta), \bar{\lambda}(\zeta)$, A_0 die Jacobimatrix im Bifurkationspunkt, sowie B und C die zweiten bzw. dritten Ableitungen von F. q ist ein Eigenvektor von A_0 zum Eigenwert $\mathrm{i}\omega_0$, p ein adjungierter Eigenvektor von A^T zum Eigenwert $-\mathrm{i}\omega_0$. $\langle\cdot,\cdot\rangle$ bezeichnet hier das Skalarprodukt auf $\mathbb{C}^n$. p und q werden so skaliert, dass $\langle p,q\rangle = 1$ und $\langle q,q\rangle = 1$ ist. $\bar{q}$ bezeichnet den komplex konjugierten Vektor. Da das polynomiale Gleichungssystem für die Gleichgewichtspunkte nur Terme höchstens zweiten Grades enthält, ist hier $C = 0$ und B konstant, was die Berechnung deutlich vereinfacht. Mit Hilfe von SymPy berechnen wir die Lyapunov-Koeffizienten zu $l_1(\zeta_1) < 0$, $l_1(\zeta_2) < 0$ Beide sind negativ, also sind beide Bifurkationspunkte nicht degeneriert, d.h. es gibt stabile periodische Orbits (nahe ζ_i). Das ist in beiden Fällen der sogenannte superkritische Fall der Hopf-Verzweigung. Wir halten als Ergebnis aus Abbildung 7 und 8 und der theoretischen Untersuchung fest.

Satz 8: Seien die oben genannten Parameterwerte benutzt. Sei $0 < \zeta < 0.26$. Dann gilt: Der endemische Gleichgewichtspunkt E_{EE} ist lokal asymptotisch stabil für $\zeta < \zeta_1$ oder $\zeta > \zeta_2$. Der endemische Gleichgewichtspunkt E_{EE} ist für $\zeta_1 < \zeta < \zeta_2$ instabil und es gibt einen stabilen periodischen Orbit. Bei ζ_1 und ζ_2 tritt eine superkritische Hopf-Verzweigung auf.

Zusammenfassung

In [8, 15] sieht man bereits eine komplizierte Dynamik in den numerischen Lösungen bei verschiedenen zwei Serotyp Modellen einer mückenübertragenen Krankheit. Hier wird jetzt erstmals konkret für $\mathscr{R}_0 > 1$ ein stabiler Grenzzyklus um einen instabilen eindeutigen endemischen Gleichgewichtspunkt nachgewiesen. Das Auftreten hängt (mindestens) von der Dauer der temporären Kreuzimmunität $1/\zeta$ ab. Bei zeitabhängigen Parametern, wie in der Realität, kann die Dynamik selbstverständlich noch komplizierter sein.

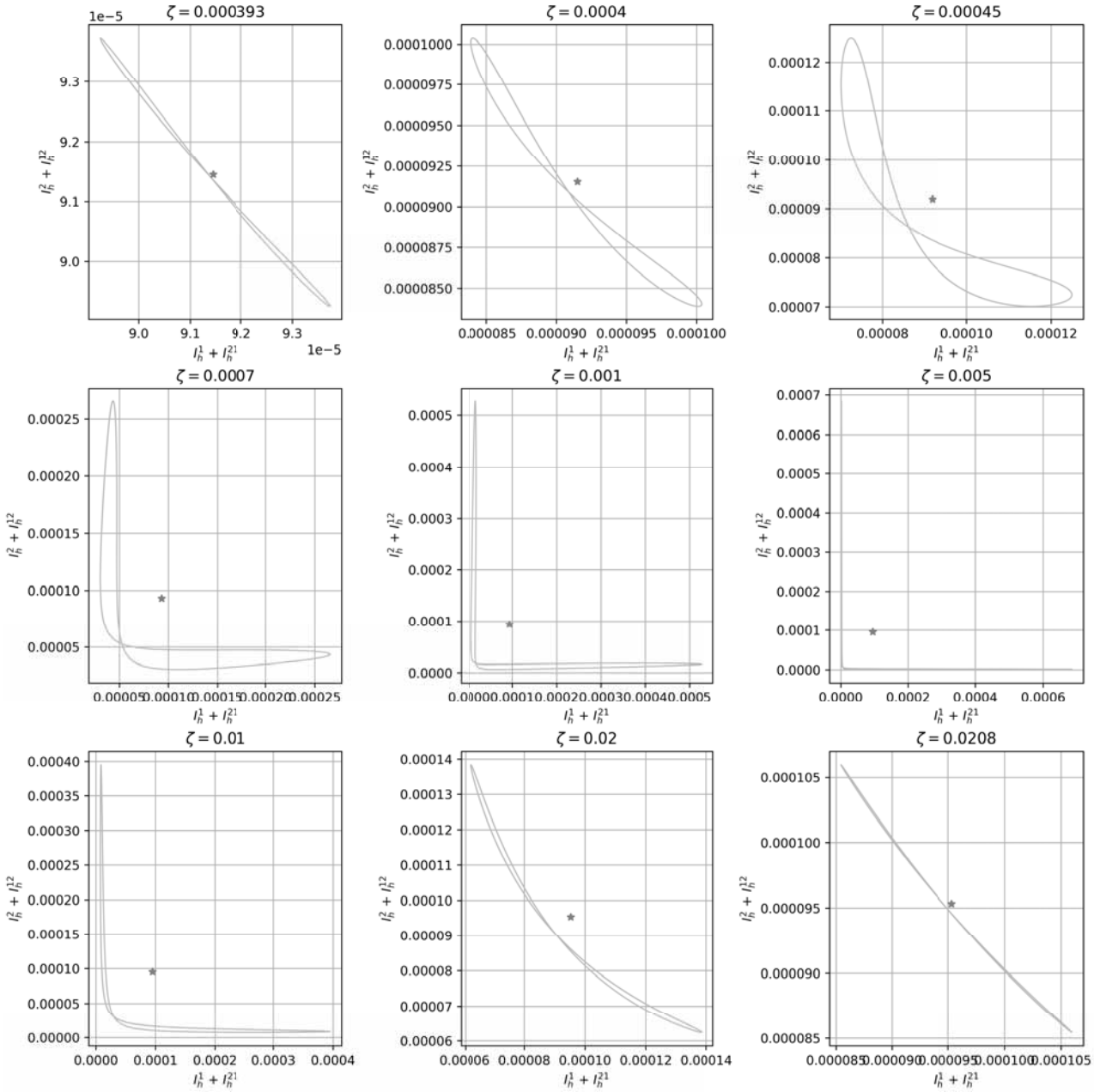


Abbildung 8: Periodische Orbits zwischen ζ_1 und ζ_2

Literatur

[1] Benelli G, Mehlhorn H. *Mosquito-borne Diseases*. Cham: Springer; 2018.

[2] Chudej K, Fischer A, Albrecht G, Herath M. Optimale Mückenbekämpfung bei einem neuen mathematischen Dengue-Fieber Modell mit 2 Serotypen. In: Wittmann J, et al editors. *Simulation in Umwelt- und Geowissenschaften Workshop Hannover 2018*. Aachen: Shaker; 2018. p 221–233.

[3] Chudej K, Rohlfs J. Theoretische Untersuchung eines Kompartimentmodells einer mückenübertragenen Krankheit mit zwei Serotypen und temporärer Kreuzimmunität. In Wittmann J editor. *Simulation in Umwelt- und Geowissenschaften Workshop Leipzig 2024*. to appear.

[4] van den Driessche P, Watmough J. Reproduction Numbers and Sub-Threshold Endemic Equilibria for Compartmental Models of Disease Transmission. *Mathematical Biosciences*. 2002; 180: 29–48.

[5] Dumont Y, Chiroleu F. Vector control for the chikungunya disease. *Mathematical Biosciences*. 2010; 7(2): 313–345.

[6] Esteva L, Vargas C. Coexistence of different serotypes of dengue virus. *Mathematical Biology*. 2003; 46: 31–47.

[7] Herath M, Chudej K. Analyse, Simulation und optimale Steuerung eines Dengue-Fieber-Modells mit temporärer Kreuzimmunität. In: Deatcu C, et al editors. *Proc. ASIM SST 2020*. Wien: ARGESIM; 2020. p 63–71. doi: 10.11128/AREP.59.A59010

[8] Folger G, *Modellierung, Analyse und Optimale Steuerung von gefährlichen Krankheiten* Diss. Univ. Bayreuth; 2021.

[9] Kugelmann B, Pulch R. Optimal control of a dengue model with cross-immunity. *Journal of Mathematics in Industry*. 2024; 14(8) 1–14.

[10] Kuznetsov YA. *Elements of applied bifurcation theory*. New York: Springer; 1998.

[11] Kuznetsov YA. Andronov-Hopf bifurcation. *Scholarpedia*. 2006; 1 (10): 1858.

[12] Rodrigues HSF, *Optimal Control and Numerical Optimization Applied to Epidemiological Models* PhD. Univ. de Aveiro; 2012.

[13] Virtanen P, et al, SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*. 2020; 17.3: 261–272.

[14] Xue L, Zhang H, Sun W, Scoglio C. Transmission dynamics of multi-strain dengue virus with cross-immunity. *Applied Mathematics and Computation*. 2021; 392: 125742.

[15] Zheng TT, Nie LF. Modelling the transmission dynamics of two-strain Dengue in the presence awareness and vector control. *Journal of Theoretical Biology*. 2018; 443: 82–91.

A meta-model for comparing carbon capture storage and utilisation technologies using life cycle analysis

J. Benz[1], K. Blank[1], S. Hötzel[1], J. Just[1], C. Lau[1], F. Schicks[1], N. von der Ahe[1],

J. Wittmann[1]

[1]HTW Berlin - University of Applied Sciences, Wilhelminenhofstraße 75A, 12459 Berlin

Abstract. This research paper comprehensively reviews current carbon capture and storage (CCS) and carbon capture and utilization (CCU) technologies with the aim of developing a metamodel to make the different approaches comparable. Furthermore, the study examines the different pathways of the technologies in a life cycle analysis (LCA) concerning their efficiency in terms of carbon footprint. The analysis shows that some processes are highly energy-intensive, underlining the need for renewable electricity to minimize CO_2 emissions. However, the study also points out several challenges, including incomplete data and unknown variables that hinder the implementation and evaluation of these technologies. In addition, the criticisms and limitations associated with CCS and CCU stress the need for further research and development in this critical area.

Introduction

The Paris Agreement [UnitedNations, 2015] is a crucial milestone in global efforts to mitigate climate change and underlines the urgent need for nations to work together to tackle greenhouse gas emissions. Carbon Capture and Storage (CCS), the process of trapping and storing Carbondioxide (CO_2) emissions, and Carbon Capture and Utilization (CCU), the transformation of CO_2 into valuable products, are crucial technologies for meeting the agreement's ambitious targets. In CO_2-intensive industries like the cement industry, capturing and mineralizing 1t of Carbondioxide equivalents (CO_2-Eq) could avoid over 1t of CO_2-Eq emissions by substitution of conventional production Ostovari et al. [2020]. However, the pressing issue is how effectively these technologies can fulfill current commitments and drive progress. Understanding the opportunities and limitations of these technologies is essential to assess their potential to achieve the goals set out in the Paris Agreement. While there is extensive research on the individual aspects of these technologies, there still is a notable gap in synthesizing this knowledge into a coherent framework. Therefore, our aim is twofold: to consolidate the existing research into a comprehensive overview by developing an over-all meta-model and to assess the collective potential of CCS and CCU to meet global emission reduction targets in terms of their carbon footprint, evaluated by a Life Cycle Assessment (LCA). In addition, our analysis will highlight existing data gaps and areas that require further investigation, thereby contributing to the ongoing discourse on climate change mitigation strategies.

1 Meta-Model

A model is required to execute a LCA calculating the environmental impact of technologies and processes. LCAs of carbon mineralization face the challenge of modeling many different processes and materials simply and comparably. For this reason, we developed a simplified meta-model. The model should consider the complexity of the various processes and feedstocks involved in carbon mineralization.

Problems creating a meta-model Carbon mineralization involves various processes depending on the technology and feedstock used. These processes include CO_2 capture, the reaction of CO_2 with mineral materials, the transport of materials and products, the pre-treatment of feedstocks, the further processing of products, and the utilization of electric energy, heat, and water. Modeling these processes in a single comprehensive LCA model can be highly complex and resource-

intensive. Some of the specific modeling issues we encountered during this study are:

Technology pathways: various technologies and approaches to carbon mineralization can differ greatly in their processes, materials, and environmental impacts. Modeling this diversity requires the development of a flexible model that can accommodate the different technologies without becoming too detailed.

Feedstock diversity: Feedstocks for carbon mineralization can vary widely and include natural minerals (like olivine or serpentine), waste products (like steel slag, ashes, etc.), and CO_2 from various sources. Each feedstock has different properties and requires different processes and conditions to optimally capture CO_2. The model must analyze the varying environmental impacts resulting from the different resources and energy amounts required by each feedstock.

Interactions between processes: The various carbon mineralization processes interact in complex ways that influence the system's overall performance and environmental impact. These interactions must be integrated and simplified into the model to enable a holistic assessment.

Requirements for the meta-model We developed a simplified meta-model to analyze the complexity of different carbon mineralization technologies and ensure comparability between different technologies and feedstocks. This meta-model should be an abstract model that simplifies the structure and behavior of the complex system by identifying the most critical factors and relationships. We intended the meta-model to have the following characteristics:

Simplification: The developed meta-model should reduce the complexity of carbon mineralization by eliminating unnecessary details and boundary effects. The goal was to abstract the different processes of the various pathways at a superordinate level and to summarise sub-processes.

Comparability: The meta-model should enable the comparability between different technologies and feedstocks by using consistent processes and a common functional unit as an assessment reference to quantify the relevant environmental impacts.

Parameterization: It should be possible to parameterize the meta-model to enable the variation of relevant variables such as energy consumption, input quantities, and outputs. In this way, the user can execute various scenarios and sensitivity analyses to investigate the effects of changes to the input parameters.

Development of the simplified meta-model A detailed literature review has been elaborated, but cannot be included in this paper for reasons of space. Based on this review we analyzed various existing LCAs of CCS/CCU processes concerning the technologies, feedstocks, and process steps described. The relevant processes were determined based on these LCAs and the described models. The goal was to identify the intersections between the different models and determine generally applicable processes for the various pathways. The identified processes for the meta-model are feedstock supply, CO_2 capture, pre-treatment, carbonation, and post- processing. With the help of these processes, we derived an initial model that serves as the basis for the LCA analysis which is shown in Figure 1. Carbonation itself and the necessary pretreatment are the main processes at the center, accompanied by feedstock supply and CO_2-supply-processes on the input side and the post-processing on the output side. A possible utilization of end-products is not covered by the proposed meta-model because no meaningful generalising model assumption can be made at this point due to the large number of usage options. The next step was to implement this model in the LCA-software Umberto [ifu, 2024] to carry out the LCA for various carbon mineralization pathways.

2 Life Cycle Assessment

LCA is a method used to comprehensively analyze a product's or technology's environmental impact over its life cycle. The ISO 14040 and ISO 14044 [ISO, 2020a,b] standardize and describe the procedure and structure of an LCA. Despite being standardized, LCAs in the field of carbon mineralization are challenging to compare, as critical factors such as the functional unit, system boundaries, and individual processes can be selected differently. For this reason, a guideline for implementing LCAs for carbon capture was used as a basis for this study [Müller et al., 2020].

Goal and scope The main objective of this LCA is to quantify and compare the environmental impact of different carbon mineralization processes with the developed meta-model. Therefore, testing the meta-model with actual data is another study objective. The assessment concentrates on the carbon foot-print caused by

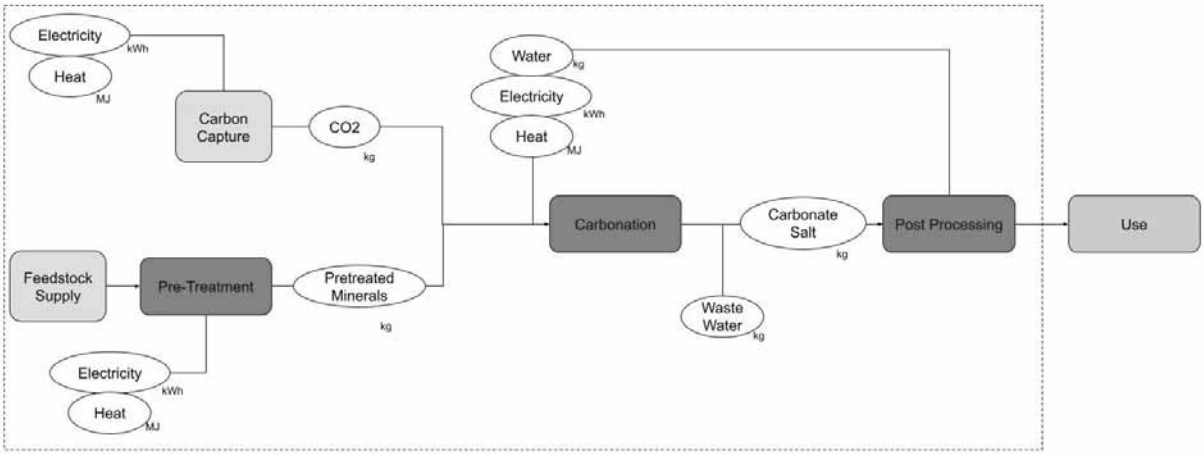


Figure 1: Developed meta-model for LCA

the different technologies. Such a comparison helps drive the development and implementation of environmentally friendly carbon capture technologies and, thus, significantly contribute to reducing global CO_2 emission. Carbon mineralization is an approach for permanently storing carbon by reacting and binding carbon dioxide in a stable mineral form [Stokreef et al., 2022]. For this reason, we defined the functional unit in this study as 1t of CO_2 bound by a carbonization process. This choice provides a direct and comparable benchmark for assessing different carbon capture technologies and processes. With this definition of the functional unit, we can cross-check the CO_2 emissions from all activities during the mineralization. If the emissions are less than 1t of CO_2 eq per tonne of bound CO_2, we can state that the emissions are net-negative, i.e., the process removes CO_2 from the atmosphere.

The system boundaries of this LCA were defined to consider the direct environmental impacts of the carboniza- tion processes. It was decided not to include the use of potential end products that could result from carbonization processes in this analysis. This focus allows a more accurate assessment of the carbonization technologies' environmental impacts without being distracted by variable application contexts, use scenarios and substitutions. The CO_2 supply was included in the system boundary, as the process of CO_2 supply can require a huge amount of energy and is, therefore, a decisive factor in determining the environmental impact of carbon mineralization. The following two types of CO_2 sources were considered in this study: direct air capture and point sources (such as industrial facilities like power plants and factories).

Umberto was used to implement the developed meta-model and calculate the environmental impact. In Umberto, the relevant process parameters and resource inputs were mapped in the model (see supplementary files). Figure 2 shows the implemented Umberto model. To simplify the modeling in Umberto, we made the following assumptions:

- Identical transport (60km by lorry) for the feedstocks was assumed in all pathways
- Electricity generation based on the German electricity mix
- For comparison, label certified electricity from Switzerland with renewable energies
- Heat from natural gas
- Non-existent extraction of olivine and serpentine in Umberto was replaced by a comparable process of limestone extraction

2.1 Life cycle inventory and data situation

We had to obtain reliable data for different carbon mineralization pathways for the analyses on the developed meta-model. Own measurements or actual data from a company were not available. For this reason, we analyzed the data from LCAs found in the literature review

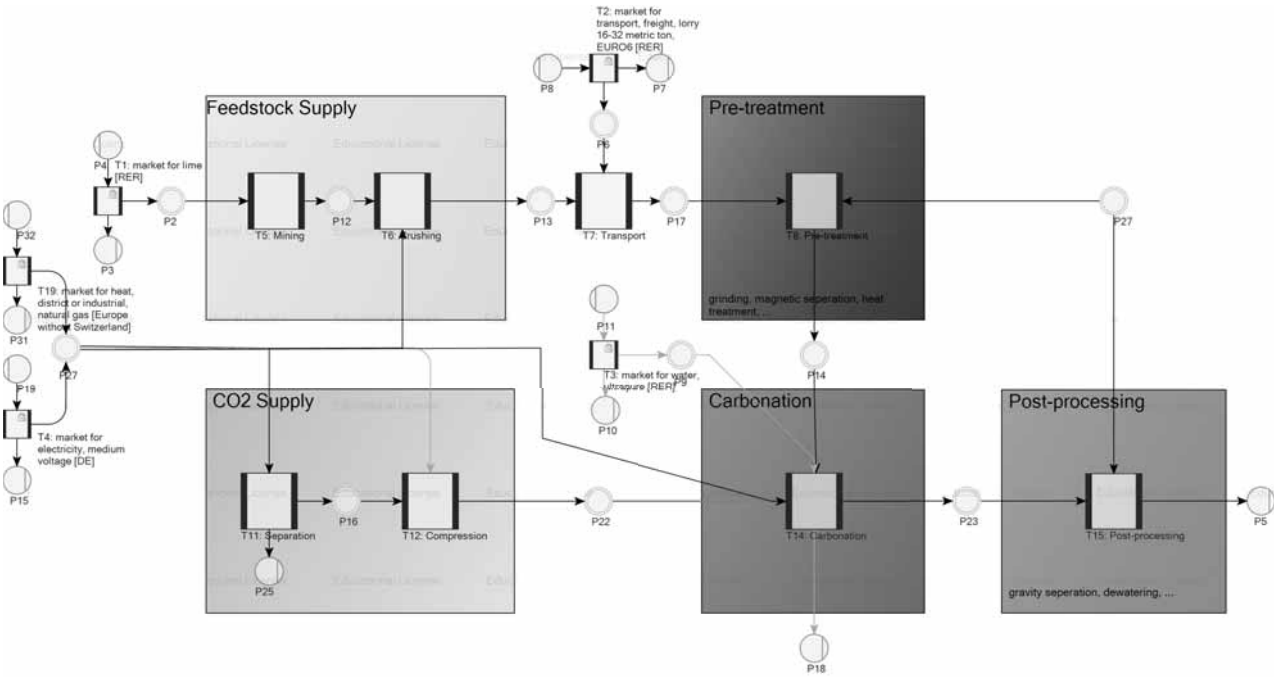


Figure 2: Developed model for LCA in Umberto

to identify relevant data. This data research results in a collection of data presented in several tables, sorted according to the processes described in the meta-model (see supplementary files). For further investigation, we found data on various technology pathways (direct and indirect). It was also important to consider data for different feedstocks, as different feedstocks used to have a considerable influence on the mass balances of inputs and outputs. Data was found for olivine, serpentine, and various waste materials such as steel slack [Bargiacchi et al., 2020, Digulla and Bringezu, 2023, Müller et al., 2020, Naraharisetti et al., 2017, Ostovari et al., 2020, Sanna et al., 2012, 2014, William Oconnor et al., 2005].

In addition to the data collected, various libraries and markets were used in Umberto to enable realistic modeling of the energy requirements [ecoinvent, 2023]:

- market for transport, freight, lorry 16-32 metric ton, EURO6 [RER]
- market for lime [RER]
- market for electricity, medium voltage [DE]
- market for electricity, medium voltage, label-certified [CH]
- market for heat, district or industrial, natural gas [Europe without Switzerland]
- market for water, ultrapure [RER] (only for serpentine)
- market for blast furnace slag [GLO] (rotary packed bed pathway)

During the data research, we identified several issues and challenges:

Up-to-date data: Up-to-date data is essential to correctly map technological developments and trends to obtain meaningful results in the LCA. Our data research revealed that in some cases, only older data (for example William Oconnor et al. [2005]) is available for individual processes and that this data formed the basis for various other LCAs found in the literature review [compare Ostovari et al., 2020, Naraharisetti et al., 2017, Kremer et al., 2022]. We must critically question whether this data is still meaningful today and reflects the current state of the art.

Accessibility of the data: In our data research, we have encountered problems with restricted access to specific datasets or data not being published in full. Reasons for this could be data protection, commercial interests, or other legal and administrative reasons. Due to the limited data available, we could not guarantee completeness across all carbon mineralization pathways in our LCA. We could only map and analyze the

processes and technologies for which data was available.

Accuracy: The accuracy of the data is crucial, as incorrect or inaccurate information can lead to false conclusions. That applies, in particular, to data on the inputs and outputs of the individual processes and on energy consumption. As we have not measured any data, we have to rely on third-party information for the data we use. Note that some of the data come from experiments under laboratory conditions [Wang and Maroto-Valer, 2011, Bodénan et al., 2014, Fabian et al., 2010, Romão et al., 2012]. For this reason, we can only make limited statements about our results for industrial and scaled applications where other conditions may exist.

Consistency: Data consistency is a critical factor in comparing different studies, technologies, and locations. As already described, there is the problem that sometimes only limited data can be retrieved, or data is only available for individual processes. For this reason, we combined our data from different sources. This results in a loss of consistency, as data generated under different conditions and for different purposes is correlated and summarized.

2.2 Life cycle impact assessment

The created meta-model enables the modeling of different CCS methods but builds on a limited data situation. In order to obtain meaningful results with the meta-model, we attempted to perform the impact analyses with data that was as coherent as possible. For this reason, this impact assessment focused on modeling the five pathways described in Ostovari et al. [2020]. We implemented the meta-model with the modeling software Umberto LCA+.

Carbonation occurs in the direct pathway with a continuously stirred tank reactor (CSTR) without any intermediate steps. The pre-treatment and carbonation conditions depend on the feedstock. Data for olivine and serpentine were available in the study. Olivine is mined and prepared by grinding and milling in the pre-treatment stage. In the subsequent carbonization, the pulverized olivine reacts with water and CO_2 from the CO_2 supply. After that, the results undergo further processing in the post-processing stage. The procedure with serpentine is the same except for the pre-treatment stage. Magnetic separation isolates the iron, and heat treatment is required. The pathway OlivineCSTR100 is based on the study by Eikeland et al. [2015]. The pathway SerpentineCSTR115 described in Ostovari et al. [2020] referenced results from William Oconnor et al. [2005].

These sources also investigate the direct process using a rotary-packed bed reactor (RPB). This RPB often uses steel slack as a feedstock. Steel slack is a waste product in various industries, requiring no additional extraction process. After grinding, it can react with CO_2. The RPB process offers several advantages, including using off-gas containing 15-20% CO_2 instead of pure CO_2. Furthermore, the waste product steel slack is utilized as feedstock, resulting in possible cost and energy savings [Ostovari et al., 2020, Pan et al., 2015].

In addition to these direct concepts, Ostovari et al. [2020] describe two indirect pathways examined in this study using the meta-model. These are the Nottingham pathway and the AA pathway. Serpentine is usually the feedstock for both of these pathways. Pre-treatment and post-processing in the AA pathway correspond to the direct concepts, while it takes intermediate steps in the carbonation. The serpentine reacts with ammonium sulfate in a solid-solid reaction, and the actual reaction with CO_2 follows afterward [Romão et al., 2012]. In the Nottingham pathway, the actual carbonation also takes place in two steps: first, an aqueous extraction and then an aqueous carbonation. In the Nottingham process, the feedstock supply and the pre-treatment stage are identical to the direct processes, as serpentine is also used as feedstock here. One exception is heat treatment during pre-treatment, which is unnecessary for the Nottingham Pathway. During the carbonation, the serpentine initially reacts with ammonium bisulfate in an aqueous reaction to generate a magnesium-rich solution that reacts with CO_2 in the second stage and binds the CO_2. The described Nottingham Pathway in Ostovari et al. [2020] is based on Wang and Maroto-Valer [2011].

With the modeling, we aimed to analyze which process is responsible for how much of the CO_2 emissions. The model was calculated once with the German electricity mix and heat from natural gas and once with a green electricity mix. Due to the data availability, which usually only contains energy consumption for a single process, we analyzed only CO_2 equivalence. However, this does not mean that other influences are irrelevant; instead, there is insufficient data to provide further information on other aspects.

Figure 3 shows the results for the German electricity mix. The CO_2 equivalence is positive in two methods, implying that the process produces more CO_2 than is stored. In the Rotary packed bed pathway, the CO_2

equivalence for the feedstock supply is negative, as it uses a blast furnace slag here. Umberto LCA+ rewards further use of this waste product with a negative CO_2 equivalence. For the Nottingham Pathway and Rotary packed bed pathway, no energy is required for the CO_2 supply, as emissions from point sources, such as industrial facilities, are used here.

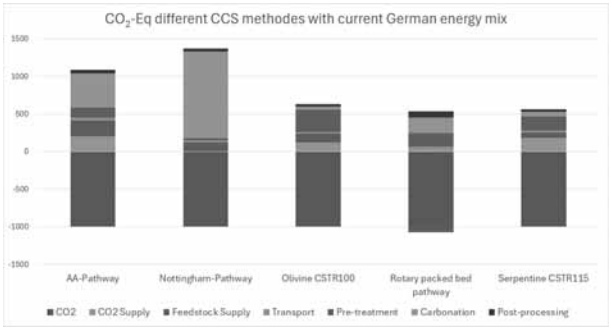


Figure 3: CO_2-eq for different mineralization pathways with German energy mix

We then calculated all five models using green electricity. Due to the lack of a sustainable heat source in the Ecoinvent database, we also used green electricity as the heat source in these models. The results can be seen in Figure 4. All energy-intensive processes now have a significantly better CO_2 equivalence, and all methods store more CO_2 than they emit. On the other hand, the feedstock supply and transport processes still generate almost the same amount of CO_2 and are therefore responsible for a large proportion of CO_2 emissions.

The latter happens because we include the prechains in these steps using data from Ecoinvent. We assume current rock extraction and transport conditions, not future conditions, that could decarbonize these steps.

In all models, CO_2 is permanently bound in rock, i.e., long-term storage. We did not investigate other methods like producing e-fuels, which are burnt later in their lifecycle, emitting CO_2 again.

2.3 Life cycle interpretation

The results from diagrams 3 and 4 show that the energy-intensive processes are responsible for the CO_2-eq in particular. Using renewable electricity can avoid a large proportion of the CO_2 emissions caused by the storage of CO_2. Until this is the case, CCS only makes limited sense. The Nottingham pathway is particularly striking, as this process emits significantly more CO_2 than

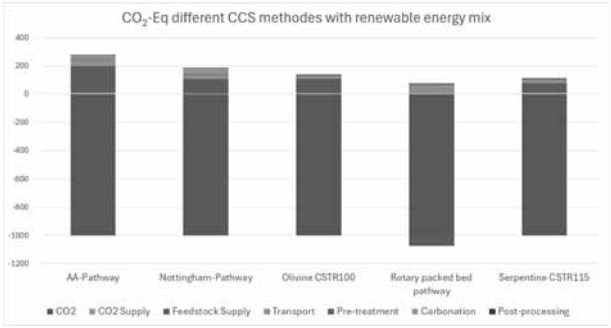


Figure 4: CO_2-eq for different mineralization pathways with renewable energy mix

is stored under the current German electricity mix. That leads to the conclusion that CCS only makes sense if the electricity mix is entirely renewable.

It is worth noting that there is an issue with the data situation. For various process steps, data from different sources, including data obtained in laboratory situations, were used in the analyzed studies. There needs to be empirical data on how energy use in large commercial systems will scale and develop.

3 Discussion

The LCA conducted in this study offers critical insights into the environmental impacts of CCS and CCU technologies. It underpins the urgent need for sustainable energy sources to power these technologies. Our findings highlight the challenges we must address to maximize the potential of CCS and CCU in effectively mitigating CO_2 emissions.

3.1 Energy Intensity and the Need for Renewable Energy

One of the most significant challenges highlighted by our analysis in Chapter 2.3 is the energy intensity of current CCS and CCU processes. The dependency on non-renewable energy sources not only undermines the overall carbon footprint reduction but also raises concerns about these technologies' sustainability and net environmental benefits. The transition to renewable energy sources is imperative to ensure that CCS and CCU technologies contribute positively to climate change mitigation efforts. This shift would align with the global push towards decarbonization and enhance

the technologies' appeal from an environmental perspective.

3.2 Data Gaps and the Importance of Comprehensive Data Collection

As elaborated in Chapter 2.1, our study also reveals substantial gaps in the available data, particularly regarding up-to-date information on the energy consumption and environmental impacts of CCS and CCU processes. These gaps hinder the ability to make informed decisions and assess the technologies' viability and effectiveness. Therefore, there is a pressing need for standardized data collection methods and increased transparency in reporting to facilitate more robust and comprehensive LCAs. Our findings align with a consensus; most papers examined in the literature review share that collaboration among academia, industry, and regulatory bodies is essential to establish uniform data collection frameworks and databases.

3.3 Technological Innovation and Scalability

The tone of the examined papers in the detailed literature review also highlights the importance of technological innovation in improving the efficiency and scalability of CCS and CCU technologies. Advances in process optimization, material sciences, and system integration are critical to overcome current limitations and reducing costs. Furthermore, exploring novel CO_2 capture and conversion pathways could open up new avenues for carbon utilization, thereby expanding the potential applications and markets for CCU products. Continued investment in research and development is crucial to accelerating these innovations.

3.4 Policy Implications and the Role of Incentives

The findings from our study show a need for supportive regulatory frameworks and incentives to promote the adoption and development of CCS and CCU technologies. Policies aimed at internalizing the cost of carbon emissions, such as carbon pricing mechanisms, can enhance the economic viability of CCS and CCU. Additionally, targeted subsidies, tax incentives, and funding for research and development can accelerate further development.

4 Summary

This research report examines various studies on CCU and CCS. In addition to reviewing already completed literature analyses, we conducted a literature search for the years 2022-2024. Based on this, we created an overview of the prevailing technology landscape. This foundation enabled us to develop a meta-model and conduct life cycle assessments on various technologies. The LCA results show that the type of electricity used significantly affects the overall efficiency of the technologies, especially in reducing the carbon footprint.

We identified numerous challenges and points of criticism. These include incomplete or outdated data, methodological weaknesses, and a lack of research in certain areas. Nevertheless, the analysis shows that CCS and CCU could contribute to achieving the 1.5-degree target of the Paris Agreement. The paper underlines the importance of further research and development in this area to fully exploit the potential of CCS and CCU and achieve the goals of the Paris Agreement.

5 Outlook

Looking to the future of these technologies, it is clear that despite the progress made in developing CCS and CCU, there still needs to be more research. In particular, existing data gaps need to be closed to make informed decisions about implementing and scaling these technologies.

A key focus of future research should be on improving the accuracy and completeness of available data. Achieving this demands increased collaboration between government agencies, research institutions, and industry partners to establish uniform standards for data collection and provide comprehensive data sets.

In addition, we need methodological improvements to enhance the robustness of life cycle assessments and better understand potential environmental, economic, and social impacts. New modeling and simulation approaches could help capture the systems' complexity and provide more accurate results.

References

Eleonora Bargiacchi, Nils Thonemann, and Jutta Geldermann. Life cycle assessment of synthetic natural gas production from different co2 sources: A cradle-to-gate study. *Energies*, 13(17):4579, 2020. ISSN 1996-1073. doi: 10.3390/en13174579.

F. Bodénan, F. Bourgeois, and C. Petiot. Ex situ mineral carbonation for co2 mitigation: Evaluation of mining waste resources, aqueous carbonation processability and life cycle assessment (carmex project). *Minerals Engineering*, 59:52–63, 2014. ISSN 0892-6875. doi: 10.1016/j.mineng.2014.01.011.

Finn-Erik Digulla and Stefan Bringezu. Comparative life cycle assessment of carbon dioxide mineralization using industrial waste as feedstock to produce cement substitutes. *Energies*, 16(10):4118, 2023. ISSN 1996-1073. doi: 10.3390/en16104118.

ecoinvent. ecoinvent version 3.6, 2023. URL https://support.ecoinvent.org/ecoinvent-version-3.6.

Espen Eikeland, Anders Bank Blichfeld, and Christoffer Tyrsted. Optimized carbonation of magnesium silicate mineral for co2 storage. *ACS applied materials & interfaces*, 7(9):5258–5264, 2015. doi: 10.1021/am508432w.

Martin Fabian, Maya Shopska, and Daniela Paneva. The influence of attrition milling on carbon dioxide sequestration on magnesium–iron silicate. *Minerals Engineering*, 23(8):616–620, 2010. ISSN 0892-6875. doi: 10.1016/j.mineng.2010.02.006.

ifu. Umberto lca software, 2024. URL https://www.ifu.com/de/umberto.

ISO. Din en iso 14040:2021-02, umweltmanagement_-ökobilanz_- grundsätze und rahmenbedingungen (iso_14040:2006_+ amd_1:2020); deutsche fassung en_iso_14040:2006_+ a1:2020, 2020a. URL https://www.iso.org/standard/37456.html.

ISO. Din en iso 14044:2021-02, umweltmanagement_-ökobilanz_- anforderungen und anleitungen (iso_14044:2006_+ amd_1:2017_+ amd_2:2020); deutsche fassung en_iso_14044:2006_+ a1:2018_+ a2:2020, 2020b. URL https://www.iso.org/standard/38498.html.

Dario Kremer, Christian Dertmann, and Simon Etzold. Ex-situ mineral carbonation – a parameter study on carbon mineralisation in an autoclave as part of a large-scale utilisation process. *Journal of CO2 Utilization*, 58:101928, 2022. ISSN 22129820. doi: 10.1016/j.jcou.2022.101928.

Leonard Jan Müller, Arne Kätelhön, Marvin Bachmann, Arno Zimmermann, André Sternberg, and André Bardow. A guideline for life cycle assessment of carbon capture and utilization. *Frontiers in Energy Research*, 8:505883, 2020. ISSN 2296-598X. doi: 10.3389/fenrg.2020.00015.

Pavan Kumar Naraharisetti, Tze Yuen Yeo, and Jie Bu. Factors influencing co2 and energy penalties of co2 mineralization processes. *Chemphyschem : a European journal of chemical physics and physical chemistry*, 18 (22):3189–3202, 2017. doi: 10.1002/cphc.201700565.

Hesam Ostovari, André Sternberg, and André Bardow. Rock 'n' use of co 2 : carbon footprint of carbon capture and utilization by mineralization. *Sustainable Energy & Fuels*, 4(9):4482–4496, 2020. doi: 10.1039/D0SE00190B.

Shu-Yuan Pan, Yi-Hung Chen, and Chun-Da Chen. High-gravity carbonation process for enhancing co2 fixation and utilization exemplified by the steelmaking industry. *Environmental science & technology*, 49(20): 12380–12387, 2015. doi: 10.1021/acs.est.5b02210.

Inês Romão, Experience Nduagu, and Johan Fagerlund. Co2 fixation using magnesium silicate minerals. part 2: Energy efficiency and integration with iron-and steelmaking. *Energy*, 41(1):203–211, 2012. ISSN 03605442. doi: 10.1016/j.energy.2011.08.026.

A. Sanna, M. Uibu, and G. Caramanna. A review of mineral carbonation technologies to sequester co2. *Chemical Society reviews*, 43(23):8049–8080, 2014. doi: 10.1039/C4CS00035H.

Aimaro Sanna, Matthew R. Hall, and Mercedes Maroto-Valer. Post-processing pathways in carbon capture and storage by mineral carbonation (ccsm) towards the introduction of carbon neutral materials. *Energy & Environmental Science*, 5(7):7781, 2012. ISSN 1754-5692. doi: 10.1039/c2ee03455g.

Stephen Stokreef, Farzaneh Sadri, and Arthur Stokreef. Mineral carbonation of ultramafic tailings: A review of reaction mechanisms and kinetics, industry case studies, and modelling. *Cleaner Engineering and Technology*, 8: 100491, 2022. ISSN 2666-7908. doi: 10.1016/j.clet.2022.100491.

UnitedNations. The paris agreement, 2015. URL https://unfccc.int/process-and-meetings/the-paris-agreement.

Xiaolong Wang and M. Mercedes Maroto-Valer. Integration of co2 capture and mineral carbonation by using recyclable ammonium salts. *ChemSusChem*, 4(9):1291–1300, 2011. doi: 10.1002/cssc.201000441.

William Oconnor, D.C. Dahlin, G.E. Rush, S.J. Gerdemann, L.R. Penner, and D.N. Nilsen. Aqueous mineral carbonation: Mineral availability, pretreatment, reaction parametrics, and process studies, 2005.

Modeling Material and Energy Flow in an Eco-Industrial Park using Discrete Event Simulation

Shufang Xie[1*], Tao Zhang[1], Tobis Uhlig[1], Oliver Rose[1], Björn Vollack[2]

[1]Institute for Computer science, University of Bundeswehr Munich, Werner-Heisenberg-Weg 39, 8557 Neubiberg, Germany; *shufang.xie@unibw.de

[2]Chair of Logistics Engineering, TUD Dresden University of Technology, Dresden, Germany

Abstract. In recent times, a specific number of eco-industrial parks have emerged as a viable solution to address the escalating environmental challenges. Within these industrial parks, factories engage in mutual interaction through the flow of materials and energy. By modeling the material and energy flow within an eco-industrial park, we gain a comprehensive understanding of how resources circulate. This understanding not only provides strategic insights but also enables the identification of optimization opportunities, fostering more efficient resource utilization, waste reduction, and significant cost savings. Therefore, this research focuses on modeling the material and energy flow in the park using the discrete event simulation technique. We will provide a detailed explanation of our modeling approach, outlining how we employ this method to optimize resource usage, reduce waste, and minimize the environmental impact of industrial activities within the park. As part of our research, we have also developed a simple virtual eco-industrial park example to validate and demonstrate the effectiveness of our modeling approach. This practical illustration will serve to showcase the real-world applicability and benefits of our research in creating more sustainable and efficient eco-industrial parks.

Introduction

In response to the increasing global environmental concerns, eco-industrial park (EIP) has emerged as a proactive and sustainable solution. EIPs are communities of manufacturing and service businesses collaborating to enhance environmental and economic performance through effective management of resources like energy, water, and materials. This collaborative approach seeks collective benefits greater than individual optimizations [1]. The concept of EIPs has recently captured significant interest from both industry and academic research communities, much of the focus has been on the planning and design stages, with limited attention to operational parks [2-5]. Boix et al. provide a comprehensive literature review on the optimization methods applied to the design of EIP [6]. As of 2011, there were over 20,000 operational industrial parks globally [7], offering substantial opportunities for material, energy, and waste savings. However, research on operational parks has been limited, often targeting specific environmental issues rather than providing comprehensive analyses for optimization. For instance, one study utilized a Monte Carlo model to simulate wastewater treatments in an industrial park in China, focusing on reducing pollution [8].

This paper aims to bridge this gap by employing modeling and simulation techniques to thoroughly investigate an operational industrial park. By employing modeling techniques, we can comprehensively analyze the intricate interactions and processes within the park. The central activity in an EIP revolves around the physical exchange of materials, energy, and services. Efficiently managing the flow of energy and materials is a cornerstone of any industrial park's operations. Prior to efficient management, the analysis and modeling of material and energy flow are essential. Consequently, this article places its primary focus on the simulation and modeling of energy and material flow within existing industrial parks, employing discrete event simulation method (DES). Widely

utilized in modeling, DES enables the study of systems that are discrete, dynamic, and stochastic [9]. It facilitates the simulation and understanding of how materials and energy traverse the system, providing valuable insights into resource management, waste reduction, and energy efficiency. A detailed explanation of our modeling approach will be provided, and a simple hypothetical park will be used to demonstrate the effectiveness of our modeling techniques.

The paper is structured as follows: a brief conceptual model is described in Section 1 while a more detailed formal model of material and energy flow are addressed in Section 2 and 3 respectively. The implementation and application of the simulation model are introduced in Section 4. The paper is concluded in the last section.

1Conceptual model of material and energy flow in EIP

1.1 Eco-industrial Park

In EIPs, common components typically include the factory infrastructure, material flow systems, and energy flow systems. These components are essential for the functioning of the park and its sustainable operations. Figure 1 illustrates a simplified EIP where multiple factories coexist. Suppliers from outside the EIP provide raw materials, and customers from outside the EIP consume the final products manufactured within the park. Instead of solely relying on externally purchased raw materials, the factories within the EIP promote resource synergy by using outputs from neighboring factories as valuable raw materials.

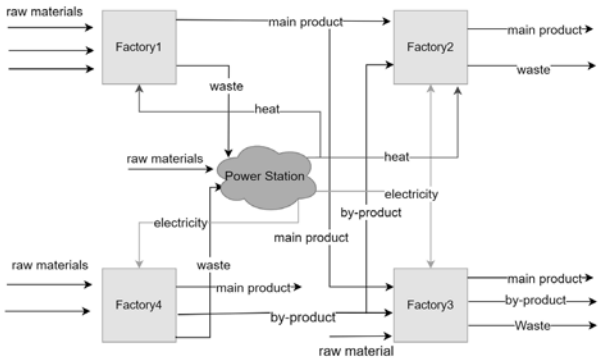


Figure 1: An EIP example

Moreover, in pursuit of sustainable waste management, the EIP adopts waste-to-energy (WTE) technology instead of depositing waste in landfills. This process recovers energy from waste sources in the form of heat, electricity, or transport fuels [10], complemented by a dedicated power plant within the park. This integrated approach minimizes environmental impact and provides a valuable energy source for the park's factories, promoting eco-friendly solutions and reducing reliance on external energy sources. Notably, in many cases, WTE plants are combined heat and power (CHP) producers [11]. In addition to WTE initiatives, the EIP enhances energy efficiency by incorporating CHP plants, known for simultaneously producing electricity and useful heat from a single fuel source. Unlike traditional power plants, CHP plants capture and utilize waste heat for heating and cooling applications, optimizing energy utilization.

1.2 Material and Energy flow in an EIP

From the perspective of each factory, materials can be categorized into two aspects: input materials and output materials. Input materials encompass not only the raw materials acquired from external suppliers but also the innovative utilization of by-products and waste generated within one factory, fostering a symbiotic relationship with another. On the output side, factories yield main products that form the core of their operations, accompanied by valuable by-products and, inevitably, waste materials. In an industrial park, energy operates in two essential forms: electricity and heat. These dual components play a crucial role in powering various processes and activities within the industrial complex.

Material and energy flow refers to the movement, transfer, or transition of materials and energy within a system or process. In the context of Industrial Park, the material flow contains the exchange of materials between factories and within a single factory. Between factories, material flow could be materials transition from the output storage of one factory to the input storage of another. Within a factory, materials move from the input storage to the machines for processing. Once processed, the resulting output materials are routed to the output storage area, awaiting delivery to their respective customers.

When considering energy flow within the context of an industrial park, two distinct types can be identified. The first type involves energy carried along with material flow. For instance, when an output material possesses a higher temperature and another factory within the park requires specific temperature conditions for its inputs. The second type of energy flow pertains to the energy

generated by the power plant within the park. The energy produced by CHP is consumed by the various factories and members operating within the industrial park. It plays a crucial role in supporting the energy needs of the park's internal processes and activities.

Material and energy flow control refers to the management, regulation, and optimization of the movement of materials and energy within EIP. Material and energy flow control encompasses decision-making at various stages of material and energy flow, addressing questions of what, where, when, and how materials and energy move within a system. Typical decision-making aspects in material and energy flow control include:

- material and energy dispatching
- supplier selection
- alternative material choice
- inventory management.

These decisions are critical in optimizing the efficiency, sustainability, and cost-effectiveness of material and energy utilization within industrial processes.

2 Formal Model of material flows

2.1 Factory

Factories or industrial facilities are common elements within EIPs, and they can be integrated into an input-output model. Input-output modelling, having the advantage of tracing all primary inputs, wastes, by-products, and main product flows of a production unit, is an appropriate tool for designing industrial symbiosis networks [12].

Input and Output. To ensure the production of high-quality products that meet user requirements, factories often rely on specific production recipes [12]. These recipes are represented by input ratios, which indicate the quantity of raw materials needed to produce one unit of the main product. However, due to uncertainties in the production process or variations in raw material quality, these input ratios can exhibit stochastic behavior. Additionally, factories may have multiple options for each input. Equation (1) illustrates a scenario where a factory has three inputs, and for the first input, there are two alternative materials to choose from. The input ratio is denoted as 'r'.

$$InputAmount = productAmount \times [[r_{1a}, r_{1b}][r_{2a}][r_{3a}, r_{3b}, r_{3c}]] \quad (1)$$

Throughout the production process, input materials undergo a transformation, resulting in the generation of the main product, by-products, and waste materials. The output from each factory can be mathematically represented using the formula (2):

$$outputAmount = productAmount \times [1\ w_1\ w_2\ w_3]^T \quad (2)$$

where the vector 'w' represents the output ratios. These ratios specify the quantities of by-products or waste materials produced in conjunction with one unit of the primary product. Given the intricate nature of the production process, it is important to recognize that numerous uncertain factors come into play. Furthermore, occasional, random changes in output ratios may occur because of advancements in production technology.

Inventory and production process. Input and output materials are typically stored within the factory's warehouse, with each type of material having its dedicated inventory. Inventory management involves defining safety levels (L_s) and target levels (L_t) for each type of material. The purpose of these levels is twofold: 1) To prevent production disruptions caused by a shortage of raw materials, the factory initiates the purchase of raw materials when the current inventory level (l) reaches the safety level (L_s), with the aim of restoring it to the target level (L_t). 2) To prevent overproduction, if the inventory of the primary product reaches the target level (L_t), production within the factory is halted. Additionally, if the quantity of waste exceeds the available inventory capacity, any excess material is disposed of in a landfill.

In our production setup, each factory has a maximum production capacity denoted as C_m, representing the highest achievable monthly or yearly production volume. Factories typically operate at a target capacity level, denoted as C_t, where C_t is less than or equal to C_m. The actual capacity, denoted as C, may vary due to factors such as machine breakdowns, worker absences, and other operational fluctuations. Specifically, the actual capacity can change as follows: $C = C + \Delta$, where Δ represents either a capacity loss or gain. Production planning follows a level strategy, assuming customer demands (d) equal to the target capacity ($d=C_t$). A production plan specifies the quantity of products to be completed on a weekly or monthly basis, which then generates a material

requirement plan divided into daily or shift-based schedules. These schedules are transmitted to the shop floor as manufacturing orders. The shop floor accepts and schedules items within the manufacturing orders for processing. Individual machines are consolidated into a single machine, and the processing time (p) for one item is calculated as p = 1/C. If raw materials for a particular item are insufficient, the item enters a queue following the "first in, first out" (FIFO) rule, ensuring items are processed in the order they arrived.

2.2 Material flow

Material flows are governed by specific protocols that dictate when and how materials are moved within the system. Each type of material adheres to its own set of rules and procedures. This section outlines three types of material flow protocols: pushing, purchasing, and ordering. Purchasing and ordering fall under pulling protocols, involving requesting materials as needed, while pushing protocols supply materials without direct requests.

Before detailing material flow protocols, let's introduce the batch concept, which is widely employed by manufacturers to optimize production. In our model, batches take various forms. Production batches, defined in the production planning system, are subdivided into smaller process batches (manufacturing orders) on shop floors, each containing multiple jobs. Suppliers deliver products in delivery batches, while customers order in order/purchase batches, typically multiples of the delivery batch.

Pushing. In this protocol, the produced material is directly delivered to customers as soon as a delivery batch is assembled, as depicted in Figure 2. When one process batch is produced in one factory, the output inventories of this factory will be updated. Push events happen regularly in the factory to check if the amount of the output inventories is enough to form a delivery batch. If sufficient material is available, it's transported to the chosen customer. However, if no customers can accept the delivery batch due to capacity constraints, and the supplier's output inventory reaches its limit, excess material may be discarded if it's considered waste.

Purchasing and Ordering. Both purchasing and ordering are fundamental pulling protocols utilized in different manufacturing contexts. Purchasing is typically employed in make-to-stock factories, where materials are stored in warehouses until customers make purchases. On the other hand, ordering is prevalent in make-to-order factories, where customers initiate orders that suppliers then produce and deliver. In both protocols, regular events are triggered based on inventory levels. When the inventory reaches a safety level, actions are taken to restore it to the target level. Supplier selection policies are often employed to optimize decision-making when multiple suppliers are available. The order details are then transmitted to the selected supplier's planning system, and the production process begins. Ultimately, orders may be split into process batches, and delivery occurs once the required quantity is fully produced.

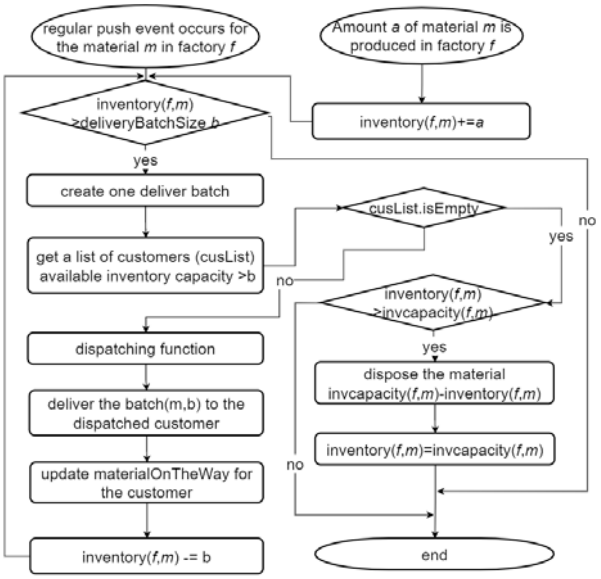


Figure 2: Pushing protocol of material flows

The choice of protocol depends on the type of materials involved. For instance, waste materials are typically not subject to purchase, and by-products cannot be ordered. The applicable protocols between external suppliers, customers, and factories are outlined in Table 1 below.

Material	From	To	Protocols
Products	Supplier	Factory	Purchasing
Products	Factory	Customer	Purchasing /Ordering
By-products	Factory	Customer	Purchasing
Waste	Factory	Disposal	pushing

Products	Factory	Factory	Purchasing /Ordering /Pushing
By-products	Factory	Factory	Purchasing /Pushing
Waste	Factory	Factory	Pushing

Table 1: Possible protocols between suppliers, factories, and customers

3 Formal model of energy flow in an EIP

In our study, we focus on energy flow which concerns the consumption of energy generated by the CHP system by the factories located within the park. It illustrates how the energy produced by the CHP system is efficiently utilized by the diverse factories operating within the park. This energy flow serves as the central and indispensable component of the park's overall energy dynamics. In the following section, we will provide an in-depth explanation of our approach to modeling this specific energy flow which can be broken down into three fundamental components: power station, power consumption, and power scheduling. We will elaborate on our approach from the perspective of each of these components.

3.1 Power station

The CHP power station element is modeled within a formal factory framework, distinguished primarily by its output type. Unlike traditional factories producing physical materials, the power station generates both heat and electricity. Its main input material comprises waste materials generated by other factories within the park. In instances where these waste materials are insufficient, the model seamlessly integrates external raw materials like coal to maintain uninterrupted operations. Additionally, our model includes parameters to configure the total number of available power generators and the capacity allocated to each individual generator within the power station.

3.2 Power consumption

Each factory's power requirement is closely tied to its production needs and is represented as a tuple (factory, start time, end time, requested power). This structured format signifies that within the specified time frame, the factory requires a specific supply of power, as quantified by the requested power value. In our study, power consumption is managed through the application of an area cutting algorithm, as depicted in Figure 3.

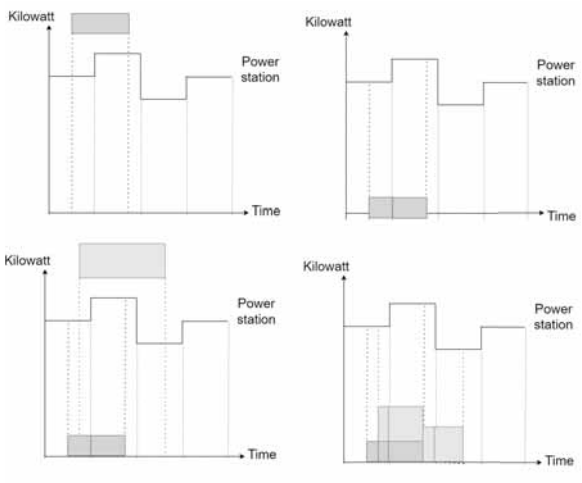


Figure 3: Area cutting algorithm

This algorithm operates by representing each power request as a rectangular area on a graph. The length of the rectangle corresponds to the duration of the power request, which is calculated based on its start and end times. Meanwhile, the width of the rectangle represents the amount of power requested. Concurrently, the power production of the station is graphically depicted as an upper black line over discrete time periods.

Upon initiation of a power request, the algorithm compares its time frame with the periods of power production at the station. This comparison determines whether the request falls within a single period or spans across multiple periods. To enhance accuracy, the periods of power production are further divided into smaller segments based on the start and end times of the request. For each power request, the algorithm updates the remaining power for the relevant segments from the start time to the end time of the request. This ensures precise tracking of power utilization during specific time intervals. Subsequent power requests undergo a similar process, where their time frames are compared with refined station periods from previous requests, and remaining power is updated accordingly.

3.3 Power scheduling

To optimize energy utilization, the implementation of power scheduling is a key strategy. This strategic approach encompasses determining the quantity of power

to be generated, and strategically coordinating how individuals or organizations utilize this power.

Our initial energy management strategy, as depicted in Figure 4, comprises several steps. Firstly, the number of generators to be activated is determined. Factories develop production plans over a specified time frame, typically a week or a month, while simultaneously creating power usage plans. These plans are transmitted to the center controller, which then calculates energy requests based on them, aggregating all requirements. The center controller formulates a power production plan, considering energy requests and the power station's capacity, including input material constraints and generator availability. This plan specifies how many generators should be activated for the upcoming period.

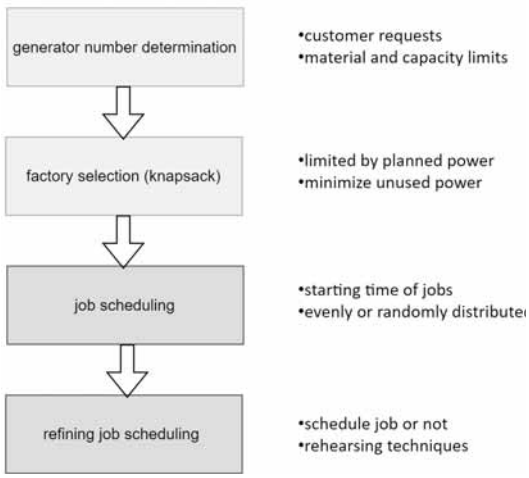


Figure 4: Energy scheduling strategy

Next, we select users in a way that minimizes unused power, similar to solving a knapsack problem with a limit on planned power. Following factory selection, we proceed to schedule jobs, determining the start time for each. This decision can be made by either evenly distributing the starting times or assigning them randomly throughout the planned period. The final step in this energy scheduling plan focuses on improving the schedule derived from earlier steps, a rehearsing technique is used here. A virtual power station is set up to check and improve the existing schedule. In this phase, jobs that are already scheduled request power according to the initial schedule. If there isn't enough power available to meet a job's power requirements, that specific job is excluded from the predetermined scheduling.

4 Implementation and application

In this section, we have used a virtual and simplified industrial park as an illustrative example to demonstrate our modeling approach. We made this choice due to our constraints in obtaining data from a specific real-world industrial park. We utilized simulation software, specifically AnyLogic, to create this example, which is depicted in Figure 5.

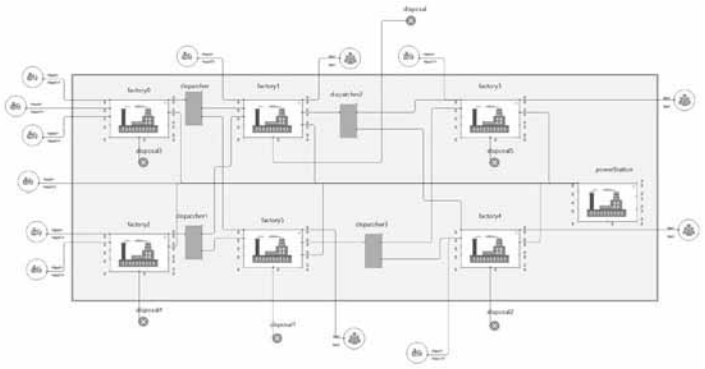


Figure 5: Instance of an EIP Created Using AnyLogic

In this example, Factories 1 and 5 serve as primary manufacturing facilities within the EIP, while the upstream two factories act as suppliers of raw materials to support the main factories. Factories 3 and 4 are responsible for managing the waste generated by the main factories. All waste from these factories is directed to a central power station within the park, which generates energy to support the park's members.

To provide context for this example, several assumptions were made: Material flows between factories follow a 'push' protocol. When multiple input sources are available, materials from upstream factories take precedence over those from external suppliers. Factories acquire materials from external suppliers through a purchasing process rather than traditional ordering. The target inventory level equals the inventory capacity. The safety inventory level is maintained at a quantity equivalent to a 30-day supply of throughputs. It's worth noting that while the power station utilizes recycled waste to generate energy, its output may not always meet the energy needs of all park members.

4.1 Experiment of material flow

The experiment on material flow aimed to establish an

efficient dispatching policy for scenarios where a single material is shared among multiple customer factories, common in industrial symbiosis networks. The application employed various dispatching policies:(1) Random dispatching. (2) Directing materials to factories with the highest available inventory. (3) Allocating materials based on the ratio of available inventory to capacity. (4) Assigning materials to factories maximizing main product output. (5) Allocating materials to minimize main product output. (6-9.) Allocation based on maximum requirements lookahead up to three planning periods.

The KPIs used to evaluate the different policies are divided into three categories: total waste disposal, total sales, and external requirements, which denote the economic and environmental objectives. "Total waste disposal" refers to the waste disposal across all factories within the park. Specifically, "Total Sale 1" represents the main product amounts that can be sold by factories 1 and 5, while "Total Sale 2" represents the main product outputs of factories 3 and 4. "External requirements" refer to the number of raw materials purchased by factories 1, 3, and 4 from external suppliers. The simulation results of these evaluations are presented in Figure 6.

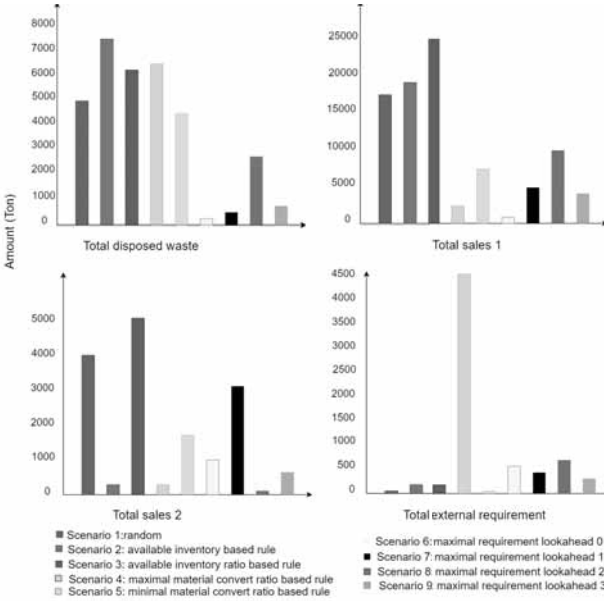


Figure 6: Experiment results of the material flow control

The simulation results emphasize the effectiveness of scenario 6, "maximal requirement lookahead 0," in minimizing waste disposal. Prioritizing factories with high demand for waste material reduces excess disposal. Conversely, scenario 2, the "available inventory-based rule," leads to higher waste disposal due to surplus allocation to plants with substantial available inventory. Regarding "total sales 1" and "total sales 2," scenario 3, the "available inventory ratio-based rule," proves most effective. This suggests that downstream factories can meet input material requirements, reducing production stoppages, enhancing throughput, and increasing sales. Scenario 4 performs worst for the total external requirement KPI. This is likely due to consistently directing waste materials to the factory with the highest material transformation ratio, forcing other factories to purchase external materials for production. Overall, the simulation provides valuable insights into how different policies impact waste disposal and sales performance.

4.2 Experiment of energy flow

In this experiment, the simulation duration spans 10 weeks, with each planning period structured as a one-week timeframe, totaling 10 planning periods. The objective was to optimize energy utilization in each planning period.

Plan period	Selected factories	Scheduled jobs ratio	Energy utilization
1	0,1,4,5	0.81	0.66
2	1,3,6	0.72	0.62
3	0,1,2,6	0.77	0.69
4	2,6	0.69	0.68
5	0,1,2,4,5,6	0.78	0.75
6	2,4,5	0.59	0.65
7	2,4,6	0.80	0.67
8	1,2,5,6	0.66	0.63
9	0,1,2,3,4,5,6	0.82	0.79
10	3,5,6	0.77	0.79

Table 2: Experiment results in each period

Table 2 presents the outcomes of the experiment for each planning period. It includes details such as the selected factories, indicating which factories were chosen to utilize the energy produced by the internal power station during the respective period. The scheduled jobs ratio represents the proportion of scheduled jobs compared to the total planned jobs across all factories. Additionally, the energy utilization ratio denotes the percentage of energy consumed by the scheduled jobs relative to the total energy generated by the power station.

5 Conclusion

In response to escalating environmental concerns, EIPs have emerged as innovative and sustainable solutions for industrial development. Given this context, effective management becomes imperative for these industrial parks. To pave the way for such management, a comprehensive understanding of the dynamic interactions and processes within EIPs is crucial. Material and energy flows are the primary representations of interactions among factories. Consequently, this paper employs discrete event simulation techniques to model the intricate material and energy flows within the park, providing a detailed account of our model's construction.

Through our modeling efforts, we aim to optimize the allocation and distribution of resources within these parks, with a specific emphasis on waste reduction and the maximization of energy utilization. The outcomes of our study can provide practical guidance for eco-industrial park management and policy development, ultimately contributing to a more sustainable and environmentally responsible industrial landscape. While our primary emphasis remains on existing operational parks, it is worth noting that our modeling approach can also be adapted to the planning phase of industrial parks, providing a means to evaluate the advantages of establishing industrial symbiosis.

Despite our comprehensive approach, it's important to note that, in this study, our management efforts are limited to a simple energy schedule. A more refined strategy is required. Therefore, our future research will specifically concentrate on energy scheduling, aiming to develop strategies that optimize energy utilization through production scheduling.

References

[1] Chertow, Marian R. (2000): I NDUSTRIAL SYMBIOSIS : Literature and Taxonomy. In Annu. Rev. Energy. Environ. 25 (1), pp. 313–337. DOI: 10.1146/annurev.energy.25.1.313.

[2] Afshari, Hamid; Farel, Romain; Peng, Qingjin (2018): Challenges of value creation in Eco-Industrial Parks (EIPs): A stakeholder perspective for optimizing energy exchanges. In Resources, Conservation and Recycling 139, pp. 315–325. DOI: 10.1016/j.resconrec.2018.09.002.

[3] Kuznetsova, E.; Zio, E.; Farel, Romain (2016): A methodological framework for Eco-Industrial Park design and optimization. In Journal of Cleaner Production 126, pp. 308–324. DOI: 10.1016/j.jclepro.2016.03.025.

[4] Leong, Yik Teeng; Lee, Jui-Yuan; Tan, Raymond R.; Foo, Ji Jinn; Chew, Irene Mei Leng (2017): Multi-objective optimization for resource network synthesis in eco-industrial parks using an integrated analytic hierarchy process. In Journal of Cleaner Production 143, pp. 1268–1283. DOI: 10.1016/j.jclepro.2016.11.147.

[5] Nuhu, Steven Kuba; Manan, Zainuddin Abdul; Wan Alwi, Sharifah Rafidah; Md Reba, Mohd Nadzri (2022): Integrated modelling approach for an eco-industrial park site selection. In Journal of Cleaner Production 368, p. 133141. DOI: 10.1016/j.jclepro.2022.133141.

[6] Boix, Marianne; Montastruc, Ludovic; Azzaro-Pantel, Catherine; Domenech, Serge (2015): Optimization methods applied to the design of eco-industrial parks: a literature review. In Journal of Cleaner Production 87, pp. 303–317. DOI: 10.1016/j.jclepro.2014.09.032.

[7] Sakr, D.; Baas, L.; El-Haggar, S.; Huisingh, D. (2011): Critical success and limiting factors for eco-industrial parks: global trends and Egyptian context. In Journal of Cleaner Production 19 (11), pp. 1158–1169. DOI: 10.1016/j.jclepro.2011.01.001.

[8] Long, Sha; Zhao, Lin; Liu, Hongbo; Li, Jingchen; Zhou, Xia; Liu, Yunfeng et al. (2019): A Monte Carlo-based integrated model to optimize the cost and pollution reduction in wastewater treatment processes in a typical comprehensive industrial park in China. In The Science of the total environment 647, pp. 1–10. DOI: 10.1016/j.scitotenv.2018.07.358.

[9] Fishman, George S. (2001): Discrete-event simulation: modeling, programming, and analysis: Springer.

[10] Cucchiella, Federica; D'Adamo, Idiano; Gastaldi, Massimo (2017): Sustainable waste management: Waste to energy plant as an alternative to landfill. In Energy Conversion and Management 131, pp. 18–31. DOI: 10.1016/j.enconman.2016.11.012.

[11] Touš, Michal; Pavlas, Martin; Putna, Ondřej; Stehlík, Petr; Crha, Lukáš (2015): Combined heat and power production planning in a waste-to-energy plant on a short-term basis. In Energy 90, pp. 137–147. DOI: 10.1016/j.energy.2015.05.077.

[12] Yazan, Devrim Murat; Fraccascia, Luca (2020): Sustainable operations of industrial symbiosis: an enterprise input-output model integrated by agent-based simulation. In International Journal of Production Research 58 (2), pp. 392–414. DOI: 10.1080/00207543.2019.1590660.

Theoretische Analyse eines Kompartiment-modells einer mückenübertragenen Krankheit mit mehreren Vogelarten

Christian Appel[1], Kurt Chudej[2,3*]

[1]Mathematisches Institut, Universität Bayreuth, 95440 Bayreuth, Germany
[2]Lehrstuhl für Wissenschaftliches Rechnen, Universität Bayreuth, 95440 Bayreuth, Germany
[3]Forschungszentrum für Modellierung und Simulation (MODUS), Universität Bayreuth, 95440 Bayreuth, Germany;*kurt.chudej@uni-bayreuth.de

Abstract. Wir untersuchen ein Kompartimentmodell einer mückenübertragenen Krankheit mit Kreuzinfektionen. Das Modell enthält verschiedene Vogelarten und eine Mückenart. Es wird gezeigt, dass es für $\mathscr{R}_0 > 1$ neben dem eindeutigen krankheitsfreien Gleichgewichtpunkt nur genau einen weiteren endemischen Gleichgewichtspunkt gibt. Zusätzlich werden Lyapunov-Funktionen für beide Equilibria unter Verwendung der Basisreproduktionszahl und deren gültige Bereiche vorgestellt.

Einleitung

Für vektorübertragene Krankheiten, wie dem West-Nil-, Usutu-, Dengue-, Chikungunya-Virus und vielen Weiteren, welche grundsätzlich in tropischen und subtropischen Arealen auftreten, besteht aufgrund des Klimawandels eine Tendenz und immanente Gefahr, dass diese innerhalb von Europa ausbrechen und katastrophale Epidemien auslösen können [1, 2, 3]. Die Infektion von gesunden Individuen ist hierbei hauptsächlich von der Anzahl infizierter Individuen einer anderen Spezies abhängig. Es spielen abhängig von der Krankheit Insekten (Vektoren) wie Moskitos, Zecken oder Flöhe eine fundamentale Rolle in der Infektionsübertragung und Verbreitung [4].
In diesem Paper wird aus diesem Grund ein generelles Kompartimentmodell einer vektorübertragenen Krankheit vorgestellt, welches, neben einer Vektorspezies, mehrere weitere Empfängerarten enthält. Der Übersicht halber werden hier stets Mücken den Vektor spielen und beliebig viele verschiedene Vogelarten die Infektionsträger, was ein mögliches Szenario der Verbreitung des Usutu- oder West-Nil-Virus beispielsweise sein könnte.
Für die zwei erwähnten Viren würde das konkret bedeuten, dass die Infektionsrate von Vögeln abhängig von der Anzahl an infizierten Mücken und die Infektionsrate der Mücken abhängig von der Anzahl der infizierten Vögeln ist. Die Möglichkeit der Unterscheidung von verschiedenen Vogelarten ist wichtig, weil einerseits verschiedene Spezies unterschiedlich anfällig sein können und auch unterschiedlich zur Verbreitung beitragen können. Eine sinnvolle Aufteilung innerhalb einer Anwendung wäre beispielsweise in anfällige heimische Vögel und stark verbreitende Zugvögel.

1 Modell

Jede einzelne Vogelart besteht im Wesentlichen aus einem klassischen SEIR-Modell [5], bei dem die Infektionsrate jedoch aus einem der jeweiligen Vogelart zugeordneten Infektionsparameter und der Anzahl der infizierten Mücken besteht.
Bei den Mücken wird dahingegen eine Aufteilung in Larven und erwachsene Mücken vorgenommen. Hierbei werden die erwachsenen Mücken ebenfalls durch ein angepasstes SEIR-Modell dargestellt, bei dem das R-Kompartiment weggelassen wird, weil diese aufgrund ihrer kurzen Lebensdauer bis an ihr Lebensende als infiziert angenommen werden können. Als Infektionsrate haben die Mücken eine Linearkombination der infizierten Kompartimente der Vögel. Daraus ergibt sich ein System von Differentialgleichungen mit nicht-negativen Anfangswerten und positiven Parametern.
Die Kompartimente E_k, I_k, E_v, I_v sind die infizierten Kompartimente, die Kompartimente I_k, I_v sind die infizierenden Kompartimente.

Anfangswertproblem (AWP):
Für $k \in \{1,\dots,n\}$:

$$\begin{aligned}
\dot{S}_k &= \Lambda_k - (\beta_k I_v + \mu_k) S_k && \text{mit } S_k(0) = S_{k_0} \\
\dot{E}_k &= \beta_k I_v S_k - (\gamma_k + \mu_k) E_k && \text{mit } I_k(0) = I_{k_0} \\
\dot{I}_k &= \gamma_k E_k - (\eta_k + \mu_k) I_k && \text{mit } E_k(0) = E_{k_0} \\
\dot{R}_k &= \eta_k I_k - \mu_k R_k && \text{mit } R_k(0) = R_{k_0} \\
\dot{L} &= \Lambda_L - (b_M + \mu_L) L && \text{mit } L(0) = L_0 \\
\dot{S}_v &= b_M L - (\sum_{k=1}^{n} c_k I_k + \mu_v) S_v && \text{mit } S_v(0) = S_{v_0} \\
\dot{E}_v &= \sum_{k=1}^{n} c_k I_k S_v - (\gamma_v + \mu_v) E_v && \text{mit } E_v(0) = E_{v_0} \\
\dot{I}_v &= \gamma_v E_v - \mu_v I_v && \text{mit } I_v(0) = I_{v_0}
\end{aligned}$$

Schnelle Berechnungen liefern, dass Lösungen des AWPs stets nichtnegativ bleiben (die Menge $\mathbb{R}^{4(n+1)}_{\geq 0}$ ist positiv invariant) [6, Kapitel 4], beschränkt sind (Betrachtung der Gesamtbevölkerung) und für alle Zeiten $t \geq 0$ existieren [7, Kapitel 4].
Zusätzlich kann die Basisreproduktionszahl $\mathscr{R}_0$ nach [8, 9] berechnet werden. Hierbei wird die Formel

$$\mathscr{R}_0 = \sqrt{\sum_{i=1}^{n} \frac{\beta_i c_i \gamma_i S_{i,DFE} \gamma_v S_{v,DFE}}{(\gamma_i + \mu_i)(\eta_i + \mu_i)(\gamma_v + \mu_v)(\mu_v)}}$$

erhalten, wobei mit dem tiefgestellten „DFE“ die Werte der jeweiligen Kompartimente im eindeutigen krankheitsfreien Gleichgewicht (DFE) gemeint sind, welches in Kapitel 2.1 weiter erläutert wird.

2 Gleichgewichtspunkte

Durch Nullsetzen der rechten Seite des Differentialgleichungssystems aus dem ersten Kapitel werden die Gleichungen

$$\begin{aligned}
0 = \quad & \Lambda_k - (\beta_k I_v^* + \mu_k) S_k^* && (1) \\
0 = \quad & \beta_k I_v^* S_k^* - (\gamma_k + \mu_k) E_k^* && (2) \\
0 = \quad & \gamma_k E_k^* - (\eta_k + \mu_k) I_k^* && (3) \\
0 = \quad & \eta_k I_k^* - \mu_k R_k^* && (4) \\
0 = \quad & \Lambda_L - (b_M + \mu_L) L^* && (5) \\
0 = \quad & b_M L^* - (\sum_{k=1}^{n} c_k I_k^* + \mu_v) S_v^* && (6) \\
0 = \quad & \sum_{k=1}^{n} c_k I_k^* S_v^* - (\gamma_v + \mu_v) E_v^* && (7) \\
0 = \quad & \gamma_v E_v^* - \mu_v I_v^* && (8)
\end{aligned}$$

erhalten. Gleichgewichtspunkte müssen diese Gleichungen alle erfüllen.
Aus (1), (5) und (6) ergibt sich für ein beliebiges Gleichgewicht $\hat{z}$ schnell:

$$\begin{aligned}
\hat{S}_k &= \frac{\Lambda_k}{\beta_k \hat{I}_v + \mu_k} \quad \forall k \in \{1,\dots,n\}, \\
\hat{L} &= \frac{\Lambda_L}{b_m + \mu_L}, \\
\hat{S}_v &= \frac{b_M \hat{L}}{\sum_{k=1}^{n} c_k \hat{I}_k + \mu_v}.
\end{aligned} \tag{9}$$

Wegen der Positivität der Parameter, der Betrachtung von Lösungen des Anfangswertproblems nur in der positiv invarianten Menge $\mathbb{R}^{4(n+1)}_{\geq 0}$ und der Beschränktheit von Lösungen sind die Kompartimente $\hat{S}_k, \hat{L}$ und $\hat{S}_v$ nach (9) echt positiv. Alle übrigen Kompartimente von Gleichgewichtspunkten in $\mathbb{R}^{4(n+1)}_{\geq 0}$ sind nichtnegativ. Das folgende Lemma liefert eine nutzvolle Folgerung für Gleichgewichtspunkte.
Lemma (Appel 2024):
Für ein beliebiges Gleichgewicht $\hat{z}$ des betrachteten Kompartimentmodells in $\mathbb{R}^{4(n+1)}_{\geq 0}$ gilt:

$$\begin{gathered}
(\exists l \in \{1,\dots,n\} : \hat{I}_l > 0 \vee \hat{E}_l > 0) \vee \hat{E}_v > 0 \vee \hat{I}_v > 0 \\
\Downarrow \\
\forall m \in \{1,\dots,n\} : (\hat{I}_m > 0 \wedge \hat{E}_m > 0) \wedge \hat{E}_v > 0 \wedge \hat{I}_v > 0
\end{gathered}$$

Beweis:
Aus (2), (3) und (8) erhält man

$$\begin{aligned}
\hat{I}_k \quad &\overset{(3)}{=} \quad \frac{\gamma_k}{\eta_k + \mu_k} \hat{E}_k \overset{(2)}{=} \frac{\gamma_k}{\eta_k + \mu_k} \frac{\beta_k \hat{S}_k}{\gamma_k + \mu_k} \hat{I}_v \overset{(8)}{=} \\
&\overset{(8)}{=} \quad \frac{\gamma_k}{\eta_k + \mu_k} \frac{\beta_k \hat{S}_k}{\gamma_k + \mu_k} \frac{\gamma_v}{\mu_v} \hat{E}_v
\end{aligned} \tag{10}$$

für ein beliebiges $k \in \{1,\dots,n\}$.
Da $0 < \hat{S}_k < \infty$ für alle $k \in \{1,\dots,n\}$ ist, ergibt sich mit (10) eine Kopplung zwischen allen infizierten Kompartimenten im Gleichgewicht. Nachdem dieser Zusammenhang für beliebige $k \in \{1,\dots,n\}$ gilt, folgt die obige Implikation. □
Das Lemma hält fest, dass sobald in einem Gleichgewicht $\hat{z}$ in $\mathbb{R}^{4(n+1)}_{\geq 0}$ ein beliebiges infiziertes Kompartiment positiv ist, müssen alle anderen infizierten Kompartimente ebenfalls positiv sein.

2.1 Krankheitsfreier Gleichgewichtspunkt

Definition 1:
Als krankheitsfreien Gleichgewichtspunkt (DFE) bezeichnet man einen Gleichgewichtspunkt, in dem die infizierten Kompartimente null sind. Also Lösungen z_{DFE} von (1)–(8), für welche $E_{k,DFE} = I_{k,DFE} = E_{v,DFE} = I_{v,DFE} = 0 \; \forall k$ gilt.
Satz 1:
Im betrachteten Kompartimentmodell existiert genau ein krankheitsfreier Gleichgewichtspunkt im $\mathbb{R}^{4(n+1)}_{\geq 0}$.
Beweis:
Die eindeutigen Werte der weiteren Kompartimente im krankheitsfreien Gleichgewicht können mit (4) und (9) sehr schnell bestimmt werden (reguläres lin. Gleichungssystem). □
Satz 2 (Appel 2024):
Für $\mathscr{R}_0 \leq 1$ ist der (eindeutige) DFE der einzige Gleichgewichtspunkt in $\mathbb{R}^{4(n+1)}_{\geq 0}$ für das betrachtete Kompartimentmodell.
Beweis:
Angenommen es gibt ein weiteres Gleichgewicht $\hat{z}$ in $\mathbb{R}^{4(n+1)}_{\geq 0}$, welches nicht der DFE ist. Wegen der Eindeutigkeit des DFEs und dem Lemma kann angenommen werden, dass $\hat{I}_v > 0$ und $\hat{I_m} > 0 \forall m \in \{1,\dots,n\}$ für das zusätzliche Gleichgewicht gelten muss. Nun wird mit (7) und (8) der Zusammenhang

$$\hat{I}_v \overset{(8)}{=} \frac{\gamma_v}{\mu_v}\hat{E}_v \overset{(7)}{=} \frac{\gamma_v}{\mu_v}\frac{\sum_{k=1}^{n} c_k \hat{I}_k \hat{S}_v}{\gamma_v + \mu_v}$$

gezeigt und dieser in die Teilgleichung von (2) und (3) aus (10) eingesetzt, was

$$\hat{I}_k = \frac{\gamma_k}{\eta_k + \mu_k}\frac{\beta_k \hat{S}_k}{\gamma_k + \mu_k}\frac{\gamma_v}{\mu_v}\frac{\sum_{l=1}^{n} c_l \hat{I}_l \hat{S}_v}{\gamma_v + \mu_v}.$$

ergibt. Zusammengefasst wurden also mithilfe von (1)–(8) zwei unabhängige Gleichungen hergeleitet, die einen Zusammenhang zwischen $\hat{I}_k$ und $\hat{I}_v$ herstellen, und zu dem eben erhaltenen Ausdruck zusammengesetzt. Dieser wird nun über alle $k \in \{1,\dots,n\}$ mit dem jeweils zugehörigen Koeffizienten c_k aufsummiert, was

$$\sum_{k=1}^{n} c_k \hat{I}_k = \sum_{k=1}^{n} c_k \frac{\gamma_k}{\eta_k + \mu_k}\frac{\beta_k \hat{S}_k}{\gamma_k + \mu_k}\frac{\gamma_v}{\mu_v}\frac{\sum_{l=1}^{n} c_l \hat{I}_l \hat{S}_v}{\gamma_v + \mu_v}$$

$$= \left(\sum_{k=1}^{n} \frac{\beta_k c_k \gamma_k \gamma_v \hat{S}_k \hat{S}_v}{(\gamma_k + \mu_k)(\eta_k + \mu_k)(\gamma_v + \mu_v)\mu_v}\right) \sum_{k=1}^{n} c_k \hat{I}_k$$

ergibt. Es muss also

$$\left(\sum_{k=1}^{n} \frac{\beta_k c_k \gamma_k \gamma_v \hat{S}_v \hat{S}_k}{(\gamma_k + \mu_k)(\eta_k + \mu_k)(\gamma_v + \mu_v)\mu_v}\right) = 1 \qquad (11)$$

für das Gleichgewicht $\hat{z}$ gelten.
Dies entspricht nahezu dem Quadrat der Basisreproduktionszahl $\mathscr{R}_0$. Die einzige Abweichung ist die Nutzung der nicht kranheitsfreien Gleichgewichtpunkte aus (9) und nicht des DFEs, wofür die nächsten zwei Gleichungen

$$\begin{aligned} S_{k,DFE} &= \frac{\Lambda_k}{\mu_k} = \frac{\beta_k \hat{I}_v + \mu_k}{\mu_k}\hat{S}_k \\ S_{v,DFE} &= \frac{b_M \Lambda_l}{(b_M + \mu_L)\mu_v} = \frac{\sum_{k=1}^{n} c_k \hat{I}_k + \mu_v}{\mu_v}\hat{S}_v \end{aligned} \qquad (12)$$

nun einen Zusammenhang liefern.
Abschließend werden die beiden Gleichungen (12) umgestellt und in (11) eingesetzt, womit dann mit Abschätzungen ein Widerspruch erlangt wird:

$$1 = \underbrace{\left(\sum_{k=1}^{n} \underbrace{\frac{\beta_k c_k \gamma_k \gamma_v S_{v,DFE} S_{k,DFE}}{(\gamma_k + \mu_k)(\eta_k + \mu_k)(\gamma_v + \mu_v)\mu_v}}_{>0} \underbrace{\frac{\mu_k}{\beta_k \hat{I}_v + \mu_k}}_{<1}\right)}_{<\mathscr{R}_0 \leq 1} \cdot \underbrace{\frac{\mu_v}{\sum_{k=1}^{n} c_k \hat{I}_k + \mu_v}}_{<1} \overset{\lightning}{<} 1.$$

2.2 Endemischer Gleichgewichtspunkt

Definition 2:
Ein vom DFE verschiedener Gleichgewichtspunkt wird in diesem Kompartimentmodell endemischer Gleichgewichtspunkt (EE) genannt.

In einem endemischen Gleichgewichtspunkt ist also mindestens ein infiziertes Kompartiment nicht null. Interessante endemische Gleichgewichtspunkte liegen in der positiv invarianten Menge $\mathbb{R}^{4(n+1)}_{\geq 0}$.
Frage:
Wieviele endemische Gleichgewichtspunkte existieren in $\mathbb{R}^{4(n+1)}_{\geq 0}$ abhängig von $\mathscr{R}_0$?

Satz 3 (Appel 2024):
Falls $\mathscr{R}_0 > 1$ gilt, existiert für das Modell nur genau ein endemisches Equilibrium (EE) in $\mathbb{R}^{4(n+1)}_{\geq 0}$.
Beweis:
Falls ein beliebiges endemisches Gleichgewicht z^* in $\mathbb{R}^{4(n+1)}_{\geq 0}$ existiert, müssen für dieses die Gleichungen $(1)-(8)$ erfüllt sein. Für den Beweis wird zunächst eine nur von I_v^* abhängige Gleichung konstruiert. Wegen dem Lemma, kann angenommen werden, dass alle infizierten Kompartimente im Gleichgewicht echt positiv sein müssen. Mit den Gleichungen (1)–(8) werden die zwei Gleichungen

$$\begin{aligned} I_k^* &\overset{(3)}{=} \frac{\gamma_k}{\eta_k+\mu_k} E_k^* \overset{(2)}{=} \frac{\gamma_k}{\eta_k+\mu_k}\frac{\beta_k I_v^* S_k^*}{\gamma_k+\mu_k} \overset{(9)}{=} \\ &\overset{(9)}{=} \frac{\gamma_k \Lambda_k}{(\eta_k+\mu_k)(\gamma_k+\mu_k)(1+\frac{\mu_k}{\beta_k I_v^*})} \end{aligned} \tag{13}$$

$$\begin{aligned} I_v^* &\overset{(8)}{=} \frac{\gamma_v}{\mu_v} E_v^* \overset{(7)}{=} \frac{\gamma_v}{\mu_v}\frac{\sum_{k=1}^{n} c_k I_k^* S_v^*}{\gamma_v+\mu_v} \overset{(9)}{=} \\ &\overset{(9)}{=} \frac{\gamma_v b_M \Lambda_L}{\mu_v(\gamma_v+\mu_v)(b_M+\mu_L)} \frac{1}{(1+\frac{\mu_v}{\sum_{k=1}^{n} c_k I_k^*})} \end{aligned} \tag{14}$$

hergeleitet, welche jeweils für ein beliebiges $k \in \{1,\dots,n\}$ den Wert I_k^* und I_v^* in Abhängigkeit zueinander setzen.
Durch Einfügen von (13) in (14) erhält man eine Gleichung, welche rein von I_v^* abhängig ist. Zuvor werden noch, der Übersichtlichkeit halber, Parameter in

$$a_k := \frac{\gamma_k \Lambda_k}{(\eta_k+\mu_k)(\gamma_k+\mu_k)}$$
$$b := \frac{\gamma_v b_M \Lambda_L}{\mu_v(\gamma_v+\mu_v)(b_M+\mu_L)}$$

zusammengefasst:

$$I_v^* \overset{(14)}{=} b\frac{1}{(1+\frac{\mu_v}{\sum_{k=1}^{n} c_k I_k^*})} \overset{(13)}{=} b\frac{1}{(1+\frac{\mu_v}{\sum_{k=1}^{n} c_k a_k \frac{1}{(1+\frac{\mu_k}{\beta_k I_v^*})}})} \iff$$

$$\mu_v = \underbrace{\sum_{k=1}^{n} c_k a_k \frac{b-I_v^*}{(I_v^*+\frac{\mu_k}{\beta_k})}}_{=:h(I_v^*)}. \tag{15}$$

Sei die Funktion $h : (lb,\infty) \to \mathbb{R}, u \mapsto h(u) = \sum_{k=1}^{n} c_k a_k \frac{b-u}{(u+\frac{\mu_k}{\beta_k})}$ mit $lb = -\min(\frac{\mu_1}{\beta_1},\dots,\frac{\mu_n}{\beta_n}) < 0$ definiert und wegen (15) ist jeder gültige Wert für $I_v^* > 0$ im Urbild von μ_v unter h. Nachdem die Funktion h für $u \in (lb,\infty)$ und damit für alle $u > 0$ definiert und stetig differenzierbar ist, kann die Ableitung

$$\frac{\partial}{\partial u} h(u) = -\sum_{k=1}^{n} c_k a_k \frac{\frac{\mu_k}{\beta_k}+b}{(u+\frac{\mu_k}{\beta_k})^2}$$

in diesem Bereich berechnet werden und es folgt sofort, dass sie für alle $u > 0$ negativ ist. Daher ist das Urbild von $\mu_v \in (0,\infty)$ höchstens einelementig, weshalb es maximal nur einen möglichen Wert für I_v^* gibt. Es gibt sogar exakt ein Urbild für μ_v unter h, weil

$$\begin{aligned} h(0) &= \sum_{k=1}^{n} c_k a_k \frac{b-0}{(0+\frac{\mu_k}{\beta_k})} = \sum_{k=1}^{n} c_k a_k \frac{b\beta_k}{\mu_k} \overset{(12)}{=} \\ &= \sum_{k=1}^{n} \frac{c_k \beta_k \gamma_v \gamma_k S_{v,DFE} S_{k,DFE}}{(\eta_k+\mu_k)(\gamma_k+\mu_k)(\gamma_v+\mu_v)} = \mathscr{R}_0^2 \mu_v > \mu_v \end{aligned}$$

gilt.
Damit existiert eine eindeutige Wahl von I_v^* für den endemischen Gleichgewichtspunkt. Aus der Gleichung (13) folgt sofort das Gleiche für I_k^* für beliebige $k \in \{1,\dots,n\}$. Mit den Gleichungen (1),(2),(4),(6) und (7) folgen, dass alle weiteren Kompartimente ebenfalls eindeutig gewählt werden müssen, was den Beweis vervollständigt. □
Satz 3a:
Für $\mathscr{R}_0 > 1$ liegt der (eindeutige) endemische Gleichgewichtspunkt (EE) aus $\mathbb{R}^{4(n+1)}_{\geq 0}$ sogar in $\mathbb{R}^{4(n+1)}_{>0}$.
Beweis:
Mit der eindeutigen Lösung für den EE in $\mathbb{R}^{4(n+1)}_{\geq 0}$ und dem Lemma und den Gleichungen (1)–(8). □
Eine geschlossene Darstellung der Komponenten des EEs in Abhängigkeit der Parameter ist leider nicht für eine allgemeine Anzahl n der Vogelarten möglich. Gleichung (15) kann zu einem Nullstellenproblem eines Polynoms vom Grad $n+1$ umgeformt werden. Eine (eindeutige) numerische Lösung des Polynoms in I_v^* oder auch von (15), jeweils zusammen mit $I_v^* > 0$, ist möglich.

3 Stabilität

Also existiert für $\mathscr{R}_0 \leq 1$ nur der DFE in $\mathbb{R}^{4(n+1)}_{\geq 0}$, welcher dann für $\mathscr{R}_0 < 1$ auch lokal asymptotisch stabil ist. Für $\mathscr{R}_0 > 1$ gibt es zusätzlich den EE in $\mathbb{R}^{4(n+1)}_{>0}$.

3.1 Krankheitsfreier Gleichgewichtspunkt

Von Z. Shuai und P. van den Driessche wurde in [10] eine Möglichkeit gezeigt, um eine Lyapunov-Funktion [6, Kapitel 5.5] auf einer positiv invarianten Menge Γ zu konstruieren. Hierbei wurde die Lyapunov-Funktion $Q(x) = \omega^T V^{-1} x$ hergenommen, wobei ω der Linkseigenvektors der Matrix $V^{-1}F$ zum Eigenwert $\mathscr{R}_0$ ist und die Matrizen F und V aus der Berechnung der Basisreproduktionszahl (z.B. in [9]) hergenommen wurden. Hierbei beschreibt x den Vektor der infizierten Kompartimente. Die Idee ist also zu zeigen, dass alle infizierten Kompartimente gegen Null gehen. Die in [10] gezeigte Variante zum Beweis der globalen asymptotischen Stabilität funktioniert jedoch nicht, weil die Matrix $V^{-1}F$ nicht irreduzibel ist und deshalb nur $\omega \geq 0$ (statt $\omega > 0$) gilt. Der nächste Satz demonstriert die Konstruktion einer strikten Lyapunov-Funktion für $\mathscr{R}_0 < 1$ auf Basis einer Erweiterung des Verfahrens aus [10].

Satz 4 (Appel 2024):
Sei $\mathscr{R}_0 < 1$. Es wird nun der Linkseigenvektor $\omega^T = \begin{pmatrix} 0 & \cdots & 0 & 0 & \frac{c_1 S_v \gamma_v}{\mu_v(\gamma_v+\mu_v)} & \cdots & \frac{c_n S_v \gamma_v}{\mu_v(\gamma_v+\mu_v)} & \mathscr{R}_0 \end{pmatrix}$ der Matrix $V^{-1}F$ zum Eigenwert $\mathscr{R}_0$ zu einem Vektor $\hat{\omega}$ erweitert, welcher wie folgt definiert wird.
Seien

$$\varphi_i := \frac{\gamma_i}{2(\eta_i+\mu_i)} \underbrace{\frac{c_i S_v \gamma_v}{\mu_v(\gamma_v+\mu_v)}}_{=:d_i} (1-\mathscr{R}_0) \quad \forall i = 1 \ldots n$$

$$\varphi_{n+1} := \frac{\gamma_v}{2\mu_v} \mathscr{R}_0 (1-\mathscr{R}_0)$$

$$\phi_i := -\varphi_i \frac{\eta_i+\mu_i}{\gamma_i} \quad \forall i = 1 \ldots n$$

$$\phi_{n+1} := -\varphi_{n+1} \frac{\mu_v}{\gamma_v}$$

$$v^T := \begin{pmatrix} \varphi_1 & \cdots & \varphi_n & \varphi_{n+1} & \phi_1 & \cdots & \phi_n & \phi_{n+1} \end{pmatrix}$$

und damit

$$\hat{\omega}^T := \omega^T + v^T =$$
$$= \begin{pmatrix} \varphi_1 & \cdots & \varphi_n & \varphi_{n+1} & \hat{d}_1 & \cdots & \hat{d}_n & \frac{\mathscr{R}_0}{2} + \frac{\mathscr{R}_0^2}{2} \end{pmatrix} > 0$$

mit $\hat{d}_i = (\frac{1}{2} + \frac{\mathscr{R}_0}{2}) d_i$ für alle $i \in \{1, \ldots, n\}$ gegeben. Dann gilt:

- Mit dem angepassten $\hat{\omega}$ ist $\hat{Q} = \hat{\omega}^T V^{-1} x$ eine Lyapunov-Funktion auf der positiv invarianten Menge $\Gamma := \{z \in \mathbb{R}^{4n+4} \mid \forall k \in \{1, \ldots, n\}, S_k + E_k + I_k + R_k \leq S_{k,DFE} \wedge L = L^* \wedge S_v + E_v + I_v \leq S_{v,DFE}\}$ und die Ableitung von $\hat{Q}$ ist entlang einer Lösungskurve nur gleich Null, wenn $x = 0$, also alle infizierten Kompartimente Null sind.
- Das DFE ist auf der Menge Γ global asymptotisch stabil.

Beweis:
Der Vektor v wurde derart konstruiert, dass $v^T V^{-1} F = 0$ gilt und $\hat{\omega} > 0$ ist. Sei $z(t)$ nun eine beliebige in Γ startende Lösungskurve des AWPs aus Kapitel 1:

$$\frac{d}{dt}(\hat{Q} \circ z)(t) = \hat{\omega}^T V^{-1} \dot{x} = \hat{\omega}^T V^{-1}((F-V)x - \hat{f}(x,y))$$
$$= \underbrace{\omega^T V^{-1} F}_{=\omega^T \mathscr{R}_0} x + \underbrace{v^T V^{-1} F}_{=0} x - \hat{\omega}^T x - \hat{\omega}^T V^{-1} \hat{f}(x,y) =$$
$$= \underbrace{(\mathscr{R}_0 \omega - \hat{\omega})^T}_{<0} x - \underbrace{\hat{\omega}^T V^{-1} \hat{f}(x,y)}_{\geq 0} \leq 0.$$

Also ist die Ableitung von $\hat{Q}(x) = \hat{\omega}^T V^{-1} x$ entlang einer Lösungskurve in Γ kleiner gleich Null. Nachdem Γ auch positiv invariant ist, ist $\hat{Q}$ eine Lyapunov-Funktion.
Für ein $x \neq 0$ ist die Ableitung echt kleiner Null, was den Beweis der ersten Aussage vervollständigt.
Für die auf der Menge Γ definierte Lyapunov-Funktion $\hat{Q}$ kann nun das Invarianzprinzip von LaSalle [11, Theorem 1] angewendet werden. Für die darin definierte Menge $E = \{z \in \Gamma \mid x = 0\}$ ergibt sich schnell die Menge aller krankheitsfreien Zustände in Γ. Nun wird noch die größte invariante Teilmenge M aus E benötigt. Offensichtlich bleiben jegliche krankheitsfreie Zustände krankheitsfrei und kombiniert mit der positiven Invarianz von Γ folgt die positive Invarianz der Menge $E = \{z \in \Gamma \mid x = 0\}$.
Lösungen des krankheitsfreien (und dadurch linearen) Differentialgleichungssystems können leicht analytisch berechnet werden, wodurch schnell gezeigt werden kann, dass nur der DFE in E negativ invariant ist, weshalb $M = \{DFE\}$ ist und nach LaSalles Invarianzprinzip die gesuchte globale asymptotische Stabilität des DFEs auf der Menge Γ gilt.

3.2 Endemischer Gleichgewichtspunkt

Nachdem nach [8, Kapitel 5, Theorem 1] für $\mathscr{R}_0 > 1$ das DFE ein instabiler Gleichgewichtspunkt ist, stellt sich nun die Frage über die Stabilität des eindeutigen endemischen Equilibriums.

Satz 5 (Appel 2024):
Im Fall $\mathscr{R}_0 > 1$ existiert für das EE z^* eine Lyapunov-Funktion W auf der Menge $\Gamma_{EE} := \mathbb{R}_{>0}^{4(n+1)}$ und das EE ist auf Γ_{EE} global asymptotisch stabil.
Beweis:
Für das EE wird mithilfe eines üblichen Standardansatzes wie zum Beispiel in [5, 12] eine Lyapunov-Funktion definiert und bewiesen. Sei

$$W(S_1,E_1,I_1,R_1,\dots,S_n,E_n,I_n,R_n,L,S_v,E_v,I_v) :=$$
$$\sum_{k=1}^{n}\Big[e_k(S_k - S_k^* - S_k^*\ln(\frac{S_k}{S_k^*})) + f_k(E_k - E_k^* - E_k^*\ln(\frac{E_k}{E_k^*}))+$$
$$+ g_k(I_k - I_k^* - I_k^*\ln(\frac{I_k}{I_k^*})) + h_k(R_k - R_k^* - R_k^*\ln(\frac{R_k}{R_k^*}))\Big]+$$
$$+ d(L - L^* - L^*\ln(\frac{L}{L^*})) + e(S_v - S_v^* - S_v^*\ln(\frac{S_v}{S_v^*}))+$$
$$+ f(E_v - E_v^* - E_v^*\ln(\frac{E_v}{E_v^*})) + g(I_v - I_v^* - I_v^*\ln(\frac{I_v}{I_v^*}))$$

ein Kandidat einer Lyapunov Funktion mit $e_k, f_k, g_k, h_k \geq 0\ \forall k \in \{1,\dots,n\},\ d,e,f,g \geq 0$ und dem EE z^*. Wegen Lemma 1 mit (9), (4) und (5) sind alle Kompartimente im EE echt größer Null. Kombiniert mit der Definitionsmenge Γ_{EE} sind alle Werte in den Logarithmusfunktionen stets positiv und deswegen ist W wohldefiniert.
Aus einer einfachen Beobachtung folgt, dass das endemische Equilibrium z^* das Minimum von Wist und W auf dem Bereich Γ_{EE} nichtnegativ ist. Nun wird die Nichtpositivität der zeitlichen Ableitung entlang einer beliebigen Lösungskurve $z(t)$, die in Γ_{EE} liegt, gezeigt:

$$\frac{d}{dt}W(z(t)) = \sum_{k=1}^{n}\Big[e_k(1-\frac{S_k^*}{S_k})\dot{S}_k + f_k(1-\frac{E_k^*}{E_k})\dot{E}_k+$$
$$+ g_k(1-\frac{I_k^*}{I_k})\dot{I}_k + h_k(1-\frac{R_k^*}{R_k})\dot{R}_k\Big]+$$
$$+ d(1-\frac{L^*}{L})\dot{L} + e(1-\frac{S_v^*}{S_v})\dot{S}_v+$$
$$+ f(1-\frac{E_v^*}{E_v})\dot{E}_v + g(1-\frac{I_v^*}{I_v})\dot{I}_v$$

Für eine bessere Übersichtlichkeit wird zuerst ein Summand der ersten Summe in einer Nebenrechnung untersucht. Sei zuvor $e_k = f_k$, $g_k = \frac{\gamma_k+\mu_k}{\gamma_k}e_k$ und $h_k = 0$, damit einige gemischte Terme in der folgenden Rechnung wegfallen. Das Weglassen der Kompartimente R_k aus der Lyapunov-Funktion ist problemlos möglich, weil die Differentialgleichungen dieser von allen anderen entkoppelt sind.
Nebenrechnung 1:

$$e_k(1-\frac{S_k^*}{S_k})(\Lambda_k - (\beta_k I_v + \mu_k)S_k)+$$
$$+ e_k(1-\frac{E_k^*}{E_k})(\beta_k I_v S_k - (\gamma_k+\mu_k)E_k)+$$
$$+ e_k\frac{\gamma_k+\mu_k}{\gamma_k}(1-\frac{I_k^*}{I_k})(\gamma_k E_k - (\eta_k+\mu_k)I_k) =$$
$$= e_k\Big[\Lambda_k - \mu_k S_k - \Lambda_k\frac{S_k^*}{S_k} + \beta_k I_v S_k^* + \mu_k S_k^* - \beta_k I_v S_k\frac{E_k^*}{E_k}+$$
$$+ (\gamma_k+\mu_k)E_k^* - \frac{\gamma_k+\mu_k}{\gamma_k}(\eta_k+\mu_k)I_k -$$
$$- (\gamma_k+\mu_k)E_k^*\frac{E_k}{E_k^*}\frac{I_k^*}{I_k} + \frac{\gamma_k+\mu_k}{\gamma_k}(\eta_k+\mu_k)I_k^*\Big] \overset{*}{=}$$

In der letzten Zeile wurde $E_k = E_k^*\frac{E_k}{E_k^*}$ einmal genutzt, was wegen Lemma 1 auch unproblematisch ist.
Nun werden die meisten Summanden, die von einem momentanen Kompartimentenwert abhängig sind, so umgestellt, dass mit angepassten Versionen der Gleichungen (1), (2) und (3) ein gemeinsamer Vorfaktor erhalten werden kann.

$$\overset{*}{=} e_k\Big[(\beta_k I_v^* + \mu_k)S_k^* - \mu_k S_k - (\beta_k I_v^* + \mu_k)S_k^*\frac{S_k^*}{S_k}+$$
$$+ \beta_k I_v^* S_k^*\frac{I_v}{I_v^*} + \mu_k S_k^* - \beta_k I_v^* S_k^*\frac{I_v}{I_v^*}\frac{S_k}{S_k^*}\frac{E_k^*}{E_k} + \beta_k I_v^* S_k^* -$$
$$- \beta_k I_v^* S_k^*\frac{I_k}{I_k^*} - \beta_k I_v^* S_k^*\frac{E_k}{E_k^*}\frac{I_k^*}{I_k} + \beta_k I_v^* S_k^*\Big] =$$
$$= e_k\Big[-\mu_k(\frac{(S_k^*-S_k)^2}{S_k}) + \beta_k I_v^* S_k^*(3 - \frac{S_k^*}{S_k} + \frac{I_v}{I_v^*} -$$
$$- \frac{I_v}{I_v^*}\frac{S_k}{S_k^*}\frac{E_k^*}{E_k} - \frac{I_k}{I_k^*} - \frac{E_k}{E_k^*}\frac{I_k^*}{I_k})\Big]$$

Damit ist die Untersuchung von diesem Summanden abgeschlossen und es wird der letzte Teil der Summe analog in einer verkürzten zweiten Nebenrechnung untersucht. Zuvor werden die Vorfaktoren günstig auf $d = \frac{b_M}{b_M+\mu_L}, e = f = 1$ und $g = \frac{\gamma_v+\mu_v}{\gamma_v}$ gesetzt.
Nebenrechnung 2:

$$\frac{b_M}{b_M+\mu_L}(1-\frac{L^*}{L})(\Lambda_L - (b_M+\mu_L)L) + (1-\frac{S_v^*}{S_v})(b_M L -$$
$$- (\sum_{k=1}^{n} c_k I_k + \mu_v)S_v) + (1-\frac{E_v^*}{E_v})(\sum_{k=1}^{n} c_k I_k S_v - (\gamma_v+\mu_v)E_v)+$$
$$+ \frac{\gamma_v+\mu_v}{\gamma_v}(1-\frac{I_v^*}{I_v})(\gamma_v E_v - \mu_v I_v) = \dots =$$

$$= \mu_v S_v^*(3 - \frac{L^*}{L} - \frac{L}{L^*}\frac{S_v^*}{S_v} - \frac{S_v}{S_v^*}) + \sum_{k=1}^{n} c_k I_k^* S_v^* \Big[4 + \frac{I_k}{I_k^*} - \\ - \frac{L^*}{L} - \frac{L}{L^*}\frac{S_v^*}{S_v} - \frac{I_k}{I_k^*}\frac{S_v}{S_v^*}\frac{E_v^*}{E_v} - \frac{I_v}{I_v^*} - \frac{E_v}{E_v^*}\frac{I_v^*}{I_v}\Big]$$

Mit der Wahl von $e_k = \frac{c_k I_k^* S_v^*}{\beta_k I_v^* S_k^*}$ ergibt die Kombination beider Ergebnisse:

$$\frac{d}{dt}W(z(t)) = \sum_{k=1}^{n}\Big\{ \underbrace{- e_k \mu_k \frac{(S_k^* - S_k)^2}{S_k}}_{\geq 0}\Big\} + \mu_v S_v^*(3 - \\ - \frac{L^*}{L} - \frac{L}{L^*}\frac{S_v^*}{S_v} - \frac{S_v}{S_v^*}) + \sum_{k=1}^{n}\Big\{ c_k I_k^* S_v^*(7 - \frac{S_k^*}{S_k} + \\ + \frac{I_v}{I_v^*} - \frac{I_v}{I_v^*}\frac{S_k}{S_k^*}\frac{E_k^*}{E_k} - \frac{I_k}{I_k^*} - \frac{E_k}{E_k^*}\frac{I_k^*}{I_k} - \frac{L^*}{L} - \\ - \frac{L}{L^*}\frac{S_v^*}{S_v} + \frac{I_k}{I_k^*} - \frac{I_k}{I_k^*}\frac{S_v}{S_v^*}\frac{E_v^*}{E_v} - \frac{I_v}{I_v^*} - \frac{E_v}{E_v^*}\frac{I_v^*}{I_v})] \leq \\ \leq \mu_v S_v^*(3 - \frac{L^*}{L} - \frac{L}{L^*}\frac{S_v^*}{S_v} - \frac{S_v}{S_v^*}) + \\ + \sum_{k=1}^{n}\Big\{ c_k I_k^* S_v^*(7 - \frac{S_k^*}{S_k} - \frac{I_v}{I_v^*}\frac{S_k}{S_k^*}\frac{E_k^*}{E_k} - \frac{E_k}{E_k^*}\frac{I_k^*}{I_k} - \\ - \frac{L^*}{L} - \frac{L}{L^*}\frac{S_v^*}{S_v} - \frac{I_k}{I_k^*}\frac{S_v}{S_v^*}\frac{E_v^*}{E_v} - \frac{E_v}{E_v^*}\frac{I_v^*}{I_v})\Big\} \leq 0.$$

Die letzte Ungleichung folgt aus der zweifachen Anwendung der AM-GM Ungleichung (z.B. [7, s.161]), welche gesondert den Ungleichungen

$$\frac{L^*}{L} + \frac{L}{L^*}\frac{S_v^*}{S_v} + \frac{S_v}{S_v^*} \geq 3\sqrt[3]{\frac{L^*}{L}\frac{L}{L^*}\frac{S_v^*}{S_v}\frac{S_v}{S_v^*}} = 3$$

$$\frac{S_k^*}{S_k} + \frac{I_v}{I_v^*}\frac{S_k}{S_k^*}\frac{E_k^*}{E_k} + \frac{E_k}{E_k^*}\frac{I_k^*}{I_k} + \frac{L^*}{L} + \frac{L}{L^*}\frac{S_v^*}{S_v} + \frac{I_k}{I_k^*}\frac{S_v}{S_v^*}\frac{E_v^*}{E_v} + \\ + \frac{E_v}{E_v^*}\frac{I_v^*}{I_v} \geq \\ \geq 7\sqrt[7]{\frac{S_k^*}{S_k}\frac{I_v}{I_v^*}\frac{S_k}{S_k^*}\frac{E_k^*}{E_k}\frac{E_k}{E_k^*}\frac{I_k^*}{I_k}\frac{L^*}{L}\frac{L}{L^*}\frac{S_v^*}{S_v}\frac{I_k}{I_k^*}\frac{S_v}{S_v^*}\frac{E_v^*}{E_v}\frac{E_v}{E_v^*}\frac{I_v^*}{I_v}} = 7$$

entspricht.

Damit die Funktion W eine Lyapunov-Funktion ist, muss nun noch sichergestellt werden, dass beliebige Lösungen, die in Γ_{EE} starten, diese Menge auch nicht verlassen. Sei nun z_0 ein beliebiger Startwert aus Γ_{EE}. Dadurch existiert $W(z_0) < \infty$ und es ist wohldefiniert. Nun muss gezeigt werden, dass die Lösung $z(t)$ mit Startwert z_0 nicht den Definitionsbereich von W verlässt. Dies würde genau dann passieren, wenn innerhalb der betrachteten Lösung $z(t)$ ein beliebiges Kompartiment gegen Null gehen würde. Als direkte Folgerung würde dann die Lyapunovfunktion W gegen unendlich gehen, was wegen der fallenden Monotonie der Lyapunov-Funktion entlang von Lösungen mit $W(z_0) < \infty$ nicht möglich ist.

Nachdem erneut keine strikte Lyapunov-Funktion gefunden wurde, muss LaSalles-Invarianzprinzip [11, Theorem 3] erneut angewendet werden.

Aus den Abschätzungen in der Berechnung der Monotonie der Ableitung von W entlang von Lösungen z und nach [7] gilt $\dot{W}(z(t)) = 0$ genau dann, wenn

$$\frac{L^*}{L} = \frac{L}{L^*}\frac{S_v^*}{S_v} = \frac{S_v}{S_v^*} = 1$$

$$\forall k \in \{1,\dots,n\}\ \frac{S_k^*}{S_k} = \frac{I_v}{I_v^*}\frac{S_k}{S_k^*}\frac{E_k^*}{E_k} = \frac{E_k}{E_k^*}\frac{I_k^*}{I_k} = \frac{L^*}{L} = \\ = \frac{L}{L^*}\frac{S_v^*}{S_v} = \frac{I_k}{I_k^*}\frac{S_v}{S_v^*}\frac{E_v^*}{E_v} = \frac{E_v}{E_v^*}\frac{I_v^*}{I_v} = 1$$

gilt. Aus der ersten Zeile folgt offensichtlich direkt $S_v = S_v^*$ und $L = L^*$. Die zweite Zeile wird zu

$$\forall k \in \{1,\dots,n\}: \quad \frac{I_v}{I_v^*}\frac{E_k^*}{E_k} = \frac{E_k}{E_k^*}\frac{I_k^*}{I_k} = \frac{I_k}{I_k^*}\frac{E_v^*}{E_v} = \frac{E_v}{E_v^*}\frac{I_v^*}{I_v} = 1 \tag{16}$$

vereinfacht. Alle Kompartimente außer R_k, die nicht mehr in dieser Gleichung auftauchen, entsprechen ihrem Gleichgewichtswert. O.B.d.A. kann wegen der Nichtnegativität der Lösungen $I_v = aI_v^*$ mit $a \in \mathbb{R}_{>0}$ gesetzt werden. Nun kann damit und den Gleichungen in (16) gefolgert werden, dass $E_k = aE_k^*\ \forall k \in \{1,\dots,n\}$ gelten muss, was mit den weiteren Gleichungen eine analoge Ausgangslage wie eben ergibt und woraus letztendlich die gesuchte Lösungsmenge

$$E = \{z \in \Gamma_{EE} \mid a \in \mathbb{R}_{>0}, z = (S_1^*, aE_1^*, aI_1^*, R_1, \dots, S_n^*, \\ aE_n^*, aI_n^*, R_n, L^*, S_v^*, aE_v^*, aI_v^*)\}$$

der Gleichungen gefolgert werden kann.

Sei nun $z(t)$ eine beliebige Lösung mit $z(t_0) \in E$. Für eine invariante Lösung der Menge E muss also für alle Zeiten $t \geq 0$ stets $S_v = S_v^*$ gelten. Also muss auch im Zeitpunkt Null die Differentialgleichung von S_v den Wert Null haben.

$$\dot{S_v}(0) = b_M L^* - (a(\sum_{k=0}^{n} c_k I_k^*) + \mu_v) S_v^* \stackrel{!}{=} 0$$

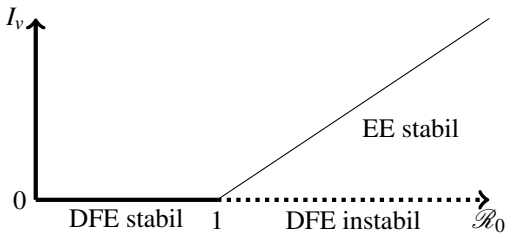
Dies ist eine von der Variable a abhängige affin-lineare Abbildung, welche deshalb nur eine Nullstelle besitzt. Für das endemische Gleichgewicht, also $a = 1$, gilt die Gleichung wegen (6) offensichtlich und für ein $a \neq 1$ daher nicht. Mit (4) muss zusätzlich auch $R_k = R_k^*$ sein. Damit ist $M = \{z \in E \mid a = 1 \wedge R_k = R_k^* \; \forall k \in \{1,\ldots,n\}\} = \{z^*\}$.
Weil jede Lösung beschränkt ist, kann [11, Theorem 3] in allen Fällen angewendet werden, woraus die asymptotische Konvergenz aller Lösungen, die in Γ_{EE} starten, gegen den endemischen Gleichgewichtspunkt folgt.

□

4 Bifurkation

Aus den bisherigen Ergebnissen wird deutlich, dass für $\mathscr{R}_0 < 1$ der (eindeutige) DFE auf Γ und für $\mathscr{R}_0 > 1$ der (eindeutige) EE auf Γ_{EE} global asymptotisch stabil sind.

Zusätzlich ist für $\mathscr{R}_0 > 1$ der DFE instabil. Also verändert sich das Lösungsverhalten an dem Punkt $\mathscr{R}_0 = 1$ fundamental. Dieser wird Bifurkationspunkt genannt und $\mathscr{R}_0$ ist hier der Bifurkationsparameter. Die Konvergenzeigenschaften der Gleichgewichtspunkte werden qualitativ in der Figur 1, welche ein Bifurkationsdiagramm zeigt, abhängig von der Basisreproduktionszahl zusammengefasst. Es liegt insbesondere eine forward Bifurkation vor.
Für die y-Achse wurde in dieser Abbildung das Kompartiment der infizierten infizierenden Mücken verwendet. Wegen des Lemmas hätte für die qualitative Zeichnung jedes beliebige infizierte Kompartiment des Modells hergenommen werden können.



Figur 1: Qualitative Graphik des Bifurkationsverhaltens.

Zusammenfassung

Es konnte das globale Stabilitätsverhalten in Abhängigkeit der Basisreproduktionszahl für ein Kompartimentmodell mit mehreren Vogelarten bei einer mückenübertragenen Krankheit bestimmt werden, obwohl die Koordinaten des (beweisbar eindeutigen) endemischen Gleichgewichtspunktes nicht immer explizit formelmässig angebbar sind.

References

[1] Semenza J, Paz S. Climate change and infectious disease in Europe: Impact, projection and adaptation. *The Lancet Regional Health - Europe*. 2021;9:100230.

[2] Fischer D, Thomas S, Suk JE, Sudre B, Hess A, Tjaden N, Beierkuhnlein C, Semenza JC. Climate change effects on Chikungunya transmission in Europe : Geospatial analysis of vector's climatic suitability and virus' temperature requirements. *International Journal of Health Geographics*. 2013;12.

[3] Rogers D, Suk J, Semenza J. Using global maps to predict the risk of dengue in Europe. *Acta Tropica*. 2014;129:1–4.

[4] Damialis A, Traidl-Hoffmann C, Treudler R. Climate Change and Pollen Allergies. In: *Biodiversity and Health in the Face of Climate Change*, edited by Marselle MR, Stadler J, Korn H, Irvine KN, Bonn A, pp. 47–66. Cham: Springer. 2019;.

[5] Martcheva M. *An Introduction to Mathematical Epidemiology*. New York: Springer. 2015.

[6] Prüss JW, Wilke M. *Gewöhnliche Differentialgleichungen und dynamische Systeme*. Cham: Springer. 2019.

[7] Königsberger K. *Analysis 2*. Berlin: Springer. 2004.

[8] van den Driessche P, Watmough J. Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission. *Mathematical Biosciences*. 2002;180(1):29–48.

[9] van den Driessche P, Watmough J. Further Notes on the Basic Reproduction Number. In: *Mathematical Epidemiology*, edited by Brauer F, van den Driessche P, Wu J, pp. 159–178. Berlin: Springer. 2008;.

[10] Shuai Z, Driessche P. Global Stability of Infectious Disease Models Using Lyapunov Functions. *SIAM Journal on Applied Mathematics*. 2013;73:1513–1532.

[11] LaSalle J. Some Extensions of Liapunov's Second Method. *IRE Transactions on Circuit Theory*. 1960; 7(4):520–527.

[12] Cangiotti N, Capolli M, Sensi M, Sottile S. A survey on Lyapunov functions for epidemic compartmental models. *Bollettino dell'Unione Matematica Italiana*. 2023;.

Recommendation Modeling for Health Self-Management Applications for People with Rheumatoid Arthritis

Nadine Schwab[1,2*], Günther Zauner[1], Christoph Urach[1], Paul Studenic[3,4], Helga Radner[3], Nasim Nakhost-Lotfi[3], Tanja Stamm[5], Thomas Hammer-Jakobsen[6], Andreas Dam[6,7], Niki Popper[1,2,8]

[1]dwh simulation services & technical solutions, dwh GmbH, Neustiftgasse 57-59, 1070 Vienna, AUSTRIA
[2]Institute for Information Systems Engineering, TU Wien, Favoritenstraße 9-11, 1040, Vienna, AUSTRIA
[3]Division of Rheumatology, Department of Internal Medicine III Medical University of Vienna, Vienna, AUSTRIA
[4]Division of Rheumatology, Department of Medicine (Solna), Karolinska Institutet, Stockholm, SWEDEN
[5]Center for Medical Statistics, Informatics, and Intelligent Systems, Medical University of Vienna, Vienna, AUSTRIA
[6]HealthBuddy, Vesterbrogade 149, DK-1620 Copenhagen, DENMARK
[7]Daman, Vesterbrogade 149, DK-1620 Copenhagen, DENMARK
[8]Institute of Statistics and Mathematical Methods in Economics, TU Wien, Wiedner Hauptstraße 8, 1040, Vienna, AUSTRIA
[*]*nadine.schwab@dwh.at*

Abstract. For the containment of chronic diseases, mHealth tools, for example mobile apps, provide great opportunities to track the disease progression and to give useful recommendations. In this project, possible data-driven enhancements for an already existing mobile app for patients with Rheumatoid Arthritis are discussed, developed and implemented. This happens in an ongoing feedback-cycle, including app developers, medical experts, patients and data scientists. The new features improve the app experience and are currently being evaluated in an observational study.

Introduction

Rheumatoid Arthritis (RA) is a chronic autoimmune disease, which affects over 3 Million people in Europe. People dealing with RA usually suffer from painful and tender joints. This and the whole plethora of also more subtle symptoms like fatigue strongly impacts their every day life [1], [2]. There is no cure for RA, but the symptoms and the progress of the disease can nowadays be halted effectively and improved by targeted pharmacological and non-pharmacological interventions. mHealth tools provide opportunities for the tracking and self-management of symptoms on a patient-individual basis [3].

RheumaBuddy is such an app that serves as an electronic diary for RA-specific symptoms: Users can track their symptoms, such as Mood, Pain, Fatigue and Stiffness, as well as activities such as walking or sleeping, on a daily basis. This helps keeping track of their disease progression, in general or for the next doctor appointment.

Throughout the project RheumaBuddy4.0 (RB4.0), it is the aim to improve this app. In order to achieve this, firstly, an extensive data analysis has been performed to figure out which additional features are possible and make sense in the current app. Next, these data-driven recommendations are implemented in the app and tested by the users.

The whole process takes place in a permanent feedback loop including the Danish app developers DAMAN, medical experts from the Medical University of Vienna (MUV), data scientists from dwh GmbH, and RA patients who use and test the app. Figure 1 shows a schematic depiction of the workflow in the project.

1 App Status at the Beginning of the Project

At the beginning of the project, RheumaBuddy is already available as a mobile app in multiple languages,

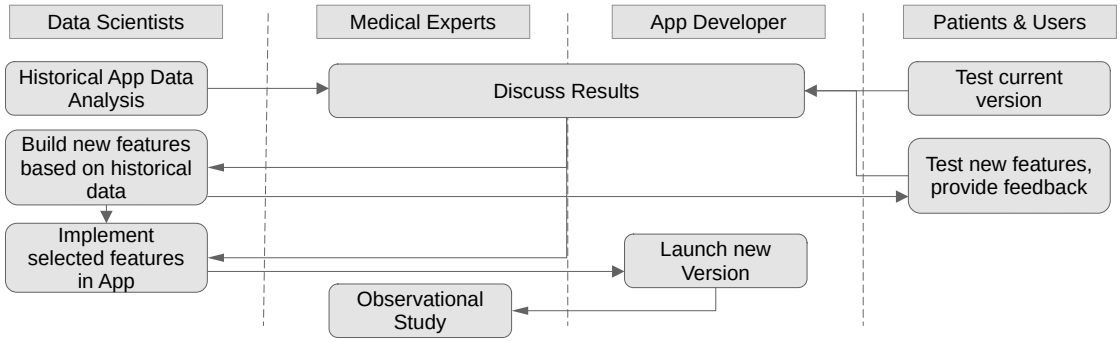


Figure 1: A schematic depiction of the general workflow of the project. The development of new features and the discussion with the experts is often repeated multiple times.

with German, English and Danish being the most popular used ones. Users have the possibility to enter RA-related symptoms, such as Mood, Pain, Fatigue and Stiffness as well as self-reported symptoms, where they could track whatever they are interested in. Every symptom can be scored on a smiley scale indicating the extent of the respective symptomatology. These smileys rank from very happy to very unhappy in five steps. Users can also enter activities, such as walking or sleeping, and how long these activities took on that day. Furthermore, they can document their joint pains on a body map. Finally, they can write free-text in a diary, as well as use the community to ask or answer to questions. The community is also available in different languages. The user's status and progression can always be seen on the Status page, where line graphs visualize the development of all the values tracked by the user.

The goal of the project is to implement a virtual coach within the app. This coach should provide users entering sufficient data into the app with personal recommendations based on their individual entries. This goal comes with the following challenges:

- **Amount of regular data:** In order to provide useful, data-based information, a sufficient amount of data per user is required, i.e. the users must enter data on a regular (ideally daily) basis over a few weeks, at the very least. Usually, it is difficult to motivate app users to provide this amount of input.
- **Subjectivity of app entries:** The app entries are subjective for every user, and it is not trivial to compare them. E.g., for one user, "good mood" means they actually had an amazing day, while for another user, it only means that everything is alright.
- **Not measurable disease parameters:** There are hardly any RA-specific parameters that can be measured objectively on a remote basis, i.e. without a medical expert. As opposed to e.g. Diabetes, where blood sugar can be objectively measured and provides a good indicator for the current disease activity, rheumatologists can only evaluate a patient's disease status by assessing the swollen, hurting and tender joints of a patient besides deviations in inflammatory parameters, blood levels, or changes in bone and joint structure with imaging modalities.

2 Data Analysis

An extensive data analysis was conducted in order to determine which app enhancements are possible with the available data, and how the results look like. The results were frequently discussed with the app developers and medical experts. Furthermore, feedback was constantly obtained from app users. Based on these feedback loops, possible app features are discussed and designed.

The data analysis with its results is presented in this chapter. The next chapter describes a new app feature, which evolved from this analysis.

2.1 User Behavior

First we analyzed, how extensively the app is used by the users. We analyzed the development of the number of unique users per day (which means they entered at least one value into the app on that day) and which

entry types are the most popular. The development of the users can be seen in Figure 2. A peak is visible in October 2020 as a result of a big marketing campaign. The number of users didn't stay that high, but remained significantly higher than before.

The most popular entry types can be seen in Figure 3. The most popular ones are the four main symptoms Mood, Pain, Fatigue and Stiffness, followed by the Pain Map. The diary and activity tracking is far less popular, as well as the self-reported symptoms. The latter shows that the four main symptoms seem to be sufficient for most app users. The least popular is writing and reading in the community.

Possible reasons for this heterogeneous behaviour were discussed with the consortium, especially with some app users. The users claimed that for those entries that are used the most often, it is much easier and faster to enter values than for the others. It goes without saying that writing into the diary or community is more time-consuming than choosing a smiley value for a symptom. The activity values, on the other hand, are not trivial to measure. Users claimed that it is hard to remember in the middle of the day, how long exactly they slept. Considering "Motion", it was not clear to them, what accounts to this, e.g. whether a walk counts already, or only exhausting physical activities, which is why they soon stopped entering values there.

2.2 Numeric Correlations

We investigated the correlations between the entries of the users. The entries are obtained directly from the app: As mentioned in section 1, a user can enter their symptoms via a smiley, which ranks from happy to unhappy in five steps. These steps are denoted in the backend with the integer values 1 to 5 (1 being the most unhappy one), which are used to compute correlations. Considering activity values (Sleep, Motion, Working hours), the actual time spans in seconds are used. These correlations are displayed in Figure 4. Strong correlations can be seen between the four main symptoms, but not between the rest of the entry types. However, no correlations between symptoms and activities might also be because the users had problems entering reasonable values for these fields, as discussed in Subsection 2.1.

2.3 Free-Text Analysis

Although these types of entries are less popular, we also analyzed the free-text entries, i.e. diary, community, and self-reported symptoms. We wanted to grasp what the users are writing about (besides the pre-defined entries), and whether some information that is relevant for all users can be extracted from the entries. To achieve this, we firstly translated all self-reported symptoms, diary and community entries to English language (using the Python Library Deep Translator), removed fill- and stop-words (using [7]) and counted which words appeared the most often in the entries. This is visualized in Figure 5.

As it can be seen immediately in the Subfigures 5b and 5a, the most important topics in the diary and in the community are RA-related symptoms or medication. Considering the self-reported symptoms in Subfigure 5c, interestingly, four of the five most frequent ones are exactly the four main symptoms that are provided in the app anyway, i.e. Mood, Pain, Fatigue and Stiffness. We assume, the users entered their data in the self-reported symptoms due to a misunderstanding - however, this indicates again that these symptoms seem to be of utmost importance. However, as the remaining symptoms (resp. the self-reported symptoms in general) are used by very few users, no further analysis was conducted in this area.

2.4 Community Crawling

The community is open for every RheumaBuddy-user, but as it was shown in Figure 3, there are only few users that use the community - even just for reading. Figure 5b indicates that many community posts and comments deal with RA-related symptoms and medication. As the community is built in a question-and-answer-style, i.e. a user can ask a question and any other user can answer to them, we assume that there are many recommendations hidden in the community. These recommendations would be of great value for many RheumaBuddy users, not only those who read and write in the community.

Therefore, we wanted to systematically collect helpful recommendations from the community. Depending on the information that could be gained, these could be forwarded to all users, or only to specific ones where it might fit (depending on the entries the users make). We collected recommendations using the following, semi-automated approach:

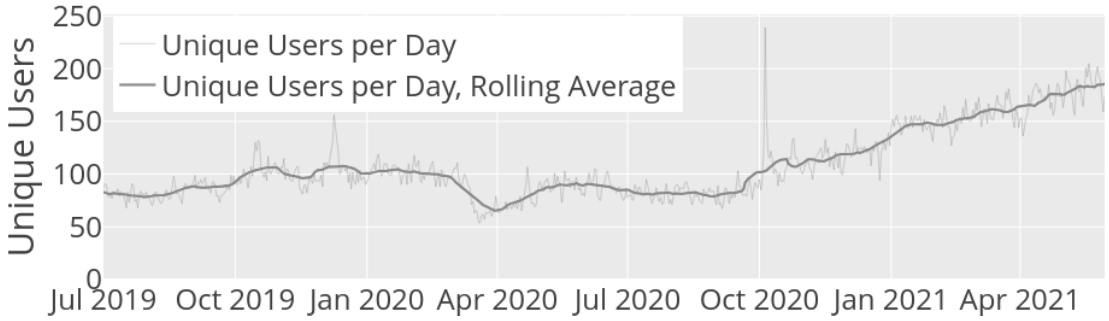


Figure 2: Plot of the development of the number of users per day. The thick line depicts a seven-day rolling average, the thin line the actual values. A peak can be seen as the result of a big marketing campaign, which led to a general increase of users per day.

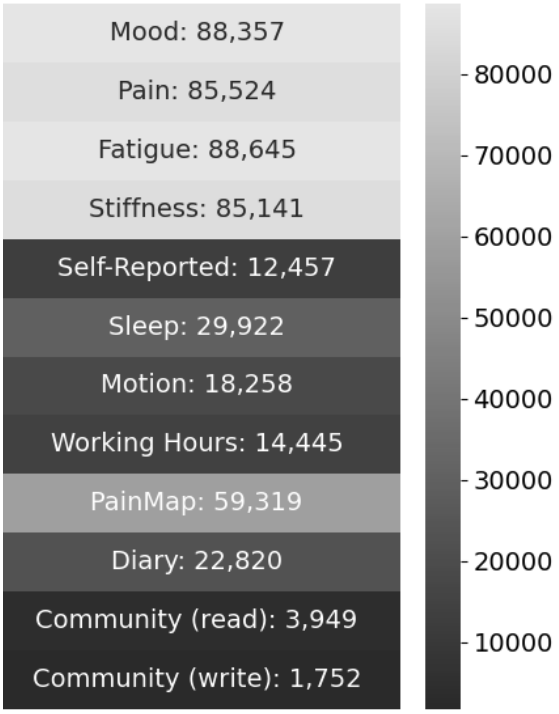


Figure 3: Depiction of the total count of entries for each entry type. The fast-to-enter symptoms as well as the pain map are the most popular entry types.

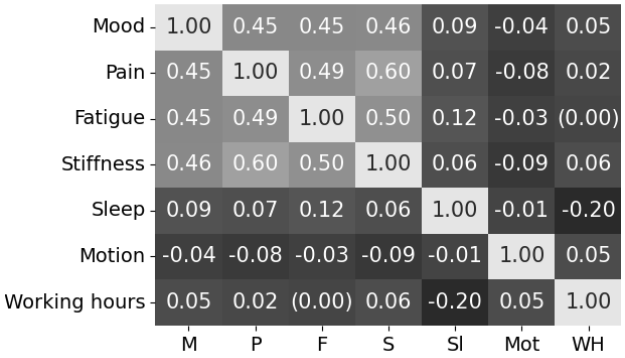
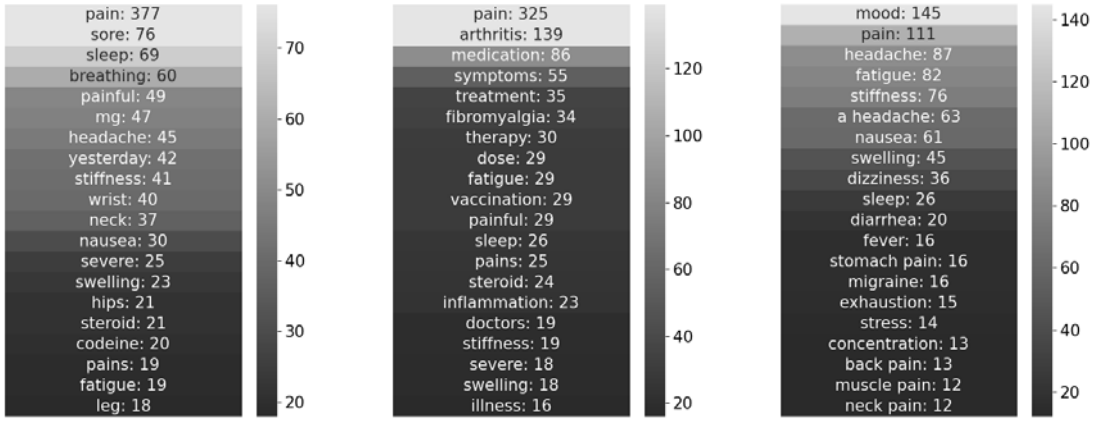


Figure 4: Linear, numeric correlations between the entry types based on 456,998 observations. Correlations between the four main symptoms are present, symptoms and activities seem to be uncorrelated.

1. Filter the community posts by posts which ask for advice on something, using regular expressions.
2. Retrieve all answers to this post and remove all answers from the user who asked the question.
3. Read the answers in combination with the question and decide whether to keep or to discard it.

A fully-automated approach was not developed within the scope of this project, as the Community Crawling was only experimental, and the amount of data available in the community is too small to justify much research in the field of automated text mining. Nevertheless, this semi-automated approach gravely reduced the manual work and enabled to quickly obtain a document with possible recommendations collected from the community.



(a) Most frequent words in the diary. (b) Most frequent words in the community. (c) Most frequent Self-Reported Symptoms.

Figure 5: Heatmaps for the most often used words in all kinds of free texts. The main topics for all sources are disease-related, i.e. symptoms and medication are the most popular.

However, the obtained recommendations cannot be used as they are as app recommendations. These recommendations are mainly tips from the users who usually do not have any medical background. Sharing what might help one user does not necessarily help another user, but could in the worst case actually harm them. And while it is fair to have recommendations in the community that might not be suitable for all users, app-provided recommendations must in any case be medically solid and do no harm to the users. Therefore, the recommendations obtained via community crawling cannot automatically be used in the app.

Still, the recommendations can be collected and further used. For example, they can be shared with medical experts which can assess on a patient-individual basis whether some of the recommendations could be helpful for some of their patients. Or specific medical studies could be started, proving (or disproving) some of the interesting recommendations. And finally, the developed method can be used in different contexts, where obtaining unsuitable recommendations is not that problematic.

2.5 User Clustering

Another idea was grouping similar users, based on their entries. Having such a group of users, the app could e.g. provide specific recommendations, based on the group characteristics. Users in such a group could also be brought together in an anonymous chat (if they consent) and share their experiences.

To achieve this, we measured the similarity between the main symptom entries (Mood, Pain, Fatigue and Stiffness) of the users. In order to recognize similar patterns even if they are shifted in time or slightly distorted, we measured the similarity using Dynamic Time Warping (DTW, [5]). We focused on one of the symptoms at a time and computed for all users with at least 10 entries the pairwise similarities between their values using DTW. Next, we grouped the users based on their pairwise distances. We tried different clustering algorithms using Scikit-Learn [6], finding that Spectral Clustering provided the most promising algorithm.

However, on the available app data, it was not possible to find useful clusters. Hardly any user provides daily entries, but rather one entry every few days. Furthermore, the values are discrete values between 1 and 5, which resulted in entry curves where no sensible pattern could be found. With the available values, it is not possible to recognize users that get steadily better or worse over some time. To achieve this, a scale where more precise entries are possible would be required.

3 User-individual Correlations

We figured that the users of the app might not only be interested in the development of their values, but also in the relationship between these values. As we showed in Subsection 2.2, significant correlations be-

tween the symptoms are present when considering all users at once.

To our surprise, no correlations between symptoms and activities could be found. However, during our consortium discussions, also including patient research partner, we agreed that relationships between activity patterns and symptoms should exist. Not finding any with the current approach could be due to several reasons:

- As discussed in Subsection 2.1, the users had problems entering reasonable values to the activity functions in the app, and as can be seen in Figure 3, the number of entries for activity values is much smaller.
- Users with positive and negative relationships could even each other out when being considered all at once.
- Also when it comes to one user, they don't necessarily have a *linear* relationship between their activities and symptoms, but rather some level of activity where they feel best/worst.

We decided to have a closer look at user-individual relationships between the values, and to not only take into account linear correlations. Additionally, the relationships between a user's individual input values is probably far more interesting to them than the correlations of all users. Therefore, we investigated on a user-individual basis, between which input values we could find the strongest significant relationships.

3.1 New Activity Values

First, we wanted to overcome the problem with entering reasonable activity values into the app. To achieve this, a step counter was added to the application. Of course, this does not cover the whole variety of physical activity (a step counter does not distinguish between walking and jogging, for example), but these values are added to the app automatically and provide a good approximation of the user's level of physical activity. Furthermore, the smiley values were enhanced by activity categories, enabling a user to provide input on a scale of 1 to 5, how physically active they were that day.

3.2 Model

We focus on the relationships between symptoms and activities per user, and consider three different kinds of relationships that could occur between the values:

- Linear relationships, e.g. "The more you walk, the better is your mood."
- Cut-Off values, e.g. "Your mood increases significantly when you walk around 6,000 steps per day."
- Extreme values, e.g. "Your mood is best when you walk around 6,000 steps per day."

These relationships are computed twice for each symptom-activity-pair, between same-day data as well as between activity data on one day and symptom data the next day. Thus, a possible output could also be "The more you walked *the day before*, the better is your mood." Additionally, every relationship can also be negative, e.g. "The *less* you walk, the better is your mood."

To achieve this, we firstly retrieved a user's recent data from their last 30 entry days. If they provided values for the considered symptom-activity-combination on at least on 10 different days (i.e. if we have at least 10 entry pairs), we fitted the user-data with three different models: A linear regression model (LR), a decision tree (DT), and a piece-wise linear function (PWLF).

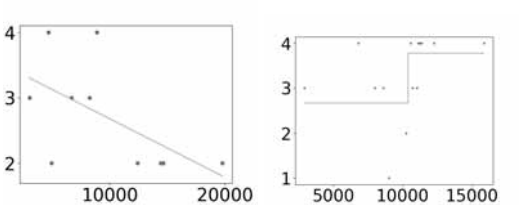
Linear Regression Model. We fitted the data with a linear regression curve of the form

$$y_i = a \cdot x_i + b + r_i$$

with y_i being the symptom values, x_i the activity values and r_i the residuals. a is the slope of the function, which is later used to determine if there is a positive or negative relationship between the two values.

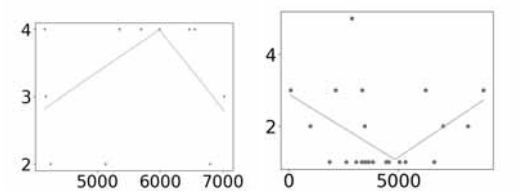
Decision Tree. We also fitted the data with a decision tree with one split node and constant values at the leaves.

Piece-wise linear function. If the user provided at least 15 value pairs in the given time frame, we also fitted the data with a continuous piece-wise linear function with one split, using the python library PWLF [8]. In order for this model to be valid in our context, the slopes of the two linear functions must have different signs, i.e. the break point must be a minimum or a maximum value. Additionally, the break point was only allowed within the inner 60% of the data range, and both



(a) A user with a negative linear correlation between Mood and Steps.

(b) A user with a cut-off value in the relationship between Mood and Steps.



(c) A user with a maximum value in the relationship between Mood and Steps.

(d) A user with a minimum value in the relationship between Mood and Steps.

Figure 6: Example plots of four users, depicting their individual relationships between number of steps and mood. The rather simple models still provide an interpretable relationship between the values.

linear functions must be supported by at least 5 data points. Otherwise, the PWLF would always fit the data better than the other two models without providing additional information.

Figure 6 shows some sample plots of the actual values of four different users, together with the model that fits their data. Note that for Subfigures 6c and 6d, only a very weak linear relationship would have been found. From each of these models, the R^2-Score is computed as a measure of fitting, and the model with the highest R^2 is kept. Thus, for every symptom-activity-combination, we compute one relationship. Finally, from all the symptom-activity-combinations, we choose the one with the highest R^2-Score and display it as an insight to the user.

3.3 Deployment

This feature was added into the RheumaBuddy app. It was deployed via a Docker Container [9] which runs on the server of the app provider DAMAN. Thus, the described relationships are computed in real-time and are communicated live to the users. A sample output from the app can be seen in Figure 7.

However, the number of entries required to obtain a significant relationship is quite high, and only few users entered enough values to obtain such a recommendation.

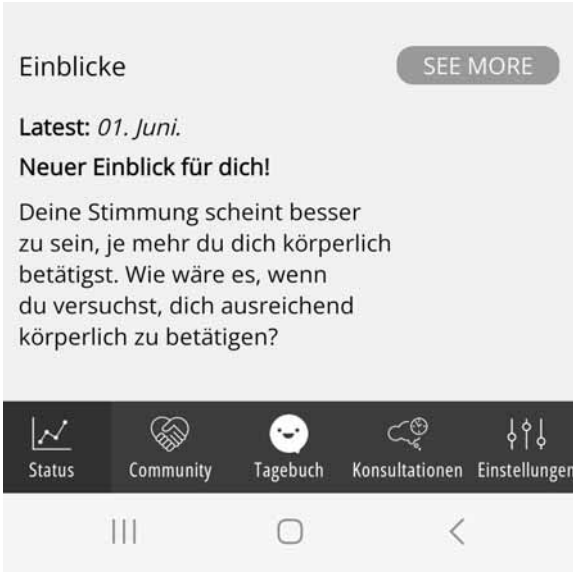


Figure 7: An actual recommendation (in German language) as provided from the RheumaBuddy app, stating that this user's mood seems to be better the more physically active they are.

4 Validation Process

The new app features are validated concerning face validity from the app developers. They check whether the app feedback makes sense and fits to the provided user input. Furthermore, they make sure that the insight is prompted to the users, and to how many. A selection of patients from their company review board tests the new features and provides thorough feedback on the functionality and the insights.

However, validation is at this point not complete and refinements would potentially be tackled in several iteration rounds. For putting RheumaBuddy4.0 on the track for CE marketing, certain aspects of a final software product would need to be submitted.

The validation process can be optimized by integrating results from the observational study at the Medical University of Vienna ("Successful Implementation of an RA management app - the IRAMA study" with ethical approval number: EK-Nr: 1846/2022) which is in process. The major aims of the IRAMA study concern the assessment of the RheumaBuddy mobile app

quality, using a standardized questionnaire to evaluate patients' experiences, preferences and needs for the use of the RheumaBuddy App in semi-structured telephone interviews, and finally to gain more detailed insight on patients' experiences, needs and preferences in semi-structured focus group sessions.

The data of this study is directly linked with a clinical registry and the app information the users are collecting. Thus, the foundation for a thorough validation via an observational study is already in place.

5 Discussion and Outlook

Overall, it can be stated that the project is progressing successfully. An interesting, helpful feature for the app users has been implemented, and we gained a lot of insights by means of the data analysis and the discussions in the consortium. We expect further interesting insights when the observational study is finalized.

One of the main challenges of the project was the lack of user input into the app. While the functions work well and can give useful insights to the users, hardly any users obtain such recommendations, because they stop entering values into the app too early. Some ways of Gamification to increase the motivation of the app users could be considered, e.g. counting how many days in a row they entered something into the app, and motivating not to break this "streak" - this has been shown to be helpful in many mobile apps [10].

The combination of passively tracked data with traditional or short PROMs as used in RheumaBuddy is potentially more promising to be supportive regarding improvement of the quality of life of people with RA when combined with the treating rheumatology team. As RA can affect people of any age, the background, i.e. experience with technology or eHealth literacy, is very heterogeneous, which makes it challenging to develop digitally assisted technologies that support every patient on an equal basis. Technology has been utterly fast paced over the past years, sharing breaking record news multiple times per year, but medical systems and people work and live on a slower pace. Thus, we would need to harmonise the slopes of development with the needs and opportunities that appear for implementation of novel technologies. This will bring benefit to people living with chronic conditions like RA as well as health care providers and system payers [4].

Acknowledgement

The project RB4.0 (FFG project number 880456) was funded by the Austrian Research Promotion Agency (FFG), the national funding agency for industrial research and development in Austria, as part of the call for proposals *Eurostars-2 CoD 13* (www.ffg.at/en).

References

[1] Symmons D, Turner G, Webb R, Asten P, Barrett E, Lunt M, Scott D, Silman A. *The prevalence of rheumatoid arthritis in the United Kingdom: new estimates for a new century.* Rheumatology (Oxford). 2002 Jul; 41(7):793-800

[2] Finckh A, Gilbert B, Hodkinson B, Bae S-C, Thomas R, Deane KD, Alpizar-Rodriguez D, Lauper K. *Global epidemiology of rheumatoid arthritis.* Nature Reviews Rheumatology. 2022 2022/09/06 doi:10.1038/s41584-022-00827-y.

[3] Winthrop KL, Mease P, Kerschbaumer A, Voll RE, Breedveld FC, Smolen JS, Gottenberg J-E, Baraliakos X, Kiener HP, Aletaha D, Isaacs JD, Buch MH, Crow MK, Kay J, Crofford L, Vollenhoven RFv, Ospelt C, Siebert S, Kloppenburg M, McInnes IB, Huizinga TW, Gravallese EM. *Unmet need in rheumatology: reports from the Advances in Targeted Therapies meeting, 2023.* Annals of the Rheumatic Diseases. 2023:ard-2023-224916 doi:10.1136/ard-2023-224916.

[4] De Cock D, Myasoedova E, Aletaha D, Studenic P. *Big data analyses and individual health profiling in the arena of rheumatic and musculoskeletal diseases (RMDs).* Ther Adv Musculoskelet Dis. 2022; 14:1759720X221105978 doi:10.1177/1759720X221105978

[5] Springer Berlin Heidelberg. Dynamic Time Warping. In: *Dynamic Time Warping*. 2007. p 69–84.

[6] *Scikit-learn: Machine Learning in Python*, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[7] Bird S., Klein E., Loper E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc."

[8] Jekel Charles, Venter Gerhard. (2019). *pwlf: A Python Library for Fitting 1D Continuous Piecewise Linear Functions.* 10.13140/RG.2.2.28530.56007.

[9] Cito Jürgen, Ferme Vincenzo, Gall Harald. (2016). *Using Docker Containers to Improve Reproducibility in Software and Web Engineering Research.* 609-612. 10.1007/978-3-319-38791-8_58.

[10] Galetta, Giuseppe. (2013). *The Gamification: Applications and Developments for Creativity and Education.* 10.13140/RG.2.2.24817.68965.

Simulationsgestützte Analyse der neuen dynamischen Strompreise und abgeleitete Empfehlungen zur Abgabenpolitik

Thomas Wiedemann[1*]

[1] HTW Dresden, Fak. Informatik/Math., F.-List-Platz 1, 01069 Dresden, *wiedem@informatik.htw-dresden.de

Abstract. Die seit 2024 verfügbaren dynamischen Strompreise werden durch die gleichzeitig gültige Gesetzgebung weitgehend negiert, da fast immer 20 oder mehr €-Cent an konstanten Abgaben anfallen. Auf der Basis von stündlichen dynamischen Strompreisen seit Anfang 2023 wurden verschiedene Simulationsszenarien zu Marktpreis- und Abgaben-Verteilungs-Alternativen modelliert und berechnet, welche höhere Anreize für Endkunden bezüglich des Speicherausbaus bewirken könnten. Natürlich würde eine reale Umsetzung eine Novellierung der Gesetze und Abgabenordnungen für Energieversorger erfordern. Ob dies im gegenwärtigen politischen als auch ökonomischen Umfeld möglich ist, wird aktuell gemeinsam mit Energiemarkt-Experten diskutiert.

Einführung

Seit dem 1.1.2024 sind große Energieversorger gesetzlich verpflichtet, Endkunden optional dynamische Strompreise anzubieten [1]. Ab 1.1.2025 gilt dies für alle Energieversorger. Diese dynamischen Strompreise werden von der Leipziger Strombörse abgebildet. Je nach Marktnachfrage und Angebot an regenerativen Energieformen schwanken die Netto-Preise zwischen negativen Werten von -10 Cent/KWh bis zu positiven Werten von 40 Cent/KWh (vgl. Abb. 1). Leider wird dieser sehr interessante neue Ansatz der Politik durch die derzeitig gültige Gesetzgebung zu Strommarktabgaben wie Netzentgelte, Konzessionsabgaben und Steuern wieder weitgehend negiert, da mindestens 20 oder mehr €-Cent an konstanten Abgaben anfallen. Bei 1 Cent Strombörsen-Netto-Preis liegt der Bruttopreis bei 21 Cent und damit die Abgabenlast bei über 2000%.

In der Folge ist zu vermuten, dass dynamische Strompreise für Kunden ohne eigenen Stromspeicher im Haus kaum attraktiv sind. Fraglich ist auch, ob bei Vorhandensein eines Stromspeichers die Einsparungen über das Jahr betrachtet, ausreichen werden, um nach einer Einsatzzeit von 10...20 Jahren den Speicher zu refinanzieren. Heutige Stromspeicher auf Lithium-Ionen-Basis sind laut deren Datenblätter nach 10 bis maximal 20 Jahren am Lebensende und müssen ersetzt werden. Weiterhin zeigen auch die zunehmend häufiger zu beobachtenden negativen Strompreise eine klare Dysfunktionalität der Strom-Marktmechanismen an. So ist in den Mittagsstunden das Angebot häufig deutlich höher als die Nachfrage. Eine genaue Datenanalyse und darauf aufsetzende Simulationen von Lösungsoptionen sollen dies genauer evaluieren.

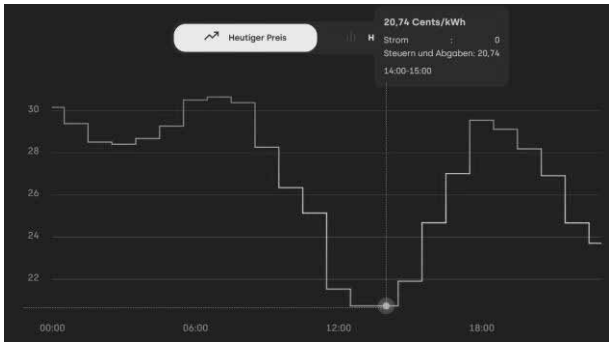


Abb. 1: Dynamische Strompreise am 28.3.24 (Quelle: www.tibber.de am 28.3.24 [2])

1 Erste Datenanalyse

Dank der OpenData-Initiative existieren im Internet sehr viele Datenquellen zum Strompreismarkt. Die vom Fraunhofer-Institut für Solare Energiesysteme ISE [3] gepflegte Website energy-charts.info bietet auf einer Unterseite [4] verschiedene Datenschnittstellen im OAS3-Format [5] an, u.a. auch **historische Börsen-Strommarktpreise [6]**.

Für die nachfolgend beschriebenen Simulationsläufe wurden nur die Daten seit dem 1.1.2023 verwendet, da die Corona-Pandemie und der Ukrainekrieg zu extremen Preisausschlägen nach unten und oben geführt haben.

Gleichwohl sind beide Ereignisse sehr wertvoll als Worst-Case-Szenarien in beiden Richtungen.

Der „Day-Ahead“ -Strompreis der Leipziger Börse wird jeweils 24 Stunden vorher festgelegt und ist online abrufbar. Damit entfallen für den Endabnehmer komplexe Prognosemodelle oder Wetteranalysen, da diese in den 24h-Preis-Vorabdaten inkludiert sind. Mit einer Minimalwert-Suche über den Preisen der nächsten 24 Stunden kann jeder Endkunde oder dessen (KI-) Steuerung leicht den besten Zeitpunkt zum Abruf von günstigem Strom für den nächsten Tag ermitteln.

1.1 Datenimport und -anreicherung der historischen dynamischen Strompreise

Nach dem Export aus [6] wurden die Daten im JSON-Format in eine Access-Datenbank-Tabelle eingelesen. Ergänzt wurden diese Rohdaten um die Stunden-, Monats- und Wochentageswerte, da der zeitliche Preisverlauf mit diesen stark korreliert. (vgl. Abb. 1). Diese Anreicherung diente vor allen der manuellen Analyse und ersten Begutachtung der Werteverläufe.
Am Wochenende führt die fehlende Industrienachfrage meist zu geringeren oder negativen Preisen. Für ca. 1,5 Jahre umfassen diese Rohdaten (vgl. Abb. 2) ca. 10.000 Datensätze und können für stundengenaue Analysen der dynamischen Strompreise verwendet werden.

Preisdaten Rohdaten

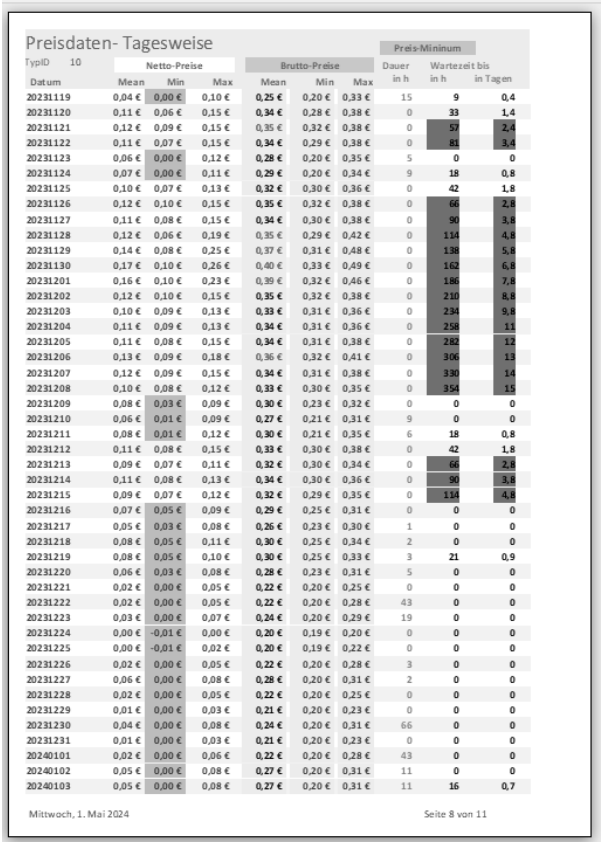
DatumZeitID	Monat	WoTag	Std.	Preis-Netto	Preis-Brutto
2023022010	2	1	10	0,02 €	0,22 €
2023022011	2	1	11	0,02 €	0,22 €
2023022012	2	1	12	0,02 €	0,22 €
2023022013	2	1	13	0,04 €	0,24 €
2023022014	2	1	14	0,05 €	0,28 €
2023022015	2	1	15	0,06 €	0,28 €

Abb. 2: Auszug aus den Rohdaten (hier vom 20.2.2023 im Zeitraum 10-15 Uhr)

1.2 Tagesweise Verdichtung der Daten

Nach dem Import der Basisdaten wird automatisch eine erste Verdichtung auf Tageswerte vorgenommen. Dies sind jeweils der Mittelwert, das Minimum und das Maximum der Netto- und Bruttopreise des jeweiligen Tages. Weiterhin wird die Zeitspanne von günstigem Strom berechnet, der zum maximalen Stromabruf für Endverbraucher und zum Laden vorhandener Speicher genutzt werden kann.

In der Datenbank ist dazu ein Grenzwert – im Beispiel bei Abb. 3 ein Wert von 0,05 € angenommen. Dieser Wert ist beliebig änderbar und kann je nach Endkunden oder Dinglichkeit des Strombedarfs unterschiedlich sein. Falls ein Kunde stark sparen will setzt er den Wert z.B. auf 0,00 €, falls auch Strompreise bis 0,07 noch akzeptabel sind, dann eben auf 0,07 €. Ab diesem Grenzwert wird der Strom als günstig (grün markiert im Report) zum Laden des Speichers definiert und gleichzeitig auch die Wartezeit bis zu einer Phase günstigen Stroms ermittelt. Im Beispiel werden Wartezeiten auf günstigen Strom größer als 48 Stunden als kritisch (rot im Report) markiert (vgl. Abb. 3).



Preisdaten- Tagesweise

TypID 10

Datum	Netto-Preise Mean	Min	Max	Brutto-Preise Mean	Min	Max	Preis-Mininum Dauer in h	Wartezeit bis in h	in Tagen
20231119	0,04 €	0,00 €	0,10 €	0,25 €	0,20 €	0,33 €	15	9	0,4
20231120	0,11 €	0,06 €	0,15 €	0,34 €	0,28 €	0,38 €	0	33	1,4
20231121	0,12 €	0,09 €	0,15 €	0,35 €	0,32 €	0,38 €	0	57	2,4
20231122	0,11 €	0,07 €	0,15 €	0,34 €	0,29 €	0,38 €	0	81	3,4
20231123	0,06 €	0,00 €	0,12 €	0,28 €	0,20 €	0,35 €	5	0	0
20231124	0,07 €	0,00 €	0,11 €	0,29 €	0,20 €	0,34 €	9	18	0,8
20231125	0,10 €	0,07 €	0,13 €	0,32 €	0,30 €	0,36 €	0	42	1,8
20231126	0,12 €	0,10 €	0,15 €	0,35 €	0,32 €	0,38 €	0	66	2,8
20231127	0,11 €	0,08 €	0,15 €	0,34 €	0,30 €	0,38 €	0	90	3,8
20231128	0,12 €	0,06 €	0,19 €	0,35 €	0,29 €	0,42 €	0	114	4,8
20231129	0,14 €	0,08 €	0,25 €	0,37 €	0,31 €	0,48 €	0	138	5,8
20231130	0,17 €	0,10 €	0,26 €	0,40 €	0,33 €	0,49 €	0	162	6,8
20231201	0,16 €	0,10 €	0,23 €	0,39 €	0,32 €	0,46 €	0	186	7,8
20231202	0,12 €	0,10 €	0,15 €	0,35 €	0,32 €	0,38 €	0	210	8,8
20231203	0,10 €	0,09 €	0,13 €	0,33 €	0,31 €	0,36 €	0	234	9,8
20231204	0,11 €	0,09 €	0,13 €	0,34 €	0,31 €	0,36 €	0	258	11
20231205	0,11 €	0,08 €	0,15 €	0,34 €	0,31 €	0,38 €	0	282	12
20231206	0,13 €	0,09 €	0,18 €	0,36 €	0,32 €	0,41 €	0	306	13
20231207	0,12 €	0,09 €	0,15 €	0,34 €	0,31 €	0,38 €	0	330	14
20231208	0,10 €	0,08 €	0,12 €	0,33 €	0,30 €	0,35 €	0	354	15
20231209	0,08 €	0,03 €	0,09 €	0,30 €	0,23 €	0,32 €	0	0	0
20231210	0,06 €	0,01 €	0,09 €	0,27 €	0,21 €	0,31 €	9	0	0
20231211	0,08 €	0,01 €	0,12 €	0,30 €	0,21 €	0,35 €	6	18	0,8
20231212	0,11 €	0,08 €	0,15 €	0,33 €	0,30 €	0,38 €	0	42	1,8
20231213	0,09 €	0,07 €	0,11 €	0,32 €	0,30 €	0,34 €	0	66	2,8
20231214	0,11 €	0,08 €	0,13 €	0,34 €	0,30 €	0,36 €	0	90	3,8
20231215	0,09 €	0,07 €	0,12 €	0,32 €	0,29 €	0,35 €	0	114	4,8
20231216	0,07 €	0,05 €	0,09 €	0,29 €	0,25 €	0,31 €	0	0	0
20231217	0,05 €	0,03 €	0,08 €	0,26 €	0,23 €	0,30 €	1	0	0
20231218	0,08 €	0,05 €	0,11 €	0,30 €	0,25 €	0,34 €	2	0	0
20231219	0,08 €	0,05 €	0,10 €	0,30 €	0,25 €	0,33 €	3	21	0,9
20231220	0,06 €	0,03 €	0,08 €	0,28 €	0,23 €	0,31 €	5	0	0
20231221	0,02 €	0,00 €	0,05 €	0,22 €	0,20 €	0,25 €	0	0	0
20231222	0,02 €	0,00 €	0,05 €	0,22 €	0,20 €	0,28 €	43	0	0
20231223	0,03 €	0,00 €	0,07 €	0,24 €	0,20 €	0,29 €	19	0	0
20231224	0,00 €	-0,01 €	0,00 €	0,20 €	0,19 €	0,20 €	0	0	0
20231225	0,00 €	-0,01 €	0,02 €	0,20 €	0,19 €	0,22 €	0	0	0
20231226	0,02 €	0,00 €	0,05 €	0,22 €	0,20 €	0,28 €	3	0	0
20231227	0,06 €	0,00 €	0,08 €	0,28 €	0,20 €	0,31 €	2	0	0
20231228	0,02 €	0,00 €	0,05 €	0,22 €	0,20 €	0,25 €	0	0	0
20231229	0,01 €	0,00 €	0,03 €	0,21 €	0,20 €	0,23 €	0	0	0
20231230	0,04 €	0,00 €	0,08 €	0,24 €	0,20 €	0,31 €	66	0	0
20231231	0,01 €	0,00 €	0,03 €	0,21 €	0,20 €	0,23 €	0	0	0
20240101	0,02 €	0,00 €	0,06 €	0,22 €	0,20 €	0,28 €	43	0	0
20240102	0,05 €	0,00 €	0,08 €	0,27 €	0,20 €	0,31 €	11	0	0
20240103	0,05 €	0,00 €	0,08 €	0,27 €	0,20 €	0,31 €	11	16	0,7

Mittwoch, 1. Mai 2024 Seite 8 von 11

Abb. 3: Historische Tageswerte dynamischen Stroms (hier vom 19.11.23 bis zum 03.01.2024)

Die rot markierten Wartezeiten sind besonders kritisch für kleinere Speicher, denn je nach Stromverbrauch und Speicherkapazität wird der Speicher unterschiedlich schnell entleert. Bei leerem Speicher ist der Haushalt gezwungen, seinen Bedarf mit teurem Strom zum Bruttopreis zu decken. Auch ein Wiederaufladen des Stromspeichers wäre dann sehr teuer.

Im oberen Teil von Abb. 3 ist ein günstiger Mix von dynamischen Strompreisen zu sehen. Ab dem 2. Drittel ist ein ungünstiger Verlauf zum Jahresende visualisiert.

Die sehr langen, rot markierten Phasen teuren Stroms zeigen indirekt eine sogenannte Dunkelflaute an. Dabei handelt es sich um ein starkes Nachlassen der Generierung aus regenerativen Quellen (keine Sonne und kein Wind). Da die Werte auf reinen Preisdaten beruhen, soll dieser Bereich zur eindeutigen begrifflichen Abgrenzung als **„Preisdunkelflaute“** bezeichnet werden. Im Beispiel vom November 2023 dauerte diese Preisdunkelflaute 15 Tage, der Maximalwert wurde erreicht gleich darauf mit 17 Tagen im Januar 2024.

Die rot markierten Tage in Abb. 3 sind eine anschauliche Visualisierung des Kernproblems der alternativen Energien und der gesamten Energiewende: Um einer solchen Preisdunkelflaute bei einem kompletten Abschalten herkömmlicher Kraftwerke zu begegnen, müsste eine Stromspeicherkapazität von mindestens 20 Tagen bereitgestellt werden, besser noch mehr! Aktuell liegt die verfügbare Speicherkapazität in Deutschland nach [8] bei ca. 55 GWh und deckt damit bei einer Spitzenlast von 80 GW (ebenfalls [8]) noch nicht einmal eine Stunde ab. Bei kleiner Nacht-Last von 20 GW dann etwa 3 Stunden. Bei der in Deutschland politisch angestrebten Vollversorgung regenerativer Energien wären jedoch 20 Tage Abdeckung notwendig, also das mindestens das Hundertfache. Dies erscheint gegenwärtig weder finanziell noch technisch durch den aktuellen Fachkräftemangel als realistisch umsetzbar.

2 Simulation eines adaptiven dynamischen Strompreises

Zur genaueren Analyse wurde ein Simulationsmodell mit den importierten Börsen-Strompreisen entwickelt.

In der aktuellen Version des Simulationsmodells wird von einem Normalhaushalt mit einem Speicher in einer Größenordnung von 6 bis 24 Kilowattstunden Speicherkapazität ausgegangen mit einem Haus-internem Verbrauch von 2 bis 12 KWh täglich. Die sehr große Spannweite des Verbrauchs entspricht den realen Annahmen eines sehr kleinen Haushalts mit nur 700 KWh/Jahr und einem größeren Mehrfamilienhaus mit Wärmepumpe und Elektroauto und 5000 KWh/Jahr. Für die generellen Ergebnisse der Simulation sind die Ergebnisse zwar bzgl. der numerischen Werte relevant, doch die ermittelten Grundaussagen bleiben gleich. Die heute häufig übliche Kombi-Installation einer Photovoltaikanlage (PV) mit integriertem Speicher macht die Betrachtung leider noch komplizierter.

Da die PV-Anlage gerade zu Zeiten sehr geringer Strompreise den meisten Strom liefert, steht die PV der sinnvollen Nutzung von dynamischen Strompreisen eher entgegen. Zumindest in der Basisversion wird daher der **Stromgewinn durch die PV-Anlage nicht berücksichtigt**, um eine reine Analyse der dynamischen Strompreise zu bewirken. Dieses Szenario kann auch real bei zu kleinen Dachflächen oder anderen Limitierungen bzgl. einer PV-Installation auftreten. Bei Vorhandensein einer PV-Anlage verschlechtern sich die finanziellen Ertragsmöglichkeiten durch dynamischen Strom. Andererseits ist der fast kostenlose PV-Strom (wenn man die Abschreibung vergisst) wiederum in der Gesamt-Stromrechnung als sehr kostensparend einzuschätzen, was aber nur für den Kunden, jedoch nicht für die dynamischen Strompreise als System vorteilhaft ist.

Im Simulationsmodell wird durch die modellierte Steuerung des Energiespeichers eine einfache Minimalwert-Analyse der jeweils nächsten 24 h dynamischer Strompreise durchgeführt. Bei mittlerer Speicherladung wird genau zum 24h-Minimum der Speicher vollgeladen. Falls die Speicherladung weniger als 25% beträgt, wird auch zu nicht ganz optimalen Zeitpunkten der Speicher auf 50% aufgeladen. Bei ungünstigen Strompreisen wird der gesamte Haushalt aus dem Speicher versorgt. Wenn die Speicherkapazität erschöpft ist, wird automatisch auf das öffentliche Stromnetz umgeschaltet, was allerdings sehr teuer sein kann.

Bei der Berechnung des Modells wurden nur echte Preisdaten ab dem 1.1.2023 verwendet. Um den Effekt dynamischer Strompreise zu bewerten, wird mit einem Standard-Stromtarif von 0,35 €/ KWh verglichen.

In Abb. 4 ist die Bedienmaske des entwickelten Energiemarkt-Simulators dargestellt. Aufgrund der starken Datenbezüge zu realen Strompreisdaten wurde eine Implementierung mit einer Microsoft-Access-Datenbank durchgeführt. Auch die Algorithmen zur Simulation sind mit dem darin verfügbaren Visual Basic von Microsoft realisiert. Portierungen zu .NET- oder webbasierten Versionen sind geplant.

Die auf ein Jahr normierte Differenz zwischen Standard-Tarif und dynamischen Strompreis von 333 € zeigt den geringen Effekt der dynamischen Strompreise unter realen Bedingungen. Mit den eingesparten ca. 333€ /Jahr ist eine Refinanzierung des Stromspeichers mit aktuell ca. 10.000 € Investitionskosten selbst nach 20 Jahren nicht möglich. Stromspeicher sind, wie leider zu Beginn vermutet, ohne zugehörige PV-Anlage, ein sehr teures Hobby und wirtschaftlich nicht vertretbar.

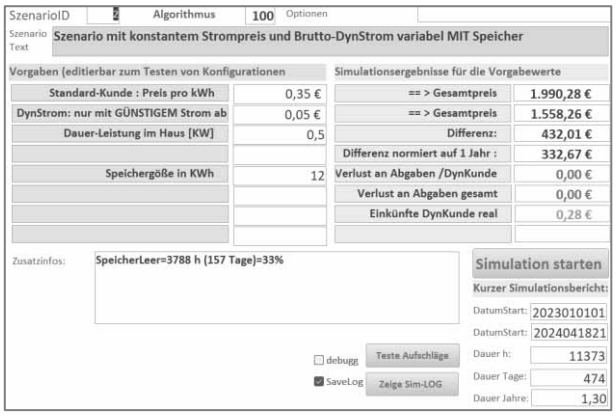


Abb. 4: Die Bedienmaske des Simulationsmodells (mit ausgewählter Standard-Konfiguration)

3 Schlussfolgerungen aus den Simulationsuntersuchungen

Bei der Berechnung weiterer Szenarien ergeben sich die nachfolgenden Ergebnisse.

3.1 Best-Case-Szenario „Ohne Abgaben“

Bei einer hypothetischen Befreiung dynamischer Strompreise von allen Abgaben ergibt sich eine Differenz von ca. +2.500 € pro Jahr. Damit wäre eine Refinanzierung und sogar Erweiterung der Stromspeicher leicht möglich. In der Summe ergeben sich aktuell nur leider Abgabenverluste in gleicher Höhe. Diese könnten für die BR Deutschland bis zu 10 Mrd. €/Jahr betragen. Eine derartige Befreiung ist bei der aktuellen Haushaltssituation nicht vorstellbar, da neben dem Staat (MwSt. u.a.) auch die Kommunen und Energieunternehmen diese Summe anteilig pro Jahr verlieren würden. Das Ist-Verhältnis von Netto- zu Brutto-Strompreisen zeigt Abb. 5.

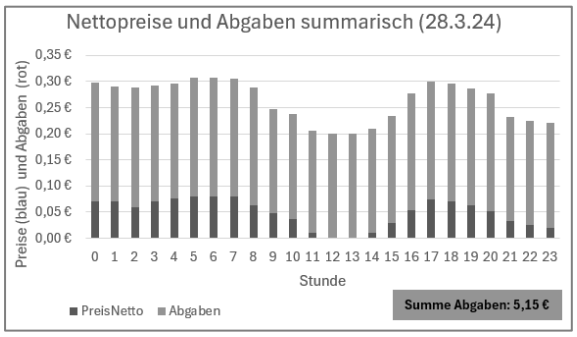


Abb. 5: Aktuelle Netto- und Brutto-Strompreise

3.2 Modellszenario „DynPlus-Strompreise“

Es erscheint sinnvoll, eine Verschiebung der Abgaben über der Zeit anzustreben. Bei geringen Nettopreisen sollten nur geringe Abgaben fällig werden und bei mittleren Preisen diese mit einem Aufschlag gegenfinanziert werden. Hohe Netto-Preise sollten unbeeinflusst bleiben. Im aktuellen Entwurf wird bei geringen Nettopreisen nur noch eine prozentuale Abgabenlast von maximal 200% angenommen. Bei 0,00 € Nettopreis oder darunter ergeben sich folglich Abgaben von ebenfalls 0,00 €, was genau erwünscht ist. (vgl. Abb. 6).

Stunde	PreisNetto	DynPlusPreis	Aufschlag	BruttoPlus	Kommentar
7	0,08 €	0,16€	0,10€	0,34€	Aufschlag
8	0,06 €	0,13€	0,13€	0,32€	Aufschlag
9	0,05 €	0,09€	0,12€	0,26€	Aufschlag
10	0,04 €	0,07€	0,03€	0,14€	Aufschlag
11	0,01 €	0,02€	0,00€	0,03€	ProzMax 200%
12	0,00 €	0,00€	**0,00€**	**0,00€**	ProzMax 200%
13	0,00 €	0,00€	**0,00€**	**0,00€**	ProzMax 200%
14	0,01 €	0,02€	0,00€	0,03€	ProzMax 200%
15	0,03 €	0,06€	0,03€	0,12€	Aufschlag
16	0,05 €	0,11€	0,12€	0,28€	Aufschlag
17	0,07 €	0,15€	0,11€	0,33€	Aufschlag

Abb. 6 Neue Preis- und Abgaben-Verteilung (Auszug)

Der resultierende Gesamtverlauf kann Abb. 7 entnommen werden. Zur Mittagszeit ergibt sich das angestrebte Brutto-Preisminimum zum Laden der Stromspeicher. Durch die Zusatzabgaben ergeben sich gleiche Abgaben *(5,15 € unten rechts)* in der Summe über den Tag, **d.h. der neue Preis-Ansatz ist für den Staat und die Energiewirtschaft aufkommensneutral!**

Simulationen dieses Ansatzes zeigen eine prinzipielle Umsetzbarkeit dieser Idee durch eine Neugestaltung der dynamischen Strompreise, nachfolgend als **DynPlus-Preise** bezeichnet. Allerdings kommt es bei längeren Preisdunkelflauten wieder zum Leerlaufen der Stromspeicher und dann zu einem Stromverbrauch in den mit den Abgaben zusätzlich belasteten Zeiten. Damit reduzieren sich die Einsparungen auf nur noch 800 € pro Jahr, was zwar noch für eine Refinanzierung gerade noch ausreichend ist, aber keine Erweiterungen mehr erlaubt.

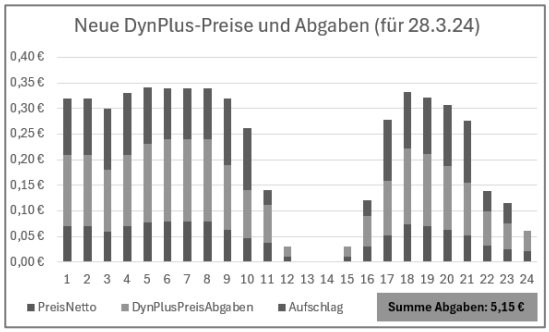


Abb. 7 Preisverlauf mit neuen DynPlus-Strompreisen

Alternativ könnte die Refinanzierung der Abgaben unter Einbeziehung der Normaltarife erfolgen. Wenn in Analogie zur Abgabenverteilung nach Abb. 7 ein Aufschlag von genau 0,01 € zu Zeiten mittlerer Strompreise erfolgt, gelingt die Refinanzierung und Speichererweiterung mit einem jährlichen Saldo von ca. +1500 € pro Jahr (vgl. Abb. 8). Möglich wird dies durch das Verhältnis gegenwärtig von 34 Kunden mit Normaltarif zu einem Kunden mit Stromspeicher. Der Aufschlag von einem Cent wird dabei nur bei von Nettopreisen kleiner 0,06 € aktiviert und auch nur bei aktiven Verlustsaldo aus den letzten Stunden.

In der Simulation nach Abb. 8 ergibt sich ein leichtes Plus von 8 € und eine Nutzung der Aufschlagszone von nur 75%, d.h. auch bei einem Anwachsen der Anzahl der Haushalte mit Speichern kann dieser Ansatz bestehen bleiben. Danach wäre über eine leichte Erhöhung des Aufschlags zu entscheiden.

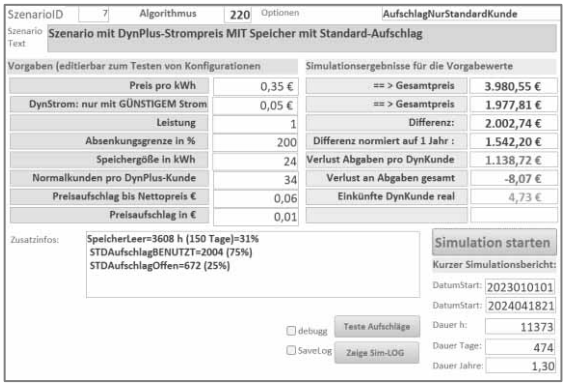


Abb. 8: DynPlus-Strompreise mit Alternativaufschlag

Bei der stundenweisen Simulation bestehen sehr viele Freiheitsgrade bzgl.

- der Preisvorgaben für die Aufschlag-Signale,
- der Abgabenbegrenzung (hier maximal 200%),
- dem mittleren Leistungsverbrauch im Haushalt,
- der Speichergröße,

welche über die Eingabefelder eingestellt und in der Szenarien-Datenbank gespeichert werden können. Dabei ist jede der gezeigten Masken mit einem Szenario nur ein Datensatz in einer beliebig großen Datenbanktabelle. Aktuell wird immer nur der aktuell simulierte Datensatz mit seinen Simulationsergebnissen (im Sinne einer 1:1-Relation) gespeichert: Das Speichern aller Simulationsergebnisse (im Sinne einer 1:N-Relation zwecks Vergleich der Ergebnisse über alle Ergebnisse über alle Szenarien) ist sehr einfach möglich, in dem das Löschen der alten Szenarien-Daten abgeschaltet wird und eine Datenreferenz zum Szenario-Datensatz aufgebaut wird.

Weiterhin ist auch eine automatisierte Neu-Berechnung aller Szenarien angedacht, damit bei einer Änderung an grundlegenden Simulationsalgorithmen, z.B. an der Art der Stromspeichersteuerung, nicht manuell gestartet werden muss Da die Simulation über die Strompreis-Zeitreihen im Stundentakt mit ca. 10.000 Datensätzen nur wenige Sekunden dauert, ist eine Komplettsimulation eine Aufgabe weniger Minuten.

3.3 Geplante Online-Validierung der Modelle

Da die ersten Gespräche mit Stakeholdern im Energiemarktbereich sehr vielversprechend waren und eine absolute Fehlerfreiheit der Berechnungen damit immer wichtiger wird, soll mit FachkollegInnen aus dem Simulations- und Energiebereich eine mehrfache „Nach-Programmierung“ der Simulationsalgorithmen aus der Access-Datenbank erfolgen.

Auch ein Projektseminar mit 6 Studenten der Wirtschaftsinformatik wird im Wintersemester 2024/25 das bisherige Modell einmal auf der Basis einer rein textuellen Beschreibung der Algorithmen zur Vermeidung simpler Programmierfehler und einmal auf der Basis des Sourcecodes analysieren und mit anderen Technologieoptionen (z.B. auch JavaScript auf NodeJS-Servern oder .NET in MS-Azure-Umgebungen) erneut erstellen.

Als Zielsystem wird ein cloudbasiertes und ggf. auch verteiltes Simulationssystem mit einer zentralen Datenhaltung und beliebig vielen Simulator-Instanzen auf unterschiedlichen Servern angestrebt. Auf dem zentralen Datenbank-Server werden Tools oder Algorithmen aus dem Business-Intelligence-Bereich (BI) zum Vergleich und zur Analyse der unterschiedlichen Simulationsergebnisse installiert. Alle am Thema (und günstigen Strompreisen) interessierten Simulations-Fachkollegen sind dazu recht herzlich eingeladen!

In die Cloudsysteme sollen auch die Simulationsmodelle von Strommarktexperten und Firmen eingehängt und dann gegen verschiedene Wetter- und Nachfrageszenarien ausgetestet werden können. Ein weiterer Ausbau könnte auch durch darauf aufsetzende Modelle zu den Steuersystemen der Energieversorger selbst erfolgen, so dass diese ihre eigenen, firmen-internen Preisbildungsmodelle und Realtime-Steuerungen vorab austesten können. Wenn die Einbeziehung der Energieversorger in diese Simulations-Infrastruktur gelingt, sollte auch die Finanzierung der Basissimulations-Infrastruktur gut abgesichert sein.

3.4 Eine Empfehlung an die Gesetzgebung

Eine reale Umsetzung des DynPlus-Preismodells würde entsprechende Novellierungen der Strommarkt-Gesetzgebung und/oder der Ausführungsbestimmungen für die Abgabenberechnung erfordern. Dazu wird gegenwärtig die Akzeptanz dieses Ansatzes bei Wissenschaftlern im Energiepreismarkt, den Vertretern der Energiewirtschaft und final der Politik geprüft.

Da die Freiheitsgrade bei der neuen Abgabenverteilung sehr groß sind und das letzte Wort die Politik und die Lobby der Energieversorger haben werden, wird folgender, sehr offener Vorschlag zur Bestimmung neuer **adaptiver dynamischer Strompreise** gemacht, wobei der enthaltene Prozentwert beliebig änderbar ist:

„Die Abgabenlast bei dynamischen Strompreisen darf 200% des Netto-Strompreises nicht übersteigen. Die summarische Differenz zur bisherigen Berechnung dynamischer Brutto-Strompreise kann zu anderen Zeiten durch Aufschläge ausgeglichen werden.“

Die Verwendung einer prozentualen Begrenzung der Strompreisabgaben führt zu einer sanften und nicht sprunghaften Reduzierung der Abgabenlast. Damit erfolgen im Ergebnis wiederum auch keine sprunghaften Lastwechsel durch die Endabnehmer und das Stromnetz kann insgesamt stabiler und besser ausgeregelt werden.

Beim obigen Prozentwert von maximal 200% Abgaben würde sich bei Netto-Stromkosten von einem Cent eine Abgabenlast von 2 Cent ergeben und damit ein finaler Bruttopreis von 3 Cent gegenüber bisher 23 Cent (vgl. Abb. 6). Ab 0.00 € Nettostrompreis und darunter gehen dann auch diese Abgaben wirklich auf 0,00 €!

Ob diese temporären „Nichtabgaben“ (auch im EU-Kontext) gesetzlich zulässig sind, wird ebenfalls mit Experten geklärt. Alternativ könnten auch bei negativen Strompreisen die gesetzlich unbedingt notwendigen Abgaben erhoben werden, möglichst jedoch auch begrenzt auf einen prozentualen Anteil und einem sehr geringen konstanten Abgabenbetrag.

Der bewusst offen gehaltene Textvorschlag kann einen Wettbewerb der Energieversorger um das **attraktivste Preismodell für zukünftige adaptive dynamische Strompreise** bewirken. Zu erwartende Folgen sind eine starke Motivation der Endkunden und Energieversorger zum Ausbau der Stromspeicher und ein nachfolgendes besseres Ausbalancieren von Angebot und Nachfrage.

4 Langfristige Energiemarkt-Simulation

Mit der konservativ ermittelten Einsparung von ca. 1.500 € pro Jahr, was ca. 50% Einsparung gegenüber einem Normaltarif bedeutet, wurde ein aufsetzendes Simulationsmodell zur langfristigen Simulation entwickelt. Die sehr konservative Bewertung ergibt sich dabei aus folgenden Punkten:

- Es wurden keine Einspeisevergütungen für eine Rückführung der gespeicherten Energiemengen in das Strom-Netz betrachtet. Angenommen wurde ein vollständiger Verbrauch der Speicherladung im Haus.
- Bei einer Betrachtung von Einspeisevergütungen ist der gegenwärtige konstante Preis von ca. 0,08 € ebenfalls als wenig förderlich zu betrachten. Analog zur massiven Reduzierung der Abgabenlast zu Zeiten eines Überangebots wäre auch eine Abgabenreduzierung bei sehr hoher Nachfrage und eine Vergütung in der Größenordnung der Nettostrompreise sehr sinnvoll. So liegt das Maximum der Nettostrompreise bei 1/3 der Tage im Jahr bei über 0,11 €! Wenn es noch einen gewissen Sonderbonus von wenigen €-Cent für die Rückeinspeisung in Zeiten sehr hoher Nachfrage gäbe, würde sich die Gesamtrechnung noch einmal stark verbessern.
- Vorhandene PV-Anlagen würden weitere, sehr günstige Energie liefern, allerdings gerade zu Zeiten sehr geringer oder negativer Nettostrompreise und günstiger als die gegenwärtigen Bruttostrompreise. Damit würden sie einem Laden des Stromspeichers mit DynPlus-Strom entgegenstehen. Andererseits würde sich bei Einspeisung dieser Strommengen bei besseren Konditionen in das Netz eine nochmals verbesserte Jahres-Rendite für die gesamte PV+Speicher-Installation ergeben.

Bei der sehr konservativen Annahme von +1.500 € Rendite pro Jahr wurde eine Langfristsimulation bis zum Jahr 2050 vorgenommen (vgl. Abb. 9 mit den Werten bis 2050). Die Simulationsdauer kann einfach in der Eingabemaske geändert werden. Allerdings werden mit zunehmender Simulationsdauer die Annahmen über die Preise pro KWh-Stromspeicher immer ungenauer, da gegenwärtig intensive Forschungen zu neuen Speichersystemen stattfinden. Starke Reduzierungen der Speicherkosten sind durchaus ebenfalls möglich.

Abb. 9: Eine Energiemarktsimulation bis 2050

Bei den Berechnungen wurde angenommen, dass die Einsparungen durch die besseren Tarife im DynPreisPlus-Konzept zu 90% wieder reinvestiert werden. Dies ist plausibel und sinnvoll, da sich:

- Jede reinvestierte Geldsumme in einem neuen Speicher niederschlägt und dieser zukünftig auch wieder zur Einsparung beiträgt.
- Im Gegensatz zu einem Kühlschrank, dessen stetiger Zubau natürlich nicht sinnvoll ist, können damit Speicher bei zukünftig vernünftiger Abgaben- und Vergütungspolitik im Gesamtmaßstab SELBST GELD VERDIENEN.

Dies sollte für den Eigentümer bei einer jährlichen Kapitalrendite von 10% genügend Anreiz sein, das eingesparte Geld wieder zu 90% zu reinvestieren. Bei kleinen Investitionsraten verschlechtern sich natürlich die Werte und es verlangsamt sich der Speicherausbau. In der Praxis wird sich ein Mix ergeben, wobei durch große Firmen-Investoren mit 100% Re-Investitionsrate und sehr großen Speicheranlagen im MW-Bereich und kleine Privatanwender mit niedrigeren Re-Investraten im eigenen Haus, bedingt auch durch Platzprobleme, sich doch ein mittlerer Wert von 90% ergeben könnte.

Im Ergebnis ergibt sich unter diesen Annahmen durch die **Umlage der eingesparten Kosten auf den Speicherausbau** ein Zuwachs der installierten Speicherkapazitäten von 1366%, also mehr als eine **Verzehnfachung**. Die Anzahl der Haushalte mit Speicher erhöht sich im gleichen Zeitraum von 2,9% in 2024 auf knapp 40% in 2050.

Der simulierte Verlauf der Entwicklung kann Abb. 10 entnommen werden. Der Anstieg der Werte hat zunächst einen exponentiellen Charakter, da die neu zugebauten Speicherkapazitäten zu einer weiteren Erhöhung der absoluten Einsparungen und damit zu stetig ansteigenden Finanzressourcen führen. Mit den prozentuellen Einstellungen für die Umlagen kann dies experimentell ausgetestet und angepasst werden.

Ungefähr ab dem Jahr 2044 kommt es bei den definierten Vorgabewerten zu einer Limitierung des Speicherzubaus durch den immer stärker werdenden Fachkräftemangel. Es wurde dabei angenommen, dass die Rekordbauleistung des Jahres 2023 von ca. 500.000 Speicheranlagen (nach [7]) pro Jahr nicht mehr als verdoppelt werden kann, da die demografische Entwicklung und die Konkurrenzsituation zu anderen Fachbereichen die absolute Anzahl der Arbeitskräfte in der Speicherbranche auf maximal das Doppelte der heutigen Anzahl limitiert. Im Ergebnis des Fachkräftemangels kann dann nur noch eine feste Anzahl von Speicheranlagen gebaut werden und der Anstieg des Gesamtspeicherleistung geht ab 2044 in einen linearen Anstieg über (vgl. Markierung in Abb. 11). Die Verfügbarkeit der Speicheranlagen (Rohstoffe, Produktion, Vertrieb etc.) wurde vorerst nicht betrachtet.

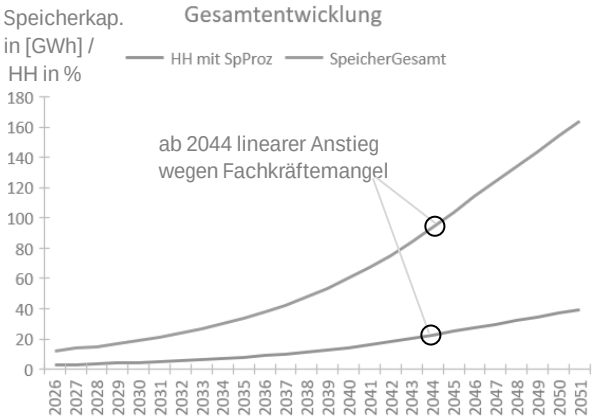


Abb. 10: Simulierte Speicherkapazitäten und Anteil von Haushalten (HH) mit Speicher bis 2050

Weitere Simulationen mit einem Vergleich von angebotener Leistung an alternativen Energien und der anwachsenden Speicherkapazität zeigen, dass selbst bei der 13-fachen Speicherkapazität noch immer ein Überangebot an den sehr volatilen Energien von 900% über den Tag besteht (vgl. gelb markierter Bereich in Abb. 9). Es müsste also noch das mindestens das 4-fache an Speicherkapazität gebaut werden, bis der Strommarkt anfangen kann, sich selbst auszubalancieren und die Phasen negativer Strompreise zurückgehen. Damit ist der Ansatz von adaptiven dynamischen Strompreisen sinnvoll und anwendbar für den gesamten Betrachtungszeittraum bis zum Jahr 2050, da kein signifikanter Rückgang der Einsparpotentiale zu erwarten ist.

Ob eine reale, weitere Vervierfachung der Speicherkapazitäten im betrachteten Zeitraum möglich ist, erscheint sehr fraglich aufgrund der limitierten Anzahl von Haushalten und dem diskutierten Fachkräftemangel. Denkbar wäre allerdings auch eine größere Anzahl von Großanlagen mit einigen GWh-Kapazität.

Eventuell könnte durch die adaptiven dynamischen Strompreise und die daraus resultierende positive Bilanz bei den Speicherinvestoren auch der finanzielle Gewinn für die Bezahlung von Fachkräften im Bereich der Stromspeichertechnik verfügbar werden. Dank der dann zu erwartenden Entwicklung in Richtung höherer Gehälter könnte wiederum ein Zufluss externer Fachkräfte erfolgen und dies die erwartete Limitierung durch den Fachkräftemangel abschwächen oder ganz aufheben.

5 Gesamtarchitektur und Ausblick

5.1 Hierarchie der Simulationsmodelle

Die Gesamtarchitektur der Simulationsuntersuchungen kann Abb. 11 entnommen werden. Ausgehend von den importierten Strompreis-Börsendaten und dem DynPlus-Preismodell wird das stundengenaue Verhalten des Stromspeichers berechnet. Nach einer Normierung der Finanzergebnisse auf ein Jahr wird der ermittelte Jahressaldo für die Langzeitsimulation verwendet.

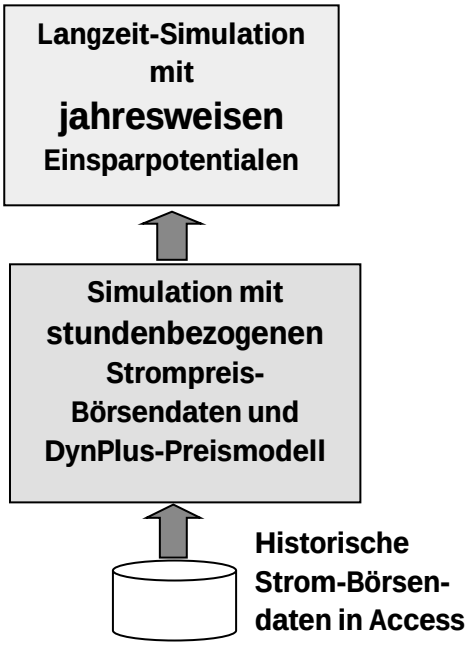


Abb. 11: Die Hierarchie der Simulationsmodelle

Bei dieser Vorgehensweise sind folgende Punkte als kritisch im Sinne der Validierung zu bewerten:

- In den Simulationsszenarien der unteren Ebene wird gegenwärtig nur der Zeitraum 1.1.2023 bis Mai 2024 berechnet. Dieser Zeitraum ist auf keinen Fall repräsentativ für alle Folgezeiträume aufgrund der vielfältigen stochastischen Rahmenbedingungen (Klimawandel, lokales Wetter, Kundennachfrage, technische Innovationen etc.). Andererseits ist mit den realen Daten der Strombörse seit dem 1.1.2023 eine genaue Simulation der hypothetischen adaptiven dynamischen Strompreise für genau diesen Zeitraum möglich gewesen.
- Das Zusammenspiel von Stromspeichern, Photovoltaik, Windgenerator-Anlagen und dynamischen Strompreisen an der Strombörse ist hoch komplex und stark abhängig von der Wetterlage und der gesamten Marktsituation. Starke Schwankungen der Nachfrage durch Feiertage oder plötzliche Kälteeinbrüche können den Verlauf der dynamischen Strompreise in beide Richtungen sehr stark verändern.
- Wie der Ausfall der Kommunikation zwischen den trans-europäischen Strombörsen am 26.6.2024 gezeigt hat, in dessen Folge die dynamischen Strompreise auf das Zehnfache gestiegen sind (vgl. [9][10]), ist die Energiewirtschaft in Deutschland allein nicht mehr voll autark überlebensfähig. Ein vollständiges Simulationsmodell für genaue, zukunftsvoraussagender DynPlus-Strompreise muss unbedingt den gesamten europäischen Markt umfassen, zumindest aber die wichtigsten Strommarktpartner von Deutschland. Die Komplexität der Modelle dürfte sich damit mindestens verzehnfachen, da in anderen Ländern wieder andere Steuer- und Abgabensysteme vorliegen!
- Veränderungen durch neue politische Einflussnahmen oder globale Krisensituationen ähnlich wie die Corona- Pandemie oder bewaffnete Konflikte können alle getroffenen Annahmen zum Ausbau der Speicherkapazitäten obsolet machen.

Trotz dieser Einschränkungen liegt mit den vorliegenden Berechnungen eine gewisse Abschätzung der zukünftigen Entwicklungen im Stromspeicherbereich vor und es besteht die begründete Hoffnung, dass mit einer Anpassung der Abgabenpolitik eine deutlich verbesserte Motivation zum Speicherausbau erzielt werden kann.

5.2 Ausblick

Die vorgestellten Untersuchungen werden aktuell gemeinsam mit Fachkollegen aus den Bereichen Energiewirtschaft und der wirtschaftswissenschaftlichen Investitionsplanung diskutiert. Wenn dabei keine grundsätzlichen gesetzgeberischen Restriktionen bekannt werden, sollen anschließend gemeinsam mit KollegInnen aus dem VDI (Verein Deutscher Ingenieure), VDE (Verband der Elektrotechnik Elektronik Informationstechnik e.V.) und anderen Fachverbänden der Wirtschaft und Gesellschaft die entsprechenden Fachexperten in den politischen Gremien angesprochen werden.

Falls keine politische oder administrative Umsetzung der adaptiven dynamischer Strompreise möglich sein sollte, dann sind die Ursachen für ein wirtschaftliches Scheitern der Energiewende in Deutschland mit den hier vorgelegten Simulations-Ergebnissen in groben Umrissen erkennbar.

References

[1] ENWG-Gesetz §41a: *Gesetz über die Elektrizitäts- und* Gasversorgung *(Energiewirtschaftsgesetz - EnWG) § 41a Lastvariable, tageszeitabhängige oder dynamische und sonstige Stromtarife*, https://www.gesetze-im-internet.de/enwg_2005/__41a.html

[2] Tibber-Website zu dynamischen Strompreisen: https://tibber.com/de

[3] Fraunhofer-Institut für Solare Energiesysteme ISE, Website: https://www.ise.fraunhofer.de/

[4] ISE-API-Überblick: https://api.energy-charts.info/

[5] *OpenAPI Specification v3:* https://spec.openapis.org/oas/v3.0.3

[6] ISE-API für “Day-ahead-price”: https://api.energy-charts.info/#/prices/day_ahead_price_price_get

[7] Statistisches Bundesamt -DESTATIS-Website: https://www.destatis.de/DE/Presse/Pressemitteilungen/Zahl-der-Woche/2023/PD23_25_p002.html

[8] Energy-Chart: Strom-Markt-Daten: www.energy-charts.info/charts/installed_power/chart.htm

[9] Heise.de: *Ursache für technischen Fehler an Strombörse gefunden* https://www.heise.de/news/Hohe-Strompreise-an-Stromboerse-Ursache-fuer-technischen-Fehler-gefunden-9780495.html

[10] Expex-spot: *Decoupling Session was run according to procedures of Single Day-Ahead Coupling* https://www.epexspot.com/en/news/decoupling-session-was-run-according-procedures-single-day-ahead-coupling

Implementierung eines Smart Grids in ein betriebsunabhängiges Simulationsmodell

Julian Stromberger[1*], Johannes Dettelbacher[1], Alexander Buchele[1]

[1]Kompetenzzentrum Industrielle Energieeffizienz, Hochschule Ansbach, Residenzstraße 8, 91522, Ansbach; [*]*j.stromberger@hs-ansbach.de*

Abstract. Diese Studie beschreibt die Entwicklung eines betriebsunabhängigen Simulationsmodells für elektrifizierte Druckgießereien, die ihren Energiebedarf mit Hilfe eines Smart Grid Systems decken. Das Modell verwendet reale Wetter- und Börsenstrompreisdaten für den Simulationszeitraum. Mit Hilfe des Modells können die Stromkosten für eine Produktion zu einem bestimmten Zeitpunkt (Tages- und Jahreszeit) sowie die Wirtschaftlichkeit verschiedener PV-Anlagen- und Stromspeichervarianten ermittelt und verglichen werden. Zudem ermöglicht es die Untersuchung des Anteils der verschiedenen Energieträger für die jeweilige Konfiguration. Dies kann mit Hilfe des Modells für Standorte in ganz Deutschland durchgeführt werden. Darüber hinaus werden in dieser Arbeit beispielhafte Simulationsstudien vorgestellt, die den breiten Anwendungsbereich des Modells aufzeigen. Aus den Ergebnissen kann ein erster Überblick über Einsparungs- und Optimierungsmöglichkeiten gewonnen werden. Perspektivisch stellt das Modell eine Grundlage dar, um mittels simulationsgestützter Optimierung optimale Anlagenlayouts und Produktionszeitpunkte zu ermitteln.

Einleitung

Vor dem Hintergrund der Klimaziele und der europäischen Lieferkettenrichtlinie gewinnt eine möglichst emissionsarme Produktion gerade für energieintensive Unternehmen zunehmend an Bedeutung. Gleichzeitig steht für die Unternehmen immer auch ein möglichst kostengünstiger Energiebezug im Fokus. Diese Aspekte stehen jedoch nicht zwangsläufig in Konkurrenz zueinander, denn aufgrund der Volatilität der erneuerbaren Energien kann emissionsfreier Strom teilweise sehr günstig bezogen werden, wodurch CO2- und kostenarmer Strombezug miteinander vereinbar sind. Darüber hinaus können produzierende Unternehmen mit entsprechenden Gewerbeflächen auch selbst erneuerbaren Strom erzeugen, was allerdings zunächst mit hohen Investitionskosten verbunden ist. Die Nutzung dieser Potenziale setzt zudem eine gewisse logistische Flexibilität voraus, die oft nicht trivial umzusetzen ist. Um entsprechende Anpassungen in Unternehmen testen und die wirtschaftlichen Auswirkungen untersuchen zu können, ohne in den realen Betrieb eingreifen zu müssen, was immer mit wirtschaftlichen Risiken verbunden ist, wird in der vorliegenden Studie die Entwicklung eines Simulationsmodells für diesen Einsatzzweck näher beleuchtet. Im Rahmen dieser Arbeit wird dieses als Erweiterung in ein Betriebsmodell für Gussbetriebe implementiert. Um Aussagen für unterschiedliche Betriebsgrößen treffen zu können und eine mögliche Übertragbarkeit auf andere Branchen zu ermöglichen, ist es betriebsunabhängig und auf Flexibilität ausgerichtet.

1 Anwendungsfeld

Die deutsche Gießereiindustrie gehört mit einem Jahresbedarf von durchschnittlich 12,6 TWh (Jahre 2010 bis 2021) zu den energieintensiven Wirtschaftszweigen. Davon wird auch im Jahr 2021 noch mehr als die Hälfte durch fossile Energieträger gedeckt [1]. Um die entstehenden Emissionen zu reduzieren, ist eine Substitution dieser fossilen Energieträger unumgänglich. Der wichtigste Ansatz hierfür ist die Elektrifizierung des Schmelzprozesses. Die Wirksamkeit dieser Maßnahme hängt jedoch stark von der Zusammensetzung des bezogenen Strommixes ab. Darüber hinaus führt diese Umstellung zu einer Änderung des Betriebsablaufs [2]. In dieser Studie wird ein Simulationsmodell eines derart umgestellten Betriebes untersucht. Ziel des Modells ist es, unter Verwendung eines Smart Grid Systems, das Wetterdaten, Stromspeicher und Strombörsenpreise berücksichtigt, die aus einer Produktion resultierenden Energiekosten für beliebige Konfigurationen eines

solchen Betriebs zu ermitteln. Das Smart Grid System ist in dieser Anwendung auf den jeweiligen Betrieb beschränkt. Aufgabe des Systems ist es, den Strombedarf in der Produktion zu überwachen und entsprechend dem ebenfalls erfassten, aktuell verfügbaren Stromangebot möglichst kostengünstig zu decken.

2 Bisherige Arbeiten

Das in dieser Studie entwickelte Simulationsmodell baut auf Vorarbeiten auf, in denen ein betriebsunabhängiges Modell für einen konventionellen Druckgussbetrieb [3] sowie ein spezifisches Modell für einen elektrifizierten Betrieb [2] entwickelt wurden. Der Fokus dieser Arbeit liegt auf der Umsetzung eines Smart Grid Ansatzes mit Hilfe von Echtzeitdaten sowie den aus der Produktion resultierenden variablen Stromkosten. Die Kostenoptimierung in der produzierenden Industrie durch den Einsatz von flexiblen Energiepreisen und Smart Grids ist ein in der Literatur viel behandeltes Thema. Für einen Überblick können Literaturanalysen wie z.B. [4] herangezogen werden. Tabelle 1 gibt einen Überblick über die jeweiligen inhaltlichen Schwerpunkte einiger in diesem Bereich publizierten Studien im Vergleich zur vorliegenden Arbeit. In der vorliegenden Studie werden erstmals reale Wetterdaten zur Ermittlung der Eigenstromerzeugung herangezogen.

Inhalt	**[5]**	**[6]**	**[7]**	**[8]**	**Diese Studie**
Simulationsstudie	*	*	*	*	*
Felxible Energiepreise	*		*	*	*
Eigenstromerzeugung				*	*
Stromspeicher				*	*
Reale Wetterdaten					*

Tabelle 1: Vergleich der inhaltlichen Schwerpunkte verschiedener Studien.

3 Simulationsmodell

Das vorliegende Simulationsmodell basiert grundlegend auf den Erhaltungssätzen der Energie und Masse und ist in der Programmierumgebung MATLAB/Simulink realisiert. Das Modell ist objektorientiert aufgebaut und kann vor dem Start einer Simulation über eine Konfigurationsdatei individuell parametrisiert werden. Die Simulation arbeitet mit realen Start- und Endzeitpunkten, zu denen Wetter- und Börsenstrompreisdaten abgerufen werden. Das Modell ist in drei Teilmodelle unterteilt. Das Zusammenspiel der Teilmodelle innerhalb eines Simulationsschrittes ist in Abbildung 1 dargestellt. Ein Simulationsschritt in der Simulation entspricht jeweils einer Sekunde der simulierten Betriebszeit.

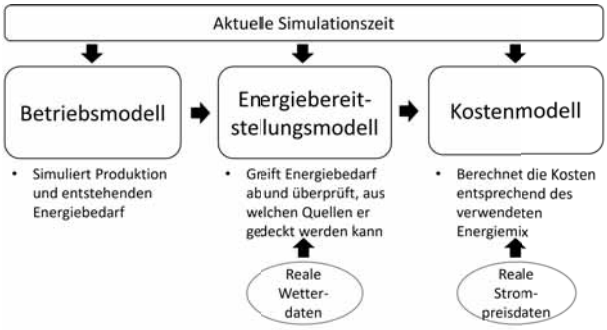


Abbildung 1: Ablauf des Simulationsmodells in jedem Simulationsschritt.

Die Kombination eines detaillierten Simulationsmodells einer Druckgießerei mit realen Wetterdaten und Strompreisen ermöglicht Aussagen über die Energiekosten und den Energiemix in verschiedenen Simulationsvarianten. Damit kann das Modell strategische Entscheidungen unterstützen und bewerten. Im Folgenden werden die einzelnen Teilmodelle näher erläutert.

3.1 Betriebsmodell

Das Modell basiert auf den Prinzipien der Energie- und Massenerhaltung und bildet den gesamten Produktionsprozess vom Schmelzen bis zum Gießen objektorientiert ab. Dabei werden Energie- und Materialflüsse gekoppelt betrachtet, während ein Steuermodul den Gesamtprozess regelt. Die Energieflüsse werden kontinuierlich simuliert, die Materialflüsse diskret. So werden die Berechnungen im Energiemodell auf Basis der inneren Energie des verarbeiteten Metalls durchgeführt. Dies ist in Abbildung 2 schematisch dargestellt. Auf der Grundlage dieser Berechnung ermittelt das Modell die aktuelle Temperatur und löst die Differentialgleichungen für die Masse des flüssigen und festen Metalls. Im diskreten Teil des Modells wird in jedem Simulationsschritt geprüft, ob bestimmte Ereignisse eingetreten sind. Gegebenenfalls werden dementsprechend Prozesse ausgelöst.

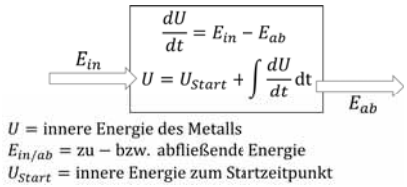


Abbildung 2: Innere Energie des verarbeiteten Metalls.

Das Modell besteht abhängig vom simulierten Betrieb aus verschiedenen Simulationsobjekten, deren Eigenschaften im Laufe der Simulation entsprechend der Simulationsparameter verändert werden.

3.2 Energiebereitstellungsmodell

In der Simulation wird nur der für den Betrieb der Schmelzöfen benötigte Strom berücksichtigt. Dieser wird in jedem Simulationsschritt durch Addition der von jedem Schmelzofen benötigten Energie, sowie der auftretenden Wärmeverluste, bestimmt. Die somit für den Betrieb benötigte elektrische Energie kann im Simulationsmodell auf drei Arten bereitgestellt werden:

1. elektrische Energie aus der PV-Anlage
2. elektrische Energie aus dem Batteriespeicher
3. elektrische Energie vom Spotmarkt

Die Reihenfolge der Nutzung dieser Energiequellen ist in Abbildung 3 schematisch dargestellt:

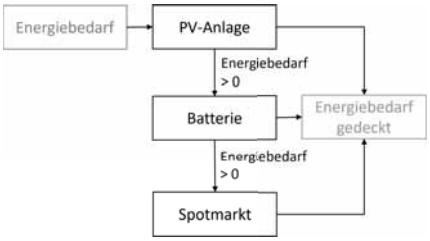


Abbildung 3: Reihenfolge des Strombezuges.

Die verschiedenen Quellen werden im Folgenden näher erläutert.

Elektrische Energie aus PV-Anlage. Abhängig von den spezifischen Anlagendaten und dem hinterlegten Betriebsstandort wird der Ertrag der PV-Anlage mit Hilfe der Wetterdaten aus dem Projekt duett [9] berechnet, das seit Anfang 2024 stündliche Daten in einer örtlichen Auflösung von 2x2 km für ganz Deutschland anbietet.

Batteriespeicher. Der Stromspeicher wird geladen, wenn mehr PV-Strom produziert wird, als der Betrieb aktuell benötigt, oder entladen, wenn mehr Strom benötigt wird, als aktuell produziert wird. Die Stromspeicherung ist mit einem gewissen Wirkungsgrad verbunden.

Spotmarkt. Kann der Energiebedarf nicht aus dem aktuell erzeugten Strom und dem Speicher gedeckt werden, wird die Differenz am Spotmarkt der European Power Exchange (EPEX SPOT) zugekauft.

3.3 Kostenmodell

Die Ausgaben für PV-Anlage und Batteriespeicher werden über lineare Abschreibungen für Anschaffungs- und Betriebskosten berechnet. Der zugekaufte Strom vom Spotmarkt geht entsprechend des aktuellen Preises im Zeitschritt und der benötigten Menge in die Gesamtkosten ein. Entsprechend setzen sich die Kosten P für die benötigte Energie in jedem Simulationsschritt aus fixen und variablen Kosten nach Formel 1 zusammen:

$$\mathrm{P} = \frac{\mathrm{P}_{Anschaffung} + \mathrm{P}_{Betrieb}}{\mathrm{T}} + \mathrm{P}_{Spot} \tag{1}$$

$\mathrm{P}_{Anschaffung}$ = Anschaffungskosten
$\mathrm{P}_{Betrieb}$ = erwartete Betriebskosten
T = erwartete Lebensdauer
P_{Spot} = Stromkosten des Börsenstroms

4 Simulationsstudie

Um das Potential des Modells zu untersuchen, werden der jeweilige Strommix und die daraus resultierenden Stromkosten einer Produktion mit vier Gießeinheiten und identischem Produktionsplan unter Variation von Batteriespeicherkapazität (4.1), max. Leistung der PV-Anlage (4.2) und des geographischen Standortes (4.3) untersucht. Die durchgeführten Simulationen betrachten jeweils einen identischen Betriebstag (24h). Die Referenzkonfiguration für alle Studien stellt ein Betrieb mit einer 1 MWp PV-Anlage und 320 kWh Speicherkapazität in Nürnberg dar. Ziel dieser Simulationsstudien ist es, die Einsatzmöglichkeiten des entwickelten Modells zu zeigen.

4.1 Variation der Speicherkapazität

Die erste Teilstudie untersucht den Einfluss der Speicherkapazität. Diese wird in vier Abstufungen zwischen

0 und 640 kWh variiert. Abbildung 4 zeigt den jeweils resultierenden Energiemix und die damit zusammenhängenden Kosten.

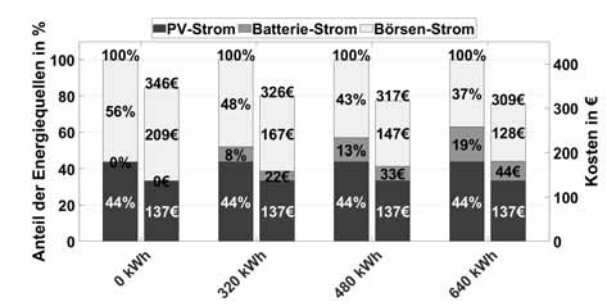


Abbildung 4: Strommix und -kosten für verschiedene Speicherkapazitäten.

In der durchgeführten Studie sinken die Gesamtkosten mit zunehmender Speicherkapazität, da ein größeres Speichervolumen mit geringeren Zusatzkosten verbunden ist, als der Strombezug am Spotmarkt. Das Ausmaß dieses Effekts hängt von den Investitionskosten des Speichers, sowie den Spotmarktpreisen im betrachteten Zeitraum, ab. Dementsprechend können Studien dieser Art im Falle einer bestehenden PV-Anlage eine Entscheidungshilfe für die optimale Dimensionierung einer Neu- oder Ersatzinvestition in einen Batteriespeicher darstellen.

4.2 Variation der PV-Anlage

Diese Simulationsstudie untersucht den Einfluss der PV-Anlagengröße. Diese wird in vier Abstufungen zwischen 100 kWp und 2 MWp variiert. Abbildung 5 zeigt den jeweils resultierenden Energiemix und die damit zusammenhängenden Kosten.

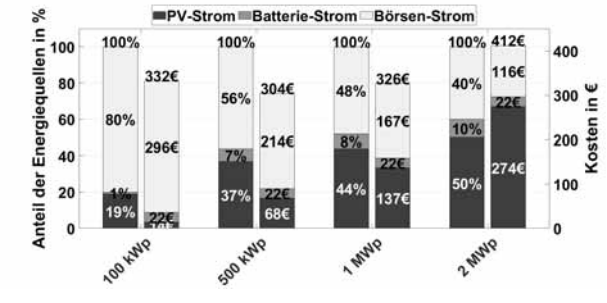


Abbildung 5: Strommix und -kosten für verschiedene PV-Anlagengrößen.

Obwohl die Konfiguration mit der größten PV-Anlage den meisten PV-Strom produziert, kann mit einer kleineren PV-Anlage (500 kWp) kostengünstiger produziert werden. Dies ist auf die begrenzte Speicherkapazität und die Produktion während der Nachtstunden zurückzuführen.

4.3 Variation des Standortes

Abschließend wird der Standort der Fabrik zwischen Bremen (BRE), Leipzig (LPZ), Frankfurt am Main (FRA) und Nürnberg (NBG) variiert. Simuliert wird mit der optimalen Konfiguration aus Simulationsstudie 4.2, also einer 500 kWp PV-Anlage und 320 kWh Speicherkapazität. Abbildung 6 zeigt den jeweils resultierenden Energiemix und die damit zusammenhängenden Kosten.

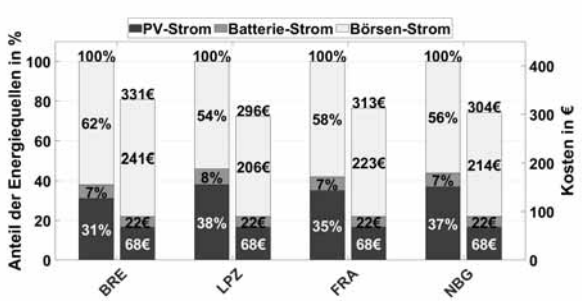


Abbildung 6: Strommix und -kosten für verschiedene Produktionsstandorte.

Die Studie zeigt, dass sich der Energiemix und die Energiekosten für ansonsten identische Betriebe an verschiedenen Standorten zum Teil signifikant unterscheiden können. Dies ist auf die unterschiedliche Sonneneinstrahlung und den daraus resultierenden unterschiedlichen PV-Ertrag zurückzuführen. Daraus lässt sich ableiten, dass optimale Anlagenkonfigurationen vom Standort des Betriebes abhängen, der daher im Simulationsmodell zu berücksichtigen ist. Die Unterstützung bei der Standortwahl ist ein weiteres potenzielles Anwendungsgebiet für solche Modelle. Für eine fundierte Aussage zur Standortwahl muss jedoch ein längerer Zeitraum simuliert werden.

5 Diskussion

Die wesentlichen Vorteile des hier beschriebenen Modells liegen vor allem in seinem flexiblen Aufbau und

in der Einbeziehung von realen Daten. Durch die flexible Modellgestaltung ist das Modell für eine Vielzahl von Anwendungen geeignet. Da das Modell mit realen Daten für Wetter und Strompreise simuliert, können auch zeit- und standortspezifische Einflussfaktoren in der Simulation berücksichtigt werden. Die vorgestellten Teststudien zeigen bereits exemplarisch die Einsatzfähigkeit des Modells für unterschiedlichste Anwendungen. So kann beispielsweise mit Hilfe des Modells für eine bestehende PV-Anlage ein optimal dimensionierter Speicher ermittelt werden und umgekehrt. Auch eine optimale Auslegung beider Komponenten lässt sich durchführen. Darüber hinaus berücksichtigt das Modell die spezifischen Wetterdaten, weshalb es sich flexibel für unterschiedliche Standorte anwenden lässt. Das Modell kann als Hilfestellung für die strategische Investitionsplanung, die Wirtschaftlichkeitsbewertung von Änderungen im Produktionsplan oder auch die Standortwahl genutzt werden. Da auch die spezifischen Anschaffungskosten im Modell variiert werden können, kann es auch zur Ermittlung derjenigen Grenzkosten, ab denen sich eine Investition lohnt, verwendet werden.

Da ein Simulationsmodell jedoch immer ein vereinfachtes Abbild der Realität darstellt, ist die Validierung anhand realer Daten eine wichtige Komponente, um die Aussagekraft eines Modells zu bestätigen. Aufgrund des Mangels an bereits elektrifizierten Gussbetrieben kann dies für das vorliegende Modell derzeit nicht erfolgen. Die aktuelle Validierung basiert daher auf Energiebilanzen und dem Vergleich mit einem bereits validierten Modell einer konventionellen Druckgießerei [3].

6 Ausblick

Perspektivisch soll das hier beschriebene Modell genutzt werden, um mit Hilfe von Optimierungsalgorithmen möglichst optimale Anlagenkonfigurationen und Produktionszeitpunkte für spezifische Betriebe zu ermitteln. Zudem kann das Modell aufgrund des flexiblen Aufbaus in viele Richtungen erweitert werden. Neben dem Einsatz in der operativen Produktionsplanung, soll insbesondere ein Einsatz in der strategischen Betriebsauslegung in weiteren Arbeiten näher untersucht werden. Darüber hinaus kann das Smart Grid Modell mit geringem Aufwand auch auf andere industrielle Anwendungsbereiche übertragen werden. Dazu ist lediglich ein Lastprofil des neuen Anwendungsfalles notwendig.

Literatur

[1] Statistisches Bundesamt. *Statistischer Bericht - Umwelt-ökonomische Gesamtrechnungen (UGR) - Energiegesamtrechnung - 2010 bis 2021*. 2023. Abgerufen am 13. Mai 2024 von https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Umwelt/UGR/energiefluesse-emissionen/Publikationen/Downloads/statistischer-bericht-ugr-energiegesamtrechnung-5850014217005.html.

[2] Dettelbacher J, Schlüter W, Buchele A. *Simulative Analyse der nachhaltigen Transformation von Gussbetrieben*. In: Bergmann, Feldkamp, Souren und Straßburger, editors. *Simulation in Produktion und Logistik 2023*. Ilmenau, 2023. https://doi.org/10.22032/DBT.57476

[3] Schlüter W, Henninger M, Buswell A, Schmidt J. Schwachstellenanalyse und Prozessverbesserung in Nichteisen-Schmelz- und Druckgussbetrieben durch bidirektionale Kopplung eines Materialflussmodells mit einem Energiemodell. In: Wenzel und Peter, editors *Simulation in Produktion und Logistik 2017*, Kassel, 2017.

[4] Hiller T, Mayerhoff J, Nyhuis P. Energy Costs in Production Planning and Control: A Categorical Literature Review and Comparative Analysis. *Journal of Production Systems and Logistics 2021*. Hannover 2021. https://doi.org/10.15488/11126

[5] Ewering C, Siebert R, Wortmann F, Youssef A. Process control with volatile electricity prices. *2014 5th International Renewable Energy Congress (IREC)*; Hammamet, Tunisia, 2014, pp. 1-5, https://doi.org/10.1109/IREC.2014.6827005

[6] Schultz C, Braun S, Braunreuther S, Reinhart G. Integration of Load Management into an Energy-oriented Production Control. *Procedia Manufacturing*. 2017. https://doi.org/10.1016/j.promfg.2017.02.017

[7] Willeke S, Prinzhorn H, Stonis M, Nyhuis P. Preconditions for applying an energy price-oriented sequencing rule. *Prod. Eng. Res. Devel.* 12, 73–81 2018. https://doi.org/10.1007/s11740-017-0782-z

[8] Roesch M, Linder C, Zimmermann R, Rudolf A, Hohmann A, Reinhart G. Smart Grid for Industry Using Multi-Agent Reinforcement Learning. *Applied Sciences*. 2020; 10(19):6900. https://doi.org/10.3390/app10196900

[9] Deutscher Wetterdienst. 2024. Verfügbar unter: `https://opendata.dwd.de/climate_environment/CDC/grids_germany/hourly/duett/radiation_global/recent/`

Analyse und Modellierung des Einflusses von ultrafeinen Partikeln auf die Entstehung von Starkregen auf der Basis von Open-Data Umweltdaten

Franziska Raabe[1], Stefanie Steinbichl[1], Jochen Wittmann[1]

[1]Hochschule für Technik und Wirtschaft, Wilhelminenhofstraße 75A, 12459 Berlin

Abstract. Die Untersuchung des Einflusses ultrafeiner Partikel (UFP) auf Starkregenereignisse zeigt eine komplexe Beziehung zwischen Atmosphärenzusammensetzung und Wetterphänomenen auf. Basierend auf einer Analyse von Starkregendaten aus der Region um den Flughafen Frankfurt am Main und UFP-Messwerten aus dem Hessischen Landesamt für Naturschutz, Umwelt und Geologie wird diskutiert, wie UFP den hydrologischen Kreislauf beeinflussen können. Obwohl eine direkte Korrelation nicht eindeutig festgestellt werden konnte, liefert die Tagesmittelwertanalyse Hinweise auf einen potenziellen Zusammenhang zwischen erhöhten UFP-Werten und bevorstehenden Starkregenereignissen. Weitere umfangreichere Analysen sind erforderlich, um diese Vermutungen zu bestätigen und die Auswirkungen von UFP auf den Wasserkreislauf besser zu verstehen.

Einleitung

Extremwetterereignisse wie Starkregen und anhaltende Dürreperioden nehmen weltweit zu. Eine wegweisende Studie von Forschenden des Karlsruher Instituts für Technologie und des unabhängigen Forschungsinstituts Airborne Research Australia untersucht den Zusammenhang zwischen Ultrafeinstaub bzw. ultrafeinen Partikeln (UFP) in der Atmosphäre und ihre invasive Wirkung auf den hydrologischen Kreislauf, insbesondere die Wolkenphysik und Regenbildung. Die Studie hebt hervor, dass anthropogene UFP-Emissionen eine Veränderung von Niederschlagsmustern verursachen und durch ihre Wirkung als Wolkenkondensationskeime Starkregenereignisse begünstigen können [1]. Zusätzlich wurde bereits belegt, dass sich die Anzahl an UFP in den letzten Jahrzehnten stark erhöht hat [2].

Unter Feinstaub versteht man luftgetragene Aerosolpartikel in Form von Staubteilchen oder Tröpfchen bestimmter Größe, die mehrere Tage in der Atmosphäre schweben können, bevor sie zu Boden sinken [3]. Je nach aerodynamischem Durchmesser der Partikel wird Feinstaub üblicherweise in die Kategorien PM_{10} (PM = particulate matter, Durchmesser < 10 µm) und $PM_{2,5}$ (Durchmesser < 2,5 µm) eingeteilt. UFP sind mit einem Durchmesser von maximal 100 nm im Vergleich bis zu tausendmal kleiner als PM_{10}-Teilchen und entstehen vor allem bei der Verbrennung fossiler Brennstoffe in Abgasreinigungsanlagen und Großfeuerungsanlagen sowie im Flug- und Schiffsverkehr. Aufgrund ihrer geringen Größe können UFP sehr tief in den menschlichen Körper bis in das Lungengewebe und den Blutkreislauf eindringen, wo sie mutmaßlich Lungen- und Herz-Kreislauf-Erkrankungen sowie Demenz, Alzheimer und Parkinson verursachen können [2][4][5].

Starkregen bezeichnet ein Wetterphänomen, bei dem in kurzer Zeit eine große Menge an Niederschlag meist aus konvektiver Bewölkung (wie z.B. Cumulonimbuswolken) auf kleinem Raum fällt. Dieser kann Überschwemmungen, Erdrutsche oder Sturzfluten verursachen, die dann wiederum enorme Schäden an Umwelt, Gebäuden und Infrastruktur zur Folge haben. Daher gibt der Deutsche Wetterdienst (DWD) beginnend bei Regenmengen von 15 Litern pro Quadratmeter innerhalb einer Stunde bzw. von 20 Litern in sechs Stunden Unwetterwarnungen heraus [6].

Die Auswirkungen des Klimawandels spielen eine nicht unerhebliche Rolle bei der Häufigkeit und dem Ausmaß dieser extremen Niederschläge. Laut einer Studie der World Weather Attribution hat sich die Wahrscheinlichkeit für das Auftreten von Starkregenereignissen in Deutschland und einigen angrenzenden Ländern

aufgrund der durch den Menschen verursachten globalen Erwärmung um das 1,2- bis 9-fache erhöht. Ebenso steigerte sich die Intensität von Starkregenfällen im Umfang von 3 bis 19 % im betrachteten Gebiet [7]. Die sich aufgrund des Kohlendioxidgehalt erwärmende Atmosphäre und ihre somit erhöhte Wasserdampfkapazität allein können die Zunahme und die hohe Variabilität in der Häufigkeit und der Verteilung von Starkregenfällen allerdings nicht zufriedenstellend erklären, da Kohlendioxid aufgrund seiner langen Lebensdauer räumlich relativ gleichmäßig verteilt ist. Hierfür ist die Einbeziehung des Wasserkreislaufs erforderlich [8].

1 Problemstellung

Aufgrund ihres Einflusses auf die Menge und Größe von Wassertröpfchen bei der Wolkenbildung in der Atmosphäre können UFP den hydrologischen Kreislauf stören, indem sie Niederschlag verzögern. Normalerweise sammeln sich Wassertröpfchen mit einem Durchmesser von ca. 0,01 bis 0,25 mm um einen Wolkenkondensationskern bzw. Cloud Condensation Nuclei (CCN), der selbst über eine Größe von ca. 0,0002 mm verfügt. Sobald sich genügend Wassertröpfchen akkumuliert und zusammen mit dem CCN eine Größe von 1 bis 2 mm erreicht haben, können diese als Regentropfen zur Erde fallen, da dann die Fallgeschwindigkeit größer ist als die Aufwindgeschwindigkeit in einer Wolke. Die UFP eignen sich allerdings nicht als CCN. Dies liegt zum einen an ihrer geringen Größe und zum anderen an ihrer stark gekrümmten Oberfläche, auf der das Wasser zu schnell verdunstet. So bleiben die akkumulierten Tropfen lange Zeit zu klein und können aufgrund des Luftwiderstands auch nicht abregnen. Durch diesen Vorgang entsteht ein zusätzliches Energiereservoir in der mittleren Troposphäre, das extreme Regenfälle begünstigt. Wenn sich die stark angereicherten Wolken schließlich abregnen, sind diese Niederschläge drastisch stärker und wasserreicher. Gebiete mit stark erhöhten UFP-Werten zeigen zunehmend extreme Starkregenfälle und weniger Gesamtniederschlag auf [1][2][9].

Im Folgenden soll anhand eines konkreten Beispiels untersucht werden, inwieweit erhöhte UFP-Werte in der Atmosphäre mit dem Auftreten von besonders intensiven Starkregenereignissen in einer ausgewählten Region zusammenhängen könnten. Als Untersuchungsgebiet dient die Region Flughafen Frankfurt am Main, zum einen, da Turbinen-Abgase der Flugzeuge bei Betrieb am Boden als ein Hauptemittent von UFP nachgewiesen wurden [10]. Zum anderen, da diese Region eine der wenigen in Deutschland ist, in der überhaupt kontinuierliche Messungen zu UFP-Werten durchgeführt werden. Dieser Sachverhalt ist nicht zuletzt auf die fehlende gesetzliche Regulierung von UFP-Emissionen zurückzuführen [11] .

Ziel der vorliegenden Untersuchung ist es, den beschriebenen Zusammenhang zwischen UFP und der Entstehung von Starkregen anhand von frei verfügbaren Daten nachzuvollziehen mit der Perspektive, mittelfristig online zugängliche Daten automatisch auszuwerten und in einem GIS-Dashboard zu visualisieren. So könnten Zusammenhänge umfassender analysiert und die Ergebnisse anschaulich präsentiert werden.

2 Datenmaterial

Die Erfassung von UFP-Werten erfordert eine spezielle Messtechnik, da die Teilchen aufgrund ihrer geringen Größe kaum zu den Partikelmassen der gröberen Feinstaubklassen $PM_{2,5}$ und PM_{10} beitragen [12]. Statt Massenbestimmung werden daher Vorrichtungen zur Partikelzählung eingesetzt [13]. Als Quelle für die in der Untersuchung verwendeten UFP-Werte diente das Messdatenportal des Hessischen Landesamts für Naturschutz, Umwelt und Geologie (HLNUG), das seit 2015 an einer variierenden Anzahl von Luftmessstationen UFP-Daten auf dem Gelände des Frankfurter Flughafens und in seiner Umgebung erhebt [14]. Dabei findet eine Kategorisierung nach Partikelgrößen zwischen 10 bis 500 nm statt. In der Untersuchung wurden die Partikelgrößen von 10 bis 100 nm betrachtet, da diese der gängigen Größendefinition von UFP entsprechen. Darunter wird dem Flugverkehr vor allem die Emission von UFP der Größe 10 bis 30 nm zugeschrieben [13].

Die für die Untersuchung verwendeten Starkregendaten stammen aus dem Katalog der Starkregenereignisse des DWD („Catalogue of Radar-based heavy Rainfall Events“ bzw. CatRaRE) [15]. In diesem Katalog sind alle Starkregenereignisse in Deutschland seit 2001 verzeichnet. Im Gegensatz zu großräumigen, länger anhaltenden Niederschlägen sind die örtlich und zeitlich meist eng begrenzten Starkregenfälle schwieriger zu messen. Um verwertbare Daten zu erhalten, verwendet der DWD daher Wetterradare, die Dauer und Intensität der Niederschläge flächendeckend erfassen können, auch abseits der regulären Messstationen durch permanentes Scannen der

Umgebung in großem Umfang um das Radargerät. Anschließend werden die Datensätze in einer mehrstufigen Qualitätskontrolle klimatologisch aufbereitet [16].

3 Methoden

Zunächst wurden die Starkregendaten in ArcGis importiert und nach dem ausgewählten Beobachtungszeitraum gefiltert. Aufgrund der besseren Datenkonsistenz wurde für die Untersuchung der Zeitraum vom 01.05.2020 bis 23.12.2022 gewählt. Zudem wurden auf Basis folgender Attribute die intensivsten Starkregenereignisse selektiert: maximaler Niederschlag innerhalb der Ereigniszone (RRmax), maximaler Starkregenindex (SRImax) und Dauerstufe des Ereignisses (Duration) (vgl. Abb.1).

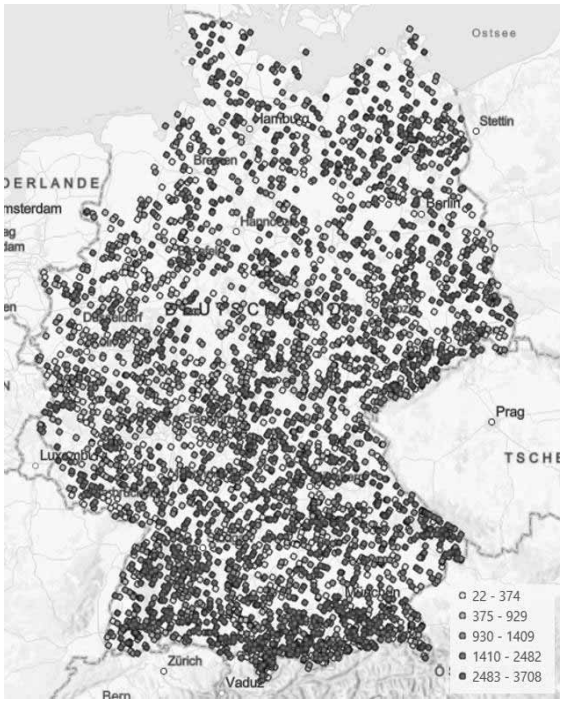


Abb. 1: Anzahl der intensivsten Starkregenereignisse in Deutschland vom 01.05.2020 bis 23.12.2022

Anschließend wurde um den Flughafen Frankfurt am Main ein Puffer von 50 km gelegt und dieser Polygon-Layer mit dem Punkte-Layer der ausgewählten Starkregenereignisse paarweise überschnitten (vgl. Abb. 2).

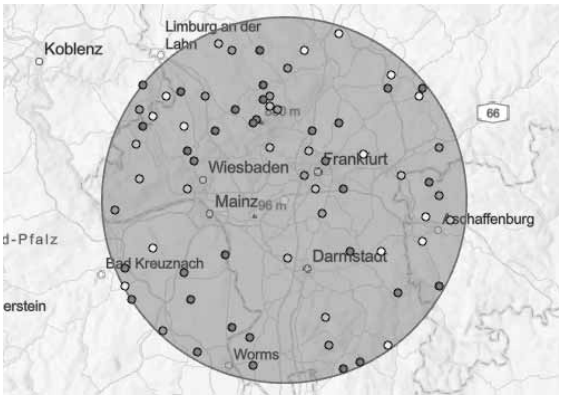


Abb. 2: Intersect von Puffer-Layer (50 km um Flughafen Frankfurt a.M.) und Starkregenereignissen

Aus den identifizierten intensivsten Starkregenereignissen in der betrachteten Region wurden im nächsten Schritt jeweils zwei Ereignisse für die Jahre 2020, 2021 und 2022 für die Stichprobenbetrachtung ausgewählt und die UFP-Messwerte der sieben Tage vor Eintreten des Starkregenereignisses importiert, wobei das Starkregenereignis im Laufe des siebten Tages (Tag 0) stattfand. Für jede Partikelgrößenkategorie wurde pro Messzeitraum das obere und untere Quartil sowie der jeweilige Tagesmittelwert bestimmt, um Aussagen über eine Verteilung der höheren und niedrigeren UFP-Messwerte in den Tagen vor dem Starkregenereignis treffen zu können (vgl. Abb. 3 bis 6).

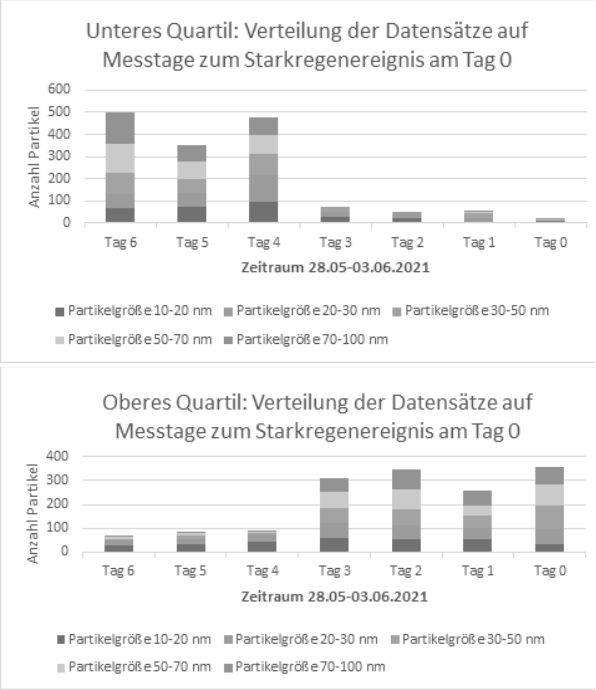


Abb. 3: Diagramme zur Verteilung der Datensätze je Messtag jeweils im unteren Quartil (Anteil Messungen mit geringsten UFP-Werten) und im oberen Quartil (Anteil Messungen mit höchsten UFP-Werten) für den Messzeitraum vom 28.05.2021 bis 03.06.2021 je Partikelgrößenkategorie

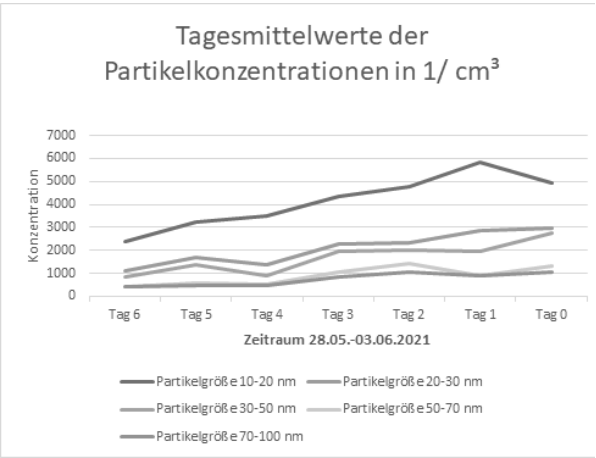


Abb. 4: Verlaufskurve der Tagesmittelwerte je Partikelgrößenkategorie für den Messzeitraum vom 28.05.2021 bis 03.06.2021

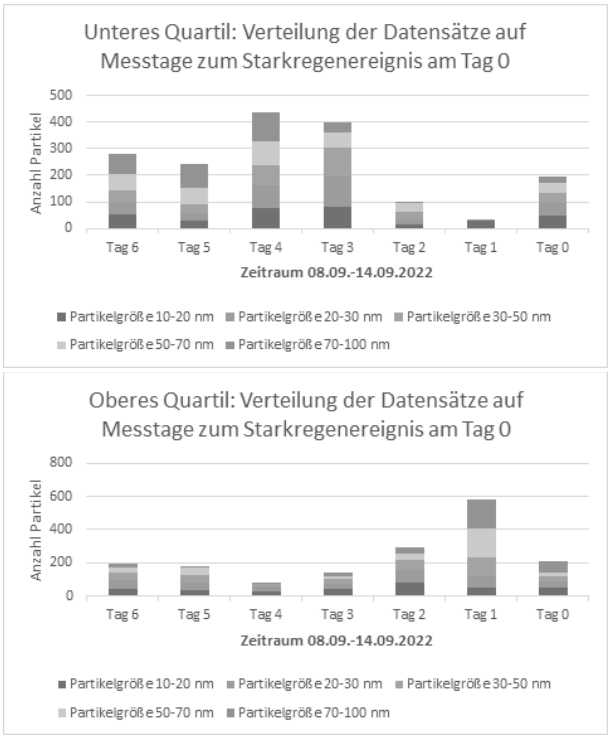


Abb. 5: Diagramme zur Verteilung der Datensätze je Messtag jeweils im unteren Quartil (Anteil Messungen mit geringsten UFP-Werten) und im oberen Quartil (Anteil Messungen mit höchsten UFP-Werten) für den Messzeitraum vom 08.09.2022 bis 14.09.2022 je Partikelgrößenkategorie

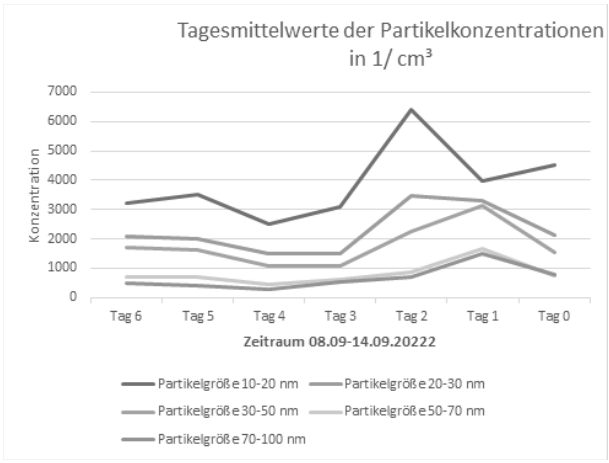


Abb. 6: Verlaufskurve der Tagesmittelwerte je Partikelgrößenkategorie für den Messzeitraum vom 08.09.2022 bis 14.09.2022

4 Resultate

In den exemplarisch dargestellen Messzeiträumen vom 28.05.2021 bis 03.06.2021 (vgl. Abb. 3 und 4) und vom 08.09.2022 bis 14.09.2022 (vgl. Abb. 5 und 6) verteilen sich die UFP-Werte in allen Partikelgrößenkategorien tatsächlich so, dass die geringeren Konzentrationen hauptsächlich zu Beginn des Zeitraums erfasst wurden (Linkslastigkeit im unteren Quartil) und die hohen Konzentrationen vermehrt zum Ende des Zeitraums (Rechtslastigkeit im oberen Quartil) auftreten. Dieses Phänomen zeigte sich auch in einigen der anderen analysierten Stichproben, allerdings nicht in einem Umfang, dass hieraus auf eine eindeutige Tendenz oder Regelmäßigkeit geschlossen werden konnte. Das Verhalten der Quartilserfassungen erwies sich vielmehr als unvorhersehbar, da sich die Messwerterfassungen je Datensatz auf sehr unterschiedliche Weise verteilten.

In der Analyse der Tagesmittelwerte der jeweiligen UFP-Konzentrationen konnte unter 30 erstellten Kurven in 20 ein Anstieg der Werte in zeitlicher Nähe zum Starkregenereignis festgestellt werden. Dabei unterschieden sich die Kurven aber in ihrer Erscheinung. Einige wiesen z.B. Höhepunkte bereits zu Beginn des betrachteten Messzeitraums auf und enthielten somit eine Senke in der Mitte des Zeitraums. Andere stiegen kontinuierlich an, um dann zum Ende des Zeitraums wieder abzufallen. Generell konnte allerdings als Tendenz beobachtet werden, dass die meisten Kurven ein bis zwei Tage vor Aufkommen des Starkregenereignisses den höchsten Tagesmittelwert innerhalb des Messzeitraums aufwiesen.

5 Schlussfolgerungen

Ein direkter Zusammenhang zwischen erhöhter UFP-Belastung in der Atmosphäre und dem Auftreten von Starkregenereignissen konnte im Rahmen der vorliegenden Untersuchung nicht hinreichend nachgewiesen werden. Aufgrund der Komplexität von Wetter- bzw. Klimaphänomenen sind weitaus umfangreichere Analysen notwendig, die die Einbeziehung anderer beeinflussender Faktoren wie Höhendaten, Geländeprofile, Temperatur, genereller Niederschlag, Luftdruck oder Windbewegungen erfordern. Beispielsweise nehmen Wind und Niederschlag Einfluss auf die lokalen UFP-Konzentrationen. Ebenso können sich Wolken je nach Windstärke und -geschwindigkeit auch hunderte Kilometer entfernt abregnen, der Starkregen muss demnach nicht direkt in der Region mit erhöhten UFP-Konzentrationen niedergehen [9].

Sicherlich kann insbesondere die Tagesmittelwertanalyse als Indiz dafür betrachtet werden, dass sich in der Zuspitzung auf ein Starkregenereignis eine Tendenz des Anstiegs von ermittelten UFP- Emissionen abzeichnet. Diese Untersuchung müsste allerdings in einem deutlich größeren Umfang stattfinden, um diese Vermutung zu bestätigen. Darüber hinaus sind für eine genauere Untersuchung auch Analysen an unterschiedlichen Orten notwendig, um nicht nur lokale Vergleiche anzustellen, sondern Ergebnisse aus unterschiedlichen Regionen mit sich ähnelnden geografischen Bedingungen miteinander in Beziehung setzen zu können.

Um eine umfassende Analyse der Zusammenhänge zwischen UFP-Belastung und Starkregenereignissen zu ermöglichen, könnte sich die Implementierung eines automatisierten Systems zur Auswertung und Visualisierung frei zugänglicher Daten über ein GIS-Dashboard eignen. Diese Herangehensweise würde es ermöglichen, nicht nur einzelne Datenpunkte zu untersuchen, sondern auch komplexe Muster und Trends in den Daten zu identifizieren sowie andere räumliche und zeitliche Skalierungen für eine Modellierung vorzunehmen. Dadurch könnten Zusammenhänge klarer dargestellt und besser verstanden werden, was wiederum eine fundiertere Grundlage für weitere Untersuchungen und Entscheidungsfindungen bieten würde. Darüber hinaus könnte ein solches System auch die Kommunikation und den Austausch von Informationen zwischen verschiedenen Interessengruppen erleichtern, was entscheidend ist für die Entwicklung von Maßnahmen zur Eindämmung der potenziellen Auswirkungen von UFP auf den Wasserkreislauf und die Umwelt insgesamt.

Nicht zuletzt wegen der seit Jahren steigenden UFP-Belastung soll mit dieser Untersuchung eine erste Grundlage und ein Anreiz für weitergehende Analysen gelegt werden, um die möglicherweise drastischen Auswirkungen von UFP auf den Wasserkreislauf zu ermitteln und darauf aufbauend diese gezielt einschränken zu können, beispielsweise mit der Festlegung eines offiziellen UFP-Grenzwerts.

Literaturverzeichnis

[1] Wolfgang Junkermann, Jorg Hacker. Unprecedented levels of ultrafine particles, major sources, and the hydrological cycle. *Sci Rep*. 2022; 12(7410): 1-8. doi: 10.1038/s41598-022-11500-5.

[2] Alexander Freund. *Ultrafeinstaub für Wetterextreme mitverantwortlich*. Deutsche Welle. 2022. https://www.dw.com/de/ultrafeinstaub-f%C3%BCr-wetterextreme-mitverantwortlich/a-61903127. Stand: 06.09.2023.

[3] Stefan Kinne, Johannes Quaas. Einfluss der Luftverschmutzung auf den Klimawandel – neue Erkenntnisse aus Satellitendaten und Klimamodellen. *Forschungsbericht 2007. Max Planck Institut für Meteorologie.* https://www.mpg.de/405815/forschungsSchwerpunkt. Stand: 06.09.2023.

[4] Initiative Klima-, Umwelt- und Lärmschutz e.V.. *Feinstaub und noch feinerer Staub – Ultrafeinstaub*. o.J.. https://www.klima-umwelt-luftverkehr.de/umwelt/ultrafeinstaub/. Stand: 22.08.2023.

[5] Umweltbundesamt. *Feinstaub*. 2022. https://www.umweltbundesamt.de/themen/luft/luftschadstoffe-im-ueberblick/feinstaub#undefined. Stand:22.08.2023.

[6] Deutscher Wetterdienst. *Wetter und Klimalexikon. Starkregen*. o.J.. https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?nn=103346&lv2=102248&lv3=102572. Stand: 04.09.2023.

[7] Frank Kreienkamp et al.. Rapid attribution of heavy rainfall events leading to the severe flooding in Western Europe during July 2021. *World Weather Attribution*. 2021: 1-51. https://www.worldweatherattribution.org/wp-content/uploads/Scientific-report-Western-Europe-floods-2021-attribution.pdf. Stand: 04.09.2023.

[8] Karlsruher Institut für Technologie. Klimaforschung: Ultrafeinstaub könnte Wetterextreme verursachen. *Presseinformation 044/2022*. 2022. https://www.kit.edu/kit/pi_2022_044_klimaforschung-ultrafeinstaub-konnte-wetterextreme-verursachen.php. Stand: 06.09.2023.

[9] Marina Weishaupt. Dürre und Starkregen: Fossile Brennstoffe verändern Wasserkreislauf. *National Geographic*. 2022. https://www.nationalgeographic.de/umwelt/2022/05/duerre-und-starkregen-fossile-brennstoffe-veraendern-wasserkreislauf. Stand: 06.09.2023.

[10] Umweltbundesamt. *Einfluss eines Großflughafens auf zeitliche und räumliche Verteilungen der Außenluftkonzentrationen von Ultrafeinstaub < 100 nm, um die potentielle Belastung in der Nähe zu beschreiben - unter Einbeziehung weiterer Luftschadstoffe*. 2021. https://www.umweltbundesamt.de/sites/default/files/medien/11850/publikationen/14_2021_texte_grossflughafen_.pdf. Stand: 25.04.2024.

[11] Umweltbundesamt. *Fragen und Antworten: Ultrafeine Partikel*. 2018. https://www.umweltbundesamt.de/themen/luft/luftschadstoffe-im-ueberblick/feinstaub/fragen-antworten-ultrafeine-partikel. Stand: 25.04.2024.

[12] Hessisches Landesamt für Naturschutz, Umwelt und Geologie. *Ultrafeine Partikel*. 2023. https://www.hlnug.de/themen/luft/luftqualitaet/sondermessprogramme/ultrafeine-partikel. Stand: 06.09.2023.

[13] Karl-Heinz Peil. Ultrafeine Partikel (UFP): Flugverkehr und Gesundheit. *Fakten - kurz und bündig erstellt vom BUND AK Luft/Klima/Lärm Hessen*. 2021. https://www.bund-hessen.de/fileadmin/hessen/Publikationen/Arbeitskreise/AK-Luft-Klima-Laerm/2021-02_BUND-Hessen_FactSheet_Ultrafeine-Partikel_final.pdf. Stand: 06.09.2023.

[14] Hessisches Landesamt für Naturschutz, Umwelt und Geologie. Messdatenportal. 2023. https://www.hlnug.de/messwerte/datenportal/ufp. Stand: 06.09.2023.

[15] Katharina Lengfeld, Ewelina Walawender, Tanja Winterrath, Elmar Weigl, Andreas Becker. *Starkregenereignisse Version 2023.01 mit Überschreitung der DWD-Warnstufe 3 für Unwetter basierend auf RADKLIM-RW Version 2017.002 Parameter und Polygone der Starkregenereignisse in Deutschland version v2023.01.*. Deutscher Wetterdienst. 2023. doi:10.5676/DWD/CatRaRE_W3_Eta_v2023.01.

[16] Tanja Winterrath. Niederschlagsklimatologie: Daten und Produkte - Mit Radardaten in die nächste Dimension. Deutscher Wetterdienst. 2021. https://www.dwd.de/SharedDocs/broschueren/DE/klima/radklim_broschuere.pdf?__blob=publicationFile&v=2. Stand: 06.09.2023.

Terrain Identification using Reaction-based Sensor Data in Simulation-driven Terrain-aware Military Logistics

Mihaela Lechner[1*], Oliver Rose[1]

[1]Chair of Modeling and Simulation, University of the Bundeswehr Munich,
Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany; *mihaela.lechner@unibw.de*

Abstract. Military planning operations require navigating constantly changing environments. To support decision-makers, innovative concepts are essential for automatically generating effective solutions tailored to specific logistics operations. These tools aim to accelerate planning procedures, minimize risks, and decrease operating costs. This paper introduces a simulation-based optimization framework designed to enhance the mobility of military vehicles through terrain-aware navigation. The paper specifically delves into a key component of the framework: terrain identification. This challenge is addressed using unsupervised methods, ensuring applicability even in unfamiliar operational settings. The experimental findings demonstrate promising results in identifying terrain characteristics, particularly in discerning surface waviness, slant, and curvature.

Introduction

The mobility of supplies, equipment, and personnel is crucial to the success of land-based military missions. Unlike civilian logistics, which often prioritize the shortest and quickest routes, military operations must consider factors such as environment uncertainty [1], route vulnerability [2], and terrain passability [3] when determining the most suitable logistics routes.

Furthermore, military operations often extend across geographically diverse regions, the condition of the terrain having a direct impact on their effectiveness [3]. Therefore, planners must carefully assess terrain characteristics such as landform features, soil conditions, and slope degree when preparing military logistics plans. The terrain encountered by military land vehicles often falls outside typical mapped areas, leaving planners with little information regarding its topology. In such scenarios, battlefield commanders lean on terrain analysts to interpret geographic features of an area and assess their impact on the military mission [4].

Over time, the process of terrain analysis evolved from a predominantly manual endeavor to one increasingly reliant on computer-based systems [5]. One facet of terrain analysis that can be solved through computational means is terrain identification. This field of research involves estimating ground characteristics (e.g., cohesion, curvature, inclination) or categorizing terrain types (e.g., gravel, asphalt, sand) by gathering diverse sensor data under various road conditions and analyzing vehicle responses to the terrain.

Numerous researchers have made significant contributions to terrain identification methodologies. Among these, supervised learning techniques such as Support Vector Machine [6, 7], Decision Tree [8], Neural Network [9, 10], or Gaussian Process Regression [11] have emerged as popular choices. Although these approaches have proven effective, they require prior human intervention or additional hardware, such as laser line striping sensors, for data labeling. Conversely, unsupervised approaches do not necessitate labeled data and can be directly applied in scenarios where the external environment is unknown.

In addition to the configuration of the learning algorithm, the accuracy of the terrain identification strategy depends on the data it receives. Various sensors can be mounted to the vehicle to gather this data. Cameras [12, 13], lidars [14, 15], and accelerometers [6, 16, 17] stand out as prominent choices in recent research. Each sensor type comes with its limitations [18]. For instance, vision-based sensors like cameras and lidars are sensitive to weather conditions that reduce visibility, such as fog or rain, whereas reaction-based sensors like accelerometers are sensitive to speed and load variations. Despite this disadvantage, reaction-based tech-

niques demonstrate great cost-effectiveness and robustness across diverse terrain types [19].

This study focuses primarily on solving the terrain identification problem, aiming to differentiate distinct terrain characteristics such as roughness, waviness, slant, and curvature. The approach involves conducting multiple test drives within military test sites to collect reaction-based data, including acceleration, roll, pitch, and angular rate, captured by an accelerometer and a gyroscope. Initially, the signal data undergoes windowing, followed by segmenting each route into predetermined lengths. Subsequently, the unsupervised learning algorithm Multivariate K-Means is utilized to differentiate between different terrain characteristics. We employ the Dynamic Time Warping (DTW) algorithm to calculate the pairwise proximity between the road segments.

Moreover, this research introduces a simulation-driven logistics framework that combines terrain identification, scheduling, and vehicle routing processes to assist path planners in conceiving terrain-aware logistics strategies. The plans generated by this framework are designed to optimize the utilization of available asset capacities by considering surface characteristics when determining efficient transportation routes. Within the broader logistics landscape, this approach presents an opportunity to improve operational efficiency and achieve substantial cost savings. In addition to immediate reductions in fuel and personnel expenses, it can also play a role in lowering long-term vehicle maintenance costs. This is achieved by implementing intelligent routing strategies that minimize vehicle wear and tear, ultimately extending their lifespan and decreasing the frequency of repairs and replacements.

The structure of the paper is as follows. Section 1 outlines the logistics framework. The method proposed for terrain identification is detailed in Section 2. Section 3 discusses the findings of the terrain identification process. Finally, Section 4 provides a summary of our conclusions and outlines future work.

1 CONCEPTUAL APPROACH OF A SIMULATION-BASED TERRAIN-AWARE LOGISTICS FRAMEWORK

Developing military logistics strategies presents a significant challenge in optimizing asset scheduling and route selection for efficiently transporting personnel, equipment, and supplies to designated destinations. This challenge is heightened by the absence of basic infrastructure at certain locations and the diverse terrain conditions encountered during transit. Additionally, different vehicles are tailored for navigating specific types of terrain. Some are designed for rough, steep terrain with obstacles, while others perform better on smooth, paved roads. To ensure effective and efficient transportation operations, it is essential to consider the mobility capabilities of vehicles across various surfaces, alongside critical logistics factors such as route length, transport duration, and delivery time requirements.

To overcome these challenges, we introduce the simulation-based logistics framework depicted in Figure 1. The primary objective of this framework is to assist planners in developing efficient military transportation systems by focusing on sustainable resource management and enhancing the mobility of military vehicles in favorable terrain conditions.

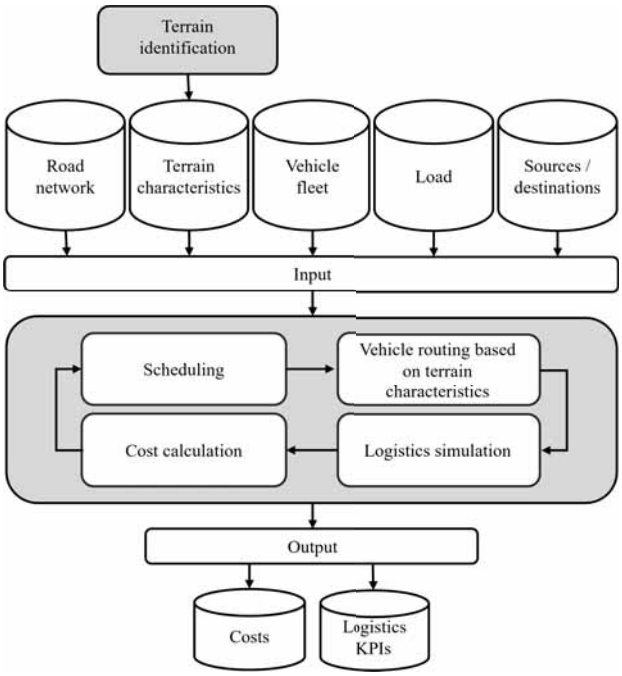


Figure 1: Conceptual model of the proposed simulation-driven terrain-aware logistics framework.

The framework begins by prioritizing the identification of terrain characteristics along the routes. These details, along with information on road networks, vehi-

cle availability, and load requirements for transportation between origin and destination points, serve as inputs.

Subsequently, the framework proceeds to optimize fleet vehicle utilization. The scheduling component determines which load should be transported by each vehicle and in what sequence, aiming to minimize costs while meeting constraints such as vehicle capacities.

Following scheduling, the routing process utilizes the scheduled assets to establish logistics routes. This process extends beyond selecting the shortest and fastest paths by considering terrain conditions. As certain terrains disproportionately affect vehicle performance, wear, and tear, selected routes must correspond to the mobility characteristics of the transporter.

During the simulation phase, logistics plans are executed, and the behavior of simulation agents is monitored. Each transportation task is evaluated using a cost function designed to minimize both transportation duration and expenses, taking into account travel feasibility on appropriate surfaces.

Acknowledging the critical role that terrain characteristics play in terrain-aware logistics, the topic of terrain identification will be explored throughout the remainder of this paper.

2 TERRAIN IDENTIFICATION

This section explores terrain identification, a key component of the logistics framework detailed in Section 1. This process is essential for enabling the computation of terrain-aware logistics routes.

2.1 Problem Description

We address the challenge of terrain identification utilizing reaction-based sensor measurements. Our primary objective is to differentiate specific terrain characteristics such as roughness (Figure 2a), waviness (Figure 2b), slant (Figure 2c), and curvature (Figure 2d), even in situations where prior knowledge about the terrain is limited. This is achieved through analyzing unique signal patterns captured by standard sensors like accelerometers and gyroscopes, which record the dynamic interaction between the vehicle and the terrain. To accomplish this task, we introduce the technique detailed in Section 2.2.

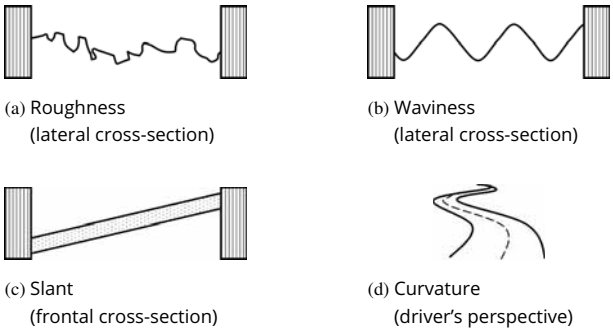


Figure 2: Terrain characteristics under investigation.

2.2 Solution Approach

We propose the methodology illustrated in Figure 3 for accomplishing terrain identification. This approach relies on data acquired from reaction-based sensors during vehicle operation on different road surfaces. In the preprocessing phase, the input data is subjected to windowing and segmentation to create frames used for feature generation. Subsequently, the unsupervised learning technique Multivariate K-Means is applied to identify different terrain characteristics.

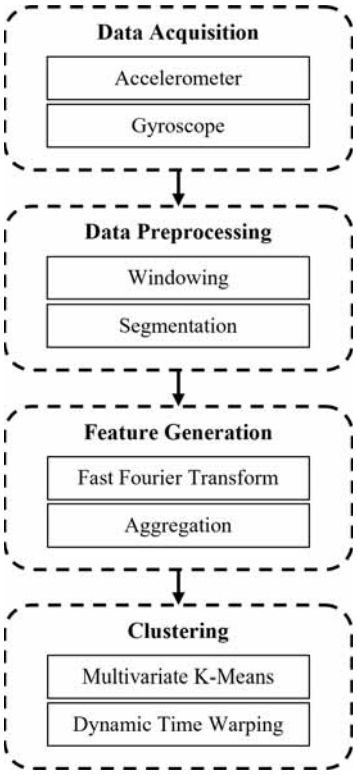


Figure 3: Proposed terrain identification methodology.

In the upcoming paragraphs, each component of the terrain identification approach will be elaborated.

Data Acquisition Over the course of 24 test runs at a military test site, data was gathered from multiple ground surfaces that exhibit different degrees of roughness, waviness, slant, and curvature. For this purpose, a military vehicle covered about 500 km, equipped with an accelerometer, a tri-axial gyroscope, and a global positioning system (GPS). Each sensor recorded data at a sampling rate of 500 Hz.

To overcome the speed dependency limitation of reaction-based terrain identification, the vehicle was driven at different speeds ranging from 5 to 45 km/h.

Data Preprocessing The preprocessing phase involves two key steps: windowing and segmentation.

Windowing is a technique essential for transforming the sequential data, such as the dataset under consideration, into a format that suits traditional machine learning algorithms [20]. Additionally, it helps reduce computational complexity. This process involves dividing the sensor data into non-overlapping frames, with each frame consisting of 500 samples, corresponding to one second of data given a sampling frequency of 500 Hz.

Clustering entire routes poses challenges in detecting local similarities among them. Conversely, clustering each observation separately fails to generate cohesive patterns and instead scatters the clusters across multiple terrain categories. To address this dilemma, we choose to partition each test drive into segments measuring 40 m, approximately five times the length of the vehicle. Each of these partitions is treated as an individual observation.

Feature Generation The sensor data in time domain, including tri-axal acceleration, tri-axal rotation rate, roll, and pitch is converted into the frequency domain using the Fast Fourier Transform (FFT) algorithm. Features are extracted by considering observations from both the original time domain representation and its frequency domain transformation within previously generated windows. Each window is aggregated to an individual output value by computing statistical measures such as mean, standard deviation, minimum, maximum, and interquartile range. In total, this process yields 80 features.

Clustering We approach the task of terrain identification by examining similar patterns within segments of routes traversed by vehicles. Since each segment contains multiple observations, the problem inherently becomes multivariate. To handle this complexity, we utilize Multivariate K-Means clustering. While deep learning clustering techniques could also be applied, they tend to be complex, challenging to interpret and can generate high computational costs. However, the K-Means method also has its limitations, particularly its sensitivity to the choice of the cluster number k. To address this issue, we employ the Silhouette Coefficient, introduced in [21], to determine an optimal number of 9 clusters.

In the clustering process, we use the DTW proximity measure, a technique proposed in [22]. This method offers advantages over the conventional Euclidean distance by effectively recognizing similarities within sequences, even in cases where they differ in length or experience slight temporal shifts.

For enhanced visualization and evaluation of the clustering results, we adopt the Multivariate T-distributed Stochastic Neighbor Embedding (m-TSNE) technique introduced by [23]. This approach enables the projection of multivariate high-dimensional data onto a lower-dimensional space while maintaining the similarity relationships between the data sequences. Consequently, sequences that are similar in high-dimensional space also remain proximate in the lower-dimensional space.

3 EXPERIMENTAL RESULTS

The solution described above has been executed and evaluated in Python 3.11.5 on a typical PC operating on Windows 11, equipped with an 11th generation Intel Core i7-11370H CPU running at 3.30 GHz and 16 GB of RAM. Training the model on a preprocessed dataset of 200MB size requires approximately 15 minutes. The complexity of training arises from the significant number of pairwise similarities that need to be computed, specifically $\binom{N}{2}$, where N represents the count of route segments.

The 24 trips are partitioned into approximately 12 000 segments, each one assigned to one of 9 clusters via the Multivariate K-Means algorithm, utilizing 80 features. To enhance visualization of the high-dimensional space, the data is reduced to two dimensions using the m-TSNE method, as shown in Figure 4.

Each point in the plot corresponds to a route segment, revealing discernible separation patterns among groups. While certain groups, particularly those at the periphery, exhibit distinct isolation from others, the observations in the central regions lack clear boundaries. The clustering method captures the underlying pattern, with only a few instances dispersed across multiple groups in the 2D space.

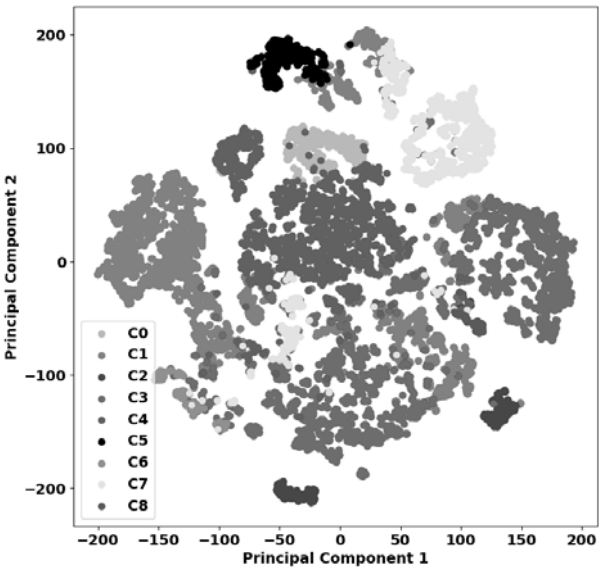
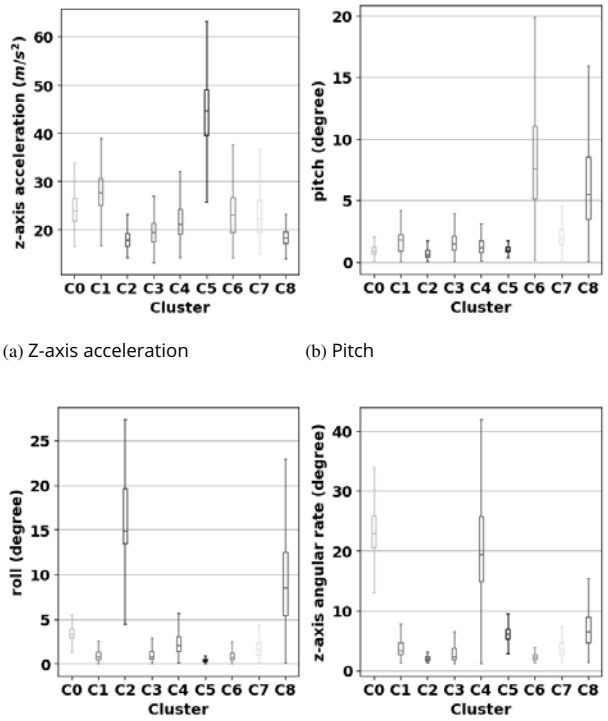


Figure 4: Representation of the two-dimensional m-TSNE components depicting route segments clustered based on the Multivariate K-Means method.

Figures 5a-5d illustrate the key features essential for cluster formation. Each displayed feature undergoes aggregation through windowing using the mean function, as detailed in Section 2.2, derived from frequency domain transformations. Analyzing these plots enables the characterization of clusters based on the distinct terrain traits outlined in Figure 2. High signal magnitudes emphasize the presence of particular terrain characteristics, while lower magnitudes indicate their absence.

The accelerometer supports the measurement of the vehicle's vertical displacement relative to the ground (z-axis acceleration), facilitating the evaluation of terrain roughness. Notably, cluster $C5$ stands out for exhibiting rough terrain, as evident in Figure 5a. Waviness, on the other hand, involves larger repetitive bumps compared to roughness, resulting in a rocking motion in the vehicle rather than just vertical acceleration. These movements are detected through pitch measurements of the gyroscope. As depicted in Figure 5b, clusters $C6$ and $C8$ highlight wavy terrain characteristics. Surface slant, indicative of tilts to the right or left, is discernible via the roll signal. Slanted terrain is observable in clusters $C2$ and $C8$ from Figure 5c. Furthermore, the gyroscope can capture the rotational motion of the vehicle, reflecting road curvature, as evident in clusters $C0$ and $C4$ from Figure 5d. The remaining clusters lack distinctive terrain characteristics based on the examined features. This suggests that the road segments within these clusters have likely smooth, straight surfaces. An overview of the characteristics exhibited by each cluster can be found in Table 1.



(a) Z-axis acceleration (b) Pitch

(c) Roll (d) Z-axis angular rotation

Figure 5: Selection of features employed in the clustering procedure, presented individually for each cluster. These features were derived from raw signals transformed into the frequency domain and aggregated using the mean function during the windowing process.

Since the test drives were conducted on a specialized test course, certain segments of the underlying surfaces have known labels. For instance, cluster $C2$ represents the inclined test track featuring an incline ranging from 20% to 30%. Cluster $C5$ encapsulates the washboard test track, while the sine-wave road is identifiable

	Roughness	Waviness	Slant	Curvature
C0	-	-	-	✓
C1	-	-	-	-
C2	-	-	✓	-
C3	-	-	-	-
C4	-	-	-	✓
C5	✓	-	-	-
C6	-	✓	-	-
C7	-	-	-	-
C8	-	✓	✓	-

Table 1: Summary of the terrain characteristics observed within the clusters.

within cluster $C6$. Cluster $C7$ encompasses cobblestone and gravel. Lastly, cluster $C8$ delineates the distortion road, characterized by alternating waves on each side.

Table 2 outlines the comparison between the clustering results and the known true labels. Upon examining the model's performance across the clusters, it becomes evident that the model shows higher performance for certain road types. Specifically, the sloped ($C2$) and the distorted ($C8$) roads are recognized with high precision, recall, and F1-score. However, the model's performance is less satisfactory in identifying the wash-board ($C5$) and the sine-waved ($C6$) tracks.

Cluster	Precision	Recall	F1-score
C2	0.99	0.95	0.97
C5	0.92	0.35	0.51
C6	0.86	0.49	0.62
C7	0.93	0.82	0.87
C8	0.93	0.93	0.93

Table 2: Summary of the model performance.

4 CONCLUSION AND OUTLOOK

In military operations, terrain-aware logistics are crucial, especially when navigating through challenging landscapes with sparse infrastructure to transport supplies, equipment, and personnel. In such contexts, logistics planning must encompass not only factors like travel distance, duration, and delivery schedules but also take into consideration the unique characteristics of the terrain traversed.

In response to this necessity, this research introduces a simulation-driven terrain-aware framework designed to support decision-makers in improving the mobility of military vehicles by enabling them to navigate more efficiently through favorable terrain conditions. The primary focus of this paper is the terrain identification process, which utilizes unsupervised methods to distinguish between different terrain characteristics even in the absence of prior knowledge about the surface conditions. The experimental findings demonstrate promising results in discerning roughness, waviness, slant, and curvature through reaction-based signals. Each terrain characteristic is represented by a dominant signal, for instance high magnitudes of the z-axis acceleration signal indicate rough terrains. Additionally, terrains with multiple characteristics can be identified by considering multiple signals; for example, higher magnitudes in the pitch and roll signals suggest a wavy and slanted road.

Despite its effectiveness, this approach requires careful consideration in certain areas. As noted in previous research [18], reaction-based terrain identification is sensitive to the speed and load of the vehicle, causing terrain signatures to vary under different operating conditions. For accurate identification, the algorithm needs to be trained with a diverse dataset that includes a wide range of speeds and loads. Additionally, unsupervised learning, though valuable when no prior knowledge of terrain conditions is available, requires human interpretation of the results. Defining thresholds for specific signals indicating particular road features is essential for precise categorization. Furthermore, the current approach focuses on identifying terrain features but does not quantify their intensity. Future work should incorporate a scoring system to evaluate terrain surfaces based on their characteristics. Identifying specific surface types, such as concrete, grass, or soil, would also significantly enhance the optimization of logistics route planning.

While this paper emphasizes terrain identification, it is imperative to implement the subsequent steps of the framework to fully realize its potential in terrain-aware logistics. This includes integrating processes for fleet scheduling, route planning based on terrain features, and simulation-based evaluation to refine and optimize military logistics operations.

Acknowledgement

We express our gratitude to the Bundeswehr Office for Defence Planning, particularly Ferdinand Rinscheid, for their invaluable assistance, generous support, and willingness to share their knowledge and expertise.

References

[1] Zhao T, Huang J, Shi J, Chen C. Route Planning for Military Ground Vehicles in Road Networks under Uncertain Battlefield Environment. *Journal of Advanced Transportation*. 2018;2018(2):1–10.

[2] Muckensturm J, Longhorn D. Assessing the Vulnerability of Military Theater Distribution Routes. *Journal of Defense Analytics and Logistics*. 2019; 3(1):60–82.

[3] Dawid W, Pokonieczny K. Methodology of Using Terrain Passability Maps for Planning the Movement of Troops and Navigation of Unmanned Ground Vehicles. *Sensors*. 2021;21(14).

[4] Headquarters Department of The Army. *ATP 2-01.3 Intelligence Preparation of the Battlefield / Battlespace*. Createspace Independent Publishing Platform. 2017.

[5] Graff LH. State-of-the-Art Terrain Analysis Capabilities for Today's Army. 1996.

[6] Weiss C, Fröhlich H, Zell A. Vibration-based Terrain Classification Using Support Vector Machines. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: IEEE. 2006; pp. 4429–4434.

[7] Oliveira FG, Santos ERS, Neto AA, Campos MFM, Macharet DG. Speed-invariant Terrain Roughness Classification and Control Based on Inertial Sensors. In: *2017 IEEE Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. Curitiba, Brazil: IEEE. 2017; pp. 1–6.

[8] Beilfuss T, Kortmann KP, Wielitzka M, Hansen C, Ortmaier T. Real-Time Classification of Road Type and Condition in Passenger Vehicles. *IFAC-PapersOnLine*. 2020;53(2):14254–14260.

[9] Csìk D, Odry À, Sàrosi J, Sarcevic P. Inertial Sensor-based Outdoor Terrain Classification for Wheeled Mobile Robots. In: *2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY)*. Subotica, Serbia: IEEE. 2021; pp. 159–164.

[10] Sunusi II, Zhou J, Sun C, Makange N. Online Terrain Parameter Estimation for Traction Control in Intelligent Electric Tractors. *International Journal of Electrical Engineering*. 2023;29(4):97–107.

[11] Inotsume H, Kubota T. Terrain Traversability Prediction for Off-road Vehicles Based on Multi-source Transfer Learning. *Robomech Journal*. 2022;9(1):6–31.

[12] Chetan J, Madhava Krishna K, Jawahar CV. An Adaptive Outdoor Terrain Classification Methodology Using Monocular Camera. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei, Taiwan: IEEE. 2010; pp. 766–771.

[13] Gao B, Zhao X, Zhao H. An Active and Contrastive Learning Framework for Fine-grained Off-road Semantic Segmentation. *IEEE Transactions on Intelligent Transportation Systems*. 2023; 24(1):564–579.

[14] Andersen JC, Blas M, Ravn O, Andersen NA, Blanke M. Traversable Terrain Classification for Outdoor Autonomous Robots Using Single 2D Laser Scans. *Integrated Computer-Aided Engineering*. 2006; 13(3):223–232.

[15] McIver CA, Metcalf JP, Olsen RC. Spectral LiDAR Analysis for Terrain Classification. In: *Laser Radar Technology and Applications XXII*. Anaheim, CA, USA: SPIE. 2017; p. 101910J.

[16] Souza JR, Marchant R, Ott L, Wolf DF, Ramos F. Bayesian Optimisation for Active Perception and Smooth Navigation. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE. 2014; pp. 4081–4087.

[17] Wang M, Ye L, Sun X. Adaptive Online Terrain Classification Method for Mobile Robot Based on Vibration Signals. *International Journal of Advanced Robotic Systems*. 2021;18(6):172988142110620.

[18] Coyle E, Collins EG, Roberts RG. Speed Independent Terrain Classification Using Singular Value Decomposition Interpolation. In: *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China: IEEE. 2011; pp. 4014–4019.

[19] Nampoothiri H, Vinayakumar B, Sunny Y, An R. Recent Developments in Terrain Identification, Classification, Parameter Estimation for the Navigation of Autonomous Robots. *SN Applied Sciences*. 2021; 3(4):480.

[20] Dietterich T. Machine Learning for Sequential Data: A Review. In: *Structural, Syntactic, and Statistical Pattern Recognition*. Ontario, Canada: Springer. 2002; pp. 15–30.

[21] Kaufman L, Rousseeuw P. *Finding Groups in Data: An Introduction To Cluster Analysis*. John Wiley. 1990.

[22] Berndt DJ, Clifford J. Using Dynamic Time Warping to Find Patterns in Time Series. In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. Seattle, WA, USA: AAAI Press. 1994; p. 359–370.

[23] Nguyen M, Purushotham S, To H, Shahabi C. m-TSNE: A Framework for Visualizing High-Dimensional Multivariate Time Series. 2017.

Pattern-Mining-Konzepte für die Analyse großer Mengen von Simulationsverlaufsdaten

Niclas Feldkamp[1*], Sören Bergmann[1]

[1]Informationstechnik in Produktion und Logistik, TU Ilmenau, Max-Planck-Ring 12, 98693 Ilmenau, Deutschland; **niclas.feldkamp@tu-ilmenau.de*

Abstract. Üblicherweise basiert die Auswertung der Ergebnisse einer Simulationsstudie auf den Kennzahlen eines abgeschlossenen Simulationslaufs oder mehrerer Simulationsexperimente. Dieser Beitrag präsentiert eine Methode zur Analyse von Simulationsverlaufsdaten, d. h. Daten und Ereignisse, die während der Simulation anfallen. Zu diesem Zweck verwenden wir in diesem Beitrag Pattern-Mining-Algorithmen, um verborgene Zusammenhänge und potenziell interessante Muster in diesen dynamischen Daten zu entdecken. Hierzu wird ein Konzept vorgestellt sowie anhand einer einfachen, akademischen Fallstudie prototypisch angewendet. Diese Methode bietet einen neuen Ansatz zur Erkenntnisgewinnung in komplexen Simulationsszenarien, insbesondere im Bereich Produktion und Logistik.

Einleitung

Die diskret-ereignisgesteuerte Simulation ist ein essenzieller Bestandteil für die Analyse von komplexen Systemen, zum Beispiel im Bereich Produktion und Logistik, aber auch in vielen anderen Anwendungskontexten. Üblicherweise basiert die Auswertung der Ergebnisse einer Simulationsstudie auf den Kennzahlen eines abgeschlossenen Simulationslaufs oder von gleich mehreren abgeschlossenen Simulationsläufen aus verschiedenen Experimenten [1, 2]. Die Auswertung von Simulationsläufen und -experimenten ist in der Simulationsforschung gut durchdrungen, daher gibt es umfassende Literatur hierzu [1, 3–5]. Weniger intensiv erforscht hingegen sind Konzepte für die Auswertung von Daten, die innerhalb eines Simulationslaufs anfallen und durch die zeitliche Dynamik innerhalb eines Simulationslaufs Schwankungen unterliegen. Dies sind in der Regel Daten zu Ereignissen und Werte von Modellzustandsvariablen. In diesem Beitrag verwenden wir hierfür als Oberbegriff den Ausdruck *Simulationsverlaufsdaten*. Am weitesten verbreitet ist hier sicherlich die Trace-File-Analyse, welche die Ereignisliste des Simulators zum Zwecke der Validierung des Modells überprüft [6]. In diesem Beitrag soll jedoch eine Methode vorgestellt werden, mit deren Hilfe versteckte, interessante und potenzielle nützliche Zusammenhänge in Simulationsverlaufsdaten entdeckt werden können. Hierfür verwenden wir das dem Data-Mining-Werkzeugkasten entnommene Pattern Mining. Pattern-Mining-Algorithmen sind in der Lage, Muster und Regeln in großen Mengen von Transaktions-, Log- und Sequenzdaten zu finden und können dabei auch zeitliche Zusammenhänge miteinbeziehen können [7].

Der Aufbau dieses Beitrags ist wie folgt: Im ersten Kapitel werden die Grundlagen zu Simulationsverlaufsdaten gegeben, als auch die Grundlagen zu diversen Vertretern von Methoden aus der Klasse von Pattern-Mining-Algorithmen. Hieran schließt sich eine Übersicht über den Stand der Forschung zum Thema Analyse von Simulationsverlaufsdaten an. In Kapitel 2 wird dann ein Konzept vorgestellt, wie solche Methoden auf Simulationsverlaufsdaten anzuwenden sind und welche Voraussetzungen und Limitationen hierbei zu beachten sind. Dies wird dann in Kapitel 3 anhand einer einfachen, akademischen Fallstudie demonstriert. Kapitel 4 beendet den Beitrag mit einem abschließenden Fazit und einem Ausblick für zukünftige, hieran anzuknüpfende Forschungsmöglichkeiten.

1 Grundlagen und Stand der Forschung

1.1 Simulation und Simulationsverlaufsdaten

Abstrakt betrachtet lösen Ereignisse der diskret-ereignisgesteuerten Simulationstechnik Zustandswechsel zu einem konkreten Zeitpunkt mit einem dazugehörigen Zeit-

stempel aus, wie beispielsweise das Auftreten einer Störung einer Maschine. Das Auftreten von Ereignissen zieht dann in der Regel Änderungen an den Werten der Zustandsvariablen des Simulationsmodells nach sich [8, 9]. Das Ereignis des Eintritts einer Entität in eine Warteschlange erhöht beispielsweise die aktuelle Länge dieser Warteschlange, beispielhaft gezeigt in Abbildung 1.

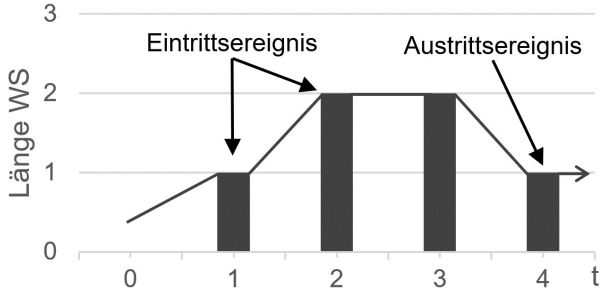


Abbildung 1: Beispiel für den dynamischen Verlauf einer Zustandsvariable.

Derartige Ereignisse und Zustandsänderungen über die Zeit zählen wir in diesem Beitrag ebenfalls zu den Simulationsverlaufsdaten. Das Persistieren von Ereignissen und Zustandsänderung über die Zeit wird in der diskret-ereignisgesteuerten Simulation oftmals auch als Eventlogs oder Tracefiles bezeichnet. Detaillierte konzeptionelle Überlegungen dazu finden sich im Kapitel 2.

1.2 Frequent Pattern Mining und verwandte Methoden

Frequent Pattern Mining ist eine bekannte Data-Mining-Methode und wird klassischerweise für die Warenkorbanalyse genutzt [7]. Hierbei wird nach Artikeln gesucht (sog. Items), die von Kunden häufig zusammen mit bestimmten anderen Artikeln gekauft werden (sog. Frequent Itemsets) [7, 10]. Daraus lassen sich wiederum Regeln generieren, die beschreiben, mit welcher Wahrscheinlichkeit beim Kauf eines Artikels auch ein anderer Artikel üblicherweise gekauft wird. Ziel hierbei ist dann zum Beispiel das Erstellen von gezielten, kundenspezifische Werbemaßnahmen und Empfehlungen. Grundlage hierfür ist das Vorhandsein einer sog. Transaktionsdatenbank. Diese speichert die Transaktionen zu jedem Kunden, wobei eine Transaktion aus einem oder mehreren Items besteht. Aus dieser werden dann algorithmisch die häufigen Itemsets berechnet [7, 10]. Eine davon abgeleitete Methode ist das Sequence Mining [11]. Hierbei wird die Transaktionsdatenbank um einen temporalen Faktor erweitert. Im Szenario der Warenkorbanalyse bedeutet dies, dass ein Kunde, der zu verschiedenen Zeitpunkten Transaktionen durchgeführt hat, somit eine Sequenz von Transaktionen generiert. Diese Daten können dann folglich nach häufigen Sequenzen durchsucht werden, also nach Itemsets, die häufig in einer bestimmten Reihenfolge auftreten [11]. Diese Methode hat neben der Warenkorbanalyse viele weitere Anwendungszwecke, beispielsweise Clickstream-Analysen zur Untersuchung von Webtraffic [12] oder Gen-Analysen im Biomedizinischen Bereich [13]. Ein weiterer, interessanter Vertreter dieser Klasse von Data-Mining-Methoden ist das Periodic Pattern Mining [14]. Im Gegensatz zum Sequential Pattern Mining werden hier nicht mehrere Sequenzen nach häufigen Mustern durchsucht, sondern eine (in der Regel längere) Sequenz nach wiederholt auftretenden Zyklen. Ziel ist es hierbei, die Regelmäßigkeit solcher Zyklen zu bestimmen, wie etwa *"alle 5 Zeiteinheiten tritt Ereignis A auf"*. Dies ist unter anderem hilfreich für die Analyse von Logfiles bei repetitiven Prozessen, beispielsweise im Kontext von Eventlogs bei Internet-of-Things(IoT)-Geräten [14].

1.3 Automatisierte Auswertung von Simulationsverlaufsdaten

Die Idee zur automatisierten Auswertung von Simulationsverlaufsdaten besteht in der Forschung im Bereich von Simulation im Allgemeinen und diskret-ereignisgesteuerter Simulation im Speziellen schon seit geraumer Zeit. So zeigten bereits Tolujev et al. Ansätze zur dynamischen Metamodellierung von Simulationsmodellen auf Basis von Trace-Dateien von GPSS-Modellen auf, zum Beispiel zur Unterstützung bei der Modellvalidierung [15]. Mit dem Aufkommen von leistungsfähigerer Recheninfrastruktur und damit einhergehendem Potenzial für größere Simulationsmodellkomplexität wurden dann auch Data-Mining-Methoden für die Verarbeitung und Auswertung der immer größer werdenden Mengen von Simulationsverlaufsdaten herangezogen. Beispielsweise zeigt Tekinay die automatisierte Analyse von Simulationsverlaufsdaten mittels Pattern Mining und Markov-Ketten für komplexe diskret-ereignisgesteuerte Simulationsmodelle im großen Maßstab auf, mit dem Ziel der automatisierten Modellabstraktion [16]. Eng verknüpft ist hierbei auch das Process-Mining, welches das Erkennen und Analysieren zugrundeliegender Prozesse basierend auf Eventdaten aus Echtsystemen ermöglicht [17], im Gegensatz zu Eventdaten (hier Simulationsverlaufsdaten genannt) aus Simulationsmodellen. Özkul et al. verwenden beispielsweise Process-Mining-Methoden

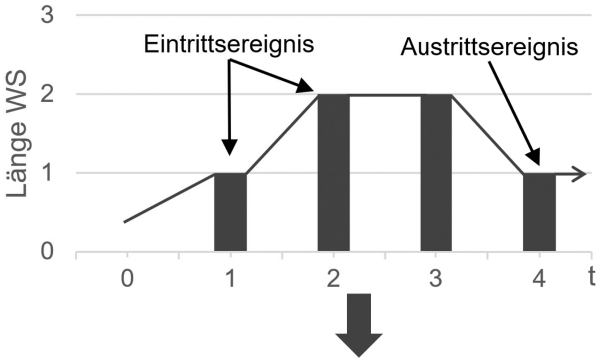
zur Verifikation von Simulationsmodellen [18]. Oftmals findet die Verknüpfung von Simulation und Process Mining allerdings eher in die andere Richtung statt. Aus Echtweltdaten, wie beispielsweise Logfiles aus Realsystemen, werden mit Hilfe von Process-Mining-Methoden Simulationsmodelle generiert, siehe zum Beispiel [19, 20].

Dieser Beitrag fokussiert sich allerdings auf die automatisierte Simulationsverlaufsdatenanalyse mit dem Ziel der Wissensentdeckung mittels Pattern-Mining-Methoden. Feldkamp et al. beschreiben die Wissensentdeckung in Simulationsdaten als Prozess zur Unterstützung der Entscheidungsfindung, indem durch das Aufdecken von unbekannten Zusammenhängen im System potenziell nützliches Wissen generiert wird [3]. Dies bezieht sich primär auf Eingangs- und Ergebnisdaten der Simulation, welche aus mehreren Läufen bzw. Experimenten stammen. Hierbei werden also aggregierte Werte am Ende eines Simulationslaufs betrachtet und nicht der zeitlich-dynamische Verlauf innerhalb eines Laufs. Eine automatisierte Analyse von Simulationsverlaufsdaten mit dem Ziel einer Wissensgenerierung findet sich in nur wenige Arbeiten. Kemper und Tepper befassen sich mit automatisierter Analyse von Simulationsverlaufsdaten für Simulationsmodelle im Kontext von Logistik. Hierzu entwickelten sie eine spezielle Technik zur visuellen Analyse. Ziel ist hierbei das Entdecken von Fehlern oder abnormalem Systemverhalten [6, 21]. Wustmann et al. verwenden Graphenanalyse, um im Kontext von Materialflusssimulationen mögliche Probleme und Ursachen im Ablauf zu identifizieren [22]. Bogon et al. verwenden Sequence-Mining-Methoden um Assoziationsregeln innerhalb eines Simulationslaufs zu entdecken [23]. Allerdings wird hier nicht näher darauf eingegangen, wie aus den Simulationsverlaufsdaten die jeweiligen konkreten Sequenzen für eine Transaktionsdatenbank generiert werden können, um Sequence-Mining-Algorithmen darauf anzuwenden.

Es lässt sich festhalten, dass ein grundsätzliches Konzept für die Anwendung von Frequent-Pattern-Mining-Methoden zur Analyse von Simulationsverlaufsdaten fehlt. Dieses Konzept wird im nächsten Kapitel beschrieben.

2 Konzeptionelle Überlegungen

In diesem Kapitel wird ein allgemeines Konzept zur Nutzung von Pattern-Mining-Methoden für die Analyse von Simulationsverlaufsdaten vorgestellt, basierend auf den drei in Kapitel 1.2 vorgestellten Methoden: Frequent Pattern Mining, Sequential Pattern Mining und Periodic Pattern Mining. Wie bereits in Kapitel 1.1 beschrieben, zählen wir in diesem Beitrag in der diskret-ereignisgesteuerten Simulation zu den Simulationsverlaufsdaten sowohl die auftretenden Ereignisse als auch den Werteverlauf von Zustandsvariablen über die Zeit. Für das Finden von häufigen Mustern und Regeln in diesen Daten mittels Frequent Pattern Mining benötigt man, wie bereits in Kapitel 1.2 dargestellt, eine Transaktionsdatenbank, welche aus Items und Itemsets besteht. Diese Items sind also gleichzusetzen mit den auftretenden Ereignissen in den Simulationsverlaufsdaten (Events). Die (numerischen) Zustandsvariablen müssen hingehen zunächst ebenfalls zu kategorialen Events diskretisiert werden, bevor diese als Items in einer Transaktionsdatenbank genutzt werden können. Abbildung 2 zeigt hierfür exemplarisch ein Beispiel. Im oberen Teil der Abbildung sieht man den Verlauf einer beispielhaften Warteschlange *WS*. In der darunter befindlichen Liste wurde der Zustandsverlauf dieser Warteschlange zu künstlichen, kategorialen Events mit Zeitstempel der Zustandsänderung diskretisiert.



t	Ereignis
1	„Länge WS = 1“
2	„Länge WS = 2“
4	„Länge WS = 1“

Abbildung 2: Diskretisierung von numerischen Zustandsvariablen zu kategorialen Ereignissen.

Hierbei ist zu beachten, dass bei Zustandsvariablen mit sehr großen Wertebereichen die Anzahl der Klassen und deren Wertebereiche für die Diskretisierung möglichst geschickt gewählt werden müssen, sodass nicht zu viele verschiedene Itemsets für die Transaktionsdatenbank generiert werden. Dies könnte sonst dem Auffinden häufiger Muster entgegenstehen. Für eine Warteschlange mit

großer Kapazität könnte man beispielsweise eine Diskretisierung auf Basis vorher definierter, prozentualer Schwellwerte bezogen auf die max. Kapazität wählen, wie etwa in folgendem Beispiel:

```
{"Länge WS ≤ 50 %",
"Länge WS > 50 % & ≤90%", "Länge WS > 90 %"}
```

Hier muss je nach individuellem Anwendungsfall eine passende Diskretisierungslösung gefunden werden. Dies erfordert eventuell weiteres Hintergrundwissen über das zugrunde liegende System sowie eine vorherige Analyse der Werteverteilung, zum Beispiel mittels einer Histogrammvisualisierung. Sind nun sämtliche zu betrachtende Simulationsverlaufsdaten zu kategorialen Ereignissen diskretisiert worden, entsteht hieraus implizit ein Zeitstrahl mit Events entlang der Simulationszeit, wie exemplarisch in Abbildung 3 gezeigt ist.

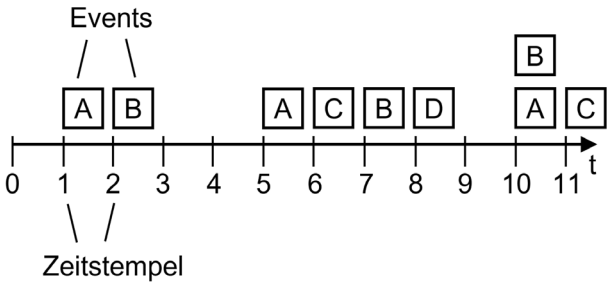


Abbildung 3: Beispielhafter Zeitstrahl mit Events im Simulationsverlauf.

Aus dieser Ereignisliste muss dann die Transaktionsdatenbank für das jeweilige Pattern-Mining-Verfahren generiert werden. Die zentrale Frage ist nun, wie man also aus der Ereignisliste eine Liste mit Transaktionen bzw. Sequenzen erzeugt. Hierbei sind jeweils bestimmte Transformationsregeln und Einschränkungen zu beachten, die sich je nach konkretem Pattern-Mining-Verfahren unterscheiden. Abbildung 4 zeigt dies jeweils exemplarisch anhand der in Abbildung 3 dargestellten Daten. Dies wird nun im Folgenden näher erläutert.

Für das klassische Frequent Pattern Mining (Abbildung 4a) können Events zu einer Transaktion zusammengefasst werden, die innerhalb einer zuvor gewählten, festen Zeitspanne liegen. Für das gewählte Beispiel liegt diese Zeitspanne bei t+2. Der Algorithmus würde hier folglich das häufige (in 80 % der Transaktionen) Muster {A, B} und die Regel *„wenn A, dann B“* finden, welche in 100 % der Fälle zutrifft. Dies ist die einfachste Variante und hat den offensichtlichen Nachteil, dass Muster mit Events, die zeitlich weit auseinander liegen, hierbei nicht gefunden werden.

(a) Frequent Pattern Mining (für t+2)

ID	Events
1	{A, B}
2	{A, C, B}
3	{C, B, D}
4	{D, A, B}
5	{A, B, C}

(b) Sequence Mining (für Startevent A)

ID	Events	Zeit
1	{A}	1
1	{B}	2
2	{A}	5
2	{C}	6
2	{B}	7
2	{D}	8
3	{A, B}	10
3	{C}	11

ID	Sequenz
1	{A}, {B}
2	{A}, {C}, {B}, {D}
3	{A, B}, {C}

(c) Periodic Pattern Mining

S(Event)	Event-Sequenz
S(A)	{1, 5, 10}
S(B)	{2, 7, 10}
S(C)	{6, 11}
S(A, B)	{1, 5, 10)

Abbildung 4: Erzeugen der Transaktionsdatenbanken aus den Simulationsverlaufsdaten für unterschiedliche Frequent-Pattern-Mining-Methoden

Sequence Mining (Abbildung 4b) berücksichtigt auch die zeitliche Dimension bzw. Reihenfolge der Items und Itemsets. Hierbei muss also die Eventliste in mehrere Sequenzen zerlegt werden. Hierfür ist ein Startevent zu definieren, welches den Beginn einer neuen Sequenz markiert. Dies kann etwa ein für den Anwender interessantes Event sein, wie beispielsweise der Ausfall einer Maschine. Alle in der Folge aufgetretenen Events werden dann zu der Sequenz hinzugezählt. Im gezeigten Beispiel wurde hier das Event A als Startevent gewählt. Der Al-

gorithmus würde hier die häufige Subsequenz {A, B} finden, selbst wenn A und B zeitlich weiter auseinander liegen. Nachteil hier ist jedoch, dass es eventuell schwierig ist, ein adäquates Startevent auszuwählen und damit eine sinnvolle Sequenzliste zu generieren.

Beim Periodic Pattern Mining (Abbildung 4c) ist das Erzeugen der Transaktionsdatenbank recht simpel zu bewerkstelligen, indem zu jedem Ereignistyp die zeitliche Sequenz aus den dazugehörigen Zeitstempeln gebildet wird. Innerhalb dieser Zeitstempelsequenzen kann der Algorithmus nun nach häufigen bzw. regelmäßigen Zyklen suchen

Periodic Pattern Mining stellt somit die beste Methode zum Analysieren insbesondere großer Mengen von Simulationsverlaufsdaten dar. Die Transaktionsdatenbank kann hier sofort und automatisiert erzeugt werden und es müssen keine Entscheidungen über notwendige Parametrisierungsoptionen bei der Erzeugung der Transaktionsdatenbank getroffen werden. Eventuell kann es aber sinnvoll sein, eine Auswahl der in die Transaktionsdaten zu überführenden Ereignistypen zu treffen, um uninteressante Eventtypen vorab herauszufiltern.

3 Fallstudie

3.1 Aufbau

Im Folgenden wird eine simple, akademische Fallstudie als Veranschaulichung der zuvor diskutierten Methodik präsentiert. Das Simulationsmodell wurde in Anylogic implementiert und besteht aus einem Warteschlangensystem mit zwei Linien, wie Abbildung 5 zeigt.

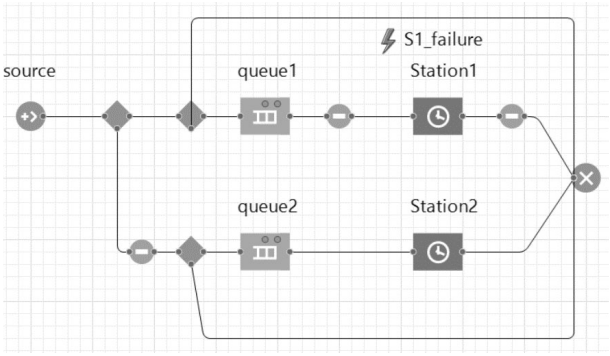


Abbildung 5: Screenshot des Simulationsmodells.

Konkret trägt die obere Bearbeitungsstation *Station1* die Hauptlast des Systems und wird über deren Warteschlange *queue1* mit Aufträgen versorgt. Im Falle einer Störung auf dieser Bearbeitungsstation wird der untere Pfad geöffnet, sodass Aufträge auf die untere Bearbeitungsstation *Station2* umgeleitet werden, bis der obere Pfad wieder zur Verfügung steht. Das Modell wurde als nicht-terminierendes System über einen längeren Zeitraum simuliert und alle auftretenden Events wurden über die Eventdatenbank von Anylogic mitgeschnitten. Dabei wurden neben den Standardereignissen der diskret-ereignisgesteuerten Simulation, wie etwa der Eintritt einer Entität oder das Auftreten einer Störung der Station, einige zusätzliche Ereignisse definiert und entsprechend geloggt, welche für die Auswertung und Analyse potenziell interessant sein könnten. Diese beziehen sich auf Ergebnisgrößen der Simulation und tauchen ohne vorherige manuelle Definition nicht notwendigerweise in der Ereignisliste des Simulators auf, wie beispielsweise eine voll- oder leergelaufene Warteschlange. Abbildung 6 zeigt exemplarisch einen Auszug des Eventlogs.

Main | events_log

	event_name	date	agent_type	agent
1	q2_empty	11-03-2024 00:00:00	Main	root
2	q1_empty	11-03-2024 00:00:00	Main	root
3	source_arrival	11-03-2024 00:00:18	Main	root
4	q1_empty	11-03-2024 00:00:18	Main	root
5	agent_exited	11-03-2024 00:01:18	Main	root
6	source_arrival	11-03-2024 00:01:53	Main	root
7	q1_empty	11-03-2024 00:01:53	Main	root
8	source_arrival	11-03-2024 00:02:59	Main	root
9	q1_empty	11-03-2024 00:02:59	Main	root
10	source_arrival	11-03-2024 00:03:01	Main	root
11	q1_empty	11-03-2024 00:03:59	Main	root
12	source_arrival	11-03-2024 00:08:06	Main	root
13	q1_empty	11-03-2024 00:08:06	Main	root
14	source_arrival	11-03-2024 00:08:09	Main	root
15	source_arrival	11-03-2024 00:08:12	Main	root
16	source_arrival	11-03-2024 00:08:16	Main	root
17	source_arrival	11-03-2024 00:08:19	Main	root
18	source_arrival	11-03-2024 00:09:15	Main	root
19	source_arrival	11-03-2024 00:10:18	Main	root
20	source_arrival	11-03-2024 00:11:32	Main	root
21	source_arrival	11-03-2024 00:12:13	Main	root
22	source_arrival	11-03-2024 00:14:22	Main	root

Abbildung 6: Beispielhafter Auszug aus dem Eventlog des Simulationsmodells.

3.2 Analyse

Zur Analyse der Simulationsverlaufsdaten verwenden wir in dieser Fallstudie ausschließlich die Methode des Periodic Pattern Mining, da diese, wie im Kapitel zuvor beschrieben, die vielversprechendste und interessanteste Methode zum Auffinden von Mustern in Simulationsverlaufsdaten darstellt. Im ersten Schritt wurde hierfür der

zuvor erzeugte Eventlog in ein allgemein kompatibles CSV-Format exportiert. Eine weitere Datenbereinigung ist nicht notwendig, da Simulationsdaten, im Gegensatz zu Echtdaten, fehlerfrei sind, sofern vorausgesetzt werden kann, dass das zugrunde liegende Simulationsmodell valide und fehlerfrei ist. Die Analyse wurde mit Hilfe der Python-Bibliothek Scikit-mine[1] durchgeführt, welche eine schnelle und effiziente API für Pattern Mining in Python bereitstellt. Konkret wurden hier insbesondere die Methoden der Klasse *PeriodicPatternMiner* genutzt.

Wie bereits zu erwarten war, findet der Algorithmus eine Vielzahl von Mustern, von denen für die Auswertung im Sinne einer Wissensentdeckung durch interessante Muster nur wenige tatsächlich relevant sind. Es benötigt also einen gewissen Grad an Hintergrundwissen über das Modell sowie ein manuelles Filtern und Suchen nach Mustern bzw. einzelnen Komponenten und Schlüsselereignissen, um letztendlich interessante Muster aufzudecken. Tabelle 1 zeigt hierzu vier ausgewählte Muster, welche im Folgenden näher erläutert werden.

Nr.	Muster	Abstand
1	(S1_failure)	3:30:32
2	(S1_failure ⇒ [d=0:30:00] ⇒ S1_repair_done)	3:29:48
3	(q2_empty ⇒ [d=0:12:00] ⇒ q2_full)	16:00:01
4	(q2_empty ⇒ [d=0:12:00] ⇒ q2_full ⇒ [d=0:17:00] ⇒ S1_repair_done ⇒ [d=0:46:00] ⇒ q2_empty)	3:30:00

Tabelle 1: Ausgewählte periodische Muster in den Simulationsverlaufsdaten. Der Wert *d* in den eckigen Klammern zeigt jeweils den durchschnittlichen Abstand zwischen zwei Teilkomponenten eines Gesamtmusters an.

Muster 1 zeigt den periodischen Verlauf des Störevents von Station 1 mit einem regelmäßigen Auftreten von 3,5 Stunden im Schnitt. Dies ist nicht überraschend, da die zugrunde gelegte Zufallsverteilung für das Auftreten der Störung mit einem Erwartungswert von 3,5 Stunden parametrisiert war. Dennoch soll dieses Muster als anekdotischer Beweis dienen, dass die gewählte Methodik in der Lage ist, solche erwarteten Regelmäßigkeiten auch tatsächlich zu finden. Als Nebeneffekt kann zudem die vorgestellte Methodik auch zur Überprüfung und Validierung des Modells herangezogen werden. Ein ähnlicher Zusammenhang findet sich in Muster 2, welches sich auf die Periodizität von Störungsauftritt und Reparatur bezieht. Der Wert *d* in den eckigen Klammern zeigt jeweils den durchschnittlichen Abstand zwischen zwei Teilkomponenten eines Gesamtmusters an. Da die im Simulator hinterlegte Entstörungszeit 30 Minuten beträgt, findet der Algorithmus auch hier korrekterweise einen regelmäßigen Zyklus aus Auftreten einer Störung sowie deren Behebung mit einem Abstand von 30 Minuten zwischen beiden Ereignissen und einer Wiederholung von im Schnitt ebenfalls wieder ca. 3,5 Stunden. Muster 3 besteht aus Ereignissen, die sich auf eine Ergebnisgröße im Verlauf beziehen, nämlich den Zustand der Warteschlange *q2*. Wie man sieht, gibt es hier eine Regelmäßigkeit zwischen Leerstand und dem Volllaufen der Warteschlange, wobei der Abstand *d* beider Ereignisse in diesem Muster im Schnitt 12 Minuten beträgt und das Muster mit einer Regelmäßigkeit von 16 Stunden auftritt. Daraus lässt sich schließen, dass bei Öffnen des unteren Pfades nach Auftritt einer Störung auf dem oberen Pfad es regelmäßig 12 Minuten dauert, bis die untere Warteschlange vollgelaufen ist. Muster 4 beschreibt schlussendlich den Zyklus zwischen dem Volllaufen der vorher leeren Warteschlange q2, dem Abschluss der Reparatur von Station 1 sowie das darauffolgende Leerlaufen derselben Warteschlange, mit den entsprechenden Zwischenabstandszeiten der jeweiligen Einzelereignisse. Daraus lässt sich also ableiten, dass nach Behebung der Störung von Station 1 die vollgelaufene Warteschlange q2 regelmäßig nach im Schnitt 46 Minuten wieder leergezogen worden ist.

Es fällt auf, dass Muster 3 einen deutlich größeren Abstand hat, also seltener auftritt, als Muster 4, obwohl die ersten beiden Ereigniskomponenten dieses Musters identisch sind. Dies liegt daran, dass in einigen wenigen Fällen zwar die Regelmäßigkeit für die ersten beiden Ereigniskomponenten von Muster 4 gegeben waren, aber nicht für die beiden hinteren, weshalb daraus in der Folge ein neues Muster (Muster 3) gebildet wurde. Solche Zusammenhänge gilt es bei der Auswertung solcher Muster auf Simulationsverlaufsdaten zu berücksichtigen. Grundsätzlich ist dies eine wesentliche Herausforderung bei der Anwendung von Periodic-Pattern-Mining, die auch in diversen prototypischen Experimenten zu diesem Beitrag festgestellt wurde: Je stärker ein stochastischer Einfluss

[1] siehe https://github.com/scikit-mine/scikit-mine

auf bestimmte Ereignisse wirkt, desto schwieriger wird es, hierzu noch Regelmäßigkeiten zu entdecken. Hierbei werden dann oftmals mehrere zyklische Muster entdeckt, die aber eigentlich derselben kausalen Ereigniskette zuzuordnen wären. Eine Abhilfe könnte hier sein, durch das geschickte Zusammenfassen mehrerer Muster hieraus wieder das originäre zyklische Muster zu errechnen.

4 Fazit und Ausblick

In diesem Beitrag wurde gezeigt, wie mit Hilfe von Pattern-Mining-Methoden Simulationsverlaufsdaten zur Wissensgenerierung verwendet werden können. Hierzu wurden diverse Pattern-Mining-Methoden vorgestellt sowie ein Konzept zur Anwendung dieser auf Simulationsverlaufsdaten eingeführt. Darüber hinaus wurde die Nutzbarkeit von Periodic Pattern Mining im Speziellen exemplarisch anhand einer einfachen, akademischen Fallstudie aufgezeigt. Allerdings verbleiben hierbei noch einige Herausforderungen und daraus abgeleiteter Forschungsbedarf. Eine große Herausforderung besteht darin, dass das Vorhandensein vieler und/oder starker stochastischer Einflüsse im Modell dem Entdecken von zyklischen Mustern entgegenstehen. Abweichungen aufgrund der Stochastik können hier möglicherweise vorhandene Muster und Regelmäßigkeiten überdecken. Eine weitere Herausforderung stellt die schiere Größe von Logfiles und damit einhergehende Masse von Ereignissen in komplexen, langlaufenden Simulationsmodellen dar. Die Grenzen der Anwendbarkeit der hier vorgestellten Methode müssen in weiteren Forschungsarbeiten noch näher untersucht werden, z. B. hinsichtlich der Komplexität des untersuchten Szenarios und des zugrundeliegenden Modells. Hierbei kann es schwierig sein, interessante, nicht-triviale Muster herauszufiltern. Es besteht weiterer Forschungsbedarf für z. B. graphische, interaktive Methoden, die den Anwender hierbei unterstützen. Ein weiteres zukünftiges Forschungsfeld könnte die Erweiterung der Methode für die Nutzung mit Daten aus mehreren Simulationsexperimenten sein. In diesem Beitrag wurde nur ein Szenario mit fixen Faktorwerten betrachtet. Wenn man nun im Sinne eines Experimentdesigns die Werte bestimmter Faktoren verändert, erhält man als Ergebnis einen Simulationsverlaufsdatensatz pro Experiment. Das Auswerten von Mustern und deren Veränderung in Abhängigkeit der Faktorwerte könnte wiederum interessantes und potenziell nützliches Wissen über das Modell hervorbringen.

References

[1] Kelton, W. D. 1995. A tutorial on design and analysis of simulation experiments. In *Proceedings of the 1995 Winter Simulation Conference*, 24–31. DOI=10.1109/WSC.1995.478700.

[2] Law, A. M. 2014. *Simulation Modeling and Analysis*. McGraw-Hill Series in Industrial Engineering and Management Science. McGraw Hill Book Co, New York, N.Y.

[3] Feldkamp, N., Bergmann, S., and Strassburger, S. 2020. Knowledge Discovery in Simulation Data. *ACM Trans. Model. Comput. Simul.* 30, 4, 1–25.

[4] Sanchez, S. M. 2014. Simulation Experiments: Better Data, Not Just Big Data. In *Proceedings of the 2014 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 805–816.

[5] Kleijnen, J. P., Sanchez, S. M., Lucas, T. W., and Cioppa, T. M. 2005. State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments. *INFORMS Journal on Computing* 17, 3, 263–289.

[6] Kemper, P. and Tepper, C. 2005. Trace Based Analysis of Process Interaction Models. In *Proceedings of the 2005 Winter Simulation Conference*. IEEE inc., Piscataway, N.J.

[7] Aggarwal, C. C. 2014. An Introduction to Frequent Pattern Mining. In *Frequent Pattern Mining*, C. C. Aggarwal and J. Han, Eds. Springer International Publishing, Cham, Heidelberg, New York, Dordrecht, London, 1–17.

[8] Schriber, T. J., Brunner, D. T., and Smith, J. S. 2014. Inside discrete-event simulation software: How it works and why it matters. In *Proceedings of the 2014 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 132–146.

[9] Pedgen, D. C. 2010. Advanced tutorial: Overview of simulation world views. In *Proceedings of the 2010 Winter Simulation Conference*. IEEE, Piscataway, NJ, 210–215.

[10] Han, J. and Kamber, M. 2006. *Data mining*. *Concepts and techniques*. The Morgan Kaufmann series in data management systems. Elsevier; Morgan Kaufmann, Amsterdam, Boston, San Francisco, CA.

[11] Agrawal, R. and Srikant, R. 1993. Mining Sequential Patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*. IEEE inc., Piscataway, N.J., 3–14.

[12] Liu, Z., Wang, Y., Dontcheva, M., Hoffman, M., Walker, S., and Wilson, A. 2017. Patterns and Sequences: Interactive Exploration of Clickstreams to Understand Common Visitor Paths. *IEEE Transactions on Visualization and Computer Graphics* 23, 1, 321–330.

[13] Cellier, P., Charnois, T., Plantevit, M., Rigotti, C., Crémilleux, B., Gandrillon, O., Kléma, J., and Manguin, J.-L. 2015. Sequential pattern mining for discovering gene interactions and their contextual information from biomedical texts. *Journal of biomedical semantics* 6, 27.

[14] Galbrun, E., Cellier, P., Tatti, N., Termier, A., and Crémilleux, B. 2019. Mining Periodic Patterns with a MDL Criterion. In *Machine Learning and Knowledge Discovery in Databases*, M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley and G. Ifrim, Eds. Lecture Notes in Computer Science. Springer International Publishing, Cham, 535–551. DOI=10.1007/978-3-030-10928-8_32.

[15] Tolujev, J., Lorenz, P., Beier, D., and Schriber, T. J. 1998. Assessment of simulation models based on trace-file analysis: a metamodeling approach. In *Proceedings of the 1998 Winter Simulation Conference. Simulation in the 21st century*. IEEE, Piscataway, N.J., 443–450.

[16] Tekinay, Ç. 2022. *Automated Abstraction of Discrete-Event Simulation Models using State-Trace Data*. Doctoral Thesis, Delft University of Technology.

[17] van der Aalst, W. M. P. 2011. *Process Mining*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[18] Özkul, F., Sutherland, R., Wenzel, S., and Spieckermann, S. 2023. Einsatz von Process-Mining zur Verifikation und Validierung von Simulationsmodellen in Produktion und Logistik. In *Simulation in Produktion und Logistik 2023*. ASIM-Mitteilung Nr. 187. Universitätsverlag Ilmenau, Ilmenau, 463–472.

[19] Spieckermann, S., Stöhr, N., Mayer, G., Özkul, F., and Wenzel, S. 2023. Fallbeispiele aus Produktion und Logistik für die Verknüpfung von ereignisdiskreter Simulation und Process-Mining. In *Simulation in Produktion und Logistik 2023*. ASIM-Mitteilung Nr. 187. Universitätsverlag Ilmenau, Ilmenau, 155–165.

[20] Jadrić, M., Pašalić, I. N., and Ćukušić, M. 2020. Process Mining Contributions to Discrete-event Simulation Modelling. *Business Systems Research Journal* 11, 2, 51–72.

[21] Kemper, P. and Tepper, C. 2009. Automated Trace Analysis of Discrete-Event System Models. *IIEEE Trans. Software Eng.* 35, 2, 195–208.

[22] Wustmann, D., Vasyutynskyy, V., and Thorsten, S. 2009. Ansätze zur automatischen Analyse und Diagnose von komplexen Materialflusssytemen. In *5. Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik*, W. M. Scheid, Ed. Universitäsverlag, Ilmenau, 1–20.

[23] Bogon, T., Timm, I. J., Jessen, U., Schmitz, M., Wenzel, S., Lattner, A. D., Paraskevopoulos, D., and Spieckermann, S. 2012. Towards assisted input and output data analysis in manufacturing simulation: The EDASim approach. In *Proceedings of the 2012 Winter Simulation Conference (WSC 2012)*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Object-Oriented Implementation of a Simulator for Linear Implicit Equilibrium Dynamics

Dirk Zimmer [1*]

[1]Institute of System Dynamics and Control, DLR, Münchener Straße 20, 82234 Weßling, Germany
[*]*dirk.zimmer@dlr.de*

Abstract. Models based on linear implicit equilibrium dynamics form a special but very useful sub-set of DAE systems that can be applied for the simulation of technical physical systems. Since these are based on differential-algebraic equations, a transformation into executable code must be performed for the purpose of model evaluation. One way to achieve this is the object-oriented formulation of the simulation code itself. To explore this path a dedicated simulator prototype has been implemented and is outlined here. The long-term goal is to define alternative compilation targets for a Modelica compilers that enable highly scalable simulation code for very large systems.

1 What is Linear Implicit Equilibrium Dynamics?

Linear Implicit Equilibrium Dynamics (LIED) is technically defined as a special class of Differential Algebraic Equation (DAE) Systems.

1.1 Formal Definition

A DAE system with potential state derivatives $\dot{\boldsymbol{x}}$, time t and algebraic variables $\mathbf{w}$

$$\mathbf{0} = \boldsymbol{F}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \mathbf{w}, t)$$

is defined as LIED system when it can be transformed into the following form:

$$\begin{bmatrix} \mathbf{w}_E \\ \dot{\boldsymbol{x}}_E \end{bmatrix} = \mathrm{g}(\boldsymbol{x}_I, \boldsymbol{x}_E, t)$$

$$\mathbf{A}(\boldsymbol{x}_I, \boldsymbol{x}_E, \mathbf{w}_E) \begin{bmatrix} \mathbf{w}_I \\ \dot{\boldsymbol{x}}_I \end{bmatrix} = \mathrm{f}(\boldsymbol{x}_I, \boldsymbol{x}_E, \mathbf{w}_E, t)$$

We see that both the algebraic variables as well as the state derivatives can be split into a fully explicit part $(\dot{\boldsymbol{x}}_E;\ \mathbf{w}_E)$ and a part $(\dot{\boldsymbol{x}}_I;\ \mathbf{w}_I)$ with a linear system in implicit form expressed by the regular matrix $\mathbf{A}$. Furthermore, the following conditions shall hold true:

$$\dot{\boldsymbol{x}}_E \cap \dot{\boldsymbol{x}}_I \subseteq \dot{\boldsymbol{x}}$$
$$\mathbf{w}_E \cap \mathbf{w}_I \supseteq \mathbf{w}$$
$$\dot{\boldsymbol{x}}_E \cap \dot{\boldsymbol{x}}_I \cap \mathbf{w}_I \supseteq \dot{\boldsymbol{x}}$$
$$\dot{\boldsymbol{x}}_E, \dot{\boldsymbol{x}}_I, \mathbf{w}_E, \mathbf{w}_I \text{ are all disjoint}$$

These conditions essentially mean that it is allowed to perform certain symbolic mechanism of index reduction such as the dummy derivative method [5] originating from Pantelides [6]. Using this method, states variables of $\boldsymbol{x}$ can be transformed to algebraic variables in $\mathbf{w}_I$ and further derivatives may be added to $\mathbf{w}_I$ or $\mathbf{w}_E$. In practice, this is important because it means that the linear implicit dynamics can be expressed by far fewer states than suggested by the vector $\boldsymbol{x}$ of the original DAE formulation.

1.2 Informal Explanation

The formal definition above may be primarily perceived as a relatively strong restriction on the model equations and not many systems may be intuitively expected to fall into this category. Surprisingly, LIED can be applied successfully for the object-oriented modelling of complex thermofluid architectures [7],[8] or to mechanical systems with stiff contacts [9].

The idea is that the non-linear behaviour of the slow mode is explicitly expressed whereas the fast dynamics that typically is needed to uphold non-linear constraints is expressed by a linear implicit system that fulfils the constraint in its equilibrium. Hence the name: linear implicit equilibrium dynamics. The equilibrium dynamics is thereby often a replacement dynamic and only an approximation of reality (as all modelling is).

2 What is LIED good for?

As the above references demonstrate, LIED has been applied using Modelica [1] for the object-oriented modeling of thermofluid or mechanical systems.

To this end, it is necessary to use triplets as interface of the model components that consist in a signal for the explicit non-linear part and a pair of potential as presented in Table 1.

Domain	Signal	Potential	Flow
trans. mechanics	position: r [m]	velocity: v [m/s]	force: f [N]
rotational mechanics	angle: φ [rad]	angular velocity: ω [rad/s]	torque: τ [Nm]
thermofluid streams	Thermo-dynamic state: $\hat{\Theta}$	inertial pressure r [Pa]	mass-flow rate: $\dot{m}$ [kg/s]

Table 1: Connection triplets for the object-oriented modelling of LIED Systems.

More background on the derivation of these triplets can be found in [10]. It goes beyond the scope of this paper how the equations are formulated in detail but the two Modelica model diagrams shown in Figure 1 and 2 may illustrate the practical usefulness.

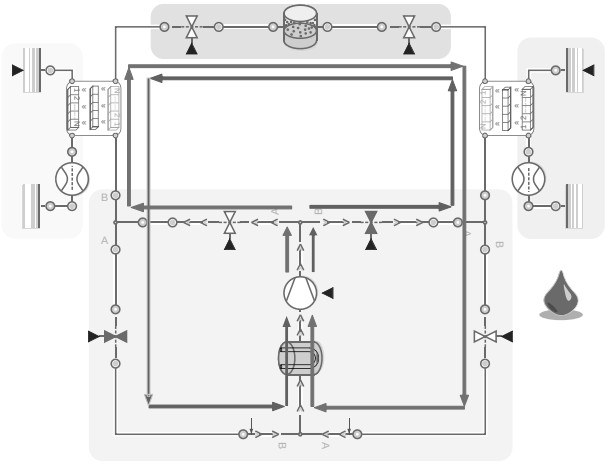


Figure 1: Model diagram of a reversible heat pump systems using the ThermoFluid Stream Library [8].

Especially the ThemoFluid Stream Library has meanwhile become a popular OpenSource library both by academia [4] and by industry [7].

LIED systems have very benevolent characteristics for object-oriented modelling. Following simple connection rules, the resulting matrix $\mathbf{A}(\boldsymbol{x}_I, \boldsymbol{x}_E, \mathbf{w}_E)$ will be regular and an a-priori statement on solvability can be given [7]. This makes this class of modeling very robust and prevents many computational simulation errors.

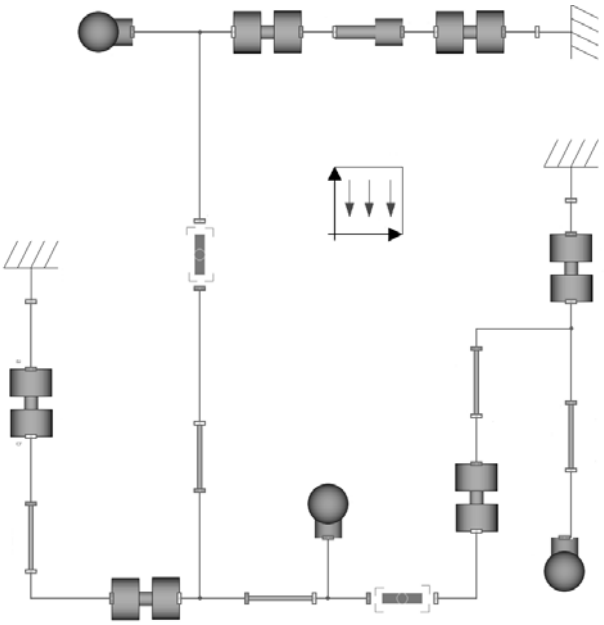


Figure 2: Model diagram of a mechanical system for a kinematic using the Dialectic Mechanics library [9].

3 A dedicated simulator

3.1 Why?

The libraries are currently developed using Modelica and since LIED systems are a subset of DAE systems and Modelica can deal with DAEs in general, any Modelica compiler can be used to generate simulation code out of LIED systems. A dedicated simulator is thus not needed. Existing and mature simulation software can be used.

However, Modelica compilers are very complex and algorithms for the processing of general DAEs involve drawbacks: all Modelica compilers create a “flat” model, where all equations are collected in a single set. For very large systems, this poses problems primarily because large amounts of code are generated.

Out of practical experience, many LIED systems have very benevolent structural characteristics that enable a compilation already on the component level. A compiler for standard Modelica cannot exploit these benefits, albeit they may greatly increase the ability to deal with large systems or with variable structure systems.

To explore alternative compile targets dedicated for LIED systems, a prototypical simulator has been developed with the name `zimsim`.

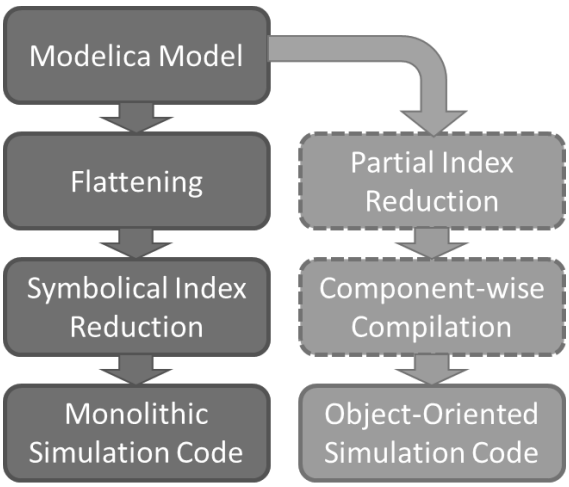


Figure 3: Illustration of different pathways for the compilation of simulation code. The blue boxes on the left illustrate the classic compilation of Modelica where all equations are collected in a single set. The green boxes on the right illustrate a potential pathway for LIED systems. This paper focuses on the compile target which is the object-oriented simulation code.

3.2 Computational structure and implementation of blocks

Figure 5 presents a simple planar mechanical system of a crane crab (a pendulum attached to a slider). To each component of the system 3 computational blocks are assigned of different color: blue, green, and orange. The blue and green blocks form thereby a computational sequence directed from the root whereas the orange signal leads to the root. For this particular domain of LIED systems, the blue signal contains the positional state and undergoes non-linear transformations. The transformation of the green and orange signal forms a linear response which can be used to solve the linear equations system that spans across the components.

```
class RevoluteJoint: public Component
  Signal flangeTo, flangeOn
  double phi, phi_der, w, z, residualTorque;
  ContinuousState position{ phi,phi_der };
  ContinuousState velocity{ w,z };
  Tearing zeroTorque{ phi_der, residualTorque };
  void evalState();
  void evalKinetic();
  void evalImpulse();
  virtual void metainfo(Meta& meta)
```

Figure 4: C++ class diagram of a revolute component.

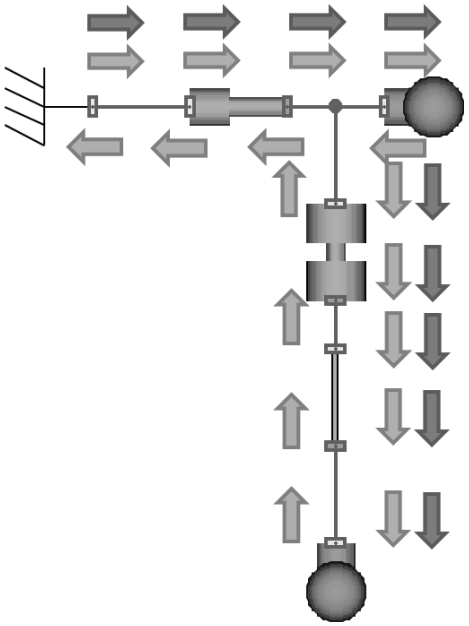


Figure 5: Crane crab modelling diagram

In practice this means that the code for each component can be represented by dedicated C++ class and the computational blocks can be expressed by C++ member functions. This is illustrated in Figure 4 with matching colors.

3.3 Handling of meta information

It is necessary to register the state-variables and their derivatives in the system. This is done by member objects. In similar vein, tearing variables and residuals are registered. The tearing variable belongs to the green signal and used to probe the system whose linear response can be assessed by the residual belonging to the orange signal flow. Furthermore, for each block (or member function) the dependence on the input and output signals needs to be registered.

For these purposes each component class must contain a virtual member function called `metainfo`. This function takes a crawler object by reference and depending on the implementation of the crawler different metainformation may be extracted.

3.4 Ordering of blocks, compacting and extracting of sub-blocks.

When instantiating a component class, all structural relevant meta data are collected. With this information, it is now possible to put all calls to member functions of all components in correct order. First a partial order is created based on the signal dependence, then the linear systems (of different dimension) marked by tearing and

residual variables are compacted and in the last stage the dynamic part is compacted (meaning that the all constant evaluations are placed upfront and all calls not needed for derivative evaluation are put last). With this ordering the linear subsystems can be constructed and solved and the overall model can be evaluated.

3.5 ODE simulation

Runge-Kutta solvers with fixed step size of order 1 to 4 as well as Backward Euler and ESDIRK23 [3] with variable step-size control have been implemented as numerical ODE solvers.

3.6 Overall Simulator Software Design

The major classes of the simulator are:

- `ModelEvaluation` instantiates the model, collects structural meta-information, orders the computational blocks and offer model evaluation
- `Simulator` implements numerical ODE solver
- `Recorder` collects references to desired outputs and generates output either to file, memory or TCP port.

4 First scaling results

4.1 The scaling experiments

The crane crab model of Figure 5 has been used to perform a simple scaling experiment. Using a logarithmic grid of factor 4, 1 to 16384 crane crabs have been instantiated and simulated using `zimsim`. The same exercise has been performed within a commercial Modelica tool. Here, this generates models ranging from 206 equations up to 3.1 million equations. In Figures 6 and 7, we use the equation number since this quantity is more familiar to Modelica users.

4.2 Scaling results in terms of memory usage

Peak memory usage for model translation, compilation and simulation is orders of magnitudes lower using `zimsim`. The likely reasons are:

- Avoidance of flattening
- On demand generation of meta-information
- More variables on stack than on heap

For small models, comparing the memory usage hardly makes sense because we compare the memory usage of a dedicated simulator (LIED) with the one of a whole modeling and simulation environment (Modelica Tool). The two orders of magnitude for small models is thus not surprising. What is surprising that this gap does not significantly close for larger models. Peak memory consumption seems to appear in the models at model translation. Zimsim has been written with efficient use of memory in mind: Meta information is generated only on-demand and also the object-oriented formulation enables to do more computation on the steak than on the heap.

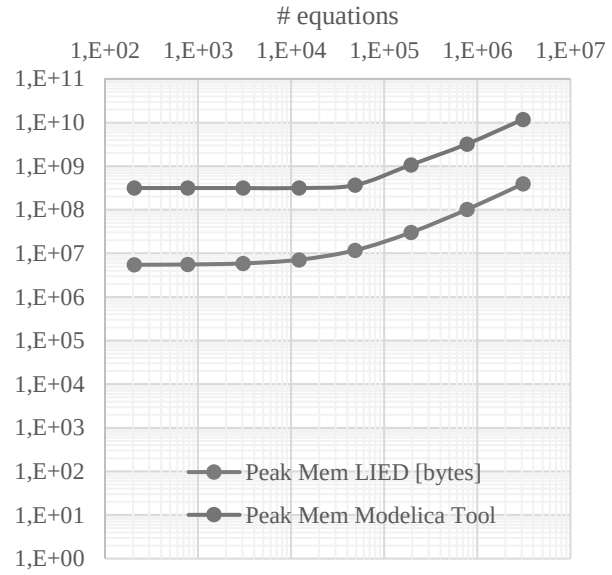


Figure 6: Memory usage in bytes

4.3 Scaling results in terms of performance

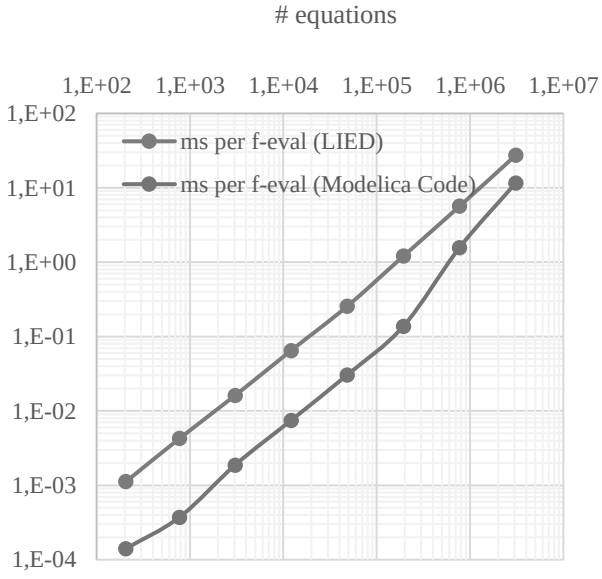


Figure 7: Time for single model evaluation in ms.

Simulation speed is up to an order of magnitude slower using `zimsim`. The likely reasons are:

- More overhead in `zimsim` due to interface variables and many function calls.
- Numerical solution of linear equation system in `zimsim` instead of symbolical transformation.

Pure simulation time is compared here. Time for translation, compilation and instantiation is ignored. Generation of output has been reduced to a negligible amount.
One has to consider that the code generation of the Modelica tool underwent decades of optimization whereas zimsim is still prototypical. A reduction of 50% in computational effort might be achievable for `zimsim`, however a certain gap will always remain.

We can observe that for larger systems, the performance penalty is smaller, the reason is unknown but is probably due the higher memory efficiency in `zimsim`.

A final remark: results for even larger systems could not be attained for `zimsim` as well as for the Modelica tool, however for completely different reasons. The Modelica tool started to hit the limits of memory leading to excessive translation time due to disk swapping. `zimsim` had no such problems but the instantiation has been implemented with a low-performing algorithm, requiring too much time. This is however a pure implementation issue and will be improved.

5 Outlook

The prototypical simulator demonstrates that LIED system can actually be coded directly in C++ and executed with acceptable efficiency and while using comparably little memory. The complexity of the software is thereby significantly lower than of any Modelica environment. The scaling capabilities are promising and can be improved by going to code dedicated for GPUs.

Although the modeling in C++ turned out to be much more natural than originally conceived, it is still not a desirable target. Instead, the existing C++ modelling libraries shall be used as an inspiration for Modelica compilers. Especially open compilers such as the OMC [2] could be modified to compile from Modelica libraries into a set of pre-compiled components.

The main, albeit preliminary, conclusion is that LIED systems not only form an interesting class for robust modelling but also an interesting class for large scale system simulation, worthy of further investigation.

References

[1] Fritzson, P., *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3*, 2nd ed., IEEE Press, Piscataway, New Jersey, 2014, pp. 1256.

[2] Fritzson, Peter A. et al. (2019) "The OpenModelica Integrated Modeling, Simulation, and Optimization Environment." *Proceedings of The American Modelica Conference 2018*, October 9-10, USA

[3] Jørgensen, J.B., Kristensen M. R. and Grove, P. (2018) A Family of ESDIRK Integration Methods. *arXiv Numerical Analysis* eprint :1803.01613

[4] Junglas, P. (2023) Implementing Thermodynamic Cyclic Processes Using the DLR Thermofluid Stream Library. *Simulation News Europe* E 33(4)

[5] Mattsson, S.E., Gustaf Söderlind (1993). "Index Reduction in Differential-Algebraic Equations Using Dummy Derivatives" In: *SIAM Journal on Scientific Computing* 1993 14:3, 677-692

[6] Pantelides, C. (1988), The consistent initialization of differential-algebraic systems, SIAM J. Sci. Statist. Comput., 9, 213–231

[7] Zimmer, D. (2020), Robust Object-Oriented Formulation of Directed Thermofluid Stream Networks . *Mathematical and Computer Modelling of Dynamic Systems*, Vol 26, Issue 3.

[8] Zimmer, D., N. Weber, M. Meißner (2022) The DLR ThermoFluid Stream Library. *MDPI Electronics - Special Issue*.

[9] Zimmer, D., C. Oldemeyer (2023). "Introducing Dialectic Mechanics". *Proceedings of the 15th International Modelica Conference*, Aachen.

[10] Zimmer (2024) Object-Oriented Modeling of Classic Physical Systems using Linear Implicit Equilibrium Dynamics *Preprints* 2024, 2024031139

Development of a Digital Twin for a Mobile Articulated Gripper Robot in Simscape Multibody

Andreas Apostolatos[1,2*], Jan Janse van Rensburg[1], Sebastian Groß[1], Stefan Kerber[1], Claudia Rubio Hervás[2], Minas Apostolakis[2], Chi-Chia Tung[2], Miguel A. Vega Torres[2], Steve Miller[1]

[1]The MathWorks GmbH, Weihenstephaner Straße 6, 81673 Munich, Bavaria, Germany; **aapostol@mathworks.com*
[2]Technical University of Munich, Arcisstraße 21, 80333 Munich, Bavaria, Germany

Abstract. Mobile articulated gripper arms are revolutionizing a range of fields, from additive manufacturing and packaging to automated assembly lines and surgical robotics. These innovations underscore the crucial role of simulation in design and development. Digital Twins are integral to this process, ensuring robustness, performance optimization, real-time monitoring and control, and adherence to industry standards. Preparing students for successful careers to develop such innovative engineering solutions leveraging state-of-the-art methods and approaches is of high importance. In this work, we demonstrate how *Project-Based Learning* (PBL) for such complex engineering tasks can be significantly enhanced and accelerated using MATLAB® and Simscape™ Multibody™. The results of this project clearly show that students can deliver high-end, robust solutions to complex engineering tasks, allowing them to efficiently familiarize themselves with advanced engineering topics by leveraging industry-mature modeling, simulation, test, and deployment ecosystems provided by MathWorks®.

Introduction

Today's job market requires highly skilled engineers equipped with the knowledge and tools to elaborate, develop, test, and eventually deploy qualified engineering solutions for diverse applications. As new technologies continuously emerge, the need for sophisticated development methods and comprehensive environments increases, enabling the handling of ever broadening application fields. Therefore, modern education necessitates the adoption of effective and efficient teaching methods such as *Flipped Classroom* and *Project-Based Learning* (PBL), see also in [1,2,3,4]. These allow students to gain practical experience using state-of-the-art platforms and methods they will eventually encounter in industry. The *Technical University of Munich* (TUM) in Munich, Germany, is one of the leading technical academic institutes in Europe and worldwide. It offers many diverse and interdisciplinary study programmes in engineering. Most these programmes, if not all, involve PBL as part of their curricula. This publication illustrates the development and investigates the results of a student project within the frame of PBL-based course "Software Lab" that took place in 2023. This lab is offered to master-level students at TUM annually. Academic and industrial partners collaborate with the participants by offering projects to pursue during this course, thus enabling students to elaborate practical projects of engineering significance in both academia and industry.

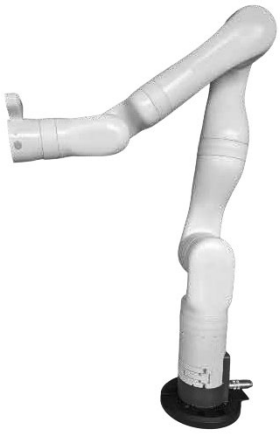


Figure 1 Kinova Gen3 robotic arm [5].

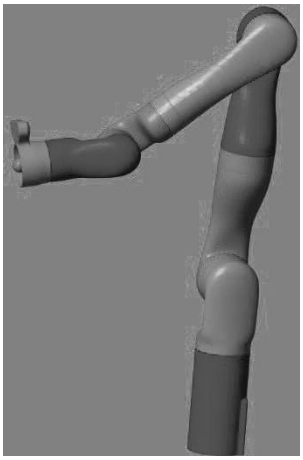


Figure 2 Kinova Gen3 robotic arm modelled in Simscape Multibody.

The results show that the use of PBL with industry-established engineering software increases motivation for the project elaboration, promotes learning outcomes, and enables students to develop practical and industry-relevant skills.

1 Problem definition

This student project was conducted in collaboration with MathWorks®, the makers of MATLAB® and Simulink® modeling and simulation platforms. Simscape™ Multibody™, an add-on product to Simulink, was used as Simulation platform. Simscape Multibody enables the modeling and simulation of multibody systems, which are integral part of robotic solutions. Coupled with the *Control Design* and *Hardware Deployment* capabilities of Simulink, Simscape Multibody enables modeling, simulation, test, and deployment of complex, industrial robotic systems.

The student team's engineering task was the development of a *Mobile Articulated Gripper Robotic Arm*, which can be used for tasks such as additive manufacturing, surgery, search and rescue missions, etc. As starting point for the project, the Multibody model [6] of the Kinova Gen3 [5] robotic arm was provided. The actual Kinova Gen3 robotic arm is shown in Figure 1, whereas its Multibody representation in Simscape Multibody is depicted in Figure 2. It is evident that the digital model exhibits strong resemblance to its physical counterpart.

The project objectives included:

- The setup of a multibody system for the mobile base with three wheels in Simscape Multibody,
- the setup of a multibody system for the mobile base coupled with the multibody representation of the articulated Gen3 robotic arm from Kinova,
- the implementation of a path following maneuvre for the mobile base,
- the implementation of a "pick and place" maneuver for the articulated robotic arm using inverse kinematics, and
- the coordination of the different tasks of the robot using a *Task Scheduler*, aiming to replicate an automated wall construction using Simscape Multibody.

The project goals were achieved in a timely manner with remarkable efficiency given the students' limited knowledge in robotics, MATLAB, and Simscape Multibody prior to the project elaboration.

The outcome of this study shows that MATLAB, Simulink, and Simscape Multibody form a strong platform for PBL due to their flexibility and adaptability. This enables students to fully leverage their skills while working on complex tasks.

Familiarization with the tools and methods needed for an engineering project like this is not trivial. The students were asked to use the MathWorks self-paced *Online Training Suite* (OTS) [7] to gain the required tools knowledge and skills. These self-paced online courses allow for an efficient and seamless familiarization with the MathWorks products and environments, necessary for the project elaboration.

2 Project elaboration

The project was expected to be elaborated and finalized throughout an academic year. Regular presentations were scheduled to allow the project participants disseminate their results, share the project status, and respond to questions from their peers, the project supervisors, and the course coordinators.

2.1 Articulated robotic arm and gripper end effector

A multibody system in `urdf`-format for the Kinova Gen3 robotic arm was provided as a starting point for the project. Function `smimport` [8] from Simscape Multibody allows for converting a CAD model (or an assembly) to a multibody system in Simscape Multibody, which considerably accelerates the development process.

In practice, one would employ torque control to move the joints of the articulated robotic arm and gripper end effector. It was suggested that motion control should be used instead to simplify the task. This modeling approach is suitable for early-stage design. Thus, the desirable motion was simply applied to the robotic joints. This suitable for early-stage design simplification allowed the students to complete the project promptly without the need for time-consuming controller tuning. Moreover, this modeling approach is equivalent to assuming that all necessary torque can be applied to the joints to achieve the desirable joint motion.

Subsequently, a set of waypoints was defined to have all joints of the articulated robotic arm operate in a coordinated manner so that the end effector can achieve the desirable end position. Using an appropriate number of waypoints allows the joints to find unique motion paths, leveraging the *Inverse Kinematics* (IK) solver in Simscape `simscape.multibody.KinematicsSolver` [9]. Just using a handful of waypoints might not be enough for the IK solver to find unique actuations for

all joints, as the inverse kinematics problem is by nature highly nonlinear.

Figure 3 shows the motion of the end effector gripper moving from the initial vertical position to finally reaching the brick on the ground. This highlights the efficiency and robustness of IK solver in Simscape.

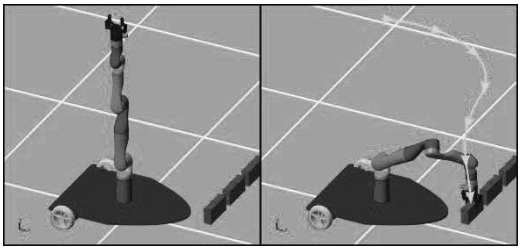


Figure 1 Inverse Kinematics Solver in Simscape (`simscape.multibody.KinemaricsSolver`).

2.2 Mobile base

The students decided to use a three-wheeled model to model the mobile base. This is convenient because it provides the necessary stability and allows for steering by controlling the rotational speed of each rear wheel. Additionally, there is no need for a suspension. This design is inspired by one of the models in the SolidWorks® tutorial; see in [10] for more information. Exporting the model from SolidWorks® to Simscape multibody can be achieved using the *Simscape Multibody Link* [11].

Various studies with different parameters were performed to find the most appropriate parameters that would allow the model to simulate stably. Moreover, the contact behaviour between the ground and the wheels was modelled using the *Spatial Contact Force* block [12] in Simscape. Tables 1 and 2 summarize the values chosen for the normal and frictional contact force coefficients used in this project, respectively.

Parameter	Value
Method	Smooth Spring-Damper
Stiffness	1e6 N/m
Damping	1e4 N/(m/s)
Transition region width	0.3 m

Table 1 Normal contact behavior

Parameter	Value
Method	Smooth Stick-Slip
Coefficient of static friction	0.9
Coefficient of dynamic friction	0.7
Critical velocity	1e-2 m/s

Table 2 Frictional contact behavior

Subsequently, the students sought to impose a path the mobile base should follow. As the robotic platform under consideration is of differential drive robot type, they decided to use the *Pure Pursuit* block in Simulink, (see in [13]), to obtain the linear and angular velocity control commands necessary for navigating the mobile base via a track defined by a provided set of waypoints. The students leveraged a *MATLAB Function* block [14] to convert these linear and angular velocity control commands to the rotational speed of the left and right wheels of the mobile base. The corresponding Simulink schematic is depicted in Figure 4. The Pure Pursuit block in Simulink is provided with the robot's pose, namely, its position and heading of the robot, and the corresponding waypoints it should navigate through. It returns the necessary control inputs for the linear and angular (rotational) velocities v and ω, respectively, that the robot must possess to navigate via the provided waypoints with the given pose.

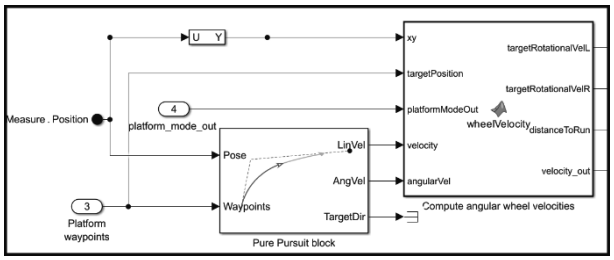


Figure 2 Pure Pursuit block and computation of the left and right wheel rotational velocities.

However, these control signals need to be converted to rotational speeds for the left and right wheel of the mobile base. To achieve that, the following formulas were employed and implemented in the MATLAB Function block with name *"Compute angular wheel velocities"*:

$$\omega_L = \frac{v - \frac{d}{2\omega}}{r} \tag{1a}$$

$$\omega_R = \frac{v + \frac{d}{2\omega}}{r} \tag{1b}$$

Please note that d and r in Eqs. (1) stand for the track width, namely, the distance between the centerline of the

two rear wheels, and the radius of each rear wheel.

It is worth noting that the pure pursuit algorithm cannot stabilize the robot once it has reached its destination. For this purpose, a triangular velocity profile reduction was implemented to stabilize the behavior of the mobile base. This ensures that the mobile base reaches its destination with a gradual reduction in its velocity, ultimately stopping at the destination. A proportional controller is used to obtain the required torque that needs to be applied at the left and right wheels to achieve the desirable rotational speed. The results of the target and achieved rotational velocities for both wheels are shown in Figure 5. A three-fold scaling of the error in terms of the rotational velocity for both wheels is used for the proportional controller. This simple *P*-controller can almost perfectly control the torque to achieve the desirable rotational velocity.

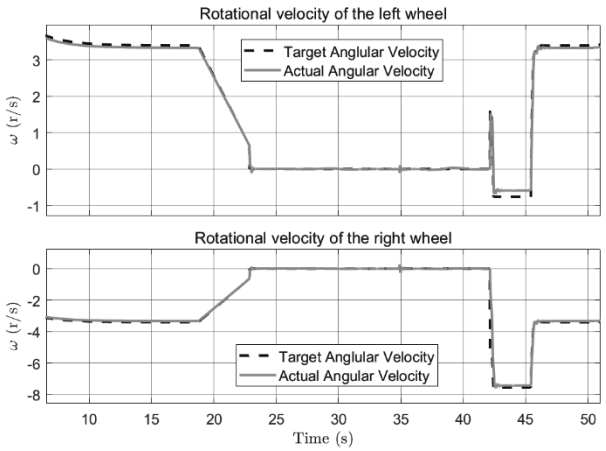


Figure 3 Target and achieved rotational velocities for both wheels of the mobile base using a proportional controller.

The triangular velocity profile employed for reducing the wheel's rotational velocity in the vicinity of the destination between can be clearly seen in Figure 5 at time instances 19 and 22.8 seconds. Employing this approach enables the mobile platform to reach its destination with stability and efficiency.

The waypoints are defined such that the mobile base navigates from the brick picking to the brick placing position. Moreover, Figure 6 depicts the track of the mobile base.

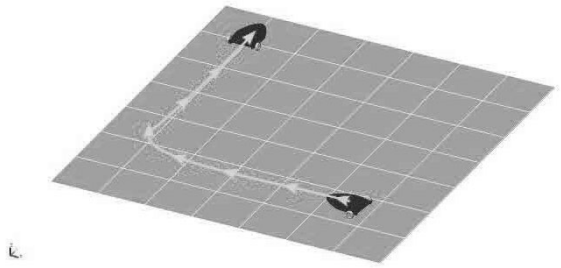


Figure 4 Mobile base navigating via waypoints using the Pure Pursuit algorithm in Simulink.

2.3 Coupling of the articulated robotic arm and the mobile base

Thus far, it has been highlighted how the students modelled and simulated the articulated robotic arm together with the gripper end effector and the mobile base separately in Simscape Multibody. A *Weld Joint* block [15] was used to establish a rigid connection between the two; see Figure 7.

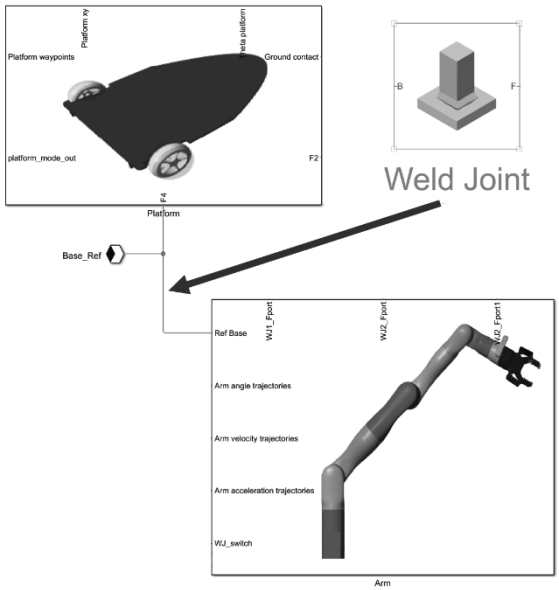


Figure 5 Rigid connection of multibody parts using the Weld Joint block in Simscape Multibody.

Joints in Simscape Multibody enable coupling different parts of a multibody system by introducing *Degrees of Freedom* (DOFs) that may restrict or enable the desirable motions depending on the application. Since the mobile base can move freely in this case, using a Weld Joint block between the mobile base of the articulated gripper arm has the desirable effect of mounting the articulated

gripper arm onto the mobile base rigidly.

2.4 Power supply for the rear wheels of the mobile base

The electrical system used to power the rear wheels of the mobile base is modelled in Simscape as a DC motor. The electrical motor is connected to a battery, shown in the blue section of the Simscape model in Figure 8. The *Motor & Drive (System Level)* block [16] was used to convert the electric current from the electrical part of the Simscape model to the mechanical part (see the green part in the Simscape mode in Figure 8). Simscape offers many blocks that allow Simscape models to be interfaced from different domains (mechanical, electrical, fluid, etc.). Finally, connecting the mechanical Simscape system in green to the Simscape Multibody system is also necessary. The torque and speed connections are denoted in Figure 8 using brown color. The *Rotational Multibody Interface* block is used [17] for this purpose.

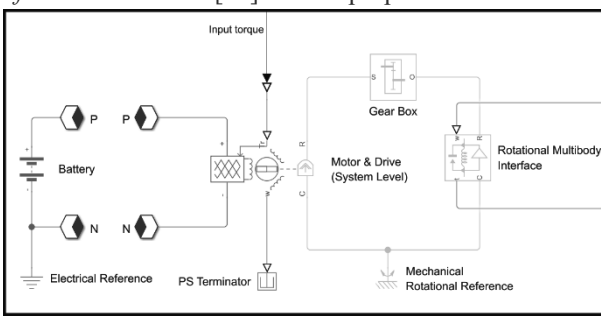


Figure 6 Modeling of a DC motor in Simscape.

The Rotational Multibody Interface block converts the mechanical force from the mechanical part of the Simscape model to torque that can be applied to the corresponding joints in the Simscape Multibody system. In this case, the torque is applied directly to the rear wheels of the mobile base. As mentioned in Section *Mobile base*, the torque is controlled using a proportional controller by scaling the error in the rotational velocity of each of the rear wheels threefold. Each rear wheel is powered by identical DC motors, as shown in Figure 8.

2.5 Task scheduler

A task scheduler had to be implemented in the simulation model to handle task management. The mobile articulated robotic arm must drive to the location of the bricks, pick up each brick, bring the robotic arm into a vertical position (idle position) minimizing the inertia while driving, drive to the location where each brick is supposed to be placed, and finally place the brick. The four main tasks that the task scheduler organizes include:

- Driving towards the destination where the bricks are stored,
- picking up one brick at a time,
- restoring the robot's arm posture to vertical (idle position),
- driving towards the destination where the bricks are to be placed,
- placing each brick to construct a wall.

These steps are shown in Figs. 9, 10, 11, 12, respectively. The task scheduler is implemented using a MATLAB Function block by means of switch branches based on a flag. The logic of the task scheduler could be also implemented graphically using Stateflow®, but the students decided to use MATLAB code in this project instead.

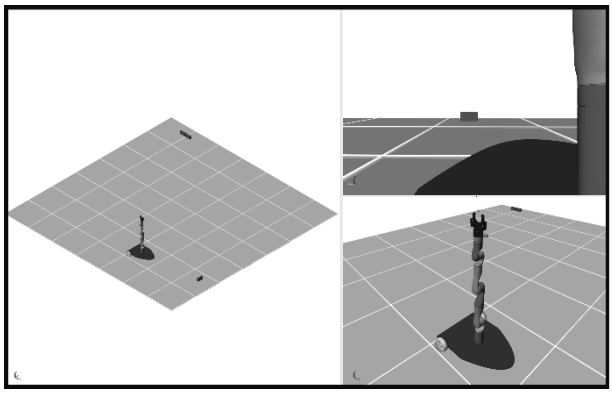


Figure 7 Task scheduler: Drive towards the destination where the bricks are stored with the arm in vertical posture (idle position).

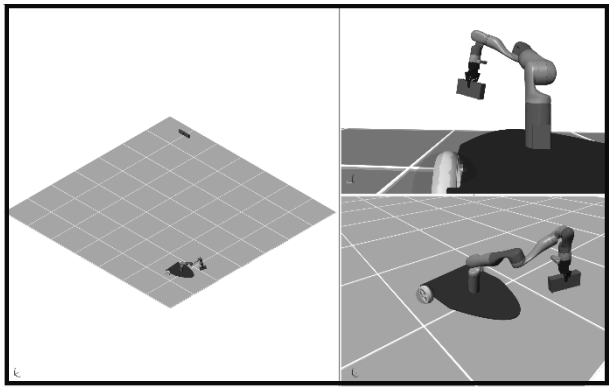


Figure 8 Task scheduler: Pick up each brick using the IK solver in Simscape.

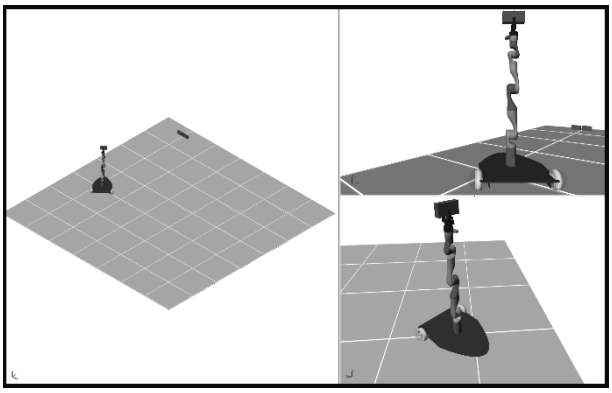


Figure 9 Task scheduler: Drive towards the destination where the bricks should be placed.

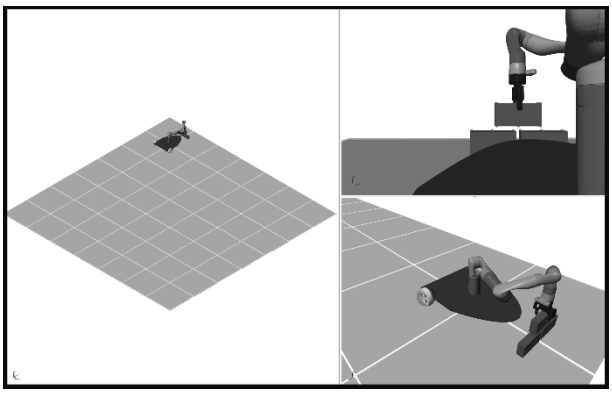


Figure 10 Task scheduler: Place bricks at the destination to construct a wall using the IK solver in Simscape.

3 Project results and learning outcomes

The main project objective of modeling and simulating a mobile articulated gripper robot that can pick and place bricks to construct a wall using Simscape Multibody has been achieved with remarkable efficiency.

Next, the power needed for the articulated gripper arm to pick the third, and last, brick (see Figure 13 accordingly) is investigated.

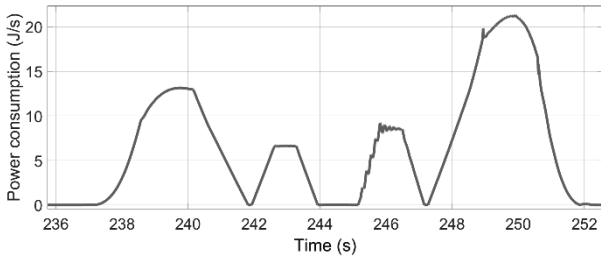


Figure 11 Power consumption of the articulated robot arm when picking up the last brick.

The power consumption for each joint can be computed using the following equation:

$$P = t \cdot \omega \tag{2}$$

where t and ω stand for the angle by which each wheel of the mobile base rotates and the corresponding angular (rotational) velocity. Simscape offers the *Rotational Power Sensor* block [18], among other sensor blocks, that can be leveraged to compute the power of a rotational mechanical system, see Eq. (2). Figure 14 shows the physical conserving ports of this sensor block, which are physical connections that conserve the mechanical system's rotational energy and the output signal port that senses the corresponding rotational power.

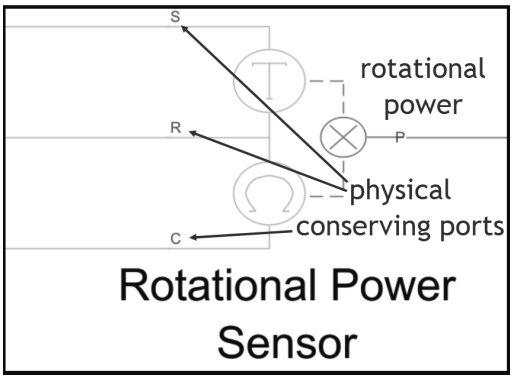


Figure 12 Rotational Power Sensor block for the computation of the power in Simscape.

Joints in Simscape Multibody enable coupling different parts of a multibody system by introducing DOFs that may restrict the motions as needed for the corresponding application. Using a Weld Joint between the mobile base of the articulated gripper arm has the desirable effect of mounting the articulated gripper arm onto the mobile base rigidly since the mobile base can freely move on the ground in this case.

The total power consumption needed for the movement of the mobile base is also computed in the same manner. The results are summarized in Figure (15).

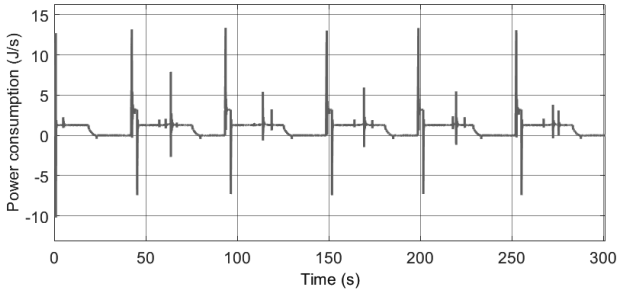


Figure 13 Power consumption of the mobile base throughout the simulation.

The power needed for the mobile base can be

exclusively attributed to the actuation of the two rear wheels according to the pure pursuit algorithm for path following and the corresponding *P*-controller. The relatively short instances where the power consumption becomes negative are worth noting. These can be attributed to the moments when the velocity is reduced as the mobile base approaches the destination using the employed triangular velocity profile using the proportional controller. When the robot's velocity is reduced by motion input, kinetic energy flows back into the electric motor. These instances can be considered as equivalent to regenerative breaking.

The students were able to ramp up quickly on the underlying topics even though they weren't experts in robotics nor familiar with the computational platforms employed prior to starting the project. They felt comfortable discussing advanced topics in robotics by the end of the project, such as physical modeling, control algorithms, inverse kinematics, and gaits among other. They proposed appropriate solutions to the challenges faced throughout the project. This project-based learning experience allowed the students to gain significant knowledge in robotics, the employed computational platforms, and team collaboration by working on a project with high practical impact.

4 Conclusions

This work underscores the importance of using adaptable and flexible simulation platforms, such as MATLAB, Simulink, and Simscape, to conduct courses based on Project-Based Learning, which enables students familiarize themselves with prominent industrial methods and workflows. Moreover, the importance of simulation for the early-stage design of robotic systems in the context of advanced engineering applications is also highlighted herein. It emphasizes the effective and efficient modeling and simulation of robotic multibody systems using Simscape Multibody, Simulink, and MATLAB. These tools offer the flexibility to choose the level of detail necessary for the required system fidelity. For example, we demonstrate that while the mobile base motors are modeled with a basic DC electric circuit, the articulated robotic arm's joint motion is simulated directly without specifying the power source. This approach simplifies early-stage design processes by focusing on essential elements. Additionally, we highlight that control design can be directly integrated within Simulink, allowing for easy computation and sensing of the required power using Simscape sensing blocks.

In conclusion, simulation is essential for any stage design of such robotic systems. It provides critical insights into the model under consideration that can be used to enhance productivity, robustness, and eventually, safety when such complex engineering solutions are deployed in practical applications. Moreover, simulations are widely used to produce data to train reinforcement learning models, create policies for optimal robot control, or train surrogate models from multibody systems via *Deep Learning* to speed up computational time, see for instance in [19]. To this end, modern teaching methods, such as Flipped Classroom, Project-Based Learning, Active Learning, etc., can be considerably enhanced by using industry-mature simulation platforms like the ones used in this study.

References

[1] D. Kokotsaki, V. Menzies, and A. Wiggins. Project based learning: A review of the literature. Improving schools, 19(3):267–277, 2016.

[2] J. A. N. Cocota, T. D'Angelo and P. M. de Barros Monteiro, *"A Project-Based Learning Experience in the Teaching of Robotics,"* in IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 10, no. 4, pp. 302-309, Nov. 2015

[3] I. D. L. Ros, A. Cazorla, J. M. Daz-Puente, and J. L. Yage, *"Project-based learning in engineering higher education: Two decades of teaching competences in real environments"*, Procedia - Soc. Behav. Sci., vol. 2, no. 2, pp. 13681378, 2010.

[4] A. Apostolatos and S. Gross, *"Teaching Advanced Topics in Numerical Engineering Using Project-Based Learning,"* 2024 IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece, 2024, pp. 01-10

[5] Kinova Robotics. *"Gen3 Robotic Arm"*. Kinova Robotics, https://www.kinovarobotics.com/product/gen3-robots. Accessed 2024-04-22

[6] Steve Miller. "*Industrial Robot Models in Simscape*". GitHub; 2024; Accessed 2024-04-22.

[7] MathWorks. "*Online Training Suite*". https://matlabacademy.mathworks.com. Accessed 2024-04-22

[8] MathWorks. *"`simimport` Function"*. https://www.mathworks.com/help/sm/ref/smimport.html. Accessed 2024-04-22

[9] MathWorks. *"KinematicsSolver"*. https://www.mathworks.com/help/sm/ref/simscape.multibody.kinematicssolver.html. Accessed 2024-04-22

[10] SolidWorks, "SolidWorks Full Tutorial – Mobile Car", YouTube video [Online]. Available: https://www.youtube.com/watch?v=U8WFaWKi3yo.

Accessed: 2024-04-22.

[11] MathWorks, *"Simscape Multibody Link"* [Online]. Available: https://www.mathworks.com/help/smlink/ug/installing-and-linking-simmechanics-link-software.html. Accessed: 2024-04-22.

[12] MathWorks, *"Spatial Contact Force Block"*. [Online] [3] Available: https://www.mathworks.com/help/sm/ref/spatialcontactforce.html. Accessed: 2024-04-22.

[13] MathWorks, *"Pure Pursuit Block"*. [Online]. Available: https://www.mathworks.com/help/robotics/ref/purepursuit.html. Accessed: 2024-04-22.

[14] MathWorks, "MATLAB Function". [Online]. Available: https://de.mathworks.com/help/simulink/slref/matlabfunction.html. Accessed: 2024-04-22.

[15] MathWorks, *"Weld Joint"*. [Online]. Available: https://www.mathworks.com/help/sm/ref/weldjoint.html. Accessed: 2024-04-22.

[16] MathWorks, *"Motor & Drive (System Level)"* [Online]. Available: https://www.mathworks.com/help/sps/ref/motordrivesystemlevel.html. Accessed: 2024-04-22.

[17] MathWorks, *"Rotational Multibody Interface"* [Online]. Available: https://www.mathworks.com/help/simscape/ref/rotationalmultibodyinterface.html. Accessed: 2023-04-22.

[18] MathWorks, *"Rotational Power Sensor"* [Online]. Available: https://www.mathworks.com/help/sdl/ref/rotationalpowersensor.html. Accessed: 2024-04-22.

[19] MathWorks, *"TWT GmbH Develops New Workflow for Tuning Automotive Suspension Designs Using Deep Learning and Multibody Simulation."* [Online] https://de.mathworks.com/company/user_stories/twt-gmbh-develops-new-workflow-for-tuning-automotive-suspension-designs-using-deep-learning-and-multibody-simulation.html. Accessed:2024-07-16

Benchmarking of Flatness-based Control of the Heat Equation

Stephan Scholz[1*], Lothar Berger[1], Dirk Lebiedz[2]

[1]Control and Process Engineering, University of Applied Sciences Ravensburg-Weingarten, Weingarten, Germany
[*]*stephan.scholz@rwu.de*

[2]Institute of Numerical Mathematics, Ulm University, Ulm, Germany

Abstract. Flatness-based control design is a well established method to generate open-loop control signals. Several articles discuss the application of flatness-based control design for (reaction-) diffusion problems in various scenarios. Beside the pure analytical derivation also the numerical computation of the input signal is crucial to yield a reliable trajectory planning. Therefore, we derive the input signal step-by-step and describe the influence of system and controller parameters on the computation of the input signal. In particular, we benchmark the control design of the one-dimensional heat equation with Neumann-type boundary actuation for pure aluminum and steel 38Si7, and discuss the applicability of the found input signals for realistic scenarios.

1 Introduction

The flatness-based control method is an open-loop technique to steer the system output along a reference trajectory [1]. In case of finite-dimensional linear and nonlinear systems the input signal $u(t)$ is found by a finite number of derivatives of a (differentially flat) output which equals the reference signal. This approach is extended to infinite-dimensional and distributed parameter systems where theoretically an infinite number of derivatives of output signal $y(t)$ is necessary to compute the input signal $u(t)$, see [2, 3, 4]. However, for practical reasons we can only consider a finite number of derivatives of the output signal. Thus, we need to show that the computation of input signal $u(t)$ converges for a certain number of derivatives of $y(t)$. In general, this estimation of convergence is not trivial because the computation of $u(t)$ depends on system and control parameters. A related approach about the controllability of the heat equation with a finite number of derivatives of y is discussed in [4].

In this contribution, we assume a one-dimensional linear heat equation with Neumann boundary actuation as depicted in Fig. 1 to discuss the impact of system and control parameters on the computation of input signal $u(t)$. For this purpose, we compare pure aluminum and steel 38Si7 to exemplify our findings. They differ in their material properties: thermal conductivity λ, specific heat capacity c and density ρ. Regarding the control parameters, we design the reference trajectory as a smooth step which is configured by the transition time and the steepness [5]. In each step of the analysis, we evaluate numerically the significance of the system and control parameters on the final control signal. Hence, we show the transition from a pure analytical towards a simulation-based control design, which enables us to distinguish whether or not a control signal is indeed applicable for a system.

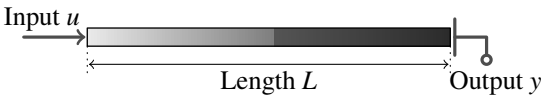


Figure 1: One-dimensional rod with heat input (left) and temperature measurement (right).

In section 2 we introduce the flatness-based modeling for the one-dimensional heat equation and derive input signal $u(t)$. The influence of the system parameters are analyzed in section 3. The trajectory planning problem and the subsequent discussion of the control parameters are described in section 4 and 5, respectively. Finally, in section 6 we present the simulation results of the open-loop system and review the applicability for realistic scenarios.

2 Flatness-based Control

We assume a one-dimensional heat conduction model as portrayed in Fig. 1 which is described by the linear

equation

$$\dot{\vartheta}(t,x) = \alpha \frac{\partial^2}{\partial x^2} \vartheta(t,x) \tag{1}$$

for $(t,x) \in (0,T) \times (0,L)$ and the Neumann boundary conditions as actuation on the left side

$$u(t) = \lambda \frac{\partial}{\partial x} \vartheta(t,x) \cdot \vec{n}_0 \Big|_{x=0}, \tag{2}$$

and thermal insulation on the right side

$$0 = \lambda \frac{\partial}{\partial x} \vartheta(t,x) \cdot \vec{n}_L \Big|_{x=L} \tag{3}$$

where the outer normal vectors are known as $\vec{n}_0 = -1$ and $\vec{n}_L = 1$. Here, we denote the temperature as ϑ, the thermal conductivity as $\lambda > 0$ and the diffusivity as $\alpha = \frac{\lambda}{c\,\rho}$ with specific heat capacity $c > 0$ and density $\rho > 0$. This heat conduction model is strongly simplified because in real world scenarios, often we have to consider two- or three-dimensional heat conduction with temperature-dependent material properties and probably thermal emissions consisting of linear heat transfer and nonlinear heat radiation towards the environment, see also [6, 7]. However, such realistic heat conduction scenarios lead to a much more complex mathematical discussion which is out of scope of this contribution, and the presented control method and its numerical analysis might not be applicable anymore. The initial temperature distribution is arbitrarily defined by

$$\vartheta(0,x) = \vartheta_0(x)$$

for $x \in [0,L]$ and the temperature is measured on the right boundary as

$$y(t) = \vartheta(t,L). \tag{4}$$

As known from the literature [2, 3, 4] the heat equation can be represented by a power series approach. So, we define power series

$$w(t,x) := \sum_{i=0}^{\infty} w_i(t) \frac{(L-x)^i}{i!}$$

and find its derivatives with respect to position x as

$$\frac{\partial}{\partial x} w(t,x) = -\sum_{i=0}^{\infty} w_{i+1}(t) \frac{(L-x)^i}{i!} \quad \text{and} \tag{5}$$
$$\frac{\partial^2}{\partial x^2} w(t,x) = \sum_{i=0}^{\infty} w_{i+2}(t) \frac{(L-x)^i}{i!}.$$

We model heat equation (1) in terms of

$$\dot{w}(t,x) = \alpha \frac{\partial^2}{\partial x^2} w(t,x),$$

identify both sides by its power series expressions as

$$\sum_{i=0}^{\infty} \dot{w}_i(t) \frac{(L-x)^i}{i!} = \alpha \sum_{i=0}^{\infty} w_{i+2}(t) \frac{(L-x)^i}{i!}$$

and yield identity

$$\dot{w}_i(t) = \alpha\, w_{i+2}(t). \tag{6}$$

Next, we apply the information of both boundary sides on identity (6) to derive the input signal. Firstly, we consider the output signal (4) as

$$y(t) = w(t,L) = \sum_{i=0}^{\infty} w_i(t) \frac{0^i}{i!} = w_0(t)$$

which implies $\frac{d^i}{dt^i} y(t) = \frac{d^i}{dt^i} w_0(t) = \alpha^i w_{2i}$ with identity (6). Secondly, the boundary condition on the right side (3) is formulated as

$$\lambda \frac{\partial}{\partial x} w(t,L) = -\lambda \sum_{i=0}^{\infty} w_{i+1}(t) \frac{0^i}{i!} = -\lambda w_1(t) = 0$$

and we find $\frac{d^i}{dt^i} w_1(t) = \alpha^i w_{2i+1} \equiv 0$. Thus, identity (6) is described by the sequences

$$w_{2i}(t) = \alpha^{-i} y^{(i)}(t) \quad \text{and} \quad w_{2i+1}(t) = 0 \tag{7}$$

for all $n \in \{0,1,\ldots,\infty\}$. In the definition of boundary actuation (2) we insert Equation (5) to derive the input signal $u(t)$ as

$$u(t) = -\lambda \frac{\partial}{\partial x} w(t,0) = \lambda \sum_{i=0}^{\infty} w_{i+1}(t) \frac{L^i}{i!}$$

and further with $i \to 2i+1$ and Equation (7) as

$$u(t) = \lambda \sum_{i=0}^{\infty} \frac{L^{2i+1}}{\alpha^{i+1}} \frac{1}{(2i+1)!}\, y^{(i+1)}(t). \tag{8}$$

3 Influence of System Parameters

We are interested in the sequence values of series (8) because for implementation reasons we need to how much

memory has to be reserved for the computation of u and at which iteration i the summation can be stopped. The power series to compute input signal $u(t)$ can be separated in sequence

$$\eta_i = \frac{L^{2i+1}}{\alpha^{i+1}} \frac{1}{(2i+1)!}. \tag{9}$$

and the derivatives of the (desired) output signal $y^{(i+1)}(t)$. In this section we discuss the influence of the physical properties length L and diffusivity α on sequence η_i, and in section 5 we analyze the parameters of (target) output $y(t)$ and its derivatives.

Sequence η_i is positive for all $i \in \{0,1,\ldots,\infty\}$ as we assume $L > 0$, $\alpha > 0$, and has a crucial influence on the computation of the input function u because it scales the derivatives $y^{(i+1)}$. Thus, we need to know the approximate values of η_i. We use a rescaled version of sequence (9) as

$$\tilde{\eta}_i := \left(\frac{L^2}{\alpha}\right)^{i+1} \frac{1}{(2i+1)!} = \frac{\gamma^{i+1}}{(2i+1)!} = L\,\eta_i$$

where $\gamma := \frac{L^2}{\alpha}$ to show that η_i and $\tilde{\eta}_i$ increase up to some index i and decrease afterwards to zero. Increasing iterator i by one we yield

$$\begin{aligned}\tilde{\eta}_{i+1} &= \frac{\gamma^{[i+1]+1}}{(2[i+1]+1)!} \\ &= \frac{\gamma^{i+1}}{(2i+1)!} \frac{\gamma}{(2i+2)(2i+3)} = \tilde{\eta}_i\,\beta_i\end{aligned}$$

where $\beta_i = \frac{\gamma}{(2i+2)(2i+3)}$ and we notice

$$\frac{\tilde{\eta}_{i+1}}{\tilde{\eta}_i} > 1 \quad \Leftrightarrow \quad \beta_i > 1$$

and

$$\frac{\tilde{\eta}_{i+1}}{\tilde{\eta}_i} < 1 \quad \Leftrightarrow \quad \beta_i < 1.$$

Due to the definition of $\tilde{\eta}$ this concept holds also for the original sequence (9) as $\eta_{i+1} = \beta_i\,\eta_i$. So, the maximum value of $\tilde{\eta}_i$ and η_i and its corresponding iterations i_{max} depend only on γ. For example, if we assume $\gamma = 100$ then $\gamma < (2i+2)(2i+3)$ holds for $i \in \{1,2,3\}$ and we find the maximum value $\tilde{\eta}_4 = \frac{100^5}{9!} \approx 27557$.

Example: Comparison Aluminum and Steel

For our numerical evaluations we consider a rod of length $L = 0.2$ for two case scenarios: a rod made of pure aluminum [8] and a rod made of steel 38Si7 [9]. The physical properties of both materials are listed in Table 1. For aluminum we have $\gamma_{al} \approx 410$ and for steel 38Si7 we have $\gamma_{st} \approx 3588$. The sequences $\eta_{al,i}$ and $\eta_{st,i}$ and their ratios $\frac{\eta_{al,i+1}}{\eta_{al,i}}$ and $\frac{\eta_{st,i+1}}{\eta_{st,i}}$ which describe evolution of the sequences by iteration are portrayed in Fig. 2 in semi-logarithmic scaling. We find that inequality $\frac{\eta_{i+1}}{\eta_i} > 1$ or equally $\log_{10}\left(\frac{\eta_{i+1}}{\eta_i}\right) > 0$ holds in case of aluminum for $i \in \{1,\ldots,8\}$ and in case of steel $i \in \{1,\ldots,28\}$. Thus the maximum values of η_i for aluminum and steel are calculated by

$$\eta_{al,9} = \frac{L^{19}}{\alpha_{al}^{10}\ 19!} \approx 5.53 \cdot 10^9$$

and

$$\eta_{st,29} = \frac{L^{59}}{\alpha_{st}^{30}\ 59!} \approx 1.59 \cdot 10^{27}.$$

As both sequences $\eta_{al,i}$ and $\eta_{st,i}$ reach such enormous maximum values, computational issues related to big numbers and data types have to be considered in the implementation process.

Moreover, sequence $\log_{10}(\eta_{al,i})$ drops below zero for $i > 27$: $\eta_{al,28} \approx 0.73$, $\log_{10}(\eta_{al,28}) \approx -0.13$; and $\log_{10}(\eta_{st,i})$ drops below zero for $i > 82$: $\eta_{st,83} \approx 0.13$, $\log_{10}(\eta_{st,83}) \approx -0.87$ (not displayed in Fig. 2).

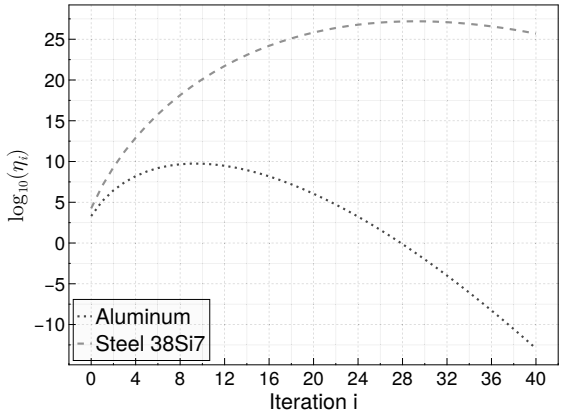
4 Trajectory Planning

According to [3, 5] we consider a transition from one fixed operating point to the next one as

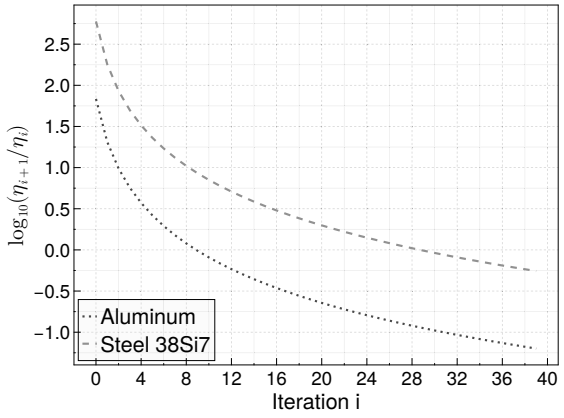
$$y(t) = y_0 + \Delta y\,\Phi_{\omega,T}(t) \tag{10}$$

Table 1: PHYSICAL PROPERTIES

	λ	ρ	c	$\alpha = \frac{\lambda}{\rho c}$
Aluminum	237	2700	900	$9.75 \cdot 10^{-5}$
Steel 38Si7	40	7800	460	$1.11 \cdot 10^{-5}$



(a) Sequence η_i



(b) Ratio η_{i+1}/η_i

Figure 2: Sequence η_i (top) and ratio $\frac{\eta_{i+1}}{\eta_i}$ (bottom) for aluminum and steel 38Si7.

where $\Delta y = y_f - y_0$, and with transition function

$$\Phi_{\omega,T}(t) = \begin{cases} 0 & t \le 0, \\ 1 & t \ge T, \\ \frac{\int_0^t \Omega_{\omega,T}(\tau)\,\mathrm{d}\,\tau}{\int_0^T \Omega_{\omega,T}(\tau)\,\mathrm{d}\,\tau} & t \in (0,T) \end{cases}$$

which uses the integral of the bump function

$$\Omega_{\omega,T}(t) = \begin{cases} 0 & t \notin [0,T], \\ \exp\left(-1/\left([1-\frac{t}{T}]\frac{t}{T}\right)^{\omega}\right) & t \in (0,T). \end{cases}$$

Parameter ω steers the steepness of transition $\Phi_{\omega,T}$ and is chosen such that the Gevrey order $go = 1 + \frac{1}{\omega} < 2$ or equally $\omega > 1$. A small value of ω, e.g. $\omega = 1.1$ means a rather flat transition, whereas a large value, e.g. $\omega = 3.0$ means a quite steep transition, as depicted in Fig. 3. To compute the input signal $u(t)$ in Equation (8) we only need to find the derivatives

$$\frac{d^i}{dt^i} y(t) = \Delta y\, \Phi^{(i)}_{\omega,T}(t) \tag{11}$$

where the derivatives of transition $\Phi_{\omega,T}$ are calculated as

$$\Phi^{(i)}_{\omega,T}(t) = \frac{\Omega^{(i-1)}_{\omega,T}(t)}{\hat{\Omega}_{\omega,T}} \quad \text{for} \quad t \in (0,T) \tag{12}$$

and $\Phi^{(i)}_{\omega,T}(t) = 0$ for $t \notin (0,1)$, using integral

$$\hat{\Omega}_{\omega,T} := \int_0^T \Omega_{\omega,T}(\tau)\,\mathrm{d}\,\tau. \tag{13}$$

In Fig. 3 trajectory $\Phi_{\omega,T}(t)$ and its first derivative are portrayed for varying $\omega \in \{1.1, 1.5, 2.0, 2.5, 3.0\}$. The derivatives $\Phi^{(i)}_{\omega,T}(t)$ can be computed symbolically using for example computer-algebra systems (see for example the MATLAB implementation [10]), numerically (which we do not recommend here). In this contribution, we compute the derivatives $\Omega^{(i)}_{\omega,T}$ with the JULIA library *BellBruno.jl* [11]. We note without a proof that an increasing order of differentiation of $\Omega^{(i)}_{\omega,T}$ leads to stronger oscillations because bump function $\Omega_{\omega,T}$ is a function composition and smooth as $\Omega_{\omega,T} \in \mathscr{C}^{\inf}((0,T))$, see also [2].

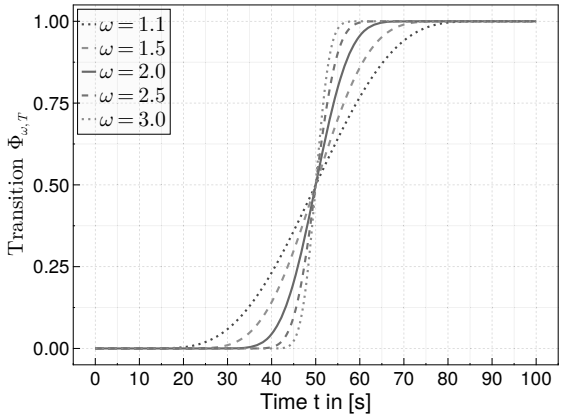
5 Influence of Control Parameters

The configuration of transition $\Phi_{\omega,T}$ and its derivatives are mainly driven by two parameters: final time T and exponent ω. In this section, we apply the L^2 norm

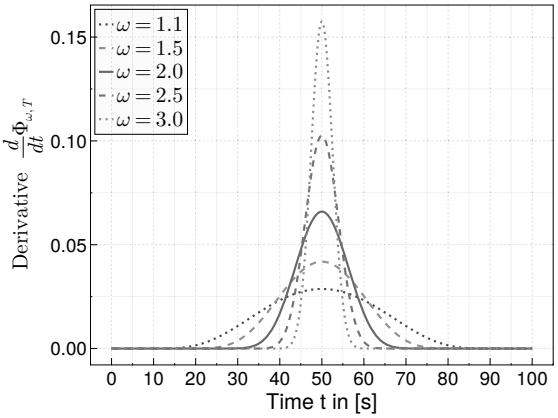
$$\|f\|_{L^2} = \sqrt{\int_0^T |f(t)|^2 dt}$$

on $\frac{d^i}{dt^i}\Omega_{\omega,T}(t)$ to unveil the influence of final time T and exponent ω on the computation of input signal (8). Noting the input signal with sequence η_n as

$$u(t) = \lambda \sum_{i=0}^{\infty} \eta_i\, y^{(i+1)}(t) \;=\; \frac{\lambda\, \Delta y}{\hat{\Omega}_{\omega,T}} \sum_{i=0}^{\infty} \eta_i\, \Omega^{(i)}_{\omega,T}(t)$$

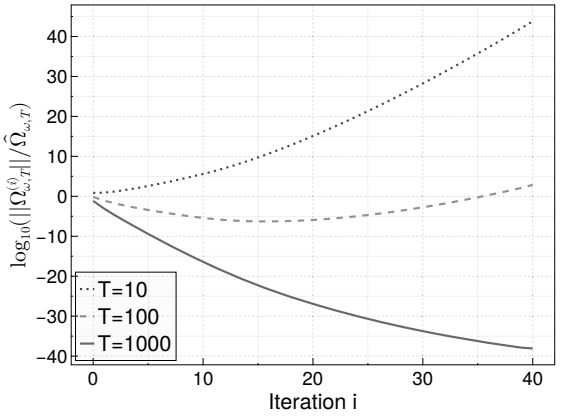


(a) Trajectory

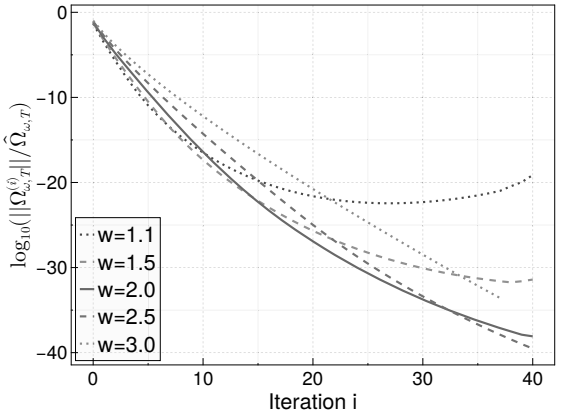


(b) First derivative

Figure 3: Trajectory $\Phi_{\omega,T}$ (top) and its first derivative (bottom) with $T = 100$ seconds and varying $\omega \in \{1.1, 1.5, 2.0, 2.5, 3.0\}$.



(a) Fixed $\omega = 2.0$, varying T



(b) Fixed $T = 1000$, varying ω

Figure 4: Norm of $\Omega^{(i)}_{\omega,T}$ with fixed $\omega = 2.0$ (top), and fixed $T = 1000$ (bottom).

using identities (11,12,13), we find the L^2 norm of $u(t)$ as

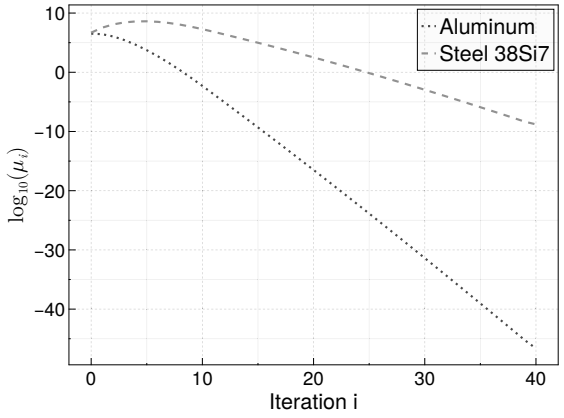
$$\|u(t)\|_{L^2} = \left\| \frac{\lambda\ \Delta y}{\hat{\Omega}_{\omega,T}} \sum_{i=0}^{\infty} \eta_i\ \Omega^{(i)}_{\omega,T}(t) \right\| \leq |\Delta y|\ \frac{\lambda}{\hat{\Omega}_{\omega,T}} \sum_{i=0}^{\infty} \eta_i\ \left\| \Omega^{(i)}_{\omega,T}(t) \right\|$$

where we assume $\lambda, \hat{\Omega}_{\omega,T}, \eta_i > 0$. We see that the power series is mainly driven by η_i (as discussed before) and derivatives $\Omega^{(i)}_{\omega,T}(t)$. Therefore, we are able to describe the quantitative behavior of the input signal by evaluating sequence

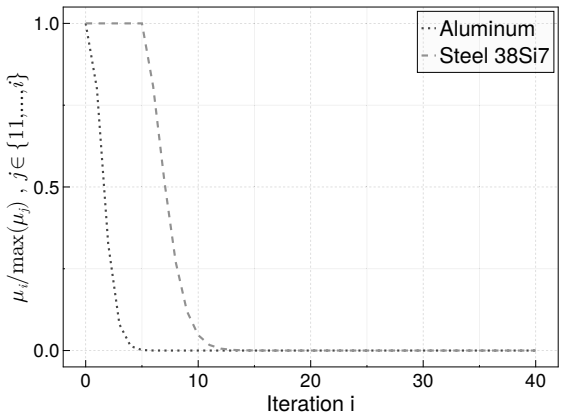
$$\mu_i := \frac{\lambda|\Delta y|}{\hat{\Omega}_{\omega,T}}\ \eta_i\ \left\| \Omega^{(i)}_{\omega,T}(t) \right\|_{L^2}. \tag{14}$$

Scaled norm $\|\frac{d^i}{dt^i}\Omega_{\omega,T}(t)\|_{L^2}/\hat{\Omega}_{\omega,T}$ is portrayed in Figure 4 in logarithmic scaling for two scenarios: fixed $\omega = 2$ and varying $T \in \{10, 100, 1000\}$; and fixed $T = 1000$ and varying $\omega \in \{1.1, 1.5, 2.0, 2.5, 3.0\}$. One notes that an increasing value only of final time T leads to a reduction of $\|\Omega^{(i)}_{\omega,T}(t)\|/\hat{\Omega}_{\omega,T}$, the influence of steepness ω may not be so clear here.

Furthermore, we take advantage of sequence μ_i to find a suitable maximum iteration number i_{max} to terminate the power series of $u(t)$ in Equation (8). Sequence μ_i consists of η_i as defined in Equation (9) and so we



(a) Sequence μ_i as in (14)



(b) $\mu_i / \max_{j\in\{1,\dots,i\}} \mu_j$

Figure 5: Sequence μ_i (top) and ratio $\frac{\mu_i}{\max_{j\in\{1,\dots,i\}} \mu_j}$ (bottom) for $\omega = 2.0$ and $T = 1000$.

distinguish aluminum and steel 38Si7 as noted in Table 1). The different values of η_i for aluminum and steel 38Si7 as in Fig. 2 lead to different values of μ_i: sequence μ_i approaching zero *faster* in case of aluminum than steel 38Si7 as depicted in Fig. 5 (a). Introducing the ratio $\frac{\mu_i}{\max_{j\in\{1,\dots,i\}} \mu_j}$ we find that the sequence elements μ_i vanish in case of aluminum for iterations approximately above $i = 5$ whereas in case of steel 38Si7 it takes at least $i = 12$ iterations - as portrayed in Fig. 5 (b).

The evaluation of μ_i and ratio $\frac{\mu_i}{\max_{j\in\{1,\dots,i\}} \mu_j}$ unveils two facts about the generation of input signal $u(t)$. Comparing the results for aluminum and steel 38Si7, we find in case of aluminum that only the very first derivatives of $\Phi_{\omega,T}$ are weighted by η_i and higher order derivatives have almost no influence on the computation of $u(t)$. Whereas in case of steel 38Si7 the weights of derivatives increase up to the fifth derivative so higher order derivatives (which tend to oscillatory behavior) influence the found input signal, too. We find an approximation of the signal input

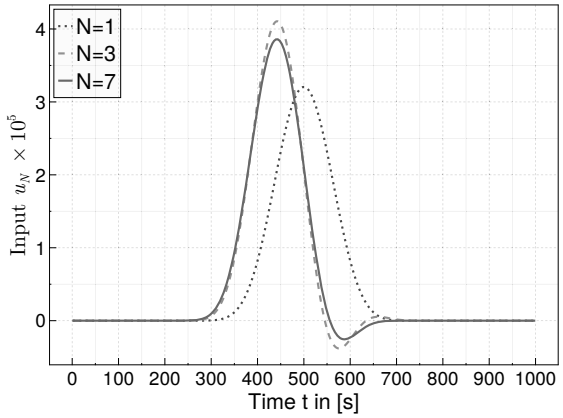
$$u(t) \approx \frac{\lambda\ \Delta y}{\hat{\Omega}_{\omega,T}} \sum_{i=0}^{N} \eta_i\, \Omega^{(i)}_{\omega,T}(t) =: u_N(t) \qquad (15)$$

where $N \in \mathbb{N}_{\geq 0}$ denotes the upper limit of iterations. Following the previous ideas, in case of aluminum a small value of N, e.g. $N = 7$, suffices to generate a good approximation. However, for steel 38Si7 we need a higher number of iterations, e.g. $N = 15$. The progress of input signals for aluminum with $N \in \{1,3,7\}$ and steel 38Si7 with $N \in \{5,10,15\}$ are presented in Fig. 6. These results confirm our previous analysis that the input signal needs more series elements for steel 38Si7 than for aluminum, and this leads the stronger oscillations in Fig. 6 (b) because higher derivatives of trajectory $\Phi^{(i)}_{\omega,T}(t)$ are necessary. In a nutshell, we find four important parameters which influence the input signal: length of rod L and diffusivity α which define sequence η_n, and final time T and steepness ω which influences the derivatives of trajectory $\Phi_{\omega,T}$ - and thus also the necessary number of summation iterations.

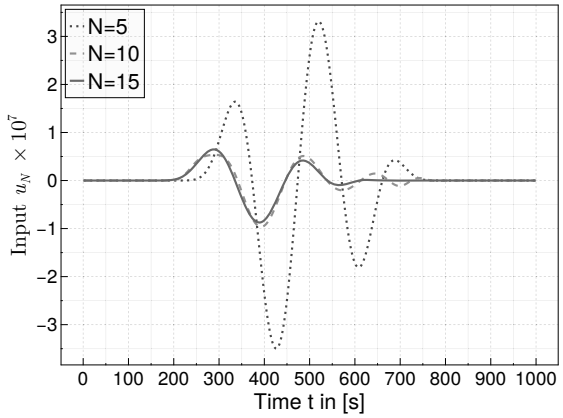
6 Simulation Results

In this section we compare the computed input signals and the resulting heat conduction simulation for aluminum and steel 38Si7. As above we assume the physical properties as listed in Table 1 and the trajectory parameters $T = 1000$ seconds and $\omega = 2.0$. So, the integral of the bump function as in identity (13) is found as $\hat{\Omega}_{\omega,T} \approx 17.06 \cdot 10^{-6}$. Further, we assume an initial temperature $\vartheta_0(x) = 300$ Kelvin which shall be increased by $\Delta y = 100$ Kelvin. A maximum iteration number of $N = 40$ is considered for approximation (15) in case of both scenarios. As explained in Section 5 lower values than $N = 40$ are also sufficient but it may rather imitate a summation until $N = \infty$. Heat equation (1) is discretized in space using finite differences with 101 grid points and is simulated using a Runge-Kutta numerical integration method for stiff systems, see [12].

The input signals and the resulting temperatures are illustrated in Fig. 7 for aluminum (a,c) and steel 38Si7 (b,d). In both cases the output, meaning the temperature at $x = 0.2$ meter, follows the reference and reaches 400



(a) Aluminum



(b) Steel 38Si7

Figure 6: Progress of approximated input signals u_N for aluminum (top) and steel 38Si7 (bottom) with $T = 1000$ and $\omega = 2$.

Kelvin. So, from a pure *mathematical* point of view the input signals are computed correctly for both scenarios. However, from a *physical* point of view we need to discuss the input signals and the resulting temperatures rather critically. Beside the fact that it may not be possible to apply negative input signals, e.g. if the actuator offers only heating and not cooling, it is in fact not possible to reach temperatures below zero Kelvin as portrayed for steel reference in Fig. 7 (d). We highlight that the strong oscillations of the input signal for steel 38Si7 in Fig. 7 (b) lead to the unrealistic temperature evolution in Fig. 7 (d). Therefore, the control parameters final time T and steepness ω have to be readjusted to decrease the necessary number of series elements, to yield a lesser or no oscillating input signal and a realizable temperature evolution.

Conclusion & Discussion

In this article we presented the computation of input signals for trajectory planning of a one-dimensional heat equation using flatness-based control design. We found in our analysis of the influence of system and control parameters on the computation of the input signal that different material properties (aluminum, steel 38Si7) result in completely different input signals and open-loop dynamics - even if all other parameters (length of rod, final time, steepness of transition) are the same. We demonstrate that strictly following the flatness-based control design may not lead to an physically realizable input signal even if the series in Eq. (8) converges. Thus, we recommend to simulate the heat equation with input signal to gain trustworthy arguments for the applicability of the computed flatness-based input signal. We motivate further research on the proposed approach for realistic scenarios in two and three dimensions including thermal convection and radiation.

Source Code

The source code is developed in JULIA programming language [13] and is available on *GitHub* and *Zenodo*: [14]. We implemented the simulations with the JULIA libraries *OrdinaryDiffEq.jl* [15] for the numerical integration in time of the spatially approximated heat equation, and *Makie.jl* [16] to create the figures.

References

[1] Fliess M, Lévine J, Martin P, Rouchon P. Flatness and defect of non-linear systems: introductory theory and examples. International Journal of Control. 1995; 61(6) (1995): 1327–1361.

[2] Laroche B, Martin P, Rouchon P. Motion planning for the heat equation. International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal. 2000; 10(8): 629–643.

[3] Rudolph J, Winkler J, Woittennek F. Flatness Based Control of Distributed Parameter Systems: Examples and Computer Exercises from Various Technological Domains. Shaker. 2003.

[4] Martin P, Rosier L, Rouchon P. Null controllability of the 1D heat equation using flatness. IFAC Proceedings Volumes. 2013; 46(26): 7–12.

[5] Utz T, Graichen K, Kugi A. Trajectory planning and receding horizon tracking control of a quasi-linear diffusion-convection-reaction system. IFAC Proceedings Volumes. 2010; 43(14): 587–592.

[6] Scholz S, Berger L. Modeling of a multiple source heating plate. arXiv preprint. arXiv:2011.14939 (2020).

[7] Scholz S, Berger L. Hestia.jl: A Julia library for heat conduction modeling with boundary actuation. Simulation Notes Europe SNE. 2023; 33(1): 27–30.

[8] Connor N. Aluminium - Periodic Table. [Online]. Available:https://www.periodic-table.org/Aluminium-periodic-table/. [Accessed: Mar. 28, 2024].

[9] Ovako AB. 38Si7. [Online]. Available: https://steelnavigator.ovako.com/steel-grades/38si7/. [Accessed: Mar. 28, 2024].

[10] Fischer F, Gabriel J, Kerschbaum S. coni-a Matlab toolbox facilitating the solution of control problems. Zenodo. 2022. Available: https://zenodo.org/record/6420876.

[11] Scholz S. BellBruno.jl. Zenodo. 2023. Available: https://doi.org/10.5281/zenodo.7685927

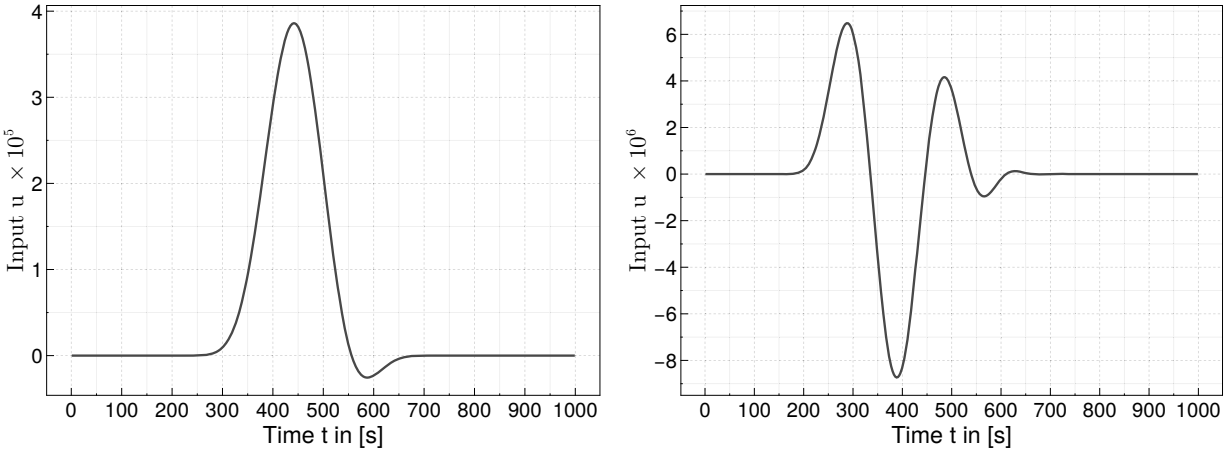
[12] Kennedy CA, Carpenter MH. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. Applied numerical mathematics. 2003; 44(1-2): 139–181.

[13] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. SIAM review. 2017; 59(1):65–98.

[14] Scholz S. BenchmarkFlatnessbasedControl.jl: BenchmarkFlatnessbasedControl.jl v0.1.0. 2024. Available https://doi.org/10.5281/zenodo.10890245

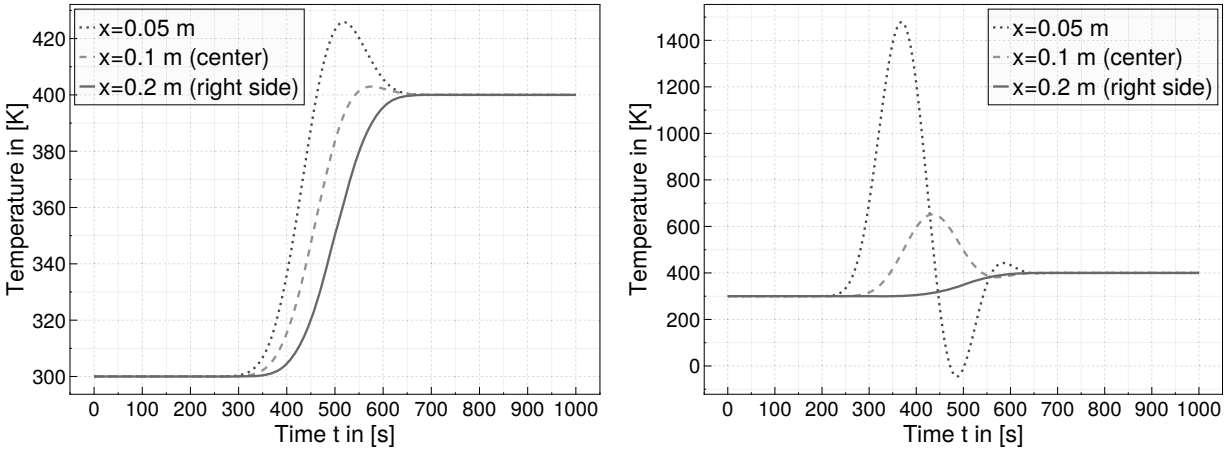
[15] Rackauckas C, contributors. SciML/DifferentialEquations.jl. Zenodo. 2022. Available: https://doi.org/10.5281/zenodo.7239171

[16] Danisch S, Krumbiegel J. Makie. jl: Flexible high-performance data visualization for Julia. Journal of Open Source Software. 2021; 6(65):3349.



(a) Aluminum

(b) Steel 38Si7



(c) Aluminum

(d) Steel 38Si7

Figure 7: Input signals and the resulting temperatures at position $x \in \{0.05, 0.1, 0.2\}$.

Comparing Different Pruning Strategies for the Evaluation Task of Virtual Stochastic Sensors

Dávid Bodnár[1*], Claudia Krull[1]

[1]Institut für Simulation und Graphik, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany; [*]*david.bodnar.ovgu@gmail.com*

Abstract. Virtual Stochastic Sensors calculate statistically relevant estimates in indirectly observable discrete stochastic systems. This is done by the proxel-based analysis that aims to reconstruct the relevant part of the state space with an iterative process. Strategically removing non-relevant proxels from the analysis (pruning) to reduce runtime overhead might potentially affect the results. And while the impact on the decoding problem has already been analysed in detail, the effect on the evaluation problem was not yet discussed.

The paper discusses three pruning strategies and compares their properties in case of the evaluation task. The theoretical statements are empirically proven using a car rental agency model in form of a Conversive Hidden non-Markovian Model.

The results show that in case of well chosen parameters all three pruning strategies are able to reach the same evaluation probability. The major difference between the strategies is due to their runtime properties which need to be carefully aligned with the use-case to reach optimal behavior. Based on the results the *fixed number of proxels pruning* strategy provides highly predictable execution time, while the *fixed threshold pruning* is very good at discovering a broader spectrum of the state space. The *variable pruning* is a very good trade-off between the previous strategies enabling lower thresholds and thorough state space analysis while maintaining acceptable execution times at the cost of more complex parametrisation.

Keywords: Virtual Stochastic Sensor, Hidden non-Markovian Model, Proxel-based Simulation, Pruning, Evaluation

Introduction

Virtual Stochastic Sensors (VSSs), introduced in [1], utilize the proxel-based analysis [2] to analyse partially observable discrete stochastic systems. The proxels, which represent a given system state at a given point in time, build a so-called proxel tree to reconstruct and represent the relevant part of the state space during the analysis. The insignificant part is pruned away.

There are different pruning strategies to define what is insignificant. In this paper three of them, *fixed threshold pruning*, *fixed number of proxels pruning* and *variable pruning*, will be discussed in detail. Similarly to [3] we aim to give an overview of the influence of the different pruning startegies on the evaluation results of VSSs. This is motivated by the generalisation and further development of the Change Adaptation Algorithm (CAA) described in [4].

As in [3], this paper utilizes the same car rental service model, presented in [5], for the evaluation. In this model, customers arrive in a premium or ordinary queue based on their membership. They use the same door for entering and leaving the shop area and this door is being observed (opening creates a signal) by the analysis. A single employee is serving both queues. Premium customers have priority over ordinary ones, but a customer is always served to the end if the processing has been started. Both queues are limited to 50 customers. Figure 1 shows the Augmented Stochastic Petri Net (ASPN) [5] of the system. ASPNs are modified Petri nets [6] that can model the emission of symbols when a transition fires. This model will also be used as an example in the following to enlighten the different concepts and tools used in this paper.

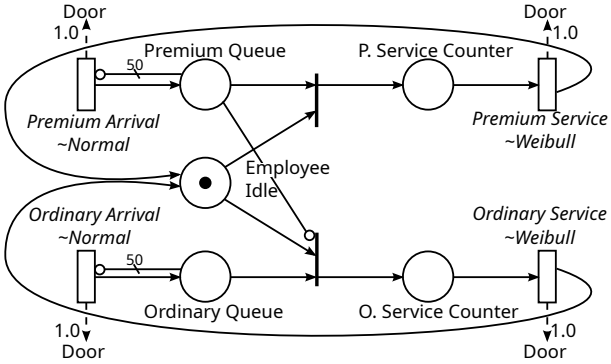


Figure 1: Car rental service Example as ASPN

This paper compares the three major pruning strategies in theory and using the above presented experi-

ment for the evaluation problem of Conversive Hidden non-Markovian Models (CHnMMs). The results show that after a given threshold there is no significant difference in the resulting probabilities, however, other major properties vary for each strategy so choosing one or the other for a given problem will always depend on the use-case.

1 Related Work

Virtual Sensors (VSs) [7] aim to collect system information that is hard or expensive to obtain in a direct way. If we combine them with stochastic processes a so-called VSS can be constructed which calculates statistically relevant estimates of system paramters that are similarly hard, inefficient and/or expensive to measure.

In this section, an introduction will be provided to such a stochastic process, the so-called CHnMM [8] and its solution algorithm, the proxel-based analysis. Additionally, proxel merging will be discussed.

1.1 Conversive Hidden non-Markovian Model

The concept of CHnMMs originates from the well-known Hidden Markov Models (HMMs) [9], where a hidden or partially hidden system is analyzed by probabilistic symbol emissions. This idea was extended to Hidden non-Markovian Models (HnMMs) by [10] to overcome the limitations of the discrete-time Markov chain in the background. HnMMs use arbitrary continuous distribution functions to describe the state changes and to create time dependence between them.

Similarly to HMMs, the HnMMs also try to solve the evaluation and the decoding problem. The decoding problem, finding the most probable generator state sequence to a given trace, and the impact of the different pruning strategies on it have already been discussed in [3]. This paper is focusing on the evaluation problem, e.g. finding the probability that a trace has been generated by a given model.

A specific subclass of HnMMs are the so-called CHnMMs which allow additional performance optimizations of the algorithm due to the fact that every state change results in an observable symbol emission. Those optimizations are powerfull enough to make CHnMMs the perfect experiment environment to test and verify new ideas in an efficient way. That is why this paper also limits its scope to CHnMMs.

In case of the presented car rental shop the hidden internal system state is the length of (the number of customers in) the ordinary and premium queues. Every state change (a customer entering or leaving the queue) results in a door opening (signal emission). This effectively means that by solving the evaluation problem, the goal of this project is to find the probabilty of a given door-opening signal protocol.

1.2 Proxels-based analysis

CHnMMs also need a solution algorithm. However, instead of using the forward algorithm [11] or the Viterbi algorithm [12] the so-called proxel-based analysis [2] can be used. The technique utilizes the encapsulation of a possible system state description (system state m, age vector τ, the probability of that state p, a timestamp t, etc.) into a so-called proxel object as it can be seen in Equation 1. One can use a collection of those proxels to describe all possible system states at a given point in time, including the ages of the relevant non-Markovian transitions. Then by creating a parent-child relationship between the timesteps one can derive the possible system states for the future timesteps in an iterative way. Of course, the proxel definition can be extended to carry additional information through the analysis. Refering back to the example model, a specific proxel represents a given number of customers in the premium and ordinary queues (m), the durations since the last customers entered each queue and while one is being served (τ) and the probability (p) of ending up in the represented state at the current simulation time (t).

$$P_x = (m, \tau, p, t) \tag{1}$$

The mentioned parent-child relation is described by the Hazard Rate Function (HRF) in Equation 2 using the Probability Density Function (PDF) $f(\tau)$ and the Cumulative Distribution Function (CDF) $F(\tau)$ of a possible state change. The equation describes the current state change rate for a given state change if this one has been active for τ and has not happened yet. So in case of our example model this translates to for example, what is the probability of a customer service being finished in this timestep, if he/she is being served right now. Or similarly, what is the probability of the next customer entering the shop now?

$$H(\tau) = \frac{f(\tau)}{1 - F(\tau)} \tag{2}$$

To keep the proxel tree at a reasonable size, very unlikely proxels can be pruned from the tree. Different strategies exist to perform this operation. They will be further discussed in Section 2.

Proxel probabilities are represented on a logarithmic scale to preserve precision as in case we represent them on normal scale, they would disappear due to arithmetic underflow very soon after the analysis was started.

1.3 Merging proxels

In case of the evaluation problem, we are not interesed in the history of a given proxel, because we are only trying to compute the probability of a given system state at time t. This means that after a couple of timesteps after starting the proxel-based analysis we can find proxels in the proxel tree, that represent the same system state, but they have been generated by different routes. These proxels can be merged by adding their probabilities (which are logarithmic probabilities, as stated before, represented by $\tilde{p}_1$ and $\tilde{p}_2$) using the Kingsbury-Rayner formula [9] shown in Equation 3. By performing this merging operation one can keep the size of the proxel tree under control very efficiently.

$$\tilde{p}_1 +_{log} \tilde{p}_2 = \tilde{p}_1 - \ln\left(1 + e^{-(\tilde{p}_2 - \tilde{p}_1)}\right) \tag{3}$$

The Kingsbury-Rayner formula additionally provides an opportunity to efficiently add probabilities that are on a different magnitude.

An example of a merging opportunity is presented in Figure 2, where the discovered state space of a three state system ($m0$, $m1$, $m2$) is reperesented over three timesteps. The two proxels marked with red at $t = 2\Delta$ represent the same internal system state with different probabilities but they were reached through different routes, so they can be merged.

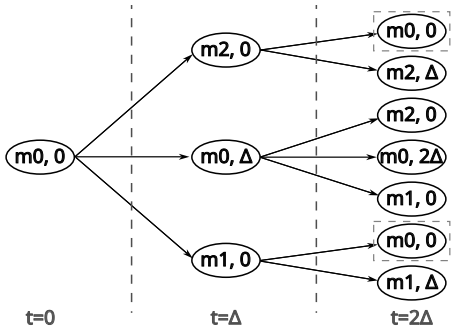


Figure 2: Example of mergable proxels

In case of our concrete example this means that the customer enter sequences (P - premium customer, O - ordinary customer) "PPOPO" and "POPPO" can be represented by the same proxel, since in case of the evaluation problem only the current system state is a subject of interest.

Merging significantely reduces the number of existing proxels over time. This also means that much less strict pruning configurations can be used than for example in case of the decoding task. These will be shown later in Section 3.2.

Until now, we have introduced CHnMMs and the proxel-based analysis as two base concepts of our research. We also gave a brief overview about the possibility of merging as the main method to keep the proxel tree under control in case of the evaluation problem. In the next section the concept of pruning and the different pruning strategies will be introduced as the main reseach interest of this paper.

2 Pruning Strategies for Evaluation

Pruning is a concept of the proxel-based analysis, in which the algorithm classifies a part of the proxel tree as irrelevant for further analysis and removes it. The different strategies encapsulate a set of logical steps to be performed on the proxel tree in order to identify and prune the irrelevant proxels.

There are two major events that can render a proxel based analysis infeasible. One of them is state space explosion, when the proxel tree grows exponentially and reaches a state where the next step is enormously expensive to compute or the ressources of the computer running the analysis are fully consumed. The other one is the exact opposite, the proxel tree can die out. In this case the remaining proxels, for example due to extreme pruning, encounter an observed symbol that cannot be emitted by their current system state and with that the proxel becomes impossible and is removed from the analysis. If all the proxels are removed from the tree, the tree dies out. Merging and a pruning strategy needs to be devised to prevent both of these extreme cases from happening

As already mentioned at the end of the previous section, merging does an excellent job at keeping the proxel tree under control. But sooner or later the analysis reaches the point where additional intervention is needed in the form of pruning. Still, merging makes it

possible to use less strict pruning strategies than in case of the decoding problem [3]. This also results in the fact that state space explosions or died out proxel trees are not common for simple models. However, with increasing system complexity a state space explosion might occur, but it is extremely unlikely.

In this section a brief overview will be given of the three investigated pruning strategies. But before diving into the details, we should list the properties of a good pruning algorithm. These are in case of the evaluation problem in our experience, the following:

1. High proxel processing throughput
2. Small amount of lost (pruned) probabilty
3. Prevents state space explosion
4. Scales the proxel tree up and down depending on how „interesting" the current part of the trace is
5. Guarantees predictable execution times
6. Is easily customizable for different needs
7. Is overall robust, which means that it does not react very differently to similar traces

These criteria will be used to evaluate the different pruning strategies.

2.1 Fixed Threshold Pruning

The *fixed threshold pruning* is a simple concept defining a pruning probability threshold $p(P_{pruned,t_i})$. Below that value every proxel is considered to be irrelevant and is then removed from the proxel tree. The threshold is defined compared to the probability of the most probable proxel $\max(p(P_{x,t_i})$ in the proxel tree for a given time t_i. It is described by a ratio r as it can be seen in the Equation 4. The parameter r is a freely selectable value with the limits $0 < r \leq 1$.

$$p(P_{pruned,t_i}) < r \max(p(P_{x,t_i})) \tag{4}$$

This pruning strategy is the most vulnerable to state space explosions and other instabilities as described in [3]. However, combined with merging it is possible to use extremely low pruning thresholds, as it will be shown in Section 3.2.

This strategy scales the proxel tree overall well and as a simple algorithm, it provides a high proxel processing throughput. But due to the unpredictable number of proxels in the proxel tree, the execution time has a high fluctuaction when the threshold is low.

2.2 Fixed Number of Proxels Pruning

The *fixed number of proxels pruning* is another simple strategy for keeping, as the name suggests, only a predefined number of proxels at the end of every timestep keeping the higher probability proxels and pruning the less likely ones. This property is the major advantage of the strategy, as it is very easy to parametrize and very robust against disturbances. However, the algorithm itself has some drawbacks.

The optimal proxel storage in case of the evaluation problem is a hash table [13], because due to the merging described in Section 1.3 one wants to retrive proxels as efficiently as possible. The cost of it is the hash tables's average search complexity, $O(1)$. However, a hash table cannot be sorted, so in order to perform the pruning, one needs to put all the proxels into a sorted array-like structure which is an additional overhead to the already complex sorting operation of $O(n\log(n))$ [13].

Of course, one can perform some implementation tricks, like storing a pointer to the proxel in the proxel storage instead of the object itself, to speed up the moving and sorting operation significantly. But in the end, one still needs to perform the moving and sorting of the entries and with a potentially higher number of proxels to keep, these can become too expensive for the analysis. Similarly, not all tricks might be universally universally available in every programming language.

This strategy provides a simple single parameter customization and very predictable execution times, but it fails to scale the proxel tree, so one really needs to find the perfect parameter with this strategy before running a long-time analysis.

2.3 Variable Pruning

Variable pruning was introduced in [4] due to the limitations of the *fixed threshold pruning* in case of the decoding problem. It is basically combining the previously discussed pruning strategies by creating a relationship between the current number of proxels in the proxel tree and the pruning threshold. This is done by defining a minimum number of proxels (r_{min}) - pruning threshold ($\#P_{min}$) pair, which prune the really unlikely proxels. Additionally, one selects a maximum number of proxels (r_{max}) - pruning threshold ($\#P_{max}$)

pair to drastically prune the tree if the proxel tree becomes too large. The two points must be connected by a strictly monotonically increasing continuous function ($r(\#P_{x,t_i})$) to guarantee a smooth transition between the two behaviors. The definition is shown in Equation 5:

$$r = \begin{cases} r_{min} & \text{if } \#P_{x,t_i} < \#P_{min} \\ r(\#P_{x,t_i}) & \text{if } \#P_{min} \leq \#P_{x,t_i} \leq \#P_{max} \\ r_{max} & \text{if } \#P_{x,t_i} > \#P_{max} \end{cases} \quad (5)$$

Even though the strategy is slightly more complex than the previous strategies due to the high factor of customization (different minumum and maximum pairs, different equations), it is a good trade-off between the advantages and disadvantages of the previous strategies. One can use overall much lower pruning thresholds while maintaining high throughput, lower risk of state space explosion and reducing the fluctuation of execution time compared to the *fixed threshold pruning*.

3 Experiments

In this section, the previously discussed properties of the different pruning strategies will be shown in an empirical way. First the experiment setup and the parametrization will be briefly discussed before describing the experiment results in details.

3.1 Experiment setup

A car rental service, presented in the introduction, was fed with the same input data as presented in [3] previously to make the result easily comparable.

A Personal Computer (PC) equipped with an AMD Ryzen 7 3800X and 64 GB of RAM has been used for the experiment execution. The RAM was sufficient to prevent swapping, which made the preformed experiments easily comparable. The implementation code utilized the C++20 standard and it was compiled using GCC 11.4.0 with the highest optimization level enabled. The application was containerized using Docker. The PC was running Manjaro Linux with the kernel 6.1.69.

3.2 Parameter selection

The experiment has been run 1620 times for every given pruning strategy with a defined parametrization, which includes 162 different model parameters with 10 randomly generated traces each. These models include only stable models, further discussed in [5].

The different pruning strategies were parametrized based on different logic. The *fixed threshold pruning* strategy (abbreviated with „th" in the following figures) was tested with various thresholds in the range of $[1e{-}1, 1e{-}150]$. Similarly, a wide range of sizes between $[50, 2500]$ were used to test the *fixed number of proxels pruning* strategy (abbreviated with „size" in the following figures). For the *variable pruning* (abbreviated with „var" in the following figures) five different equations were used, all with the minimum threshold (r_{min}) of $1e{-}300$ for proxel tree sizes below 1000 ($\#P_{min}$), and a maximum threshold (r_{max}) of 0.1 above proxel tree size of 100000 ($\#P_{max}$). All equations describe a mapping between the pruning threshold and the logarithm of the current tree size in the following way:

$$\begin{aligned} n &= \log_{10} \#P_{x,t_i} \\ r(n)_1 &= 10^{\frac{\log_{10} 0.1 - \log_{10} 10^{-300}}{\log_{10} 100000 - \log_{10} 1000}(n - \log_{10} 1000) + \log_{10} 10^{-300}} \\ r(n)_2 &= 10^{74.5n^2 - 446.5n + 369} \\ r(n)_3 &= 10^{49.5n^2 - 246.5n - 6} \\ r(n)_4 &= 10^{-50.5n^2 + 553.5n - 1506} \\ r(n)_5 &= 10^{-70.5n^2 + 713.5n - 1806} \end{aligned}$$

Figure 3 visualizes the previous equations between the previously described minimum and maximum pairs to make them easier to follow for the reader. Please be aware that the x and the y axis are using logarithmic scale.

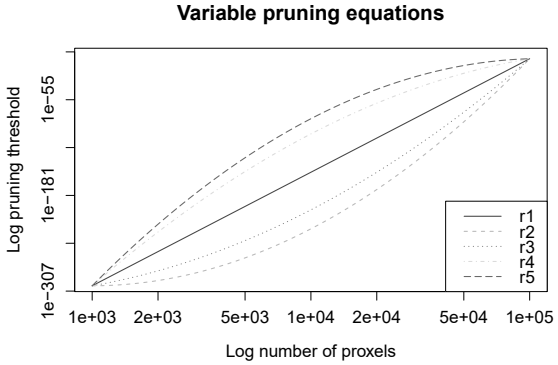


Figure 3: Variable pruning equations

The different equations make it possible to scale the *variable pruning* with different „speed" between the

minimum and maximum values. As the results in the next subsection will show, this was already enough to reach significantly different behavior.

3.3 Experiment results

For easier representation, the experiment results were visualized in the Figures 4 - 6 in a unified way. Yellow background („th" on the X-axis) marks the *fixed threshold pruning* strategies with different thresholds in scientific notation. Green background („size" on the X-axis) marks the *fixed number of proxels pruning* strategies with different fixed proxel tree sizes, while blue background („var" on the X-axis) marks the *variable pruning* strategies with different equation IDs.

During the tests, we did not experience any state space explosions, even with extremely small pruning thresholds. This means that the merging efficiently eliminates this problem, however, this is not a proof that in case of a more complex model we would not experience any problems with the *fixed threshold pruning* strategy, as this is the most vulnerable strategy regarding state space explosions.

Figure 4 visualizes the performance criteria of the experiment. Here we see that the *fixed threshold pruning* needs to deal with an extreme uncertainty regarding the number of proxels with decreasing pruning threshold. This leads to a similar fluctuation/variation in the execution time. The *variable pruning* efficiently copes with this problem while using a significantly lower minimal threshold. The *fixed number of proxels pruning* maintains predictable and low execution times.

In case of the proxel processing efficiency (Figure 4, top right) the *variable pruning* and the *fixed threshold pruning* with lower thresholds outperform the *fixed number of proxels pruning* by about $25-30\%$. This results from the fact that sorting the proxel tree is an operation that is hard to parallelize.

The last graph in the Figure 4 visualizes the probability lost through pruning. Please be aware that the probabilities are visualized on a logarithmic scale. One would like to minimize that in order to get the possibly most complete analysis of the state space. However, the first steps in case of the analysis play a crucial role in this case, because the proxel probabilities are decreasing drastically over the analysis time domain. This means that the first pruned away proxels basically determine the amount of lost probability. As we can see in the picture, the *fixed number of proxels pruning* has a really high spread for these values. This shows that it could potentially throw away important proxels. The *fixed threshold pruning* copes with the problem well at lower pruning thresholds. The *variable pruning* provides the best values with somewhat higher spread than the *fixed threshold pruning*. This is the indicator of being a good trade-off between the two other techniques.

Evaluation tries to compute the probability that a given trace has been generated by a given model. Interestingly enough, most of the strategies provided the same result to that question as it can be seen in Figure 5. Only the *fixed threshold pruning* strategies with the thresholds $1e{-}1$ and $1e{-}2$ failed to reach the same results, but in these cases only 1 or 2 proxels survived the timesteps on average. In case of the fixed pruning threshold $1e{-}10$, which is the first test case that came to the common solution, on average about 35 proxels survived the timestep after pruning, so we expect that a *fixed number of proxels pruning* with values under 50 could have also reached this result.

Not being able to reach a better result by processing more proxels has to do with the fact that the proxel probabilities are represented on a logarithmic scale. At the end of the analysis a subset of proxels became dominant (they had significantly higher probabilities) over the remaining proxels. That part of the proxel tree computationally defined the results.

This means that there is a sweet spot in the computation and with an optimal number of proxels one is able to compute the end result of the analysis very efficiently. In our case this sweet spot is somewhere between $35-50$ proxels. However, this cannot be stated universally, because more complex systems might have a higher optimal proxel number. Generally said, with a threshold based pruning strategy one can get to the optimum more easily than with the *fixed number of proxels pruning* strategy, because it is very hard to find the optimal number of proxels without performing multiple experiments.

Another important aspect is to keep enough proxels in the tree to support diversity and to prevent the proxel tree from dying out if something very unexpected happens, for example due to very strong pruning all the existing proxels become impossible in the next timestep. This quality can be visualized with the pruning ratio, so which amount of the proxels are kept on average after the pruning step. This can be seen in Figure 6.

Here we see that the strategies that failed to reach the common probability result have thrown away more than 50% of the proxels on average from timestep to

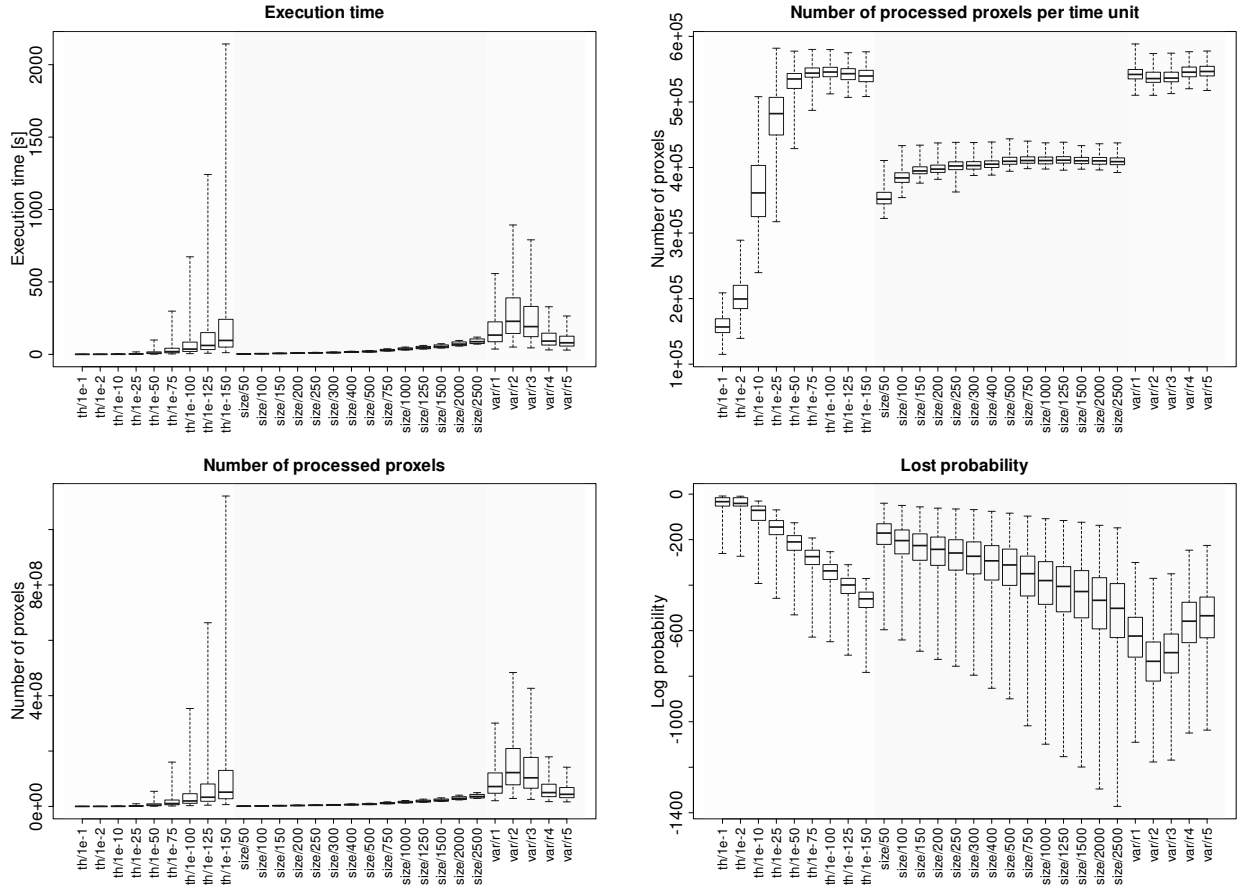


Figure 4: Quality Measures of Different Pruning Strategies
Note: Yellow background („th" on the X-axis) marks the *fixed threshold pruning* strategies with different thresholds. Green background („size" on the X-axis) marks the *fixed number of proxels pruning* strategies with different sizes. Blue background („var" on the X-axis) marks the *variable pruning* strategies with different equations.

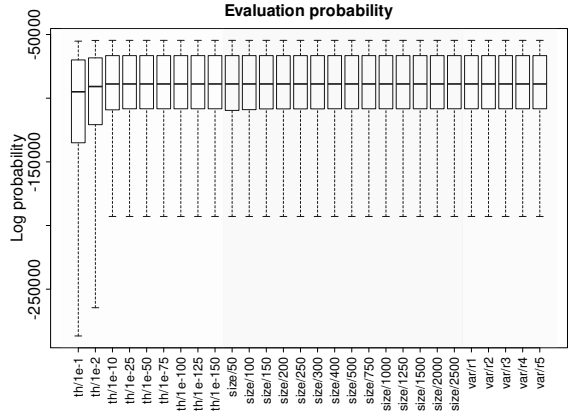


Figure 5: Probability of a given door protocol
Note: Information about the colors and abbreviations can be found in the note below the caption of Figure 4

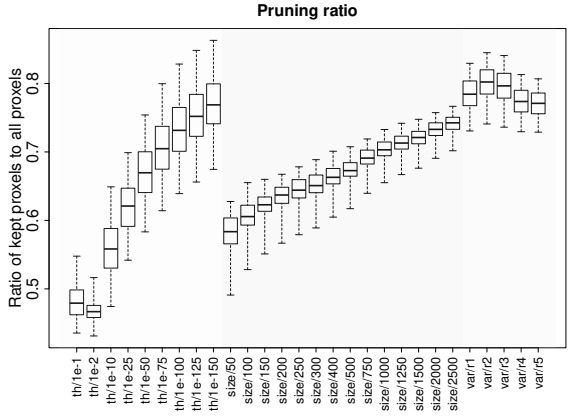


Figure 6: Pruning rate with different prunings
Note: Information about the colors and abbreviations can be found in the note below the caption of Figure 4

timestep. Generally, this is caused by the relatively small number of proxels. Of course, with the increasing number of proxels the different strategies are keeping a higher amount of proxels. This generally means also higher fault tolerance toward very unlikely events in the proxel-based analysis and also indicates higher robustness for similar traces.

4 Discussion & Conclusion

The goal of this paper was to give a general overview of different pruning strategies for the evaluation problem and to compare their properties.

All three of the analyzed pruning strategies are suitable for the evaluation problem. Problems like state space explosion or dying out proxel trees are very unlikely, as merging keeps the proxel tree under control enabling the use of less strict pruning strategies compared to the decoding problem [3]. We have not encountered them during our experiments. This generally also shows that the pruning strategy has a smaller impact on the results in case of the evaluation problem, as long as they are parametrized in a reasonable way.

There is generally an optimum where the proxel-based analysis reaches the final evaluation result with a minimum number of proxels. However, finding this sweet spot is not trivial and it surely requires multiple test runs, which is not always possible in the real world.

In case of all three strategies the parameters need to be chosen carefully. But with a good rule of thumb solution like „one should keep at least $50 - 100$ proxels on average, but one should aim for more" one should be on the safe side to get a good general solution with acceptable execution times. However, for more complex systems a higher number of proxels might be needed.

The *fixed number of proxels pruning* performs good enough to be a robust solution if one values highly predictable execution times over other properties. However, the proxel processing throughput might be a limiting factor. If the desired property of the analysis is to explore the state space as thoroughly as possible, one should use the *fixed threshold pruning* or tweak the *variable pruning* to speed up the execution time and get some additional advantages, like less lost probability, more diverse proxel tree, etc.

From a practical point of view there is no real limitation that would prevent the user from utilizing any of these pruning strategies in an experiment with artificial data or in real world use-cases.

References

[1] Krull C, Buchholz R, Horton G. Virtual Stochastic Sensors: How to gain insight into partially observable discrete stochastic systems. *The 30th IASTED International Conference on Modelling*. 2011;.

[2] Lazarova-Molnar S. *The Proxel-Based Method: Formalisation, Analysis and Applications*. Magdeburg: Otto-von-Guericke-Universität. 2005.

[3] Bodnár D, Krull C. Comparing Different Pruning Strategies for the Decoding Task using Virtual Stochastic Sensors. In: *The European Simulation and Modelling Conference 2023*, edited by Vingerhoeds R, de Saqui-Sannes P. Toulouse: EUROSIS-ETI. 2023; pp. 37–42.

[4] Bodnár D, Krull C. Adapting to Change of Model Transitions in Proxel Based Simulation of CHnMMs. In: *Proceedings Langbeiträge ASIM SST 2022, 26. Symposium Simulationstechnik*, edited by Breitenecker F. Vienna: TU Wien. 2022; pp. 101–108.

[5] Krull C. *Virtual Stochastic Sensors: Formal Background and Example Applications: Reconstructing the Behavior of Partially Observable Discrete and Hybrid Stochastic Systems*. Shaker. 2021.

[6] Bobbio A, Puliafito A, Telek M, Trivedi KS. Recent Developments in Non-Markovian Stochastic Petri Nets. *Journal of Systems Circuits and Computers*. 1998; 8(1):119–158.

[7] Wilson E. Virtual sensor technology for process optimization. 1997. Presentation at the ISSCAC 1997.

[8] Buchholz R. *Conversive Hidden non-Markovian Models*. Magdeburg: Otto-von-Guericke-Universität. 2012.

[9] Fink GA. *Markov Models for Pattern Recognition*. London: Springer London. 2014.

[10] Krull C, Horton G. Hidden non-Markovian Models: Formalization and solution approaches. *6th Vienna International Conference on Mathematical Modelling*. 2009;.

[11] Rabiner L. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*. 1989;77(2):257–286.

[12] Viterbi A. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *Information Theory, IEEE Transactions on*. 1967; 13:260 – 269.

[13] Skiena SS. *The Algorithm Design Manual*. Cham: Springer International Publishing. 2020.

Virtual Stochastic Sensors for Ambient Assisted Living - Analyzing the Effect of Generalized Resident Behavior

Vishwajeet Karumuri[1], Claudia Krull[1*]

[1]Fakultät für Informatik, Otto-von-Guericke-Universität, Postfach 4120
39016 Magdeburg, Germany; *claudia.krull@ovgu.de*

Abstract. The advancements in Ambient Assisted Living (AAL) have been prompted by the growing population of elderly individuals facing diagnoses such as Dementia or Alzheimer's, aiming to enhance their overall quality of life. To provide support it is important to know their daily activities and support them. A large portion of research in the field of Human Activity Recognition uses black box learning approaches such as deep learning, but there are cases where model based methods, such as Virtual Stochastic Sensors (VSSs) are competitive. This is possible because the model based methods can include system structure in the modeling process if it is known. VSS's are derived from Hidden Markov Models (HMM) and applied to a CASAS single resident dataset, which is an apartment fitted with different types of ambient sensors. For future applications a generalization of behavior, sensors or models is necessary so that models are not just trained and used for one specific apartment and setup. In this paper we analyze the effect of generalizing the residents behavior on the reconstruction accuracy. The generalization did lead to some improvements in the reconstruction accuracy, but the implications for the actual application need to be considered.

Introduction

Advancements in the medical field led to long and healthier lives, roughly around 20% of the world population will be aged above 60 by 2050 [1], seeking to explore effective solutions that empower elderly individuals to maintain independent living. Studies of Counsel and Care in UK showed that elderly people have a preference to stay in their apartments rather than nursing homes [2]. Researchers have shown that having clinical therapy at home has no negative effect on the process [3]. There are multiple ways to make this happen, one way is Ambient Assisted Living, where some ambient sensors are installed, e.g. motion sensors, to monitor the behavior of elderly residents, a model is used to guess the behavior using the sensor readings, then, this is used to identify if everything aligns with the usual behavior, and if not, assistance can be provided. This ensures a safer living space without unduly intruding on the privacy of the residents.

For replicating human behavior [4], [5] and [6] successfully implemented machine and deep learning algorithms for this task. In [7] Virtual Stochastic Sensors (VSSs) are used, which are designed to facilitate the reconstruction of partially observable stochastic systems and enable solving backward problems in the realm of stochastic modeling and simulation. The model is based on the ideas of Hidden Markovian Models (HMM) but extends these by arbitrary non-Markovian distribution functions for multiple concurrent processes and symbol outputs at arbitrary points in time [8]. VSS discretize the time domain and use a simple iterative algorithm to discover the reachable state space of the model, therefore being very flexible. However, they cannot be applied in real-world scenarios on a large scale yet, because model parametrization is not automated, and the model needs to be trained for a specific system to be used for reconstruction. The model generalization tested in this paper is one step towards pre-trained models for unknown systems.

This research aims to develop a conceptual model for a generalized set of activities, transform the dataset (CASAS Single resident apartment (HH101)) and feed the transformed data into the model to reconstruct human behavior, and later evaluate the results using different metrics. This dataset contains activities performed by the residents and the corresponding active sensors for the activities performed. The rest of the paper is structured as follows, Section 1 contains the details of

related research, Section 2 underlines the details of the dataset, Section 3 explains the conceptual model and the algorithm design, Section 4 shows the outcome of the research.

1 Related Work

This section explores different approaches for reconstructing human behavior in the field of ambient assisted living that are similar in approach or goal to this research. Additionally, it presents relevant information for Virtual Stochastic Sensors (VSSs).

[9] has mainly emphasized the duration of the activity to find the abnormality in human behavior using Explicit State Duration Hidden Markov Model (ESD-HMM). They checked the deformity of current activity which might be shorter or longer than the usual routine, the dataset used for this research was limited to the kitchen. [10] introduced a new observation probabilistic model to recognize daily activities, they incorporated temporal data which had information regarding 77 sensors. [4] tried to identify the critical features from the sensor data, these features are used to classify overlapped activities. Unsupervised K nearest neighbor (KNN) was applied to day-to-day activities but with a small set of activities. [5] implemented long short-term memory (LSTM) recurrent neural network (RNN) to perform activity recognition from wearable sensors, this implementation was not tested for activities of daily living. [11] suggested another approach using RNN on three different datasets, this approach outperformed similar approaches concerning accuracy and speed, but the dataset is not publicly available. [6] proposed an unobtrusive activity recognition classifier using deep convolutional neural network (DCNN) and publicly available CASAS Aruba dataset. [12] used a knowledge-driven approach, including a Partially Observable Markov Decision Process (POMDP) and exploited the task information, while the location is combined with the sensor events in the smart home, but the series of conditions are used to classify activity. This shows that AAL is a current research field with several approaches all with their individual features.

1.1 Virtual Stochastic Sensors

Virtual Stochastic Sensors represent a framework for analyzing partially observable stochastic systems, including different modeling paradigms and solution methods [13]. VSS can compete with some black box models when the hidden system structure information is available [7], and can incorporate such information to accurately represent dynamic system behavior and its relationship with the observable output. VSS use augmented stochastic Petri nets (ASPN) as user models that contain multiple concurrent non-Markovian transitions [14]. ASPNs generate observable output by the firing of transitions depending on the discrete system state, the discrete system output is collected in a protocol with associated time stamps, since in contrast to the Hidden Markovian Model (HMM), the model is defined in continuous time and can produce output at arbitrary points in time [7].

[7] has applied VSS on CASAS Aruba 2010 dataset and produced a very promising result, based on this VSS is considered to be a viable option for activity classification. However, the previous implementations were all trained and tested on the same use case, which is not a feasible approach for broad scale applicability. Therefore we are examining different methods of generalization to eventually enable generalized models to be applied on systems not used for the training. This paper is focusing on the impact of generalization of the activity set on the reconstruction accuracy for a single system.

2 CASAS Dataset and VSS Model

In this paper we are using a data set from the CASAS Research Project of Washington State University [15]. There are different types of datasets, one contains the daily activities of 20 participants, few other datasets include pets for single or multiple residents, and finally, HH datasets are mostly single residents but a small portion of them are two-resident apartments. In this research, the HH101 single resident apartment dataset is considered because it is multivariate, sequential, and time series. The data is collected using different kinds of sensors, like motion, door/temperature, and light switch sensors placed throughout the apartment, while the residents perform their normal routines. The dataset format is Date, Time, Sensor, Translate, and Activity (Table 1).

- Date is when the information is recorded in MM/DD/YYYY
- Time is in the 24-hour format
- Sensor is the name of the sensor

Date	Time	Sensor	Translate	Activity
08/01/2012	00:00:06	M008	LivingRoom	Watch_TV
08/01/2012	00:00:07	M008	LivingRoom	Watch_TV
08/01/2012	00:00:09	M008	LivingRoom	Watch_TV
		.....		
08/01/2012	01:12:51	M012	Bedroom	Sleep
08/01/2012	01:12:54	M012	Bedroom	Sleep
08/01/2012	01:12:55	M012	Bedroom	Sleep
		.....		

Table 1: Sample extract from single resident HH101 dataset

- Translate is room-level location
- Activity is what the person is doing at that specific date and time, and was tagged by the residents

The time frame of this dataset is from 20th July 2012 to 17th September 2012. From this time frame, only the data of September 3rd to 9th is used here.

2.1 Generalization of Activities

In this dataset a total of 34 activities were recorded (Table 2), describing the daily routine of a person living in this residence. For generalization, we grouped activities with similar characteristics and similar sensor outputs. These 34 activities are first generalized to 11 activities and further generalized to 6 activities. The smaller set of activities was created, to test the effect of different levels of generalization.

Before grouping similar activities, some were removed from the dataset. *Leave_Home*, *Enter_Home*, all medicine activities have very short durations are therefore not easily detectable through the sensor readings. The activities *Work_At_Table* and *Entertain_Guests* occurred very rarely and and were therefore ignored in the analysis. Some activities like *Go_To_Sleep*, *Wake_Up*, *Eat* also have very low occurrence, but can be combined with *Sleep*, other *Eat_* activities respectively, and will therefore not be omitted.

For the first type of grouping, activities like *Cook_Breakfast*, *Cook_Lunch*, *Cook_Dinner*, and *Cook* are grouped to *Cook*, even though these activities occur during different times of the day they essentially give the same idea. Similar to cook, we can combine all eat, wash dishes and sleep activities into *Eat*, *Wash_Dishes* and *Sleep* respectively. *Bed_Toilet_Transition* takes place for very little time and occurs during the *Sleep* activity hence it is grouped with sleep. *Relax* and

Activity	Occurences %
Watch_TV	26.59%
Sleep_Out_Of_Bed	7.21%
Bathe	7.06%
Cook_Breakfast	7.06%
Dress	6.09%
Toilet	5.95%
Personal_Hygiene	5.73%
Sleep	5.10%
Read	3.73%
Relax	2.51%
Cook_Dinner	2.41%
Drink	1.65%
Eat_Breakfast	1.55%
Morning_Meds	1.54%
Evening_Meds	1.46%
Wash_Breakfast_Dishes	1.39%
Cook_Lunch	1.39%
Wash_Dishes	1.39%
Leave_Home	1.35%
Cook	1.20%
Enter_Home	1.12%
Entertain_Guests	1.11%
Wash_Dinner_Dishes	1.07%
Phone	0.80%
Groom	0.78%
Step_Out	0.65%
Eat_Dinner	0.47%
Eat_Lunch	0.38%
Wash_Lunch_Dishes	0.34%
Bed_Toilet_Transition	0.31%
Eat	0.22%
Go_To_Sleep	0.18%
Wake_Up	0.16%
Work_At_Table	0.08%

Table 2: All 34 activities and their total share in occurence time the CASAS HH101 dataset

Sleep_Out_Of_Bed are combined to *Relax* because both activities are done in similar places and during similar times of the day, in similar sense *Personal_Hygiene* and *Bathe* are combined to *Freshen_Up*. This various grouping leads to 11 activities. The first grouping is shown in table 3, where the groups are separated by dashed and solid lines.

To obtain the smaller set, the activities are grouped in such a way that the exact information is not known but the general idea of the activities is not lost. *Freshen_Up* and *Toilet* are combined into *Freshen_Up*, activities like *read*, *Phone*, *Relax*, *Watch_TV* into *Personal_Activity*, as well as *Cook* and *Wash_Dishes* into *Meal_Prep*. This results in a total of 6 activities. This second grouping is shown in table 3, where the groups are separated by solid lines.

2.2 Data preparation

After cleaning the data, the next step is to modify the data into distributions and probabilities so that it can be used as input in the model. Each activity has breaks, short and long breaks for one type of Petri net and morning, evening and night breaks for another type of Petri net.

Distributions for all the breaks and activities are based on their duration in minutes. Distributions are estimated with the help of the MATLAB distribution fitter app. Probabilities for an activity to go to breaks are needed to be calculated. As mentioned in Section 3 if a break is less than or equal to 60 minutes it is considered a short break and anything longer is a long break. Depending on this probability for an activity to go to short or long break is obtained.

Once all the distributions and state transition probabilities are determined, probabilities for output symbols are calculated. This is the final input required to run the model, and this is achieved with Equation (1).

$$P(S_i|A_i) = \frac{(\Delta t|\forall S = S_i \cap A = A_i)}{(\Delta t|\forall A = A_i)} \tag{1}$$

- S_i the sensor
- A_i the activity
- $P(S_i|A_i)$ is the probability of sensor given activity

For evaluation, an unlabeled trace, with an existing ground truth, is required this contains the information regarding the time and the active sensor. This trace is created for a day, from data which is not included for distribution and probability calculation.

34 Activities	**11 Activities**	**6 Activities**
Watch_TV	Watch_TV	Personal_Activity
Sleep_Out_Of_Bed	Relax	
Relax		
Phone	Phone	
Read	Read	
Bathe	Freshen_Up	Freshen_Up
Personal_Hygiene		
Toilet	Toilet	
Dress	Dress	Dress
Groom		
Cook_Breakfast	Cook	Meal_Prep
Cook_Dinner		
Cook_Lunch		
Cook		
Wash_Dinner_Dishes	Wash_Dishes	
Wash_Lunch_Dishes		
Wash_Breakfast_Dishes		
Wash_Dishes		
Eat	Eat	Eat
Eat_Breakfast		
Eat_Dinner		
Eat_Lunch		
Drink		
Sleep	Sleep	Sleep
Bed_Toilet_Transition		
Go_To_Sleep		
Wake_Up		
Morning_Meds	–	
Evening_Meds	–	
Leave_Home	–	
Enter_Home	–	
Entertain_Guests	–	
Step_Out	–	
Work_At_Table	–	

Table 3: Activity grouping, separated by dashed lines for first grouping or solid lines for second grouping

3 Conceptual Model and Algorithm Design

3.1 Conceptual Model

There are different ways to model the daily activities of a resident. In this research, each activity has a dedicated Augmented Stochastic Petri net (ASPN) because these are simple and demand less computational power [7]. So the activities are grouped into sets, one set has 2-breaks (Figure 1) design, with a long and a short break,

and the other set has 3-breaks design (Figure 2) with short, medium and long breaks. Activities *Eat*, *Cook* and *Wash_Dishes* have 3-breaks design, the remaining activities all have 2-breaks design. Since there is an extra place for 3-breaks it also has extra transitions. Both standard Petri nets have *Activity* place and *No_Activity* place. These individual Petri nets are independent of each other.

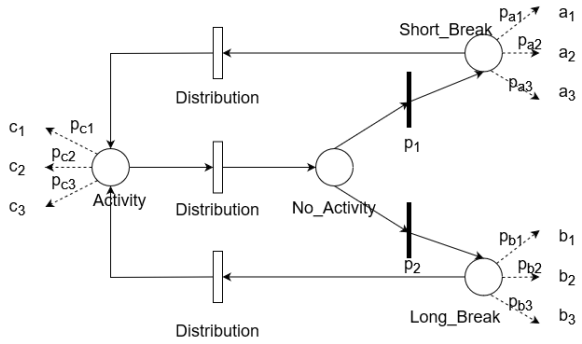


Figure 1: 2-Breaks Augmented Stochastic Petri Net

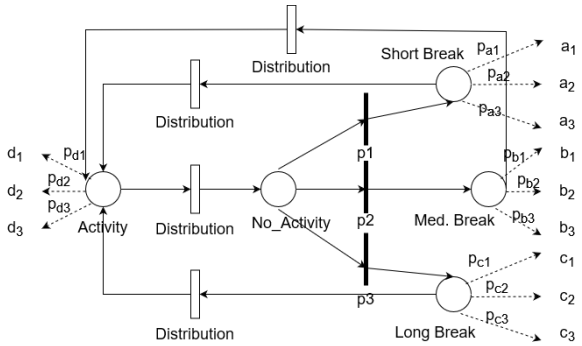


Figure 2: 3-Breaks Augmented Stochastic Petri Net

Once the structure of the Petri net is finalized, the output symbol emissions for the behavior reconstruction algorithm are added. These output symbols are linked with the Petri net places [8], as they occur, when an activity is being performed rather then when a state change occurs. From Figure 1 and 2 the output symbols are connected to all places except *No_Activity*. The p_k denote transition probabilities, p_{ak} denotes the output probability of symbol a_k. These output symbols play a crucial role in Virtual Stochastic Sensors (VSS), as they connect an observed symbol to the unobserved system state, and thus enable a behavior reconstruction.

3.2 Algorithm Design

To reconstruct the unobserved system behavior, here the residents activities, the Proxel algorithm is used. The Proxel algorithm determines possible development paths of the system and their probability [16, 17, 8]. A Proxel is a 5-tuple, which represents one point in the expanded system state space, this tuple consists of the state of a system, age intensity, current point in simulation time, route through the state space and probability. All individual models for activities are executed independently and determine output paths for all activities. This output path contains the probability of an activity occurring at a certain point in time.

Once the probabilities for each activity are computed the next thing is to classify the activities. For classification a simple decision system is incorporated. This system outputs the activity which has the highest probability for all individual models for their activity state at that point in time. For every timestep of the protocol, this decision system results an activity. This type of system works because all the activities have individual ASPNs independent of each other. If for a particular timestep the probability of all activities is zero then the model returns *Other_Activity*. Details of the solution method can be found in [7]. This procedure was employed for the two reduced activity setups and the results will be shown and discussed in the following section.

4 Experiments and Results

In this section, the performance of the models is evaluated. The metrics precision, recall and F1 score are calculated for individual activities, and then they are averaged in two different ways one by averaging for all the activities (*Average*) and the other by adding weights depending on the activity occurrences (*Weighted Average*). The average recall is also often used as an overall accuracy measure. These results are correctly classified if at a given time the reconstructed activity corresponds to the trace's ground truth. The above-discussed metrics are evaluated for all days of the week (03rd September to 9th September), to make sure that the model is not biased for a few specific days of the week. A confusion matrix is also created to get a comprehensive breakdown of the model's behavior, allowing for a detailed assessment of its performance.

All the metrics discussed above are applied for two different sets of activities, one set has 11 and the other

set has 6. The experiments are preformed in a k-fold cross-validation fashion, using n-1 days for training and the one left out for testing. For both sets the range of results for F1 and Accuracy are given, and an example result is examined in detail.

4.1 Evaluation metrics with 11 Activities

The model is applied for the larger set of activities for a week using k-fold cross validation. One of the results, which was obtained on Tuesday (4th of September) of that week was chosen to be investigated in detail here.

Activity	Precision	Recall	F1 Score
Watch_TV	0.51	0.16	0.24
Dress		0.00	
Freshen_Up	0.45	1.00	0.62
Sleep	0.96	0.96	0.96
Read	0.00		
Relax		0.00	
Toilet		0.00	
Phone	0.00		
Cook	0.60	0.98	0.75
Eat		0.00	
Wash_Dishes		0.00	
Average	0.42	0.34	0.64
Weighted_Average	0.61	0.49	0.50

Table 4: Individual activity and average model performance metrics for Tuesday 4th of September, larger activity set

Table 4 contains the precision, recall and F1 measures for the same day, some values are not filled, as they cannot be calculated if there are no classifications in that specific class. Only few of the 11 generalized activities are classified well, such as *Sleep* or *Cook*, many of the less frequent activities are grossly mis-classified, such as *Relax* or *Dress*. One can see, that for precision and recall, the weighted average is much better than the average, since the weighted average is impacted by the activities with longer durations, which were classified more accurately, such as *Sleep*.

On the other hand, the model is having trouble classifying some shorter activities such as *Relax* and *Eat*. The reason is that the model is unable to distinguish between some activities because the activities take place during similar times and activate similar sensors.

Table 5 shows the minimum and maximum results from this experiment for average and weighted average recall (also accuracy) as well as F1 measure. One

Evaluation Metric	Min	Max
Average_Recall	0.27	0.44
Weighted_Average_Recall	0.47	0.62
Average_F1	0.42	0.67
Weighted_Average_F1	0.46	0.57

Table 5: Range for average model performance metrics for Tuesday tth of September, larger activity set

can see, that there is quite a large range, which shows that for further usage, more training data has to be used to make the model performance more reliable and predictable. Furthermore, the average and weighted average recall differ quite considerably, this points to the same problem shown in the single day evaluation, that activities with longer durations are classified well, but shorter activities with overall less time are not reconstructed well.

Another experiment was conducted for a set of 6 activities. The grouping procedure was explained in 2.1.

4.2 Evaluation metrics with 6 Activities

Analogous to the experiment with 11 activities, the model is applied for the smaller set of 6 activities for a week using k-fold cross validation. One of the results, which was obtained on Monday (3rd of September) of that week was chosen to be investigated in detail here.

Activity	Precision	Recall	F1 Score
Personal_Activity	0.95	0.81	0.88
Dress		0.00	
Freshen_Up	0.96	1.00	0.98
Sleep	0.89	0.90	0.90
Eat		0.00	
Meal_Prep	0.26	1.00	0.41
Average	0.77	0.62	0.79
Weighted_Average	0.90	0.85	0.86

Table 6: Individual activity and average model performance metrics for Monday 3rd of September, smaller activity set

The results for the smaller group of activities are better.

Table 6 contains the precision, recall and F1 measures for the same day, some values are not filled, as they cannot be calculated. Four of the six generalized activities are classified well, only two activities, *Dress* and *Eat*, are not classified at all, reducing their recall to

0. As in the larger activity set, for precision and recall, the weighted average is much better than the average, since the mis-classified activities are rather short and impact the weighted average less. This second experiment shows that the further generalization, combining activities with similar sensor footprint, has improved the reconstruction performance.

Evaluation Metric	Min	Max
Average_Recall	0.55	0.63
Weighted_Average_Recall	0.66	0.90
Average_F1	0.56	0.89
Weighted_Average_F1	0.68	0.89

Table 7: Range for average model performance metrics for Monday 3rd of September, smaller activity set

Table 5 shows the minimum and maximum results from this experiment for average and weighted average recall (also accuracy) as well as F1 measure. As in the previous experiment, there is quite a large range. Therefore, more training data has to be used to make the model performance more reliable and predictable.

Furthermore, the average and weighted average differ quite considerably, both for recall as well as F1 measure, for the same reason as in the larger data set. However, the overall performance for the whole week is better than for the larger set.

4.3 Experiment Discussion

The two experiments show that generalizing activities has an impact on the overall reconstruction performance of Virtual Stochastic Sensors. Leaving aside the shortcomings of the experiments conducted one can conclude the following: A semantic grouping of activities by similar time of day and living area, which corresponds to a similar sensor footprint, leads to an improvement in accuracy. However, the information content of the reconstruction is considerably less, when more activities are combined under the same label. It has to be investigated with the help of domain experts at which point the generalized set of activities with better performance measures still holds enough information to assess the residents status. Ultimately the goal is to use this ambient sensor observation to decide, whether the residents behavior is still within normal bounds, or whether outside assistance or intervention is necessary.

5 Conclusion

In conclusion, this research has a few issues which need to be overcome, but even with these issues, there are a few positive takes. This research used the a CASAS dataset for single-resident apartment, there were various data transformations for activities and sensors. Then probabilities and distributions were extracted to input into the model. As for the model, Virtual Stochastic Sensors (VSS) are used to reconstruct resident activities. Two experiments were carried out to evaluate the model behavior, one had 11 activities and the other had 6 activities. Experiments with a smaller set of activities resulted in better performance Whether the better reconstruction accuracy comes at the expense of less information on the residents behavior, has to be investigated.

5.1 Future Work

The analysis presented in this paper is only part of the research aiming at providing generalized models to be pre-trained and then applied for the reconstruction of previously unknown apartments, or systems in general.

Further steps in this direction include expanding the research to different model architectures, for example, creating Augmented Stochastic Petri nets (ASPNs) for room-level, or extend this concept to generalize multiple single-resident apartments. This implies applying data analysis and models to a broader context, like adapting algorithms for variations in apartment layouts, sensor placements and resident behaviors.

References

[1] Rashidi P, Mihailidis A. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*;17(3):579–590.

[2] Sun H, Florio VD, Gui N, Blondia C. Promises and Challenges of Ambient Assisted Living Systems. In: *2009 Sixth International Conference on Information Technology: New Generations*; pp. 1201–1207.

[3] Deutsch JE, Lewis JA, Burdea G. Technical and patient performance using a virtual reality-integrated telerehabilitation system: preliminary finding. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*;15(1):30–35.

[4] Fahad LG, Tahir SF, Rajarajan M. Feature selection and data balancing for activity recognition in smart homes.

In: *2015 IEEE International Conference on Communications (ICC)*. IEEE; pp. 512–517.

[5] Ordóñez FJ, Roggen D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*;16(1):115.

[6] Gochoo M, Tan TH, Liu SH, Jean FR, Alnajjar FS, Huang SC. Unobtrusive activity recognition of elderly people living alone using anonymous binary sensors and DCNN. *IEEE journal of biomedical and health informatics*;23(2):693–702.

[7] Mueller LF. Feasibility and Applicability of Virtual Stochastic Sensors for Human Activity Recognition in the Context of Ambient Assisted Living. Master's thesis, Otto-von-Guericke-University Magdeburg. 2021.

[8] Krull C. Virtual Stochastic Sensors: Formal Background and Example Applications Reconstructing the Behavior of Partially Observable Discrete and Hybrid Stochastic Systems. Habilitation thesis, Otto-von-Guericke-University Magdeburg. 2021.

[9] Lühr S, Venkatesh S, West G, Bui HH. Explicit state duration HMM for abnormality detection in sequences of human activity. In: *PRICAI 2004: Trends in Artificial Intelligence: 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, August 9-13, 2004. Proceedings 8*. Springer; pp. 983–984.

[10] van Kasteren T, Kroese B. Bayesian Activity Recognition in Residence for Elders. In: *3rd IET International Conference on Intelligent Environments*. 2007; .

[11] Singh D, Merdivan E, Psychoula I, Kropf J, Hanke S, Geist M, Holzinger A. Human activity recognition using recurrent neural networks. In: *Machine Learning and Knowledge Extraction: First IFIP TC 5, WG 8.4, 8.9, 12.9 International Cross-Domain Conference, CD-MAKE 2017, Reggio, Italy, August 29–September 1, 2017, Proceedings 1*. Springer; pp. 267–274.

[12] Chen L, Nugent CD, Wang H. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*; 24(6):961–974.

[13] Krull C, Horton G. Hidden Non-Markovian Models: Formalization and Solution Approaches. In: *Proceedings of 6th Vienna Conference on Mathematical Modelling, Vienna, Austria*. 2009; pp. 682–693.

[14] Krull C. A Hybrid User Model for Virtual Stochastic Sensors. *Simulation Notes Europe SNE33*;33(1):35–43.

[15] Cook DJ, Crandall AS, Thomas BL, Krishnan NC. CASAS: A smart home in a box. *IEEE Computer*. 2013;46(6):26–33.

[16] Horton G. A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times. In: *Proceedings of the European Simulation Symposium 2002*. SCS European Publishing House. 2002; pp. 129–136.

[17] Lazarova-Molnar S. The Proxel-Based Method: Formalisation, Analysis and Applications. Ph.D. thesis, Otto-von-Guericke-University Magdeburg. 2005.

AURONA – ein Autonomer rekonfigurierbarer Funktionsträger für nachhaltige Mobilität

Sven Jacobitz[1*], Marian Göllner[1], Taihao Li[1], Paul Ole Flender[1], Xiaobo Liu-Henke[1]

[1]Fachgruppe für Regelungstechnik und Fahrzeugmechatronik, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel; [*]*sve.jacobitz@ostfalia.de*

Kurzfassung. Der vorliegende Beitrag stellt den Autonomen rekonfigurierbaren Funktionsträger für nachhaltige Mobilität AURONA vor. Die mit dem Funktionsträger forcierte Forschung konzentriert sich auf die Entwicklung autonomer Fahrzeugtechnologien und die Integration fortschrittlicher Sensordatenerfassung und -verarbeitung, unterstützt durch V2X-Kommunikation. AURONA ist modular aufgebaut und basiert auf einer rekonfigurierbaren und flexiblen Architektur, die es ermöglicht, verschiedene Komponenten und Systeme effizient zu integrieren und zu testen. Der Beitrag beschreibt die Konzeption, die Systemstrukturierung, die Sensorik und Kommunikationsansätze sowie erste Schritte zur Realisierung und Verifikation des Funktionsträgers. Durch seine hochmoderne Ausstattung und Fähigkeit zur Echtzeitdatenverarbeitung dient AURONA als wegweisende Plattform für die Mobilität der Zukunft.

Einleitung

Autonome Fahrzeuge sind ein wesentlicher Bestandteil zukünftiger Mobilitätsszenarien. Sie sorgen für eine Steigerung der Verkehrssicherheit und bieten zudem weitreichende Möglichkeiten in Bezug auf Effizienz und Umweltschutz. Durch Vernetzung und Vehicle-to-everything (V2X) -Kommunikation können die zukünftigen intelligenten Fahrzeuge Informationen austauschen und kooperativ interagieren. Dies führt zu reduziertem Energieverbrauch und stellt die Grundlage für eine bedarfsgerechte nachhaltige Mobilität dar. Hierbei entstehen cyber-physische Verkehrssysteme mit hoher Komplexität [1].

Die Entwicklung solcher Fahrzeuge stellt eine enorme Herausforderung dar und ist mit konventionellen Methoden nicht mehr realisierbar. Daher werden vermehrt szenarienbasierte Entwicklungs- und Testmethoden eingesetzt [2]. Insbesondere die Echtzeitsimulation der entstehenden komplexen Systeme ist ein wesentlicher Bestandteil der Entwicklung. Hierfür müssen sämtliche relevanten Verkehrsteilnehmer sowie die Sensordatenerfassung und Kommunikation in der Simulation abgebildet werden.

Zur Forschung an den entstehenden neuartigen intelligenten Fahrzeugen und Systemen wird an der Ostfalia sukzessive im Rahmen verschiedener Forschungsprojekte der Autonome rekonfigurierbare Funktionsträger für nachhaltige Mobilität (AURONA) entwickelt und aufgebaut. Dieser Beitrag fasst das Konzept sowie erste Ansätze der Realisierung zusammen.

Der weitere Beitrag ist wie folgt aufgebaut. Zunächst wird im Abschnitt 1 der strukturierte, modellbasierte, verifikationsorientierte Rapid Control Prototyping (RCP)-Entwicklungsprozess für vernetzte cyber-physische Systeme detailliert beschrieben. Anschließend werden in Abschnitt 2 relevante Forschungen zu Funktionsträgern und Sensorik für autonomes Fahren beleuchtet. Der dritte Abschnitt umfasst die Anforderungsdefinition, Systemstrukturierung und Aufbau der Systemarchitektur von AURONA. In Abschnitt 4 wird schließlich die Realisierung der Konzepte vorgestellt, einschließlich Details zu Antrieb, Tragstruktur und Echtzeitdatenverarbeitung. Eine grundlegende Funktionsvalidierung wird in Abschnitt 5 behandelt. Der Beitrag schließt in Abschnitt 6 mit einer Zusammenfassung und einem Ausblick auf zukünftige Forschungen und Entwicklungen.

1 Methodik

Zur Beherrschung der Systemkomplexität wird der strukturierte, modellbasierte, verifikationsorientierte Rapid Control Prototyping (RCP) Entwicklungsprozess eingesetzt. Hierbei erfolgt zur Reduktion der Komplexität zunächst die Strukturierung des Gesamtsystems unter Anwendung von Modularisierung und Hierarchisierung in intelligente gekapselte Teilsysteme mit definierten Schnittstellen. In einem Top-Down-Verfahren werden

die Funktionalitäten systematisch zerlegt und in den vier Ebenen Mechatronisches Funktionsmodul (MFM), Mechatronische Funktionsgruppe (MFG), Autonomes Mechatronisches System (AMS) und Vernetztes Mechatronisches System (VMS) hierarchisch angeordnet [3].

Der anschließende ganzheitliche, durchgängig modellbasierte Entwicklungs- und Absicherungsprozess beinhaltet Model-in-the-Loop- (MiL-), Software-in-the-Loop- (SiL-), und Hardware-in-the-Loop- (HiL-) Simulationen (vgl. Abbildung 1). Die HiL-Simulation ist dabei zu einem wesentlichen Bestandteil des Freigabeprozesses für die Software von Steuergeräten geworden [4].

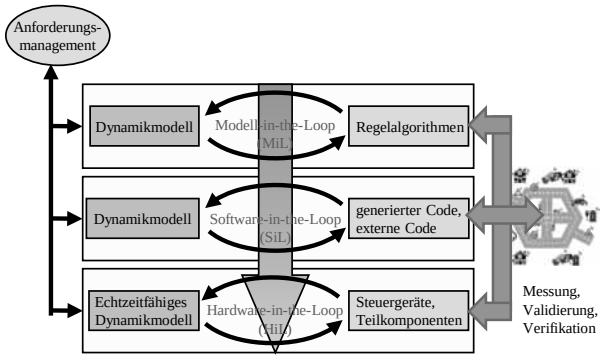


Abbildung 1: Ganzheitlicher modellbasierter Funktionsentwicklungs- und Testprozess [4].

In den Ebenen MiL und SiL werden die Regelfunktionen modellbasiert entworfen und mit definierten Schnittstellen in die Modelle der zu regelnden Strecke integriert. Entsprechend den vorgegebenen Spezifikationen und gewünschten Funktionen werden bereits in frühen Entwicklungsphasen Tests durchgeführt. Die Simulationsergebnisse aus MiL / SiL werden mittels HiL-Simulation in einer Echtzeitumgebung weiter validiert und optimiert. Dabei wird immer wieder auf die vorhergehende Ebene zurückgegriffen, um Fehlerquellen zu beseitigen und gewünschte Funktionen zu realisieren. Während aller Prozessschritte finden Identifikations- und Validierungsmessungen am realen System statt [5]. Hierzu werden Prüfstände und Funktionsträger eingesetzt.

2 Stand des Wissens

Die Entwicklung autonomer Fahrzeugtechnologien hat in den letzten Jahren erhebliche Fortschritte gemacht, insbesondere in städtischen und komplexen Verkehrsumgebungen, wo die Herausforderungen besonders groß sind. Trotz der Fortschritte in der Sensortechnologie, der Datenverarbeitung und der künstlichen Intelligenz bleibt die sichere Navigation autonomer Fahrzeuge in solchen Umgebungen eine ungelöste Problemstellung, die fortlaufende Forschung und Innovation erfordert. Im Folgenden werden relevante Forschungen zu Funktionsträgern sowie zur Sensorik autonomer Fahrzeuge eingeführt.

2.1 Funktionsträger für autonomes Fahren

Im Rahmen der Forschung an autonomen Fahrfunktionen sind verschiedene Versuchsträger und Prototypen entwickelt worden, um innovative Ansätze aufzuzeigen. Im Folgenden wird kurz auf ausgewählte relevante Fahrzeuge eingegangen.

Das Research Vehicle for Automated and Intelligent driving in ON-street applications (RAION, ehemals PLUTO) der TU Braunschweig soll als Demonstrator für ein autonomes Shuttle fungieren. Durch mehrere Radar-, LiDAR- und Kamerasensoren wird, wie beim Vorgängerfahrzeug „TEASY 3“, eine 360° Rundumsicht ermöglicht [6]. Als Zentralrechner wird eine Scalexio AutoBox von dSPACE eingesetzt [7].

Das Verbundprojekt UNICARagil forciert die Erforschung neuartiger modularer Strukturen für automatisierte Fahrzeugkonzepte [8]. Die Ergebnisse sehen eine dezentrale intelligente Struktur vor, welche dem menschlichen Gehirn nachempfunden ist. Aufbauend auf den Projektergebnissen ist unter anderem der Versuchsträger Edgar der TU München (vgl. [9]) entstanden.

Das OPA^{3}L-Projekt an der Universität Bremen setzt ein Hybridfahrzeug zur Erforschung urbaner autonomer Fahranwendungen ein [10]. Dieses Forschungsfahrzeug ist mit einer Vielzahl von Sensoren ausgestattet. Dazu gehören sechs Ibeo ScaLa LiDAR-Sensoren, eine Kamera im Windschutzscheibenbereich, vier ValeoVis Flächenkameras und ein Global Navigation Satellite System (GNSS). Zusätzlich liefert eine ADIS 16488 Inertiale Messeinheit (IMU) präzise Beschleunigungs- und Gyroskopdaten. Die Berechnungen werden von einem Bordcomputer mit einem Intel Core i9-9900K CPU und zwei NVIDIA GeForce RTX 2080 Ti GPUs durchgeführt.

2.2 Sensorik für autonome Fahrzeuge

Im Bereich der Sensorik und Sensordatenverarbeitung für autonome Fahrzeuge haben sich in den letzten Jahren bedeutende Fortschritte vollzogen. Durch den Einsatz moderner Sensordatenfusionen und künstlicher Intelligenz lassen sich Messergebnisse der eingesetzten LiDAR-, Radar-, Kamera- und Ultraschallsensoren auswerten und zu einem digitalen Zwilling der realen

Umgebung integrieren. Der Stand des Wissens in diesem Bereich spiegelt sich in kontinuierlichen Innovationen und Verbesserungen, die nicht nur die Leistungsfähigkeit und Zuverlässigkeit der Sensoren erhöhen, sondern auch neue Anwendungsfelder in der Mobilität und darüber hinaus erschließen wieder.

Wie bereits aus [11] hervorgeht, lassen sich nicht alle für die autonome Fahrt notwendigen Informationen aus einem Sensor ermitteln. Daher ist stets eine Sensordatenfusion notwendig. Abbildung 2 verdeutlicht den Hintergrund, durch Analyse verschiedener Sensortypen. Hierbei wird die Eignung zur Messung von Entfernung und Geschwindigkeiten sowie die erzielte Auflösung und Framerate beurteilt. Weitere Randbedingungen wie Wetterabhängigkeit und Robustheit geben einen Überblick über mögliche Einsatzgebiete. Nach [12] werden für autonome Fahrzeuge der Stufen 4 und 5 im Straßenverkehr unter anderem 8 Radar-, ein Lidar-, 5 Kamerasensoren benötigt.

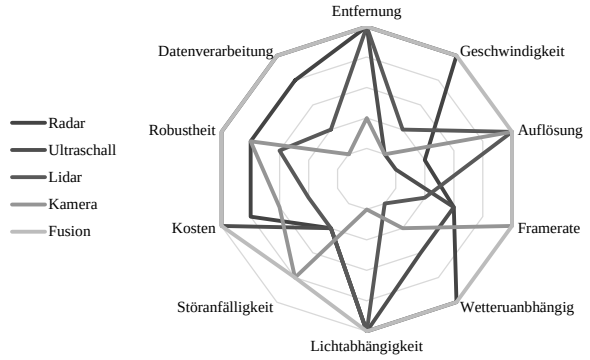


Abbildung 2: Analyse verschiedener Sensoriken für autonome Fahrzeuge in Anlehnung an [13].

Weiterhin kommt GNSS zur Positionsbestimmung zum Einsatz. Dessen Genauigkeit lässt sich durch Einsatz eines Real Time Kinematic (RTK) Systems auf unter einem Meter steigern [14]. Ein weiterer wesentlicher Faktor zur Informationsbeschaffung ist die V2X-Kommunikation, welche insbesondere durch Einsatz von 5G-Technologie einen Datenaustausch über weite Entfernungen ermöglicht [15].

3 Konzeption

Im folgenden Abschnitt erfolgt die Konzeption des Funktionsträgers AURONA, indem zunächst Anforderungen definiert und anschließend eine Systemstruktur abgeleitet wird. Zuletzt wird die Systemarchitektur aufgebaut.

3.1 Anforderungen an den Funktionsträger

AURONA soll als Funktionsträger für autonome Fahrfunktionen und nachhaltige Mobilität agieren. Für die Forschung in diesem agilen Themenfeld ist die Rekonfigurierbarkeit und Flexibilität von besonderer Bedeutung.

Im Kern werden zur Erfüllung der übergeordneten Ziele folgende Anforderungen an AURONA gestellt:

1. Modulare Architektur mit offenen energetischen und informationstechnischen Schnittstellen sowie eingebauten Sicherheitsvorrichtungen.
2. Leistungsfähiges RCP-System mit KI-Eignung zur prototypischen Realisierung und Untersuchung innovativer Funktionen.
3. Sensorik zur 360° Umfelderfassung im Nah- und Fernbereich sowie 5G-basierte V2X-Kommunikation.
4. X-by-Wire-Aktorik für Antrieb, Bremse und Lenkung.
5. Elektrischer Antriebs mit radindividuellem Traktionsmotor.

3.2 Beitrag zur nachhaltigen Mobilität

Autonomes kooperatives Fahren ermöglicht eine optimale Nutzung vorhandener und zukünftiger Verkehrsressourcen. Fahrzeuge können in Echtzeit miteinander kommunizieren, um den Verkehrsfluss zu verbessern und Staus zu vermeiden. Dies reduziert nicht nur die Fahrzeiten, sondern auch den Energieverbrauch erheblich. Ein weiteres Beispiel für nachhaltige Mobilität sind bedarfsgerechte autonome Shuttles, die in Kolonne fahren können. Im Gegensatz zu herkömmlichen, getakteten Verbindungen, die insbesondere zu Randzeiten wenig ausgelastet und ineffizient sind, passen sich diese Shuttles flexibel an den tatsächlichen Bedarf an. Dies führt zu einer effizienteren Nutzung der Fahrzeuge und einer signifikanten Reduktion des Energieverbrauchs.

Durch den modularen Aufbau und die flexible Architektur von AURONA können verschiedene Funktionen für nachhaltige Mobilität erforscht werden. Dies umfasst unter anderem die Integration und Testung von fortschrittlichen Sensordatenerfassungs- und -verarbeitungssystemen sowie die Entwicklung neuer kooperativer Fahrfunktionen. AURONA dient als Plattform zur Erforschung und Validierung dieser Technologien, die das Potenzial haben, die Umweltbelastung durch den Verkehr erheblich zu reduzieren. Somit wird durch AURONA ein direkter Beitrag zur Entwicklung zukunftsorientierter nachhaltiger Mobilität geleistet.

3.3 Strukturierung

Wie in Abschnitt 1 eingeführt ist die Strukturierung des Systems eine wesentliche Voraussetzung zur Reduktion der Komplexität und ermöglicht eine Rekonfigurierbarkeit. Abbildung 3 illustriert daher die zur Erfüllung der im Vorfeld beschrieben Anforderungen forcierte mechatronische Strukturierung des Funktionsträgers AURONA. Auf der untersten und vitalsten Ebene MFM sind grundlegende Funktionen für Antrieb, Bremse, Lenkung, Batteriemanagement, etc. vorhanden. Durch informationstechnische Kopplung werden auf der Ebene MFG höherwertige Funktionalitäten wie die integrierte Fahrdynamikregelung oder das Energiemanagement erzielt. Die übergeordnete Ebene AMS integriert diese MFG zum intelligenten autonomen Fahrzeug AURONA. Durch Vernetzung mit anderen autonomen Fahrzeugen entsteht schließlich die oberste Ebene VMS.

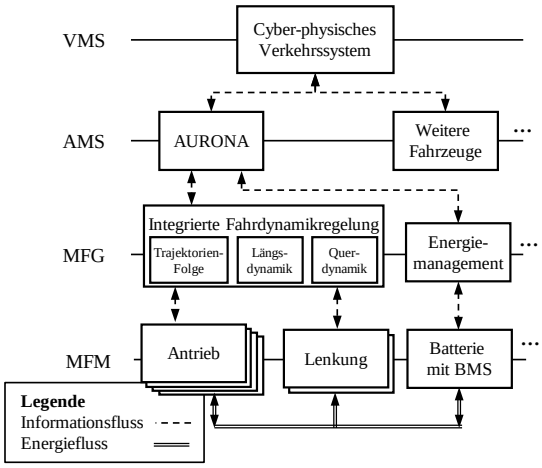


Abbildung 3: Mechatronische Strukturierung des Funktionsträgers AURONA.

3.4 Systemarchitektur

Die Systemarchitektur des Funktionsträgers AURONA ist speziell für die Integration und das Testen autonomer Fahrfunktionen konzipiert. Der zentrale Aspekt dieser Architektur ist ihre Modularität und Skalierbarkeit, welche es ermöglicht, verschiedene Sensoren und Aktoren effizient zu integrieren. AURONA verwendet eine fortschrittliche Elektrik/Elektronik-Architektur (E/E-Architektur). Abbildung 4 fasst die Systemarchitektur des Versuchsträgers zusammen. Diese erlaubt es, eine Vielzahl von Sensoren für die Umgebungserfassung wie LiDAR, Radar, Kamera und Ultraschallsensoren zu unterstützen. Zusätzlich verfügt AURONA über ein echtzeitfähiges Bussystem, welches die Kommunikation zwischen den einzelnen Systemkomponenten ermöglicht.

Die Architektur von AURONA setzt ebenfalls auf eine serviceorientierte, zentralisierte Verarbeitungsstruktur mit definierten Verantwortlichkeiten, die eine effiziente Datenverarbeitung und eine hohe Rechenkapazität bereitstellt. Dies wird durch den Einsatz leistungsfähiger Recheneinheiten unterstützt, die in der Lage sind, komplexe Algorithmen für das autonome Fahren zu verarbeiten. Der Einsatz eines RCP-Systems ermöglicht es, Simulationen und reale Tests nahtlos zu integrieren, wodurch eine kontinuierliche Verbesserung und Validierung der autonomen Systeme gewährleistet wird. Diese robuste und erweiterbare Systemarchitektur ermöglicht es AURONA, als hochflexibler Versuchsträger für Forschung und Entwicklung in der Mobilität der Zukunft zu dienen.

4 Realisierung

Der Funktionsträger AURONA wurde als hochmoderne Plattform zur Erforschung und Entwicklung autonomer Fahrfunktionen konzipiert. Nach der detaillierten Planung und konzeptuellen Entwicklung, die in den vorherigen Abschnitten beschrieben wurden, fokussiert dieser Abschnitt die technische Umsetzung der Systemarchitektur, einschließlich der Integration von Antriebskomponenten, Sensorik und Kommunikationssystemen.

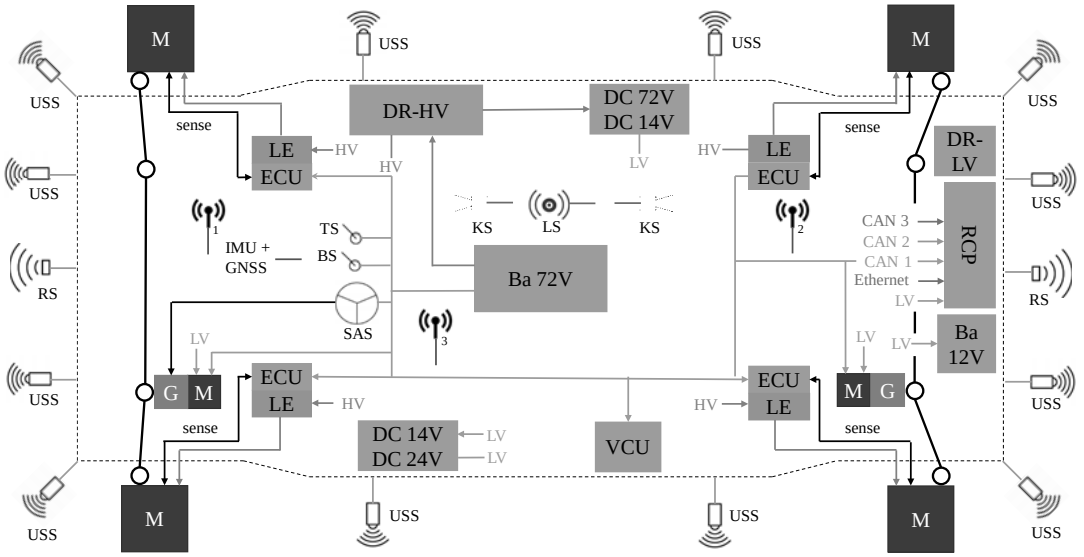
4.1 Antrieb und Tragstruktur

AURONA verfügt über vier Radnabenantriebe mit einer Gesamtleistung von 15,2 kW sowie separat lenkbare Front- und Hinterachsen. Antrieb, Bremse und Lenkung lassen sich by-Wire und somit durch die Informationsverarbeitung ansteuern.

Abbildung 5 illustriert den realisierten Funktionsträger. Dieser bietet sowohl die Möglichkeit autonom zu fahren als auch manuell über eine Fernsteuerung oder ein klassisches Lenkrad / eine Pedalerie bedient zu werden.

4.2 Echtzeitdatenverarbeitung

Auf der Verarbeitungsseite verfügt AURONA über einen leistungsstarken RCP fähigen Rechner vom Typ dSPACE Autera, der mit einem Intel Core i9-Prozessor und einer NVIDIA GeForce RTX A2000 Grafikkarte ausgestattet ist. Diese Hardwarebasis ermöglicht es,



Legende

- USS – Ultraschallsensor
- IMU – Beschleunigungssensor
- GNSS – globales Navigationssatellitensystem
- LS – Lidarsensor
- RS – Radarsensor
- KS – Kamerasensor
- TS – Gaspedalsensor
- BS – Bremsensensor
- SAS – Lenkwinkelsensor
- Ba – Batterie
- HV – Hochvoltnetz
- LV – Niedervoltnetz
- LE – Leistungselektronik
- ECU – Motorsteuergerät
- VCU – Fahrzeugsteuergerät
- Autera – RCP-System
- DS – Verteilerschiene
- RCP – RCP-System
- M – Motor
- G – Getriebe
- Antenne 1 - Vector Antenne
- Antenne 2 – Position Antenne
- Antenne 3 – 4G

Abbildung 4: Systemarchitektur von AURONA.

komplexe Algorithmen zur Datenverarbeitung und Entscheidungsfindung und KI-basierte Funktionen in Echtzeit auszuführen. Weiterhin ist das Fahrzeug mit einem 5G-basierten V2X-Kommunikationssystem ausgestattet.

Abbildung 5: Realisierter Funktionsträger.

4.3 Sensorik

Der Funktionsträger AURONA wurde speziell für die Erforschung und Entwicklung intelligenter autonomer Fahrsysteme entworfen. Zu den zentralen Komponenten des Fahrzeugs gehören ein zentraler LiDAR-Sensor des Typen Robosense Helios-16P, welcher eine 360-Grad-Erfassung der Umgebung ermöglichen, sowie zwei Radarsensoren vom Typ Continental ARS 408-21 für die frontale und rückwärtige Fernerkennung. Diese werden ergänzt durch zwei hochauflösende Kamerasysteme Leopard Imaging IMX490-GW5400-GM, die sowohl an der Front als auch am Heck des Fahrzeugs angebracht sind, um visuelle Daten zu erfassen und Verkehrszeichen sowie Hindernisse zu identifizieren und zu klassifizieren. Zudem ist ein integriertes Sensorsystem CGI-610 mit hochauflösender IMU zur Messung der Beschleunigung und Gierraten in 6 Freiheitsgraden sowie einem GNSS-System mit RTK-Funktionalität zur hochgenauen Positionsbestimmung vorhanden.

5 Verifikation

In diesem Abschnitt wird die Verifikation der grundlegenden Funktionen des Versuchsträgers diskutiert. Exemplarisch wird das Vorgehen anhand ausgewählter Sensoriken aufgezeigt. Die Verifikation der Sensorfunktion ist ein erster entscheidender Schritt in der Entwicklung und Implementierung von autonomen Systemen, insbesondere Fahrzeugen. Dieser Prozess stellt sicher, dass die Sensoren unter verschiedenen realen Bedingungen präzise und zuverlässig funktionieren. Durch systematische Tests und Analysen werden die Leistungsfähigkeit und die Genauigkeit der Sensoren in verschiedenen Umgebungen und Szenarien überprüft.

Im ersten diskutierten Versuch wird die Längsbeschleunigung des Fahrzeugs betrachtet. Zum einen erfolgt die direkte Messung über die integrierte IMU zum anderen die Berechnung auf Basis der Raddrehzahlen. Abbildung 6 stellt die zeitlichen Verläufe beider Signale während einer Testfahrt gegenüber. Beide Verläufe stimmen mit hoher Genauigkeit überein.

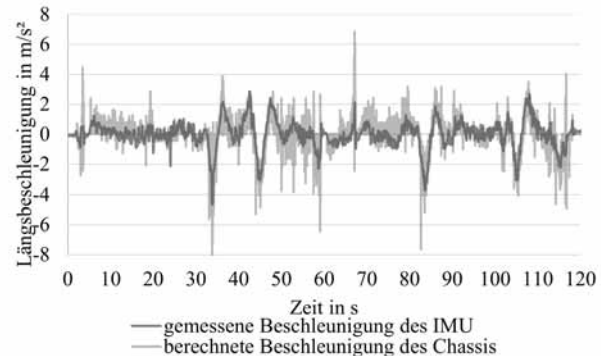


Abbildung 6: Zeitliche Verläufe der Längsbeschleunigung während einer Messfahrt.

Weitere Versuche wurden zur Analyse der Querdynamik durchgeführt. Abbildung 7 illustriert die zeitlichen Messverläufe von Querbeschleunigung, Lenkwinkel und Längsgeschwindigkeit über der Zeit. Die zeitlichen Verläufe sind konsistent und weisen eine Korrelation zu den angenommenen physikalischen Prinzipien auf. So ist beispielsweise der geschwindigkeitsabhängige Zusammenhang zwischen Lenkwinkel und Querbeschleunigung deutlich erkennbar.

Die gemessenen Daten passen sehr gut zu den theoretischen Annahmen. Weitere Versuche zur Validierung der Funktionen zeigten ähnliche vielversprechende Ergebnisse.

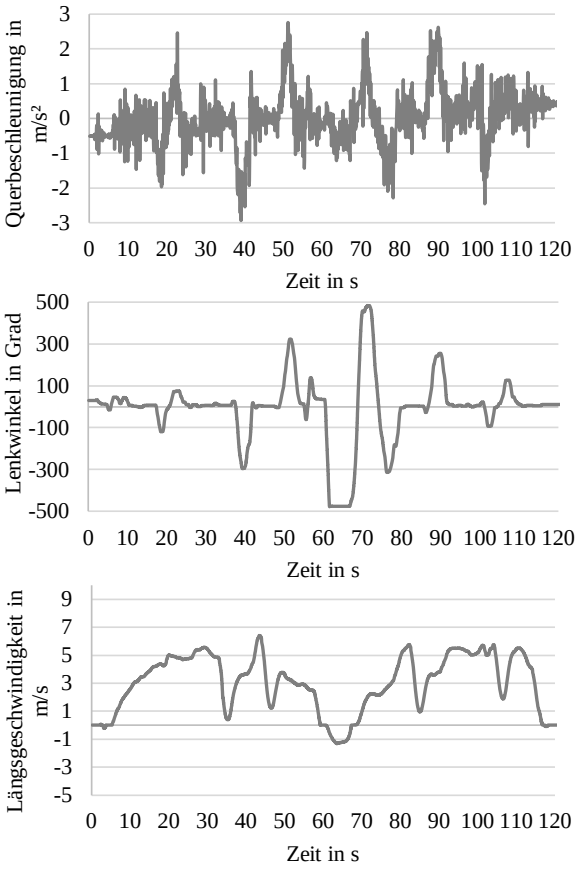


Abbildung 7: Zeitliche Messdatenverläufe der Querdynamik während einer Probefahrt.

6 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wurde der autonome rekonfigurierbare Funktionsträger für nachhaltige Mobilität, kurz AURONA, vorgestellt. Durch den Einsatz modernster Technologien und die Anwendung modellbasierter Methoden konnte ein rekonfigurierbares Forschungsobjekt für die Echtzeituntersuchung geschaffen werden. Die Kombination aus fortschrittlicher Sensorik und V2X-Kommunikation ermöglicht eine umfassende Umgebungserfassung und Interaktion mit anderen autonomen Fahrzeugen in einem vernetzten cyber-physischen Verkehrssystem.

Die Ergebnisse der ersten Echtzeitsimulationen und Tests belegen, dass AURONA die gestellten Anforderungen an Flexibilität und Rekonfigurierbarkeit erfüllt und sich als hochmoderner Versuchsträger für verschiedene Mobilitätsszenarien eignet. Dies unterstreicht das Potenzial von AURONA, zentrale Herausforderungen in der

autonomen Fahrzeugtechnologie anzugehen und zur Sicherheit und Effizienz im Verkehr beizutragen.

Zukünftige Arbeiten werden sich darauf konzentrieren, die Interoperabilität von AURONA in größeren, vernetzten cyber-physischen Verkehrssystemen zu testen und zu optimieren. Dies beinhaltet auch die Weiterentwicklung von Algorithmen der modernen Regelungstechnik und der künstlichen Intelligenz, um autonome Fahrfunktionen unter variablen Umgebungsbedingungen zu verbessern. Langfristig soll AURONA als Basis für die Entwicklung weiterer autonomer Systeme dienen, die spezifische Anforderungen unterschiedlicher Mobilitäts- und Transportaufgaben erfüllen können und somit einen wesentlichen Beitrag zur nachhaltigen Mobilität leisten.

Danksagung

Gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur unter Fördernummer ZN3495 im Niedersächsischen Vorab der VolkswagenStiftung und betreut vom Zentrum für digitale Innovationen Niedersachsen (ZDIN).

Literatur

[1] Göllner M, Jacobitz S, Li T, Liu-Henke X. Modular Platform for Route Guidance in the Cyber-Physical Laboratory Test. *SNE*. 2023;33(1):45–52.

[2] Liu-Henke X, Jacobitz S, Göllner M, Zhang J. Concept of a holistic HiL test system for autonomous vehicles in cyber-physical Traffic Systems. In: *The 3rd International Conference on Computers and Automation (CompAuto 2023)*., 2023.

[3] Liu-Henke X. *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab*. Dissertation, Universität Padaborn, 2005. Fortschritt-Berichte VDI Reihe 12, Verkehrstechnik/Fahrzeugtechnik. Vol. 589. Düsseldorf: VDI-Verl., 2005.

[4] Liu-Henke X, Jacobitz S, Scherler S, Göllner M, Yarom O, Zhang J. A Holistic Methodology for Model-based Design of Mechatronic Systems in Digitized and Connected System Environments. In: *Proceedings of the 16th International Conference on Software Technologies*. SCITEPRESS - Science and Technology Publications, 2021:215–223.

[5] Scherler S. *Zeit- und energieoptimierter Fahrbetrieb eines Elektrofahrzeugs mit Brennstoffzellen-Range-Extender im digital vernetzten Verkehr*. Dissertation. Technische Universität Braunschweig, Shaker Verlag, 2022.

[6] Kascha M, Henze R. Modular Decision Making Framework for Level 4 Applications in Automated Driving. In: *2023 29th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 2023:1–6.

[7] Everding L, Aslam I, Raulf C, Aviv Yarom O, Fritz J, Jacobitz S, Hegerhorst T, Pethe C, Şahin T, Iatropoulos J, Vietor T, Rausch A, Liu-Henke X, Henze R. Dynamically Configurable Autonomous Vehicles for Urban Cargo Transportation. In: Proff H, editor. *Towards the New Normal in Mobility*. Wiesbaden: Springer Fachmedien Wiesbaden, 2023:851–869.

[8] Woopen T, Eckstein L, Kowalewski S, Moormann D, Maurer M, Ernst R, Winner H, Katzenbeisser S, Becker M, Stiller C, Furmans K, Bengler K, Lienkamp M, Reuss H-C, Dietmayer K, Lategahn H, Siepenkötter N, Elbs M, v. Hinüber E, Dupuis M, Hecker C. UNICARagil - Disruptive modulare Architektur für agile, automatisierte Fahrzeugkonzepte. In: Eckstein L, Pischinger S, Hammermüller B, Wolsfeld R, editors. *27. Aachen Colloquium Automobile and Engine Technology*. Aachen: Institute for Automotive Engineering RWTH Aachen, 2018.

[9] Lehrstuhl für Fahrzeugtechnik, TU München. EDGAR - Excellent Driving Garching. 2024. https://www.mos.ed.tum.de/ftm/forschungsfelder/team-av-perception/edgar/. [Accessed 11 April 2024].

[10] Folkers A, Wellhausen C, Rick M, Li X, Evers L, Schwarting V, Clemens J, Dittmann P, Shubbak M, Bustert T, Zachmann G, Schill K, Buskens C. The OPA³L System and Testconcept for Urban Autonomous Driving. In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022:1949–1956.

[11] Göllner M, Liu-Henke X. Realisierung einer Datenfusionsstruktur für die Umfeldperzeption autonomer Fahrzeuge. In: *Tagungsband ASIM Workshop STS/GMMS 2019*. ARGESIM, 2019.

[12] Karil M. *Auto tankt Internet: Auswirkungen des automatisierten und vernetzten Fahrens auf den Energieverbrauch von Fahrzeugen, Datenübertragung und Infrastruktur*., 2020.

[13] Rice C. AV and ADAS Sensors. 2021. https://community.sw.siemens.com/s/article/AV-and-ADAS-Sensors. [Accessed 19 April 2024].

[14] Zhang Z, Li Y, He X, Hsu L. Resilient GNSS real-time kinematic precise positioning with inequality and equality constraints. *GPS Solut.* 2023;27(3).

[15] Scherler S, Jacobitz S, Liu-Henke X, Henke M. Cloud-based V2X communication for dynamic intelligent guidance in connected traffic systems. In: *2021 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, 2021:1–7.

Development of a Simulation Model for Predicting Energy Consumption of Battery-Electric Buses

Anja Konzept[1*], Arne Hitz[1], Benedikt Reick[1]

[1]Institute for E-mobility, Ravensburg-Weingarten University, Doggenriedstraße 70, 88250 Weingarten, Germany; [*]*anja.konzept@rwu.de*

Abstract. This paper presents the development of a simulation model for battery-electric public transportation buses aimed at accurately predicting energy consumption and state of charge with minimal input data. The model considers driving resistances, elevation profiles, temperatures, and load conditions to closely replicate real-world operational scenarios. Validation with data from a Swiss public transport company shows high accuracy in the prediction of energy consumption and state of charge. The model facilitates precise route and charging infrastructure planning, enhancing efficiency and optimizing costs for public transport operators.

Introduction

In order to reduce CO2 emissions at European level, as outlined in the Paris Agreement, the EU Climate and Energy Framework, and the White Paper on Transport, an increasing number of companies in the public transportation sector are transitioning from combustion engine buses to battery-electric buses. This transition represents an important shift in the transportation sector [1, 2]. A significant challenge in this context is the variability in range depending on the driven route (elevation profile, rural track, urban track), temperature, and load [3]. To facilitate an effective transition to electromobility in public transport, it is important to consider the energy supply of the buses in advance [2]. To enable precise planning of the charging strategy (depot or opportunity charging) and the associated charging stations as well as network utilization, even before the ordering of a new fleet, a simulation model of an electric bus is created as part of this publication. A challenge in this task is that the transport companies usually have limited technical data and information about the electric buses. Thus, the planning of fleets size and charging stations is often done by assuming average consumption values determined in Standardized On Road Test cycles (SORT) [4, 5]. These data are often very inaccurate and do not reflect worst case scenarios or the exact local conditions (e.g. low environmental temperatures or elevations). The aim of this work is to develop a bus model with few data that reflects as accurately as possible the consumption and state of charge (SoC) of a battery-electric bus, including powertrain data, heating, air conditioning, and auxiliary consumers for different route sections taking into account altitude data, outside temperature and load. This should enable precise mapping of route sections, schedules, and extreme scenarios. The model should also be easily adaptable to different bus types. In the following sections, the functions and individual components of the simulation model are described in more detail. Furthermore, the input parameters and the verification of the model are discussed, an outlook for improvements and further applications is given.

1 Model Overview

This section provides an overview of the simulation model. The structure and individual components are described in more detail. The first step in developing the model is to identify the available data and determine how detailed the model should be.
For the initial setup of the model, the MAN Lion's City 12 E low-floor bus is selected, as data for verification purposes are available from a battery-electric bus fleet that is already in operation. The model is designed so it can be easily re-parameterized for other bus models through an initialization script. Due to the limited availability of technical data, electric drive train components such as the inverters are not physically simulated in detail, this also ensures fast computation times. Instead, an efficiency map based, purely longitudinal dynamic model was developed. Since the main focus is energy consumption, the longitudinal dynamics prove

to be sufficiently accurate, as can be seen in the verification process.
Figure 1 provides an overview of the model. The model inputs are a speed profile over time, elevation profile over the distance traveled and average outside temperature. Due to the monitoring system, the fleet operators in this study have only access to sparsely sampled GPS positions (5.3 samples per minute) without altitude information. Consequently, these GPS points require a special preprocessing methodology to reconstruct the route data (including speed and road slope) to a sufficient sampling frequency of 1 Hz. The detailed methodology employed is described in [6].

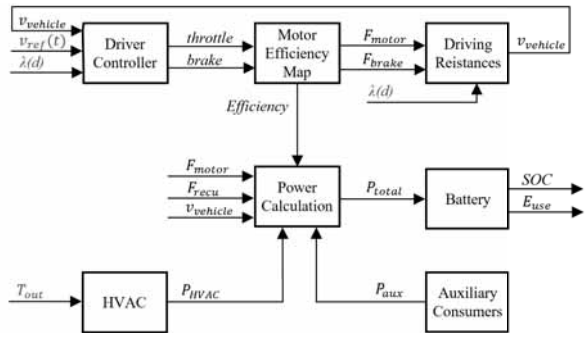


Figure 1: Simulation model overview

As driver controller, the "Longitudinal Driver" component of the Simulink vehicle dynamics blockset is used as a PI controller [7]. This controls the vehicle speed and outputs throttle and brake pedal position.
In the motor efficiency block, the current motor torque is then determined using the accelerator pedal position and the maximum possible torque for the current operating point. The maximum torque can be determined from a motor map using the motor speed calculated from the vehicle speed and the gear ratio.
From the motor torque, the motor force F_{motor} that drives the vehicle is then calculated using tire radius and gear ratio. The recuperation force F_{recu} of the motor is also calculated in this way. For strong decelerations, an additional mechanical brake is used. This results in the total braking force F_{brake}. Furthermore, the efficiency during driving and recuperation is determined using the motor map stored in a lookup table. As the specific motors installed in the MAN bus are not known, the map of an asynchronous electric motor with similar power data is used. Exemplary motor maps are available in publications e.g. [8, 9] or can be generated via simulation tools such as Matlab or Motor CAD.

The driving resistances block uses F_{motor}, the total braking force F_{brake}, and the road gradient λ(d) as inputs. Here, all necessary driving resistances of the bus are calculated. These are rolling resistance F_{RW}, air resistance F_{RA}, gradient resistance F_{RC}, and inertia resistance F_{RI}. This results in the total driving resistance F_R calculated in equation 1 using the individual resistances from equation 2. [10]

$$F_R = F_{RW} + F_{RA} + F_{RC} + F_{RI} \tag{1}$$

$$\begin{aligned} F_{RW} &= m_{ges} * g * \mu \\ F_{RA} &= cw * A * \frac{\rho_A}{2} * v^2 \\ F_{RC} &= m_{ges} * g * sin(\lambda) \\ F_{RI} &= a_x \left(e_i * m_{net} + m_{pl}\right) \end{aligned} \tag{2}$$

The result is the longitudinal vehicle velocity, which is fed back into the driver controller as $v_{vehicle}$. Since there are certain deviations between v_{ref} and $v_{vehicle}$ due to the driver controller, the road gradient λ(d) is given over distance and not over time. This way, the actual distance traveled can be calculated by integrating the vehicle velocity, and the appropriate road gradient at that position can be used.
To map power and energy consumption of the vehicle, the power calculation block is introduced. The driving power is initially calculated from motor force (F_{motor}), recuperated force (F_{recu}), and vehicle velocity ($v_{vehicle}$). Additionally, motor efficiency, determined from the efficiency map, is used to calculate the required driving power. For the gearbox and power electronics, a constant efficiency is assumed.
The consumption of the Heating-Ventilation Air Conditioning (HVAC) components also have a significant impact on the range of electrically operated buses. Consequently, the power of the heat pump is mapped depending on outside temperature via a lookup table [11]. Auxiliary consumers such as display boards, doors, and compressors are taken into account with a constant power consumption. The power of HVAC (P_{HVAC}) and auxiliary consumers (P_{aux}) are introduced into the power calculation block. Thus, the total power (P_{total}) can be determined by summing the power of the different components. By integrating P_{total}, it is also possible to determine the energy (E_{use})consumed for the driven cycle.

1.1 Battery Model

To be able to simulate the SOC of the vehicle, a battery model is added to the simulation model. Here, the table-based battery model from the Simscape library is used and populated with parameters from a nickel-manganese-cobalt battery cell, which was used in a previous work [12]. The cell has a maximum Open-circuit voltage (OCV) of 4.15 V (V_{ocv}) and a capacity of 14.44 Ah (C_{nom}). In the battery model, the OCV is calculated as a function of the SoC and temperature. The internal resistance also depends on the SoC and temperature (T) [13]. Self-discharge and aging are initially neglected. Only limited information is available about the battery installed in the MAN Lion's City, though it is known to be an 800 V system with 65 % of the 480 kWh battery capacity usable. Since the Simscape battery model is an electrical model, a current must be calculated from the previously determined total vehicle power (P_{total}). This is done using equation 3.

$$I_{bat} = \frac{P_{total}}{U_{bat}} \tag{3}$$

Here P_{total} is the total power of all consumers determined in the power calculation block. U_{bat} is the battery voltage measured at the battery model and delayed by one time step. This approximates the battery current. A controlled current source is then used to charge or discharge the battery model. The SoC of the battery is then calculated in the Simscape block using equation 4.

$$SoC(t) = SoC(t=0) - \frac{1}{C_{nom}(T)} \int_0^t (I_{bat}(t))\, dt \tag{4}$$

This enables simulation of any route sections and calculation of energy consumption as well as battery SoC.

1.2 Verification

To verify the functionality and parameterization of the model, precise data must be collected. For this purpose, data is provided by the Swiss transport company "Verkehrsbetriebe Zürich" which is already operating numerous electric buses and has a detailed monitoring system. The data used for verification are GPS position, vehicle speed, elevation profile, outside temperature, power of the heatpump, power of the powertrain, energy of auxiliary consumers and the SoC.
With this data, the exact consumption, divided into drivetrain (driving and recuperation), heat pump depending on the route and auxiliary consumers can be determined. The same routes are also simulated with the longitudinal bus model. A comparison between the consumption for driving, recuperation, air conditioning, and auxiliary consumers, as well as the SoC is made.
Since the available test data was collected in winter, the installed diesel heater was often recorded as active (activated at temperatures below 7 °C). The energy consumed by the heatpump is therefore very low. Further verification of the HVAC modeling for other seasons is planned.
Figure 2 shows the SoC progression of the real vehicle compared to the SoC progression of the simulation model. For this comparison, the speed recorded by the monitoring system is used as the input for the simulation model. Additionally, the altitude data is used to determine the incline over distance and also serves as input. The environmental temperature is also taken into account in the simulation. Here it can be seen that the two SOC curves show only small deviations.

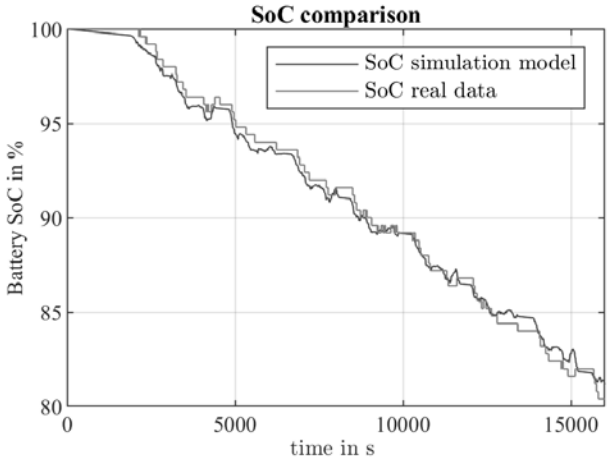


Figure 2: State of charge of the battery model compared to real vehicle data driving the same route

As route input, a round trip of 44 km length until a charger is approached is used. To quantify the differences, the maximum deviation and the Mean Squared Error (MSE) is calculated. The maximum deviation between the two SoC curves is 1.4469%. The calculated MSE is 0.31737, which indicates a high level of concordance between the two datasets.
Table 1 presents the energy consumption of the buses main consumers on the same 44 km round trip as in figure 2, broken down by drivetrain, HVAC and aux-

iliary consumers. This data also demonstrates that the energy consumption per component in the simulation model corresponds to the real-world data.

	$E_{drivetrain}$	E_{HVAC}	E_{aux}
Real data	57.07 kWh	0.84 kWh	11.98 kWh
Simulation	57.56 kWh	0 kWh	11.34 kWh

Table 1: Energy consumption by component for simulation and real data

2 Discussion

In this paper, a simulation model for a battery-electric bus is presented. This model is capable of accurately predicting energy consumption and state of charge with minimal input data. By considering driving resistances, elevation profiles, temperatures, velocity and various load conditions, a model is created which accurately represents a real electric bus. The validation of the model with data from a Swiss public transport company demonstrates high accuracy, highlighting the model's relevance and reliability, although only a minimal set of parameters are available.

The model enables public transport operators to precisely plan their bus routes, battery sizing and charging infrastructure before electrifying their fleets. This not only enhances efficiency but also optimizes costs and relieves the power grid. Additionally, the model allows for the simulation of extreme scenarios, such as full load at cold temperatures, which would not be possible with average consumption values.

In the future, the model is to be extended and includes battery aging, enabling better assessment of long-term performance. This will determine whether an old bus is still capable of covering all routes with the available charging points. Furthermore, additional parameter files for other buses should follow. Moreover, adding a more intelligent controller that can map different driver types is conceivable. Integrating different driving styles is a valuable addition to simulate the impact of driving behavior on energy consumption.

Overall, this battery-electric bus model represents a powerful tool for planning and optimizing electric bus fleets, supporting the sustainable transformation of public transport, as the application in the project FreeE-Bus funded by Interreg Alpenrhein-Bodensee-Hochrhein shows.

References

[1] Bundesministerium für Digitales und Verkehr. *Gesetzeskarte Elektromobilität im ÖPNV*. Berlin; 2022.

[2] Statista. *Anzahl der elektrifizierten Busse im öffentlichen Personennahverkehr in Deutschland nach Antriebsart von 2019 bis 2022*. retrieved: 05.28.2024, https://de.statista.com/statistik/daten/studie/1048269/umfrage/anzahl-der-elektrobusse-in-deutschland-nach-antriebsart/

[3] Zank P, Heininger P. *VDE Renewables GmbH: Elektrifizierung von KMU-Busunternehmen*. Alzenau; 2023.

[4] Seeliger A, et al. *Elektrobusse im ÖPNV - Eine technisch/wirtschaftliche Analyse unter Berücksichtigung praktischer Anwendungsbeispiele*. Hochschule Niederrhein, Krefeld; 2016.

[5] Jahic A, et al. *Energy Consumption of Battery- Electric Buses: Review of Influential Parameters and Modelling Approaches*. B&H Electr. Eng.; 2023. doi: 10.2478/bhee-2023-0007.

[6] Hitz A, Konzept A, Reick B, Rheinberger K (in press). *Efficient GPS route matching method for battery electric bus fleets*. SAE Technical Paper; 2024.

[7] Mathworks. *Longitudinal Driver*; retrieved: 05.28.2024, https://de.mathworks.com/help/autoblks/ref/longitudinaldriver.html

[8] Hayes G. J, Goodarzi A. G. *Electric Powertrain: Energy Systems, Power Electronics and Drives for Hybrid, Electric and Fuel Cell Vehicles*. John Wiley & Sons Ltd; 2017. doi: 10.1002/9781119063681.

[9] Winzer P, Doppelbauer M. *Berechnung von Wirkungsgradkennfeldern von Asynchronmaschinen mit Hilfe der Finite-Elemente-Methode*. Karlsruhe Institute of Technology (KIT); 2013.

[10] Mitschke M, Wallentowitz H. *Dynamik der kraftfahrzeuge*. Springer Vieweg Wiesbaden; 2014. doi: 10.1007/978-3-658-05068-9.

[11] Heß L, et al. *Analysis of the Specific Energy Consumption of Battery-Driven Electrical Buses for Heating and Cooling in Dependence on the Technical Equipment and Operating Conditions*. World Electric Vehicle Journal; 2023. doi:10.3390/wevj14050126.

[12] Konzept A, Reick B, Pintaric I, Osorio C. *HIL based Real-Time Co-Simulation for BEV Fault Injection Testing*. SAE Technical Paper; 2023. doi: 10.4271/2023-24-0181.

[13] Mathworks, *Battery(Table-Based)*; retrieved: 05.28.2024, https://de.mathworks.com/help/sps/ref/batterytablebased.html

Appendix

A	Cross sectional area
a_x	Acceleration in x direction
C_{nom}	Nominal battery capacity
CO2	carbon dioxide
cw	Drag coefficient
E_{aux}	Energy consumption of auxiliary consumers
$E_{drivetrain}$	Energy consumption of drivetrain
E_{HVAC}	Energy consumption of HVAC components
e_i	Moment of inertia addition-factor
E_{use}	Energy consumption total
F_{brake}	Brake force
F_{motor}	Motor Force
F_R	Total driving resistance
F_{RA}	Aerodynamic drag
F_{RC}	Climbing resistance
F_{recu}	Recuperation force
F_{RI}	Inertial resistance
F_{RW}	Wheel resistance
I_{bat}	Battery current
m_{ges}	Total vehicle mass with load
m_{net}	Vehicle mass without load
m_{pl}	Mass of load
MSE	Mean Squared Error
OCV	Open circuit voltage
P_{aux}	Power auxiliary consumers
P_{HVAC}	Power Heating-Ventilation, Air Conditioning
P_{total}	Total power
SoC	State of Charge
SORT	Standardized On Road Test cycle
T_{out}	Environmental Temperature
U_{bat}	Battery voltage
$v_{vehicle}$	Vehicle longitudinal velocity
v_{ref}	Reference velocity
$\lambda(d)$	Road gradient as a function of distance
ρ_A	Air density

Architecture of Simulation-Based Representations for Digital Twins

Günther M. Gaßner[1*], Jenny Rüffer[2*], Madlene Leißau[2]

Tobias Voigt[1], Christoph Laroque[2]

[1]Technical University of Munich, School of Life Sciences, Chair of Brewing- and Beverage Technology, Weihenstephaner Steig 20, 85354 Freising, Germany; *guenther.gassner@tum.de*

[2]University of Applied Sciences Zwickau, Chair for Business Informatics, Kornmarkt 1, Zwickau, 08056, Germany; *jenny.rueffer@fh-zwickau.de*

Abstract. The world faces significant sustainability challenges, from resource consumption to supply chain inefficiencies. A comprehensive simulation-based approach for creating Digital Twins (DTs) is developed to address these issues. This paper contributes to the gradual development of DTs and presents a novel architecture for a simulation tool. The proposed architecture enables the creation of a holistic virtual representation of the system by incorporating linear and circular production models, supporting various scenarios, identifying optimization potential, and making data-driven decisions. Using simulation-based DTs indicates the potential to drive sustainable transformation in the beverage industry and beyond. As the approach progresses, it aims to provide a blueprint for leveraging digital technologies, fostering a more sustainable and resilient future.

Introduction

The world is confronted with unprecedented sustainability hurdles, including climate change, resource depletion, social inequality, and economic instability. There is a desperate need to tackle these intricate and interrelated issues. To combat these problems, the United Nations has established the Sustainable Development Goals (SDGs), a global framework for achieving a more sustainable, equitable, and prosperous future for all [1]. Digital technologies, such as the Internet of Things (IoT), Artificial Intelligence (AI), and simulation, are increasingly recognized as powerful tools for supporting the achievement of the SDGs [2, 3].

In particular, SDG 9 emphasizes promoting inclusive and sustainable industrialization, fostering innovation, and the need for resilient and sustainable infrastructure. By using digital technologies to design, optimize, and manage auch infrastructures and industrial processes, organizations can make data-driven decisions that minimize environmental impact and promote responsible production and consumption, as highlighted in SDG 12 [3, 4]. This aligns with the emerging paradigm of Industry 5.0, which builds upon the technological advancements of Industry 4.0 while placing a strong emphasis on the collaboration between humans and machines to create more sustainable, resilient, and human-centric industrial processes [5, 6].

One promising application of digital technologies for sustainable development is the concept of virtual representations of physical systems that are capable of real-time data to mirror the behavior and performance of their real-world counterparts [7, 8]. By integrating sustainability metrics and indicators into these systems, organizations can assess their operations' environmental impact, identify improvement opportunities, and make informed decisions to reduce their ecological footprint [9].

Manufacturing also encounters critical sustainability concerns, from energy and water consumption to waste generation and supply chain inefficiencies. To address these, the BeverGreen project [10], a collaborative effort involving the beverage industry and academic partners, aims to develop an extensive methodology for creating sustainability systems in this field. Aided by simulation techniques and integrating sustainability metrics, the project seeks to optimize resource efficiency, minimize environmental impact, and promote circular economy principles.

We aim to provide valuable insights and practical guidance for organizations seeking to harness technologies for driving sustainable development and Industry 5.0 principles through examining the intersection of simulation, DTs and sustainability,

1 Digital Twins and Simulation

1.1 Digital Twin Fundamentals

DTs are virtual representations of physical systems that use historical data, real-time data, and different data processing methods to act as a system or part of it. It also allows for simulation, analysis, and optimization of the physical counterpart. The concept of DTs was first introduced by Michael Grieves in 2002 in the context of product lifecycle management (PLM) [11]. Since then, they have evolved to encompass various applications, from manufacturing to healthcare [12]. A key characteristic of DTs is their ability to integrate data from multiple sources, such as sensors, historical data, and domain knowledge, to create an accurate and up-to-date representation of the physical system [13]. Simulation plays a crucial role in developing DTs by enabling the prediction of system behavior under different conditions and scenarios [14].

1.2 Simulation Paradigms for Digital Twins

Several simulation paradigms are employed in developing DTs, each with strengths and limitations. Discrete event simulation (DES) is widely used to model systems where state changes occur at discrete points, such as manufacturing systems and supply chains [15]. Agent-based simulation (ABS) focuses on modeling the behavior and interactions of individual agents within a system, making it suitable for modeling complex adaptive systems [16]. System dynamics (SD) models the feedback loops and nonlinear relationships in complex systems, such as ecosystems and social systems [17]. Hybrid simulation approaches combine multiple paradigms to leverage their strengths and address individual paradigms' limitations [18].

1.3 Simulation-Based Digital Twin Development Process

Developing, commissioning, running, and maintaining DTs using simulation typically involves four main stages: conceptual modeling, implementation, verification, and validation [19], and experimentation. Conceptual modeling consists of defining the scope, objectives, and assumptions of DTs, as well as identifying the key components and relationships within the system [20].

Implementation involves translating the model into a computer model using appropriate simulation software and programming languages [21]. Verification and validation ensure that DTs accurately represents the real-world system and produces reliable results. Experimentation involves exploring different scenarios, optimizing system performance, and supporting decision-making.

The concept of Green Digital Twins (GDTs) [22] has recently emerged as a promising approach to address as of SDGs. These expanded DTs with a subclass containing properties and requirements mainly characterized by integrating sustainability-related specifications. A GDT based on this concept requires digital models of resources, processes, domain knowledge, and energy networks, such as energy sources and sinks, as well as models of their relationships as well as a component emission repository, by which emission factors can be used for calculating CO2-equivalents or other metrics. [9]. This enables organizations to assess their operations' environmental impact, identify improvement opportunities, and make data-driven decisions to reduce their ecological footprint [4]. Across economic, environmental, and social dimensions, GDTs facilitates the alignment of business objectives with sustainability goals, promoting responsible production and consumption practices by providing a holistic view of the system's performance.

The emergence of AI has been a significant driver in the development and application of DTs [23]. Its techniques, such as machine learning, enable them to learn from data, adapt to changing conditions, and make autonomous decisions [24]. The integration of AI has opened up new possibilities for optimizing system performance, predicting failures, and enabling predictive maintenance [8]. AI-powered DTs can analyze vast amounts of data in real time, identify patterns and anomalies, and provide actionable insights for improving operational efficiency and reducing downtime. This synergy between AI and DTs is particularly relevant for complex, dynamic systems, where traditional simulation approaches may struggle to capture the full range of system behaviors and interactions.

Addressing these challenges requires collaborative efforts across disciplines to develop robust, scalable, and interoperable platforms. The BeverGreen project aims to address some of these challenges by developing a thorough methodology for creating GDTs in the beverage industry, focusing on integrating sustainability metrics and reducing environmental impact.

2 Methodology

2.1 Overview and Objectives

Our simulation-focused architecture builds on assembling a team of domain experts in production and distribution, software developers, and scientific researchers to develop comprehensive DTs that map a supply chain. Within the BeverGreen project, we aim to digitally interlock linear production chains within companies and circular processes involving multiple actors to provide a foundation for wide-ranging optimization potential. In line with global sustainability goals and resilient economics, the project focuses on critical areas such as energy management, production planning, inventory management, and transport logistics, seeking to identify adaptable parameters that contribute to these objectives [9].

2.2 Simulation-Focused Approach for DTs

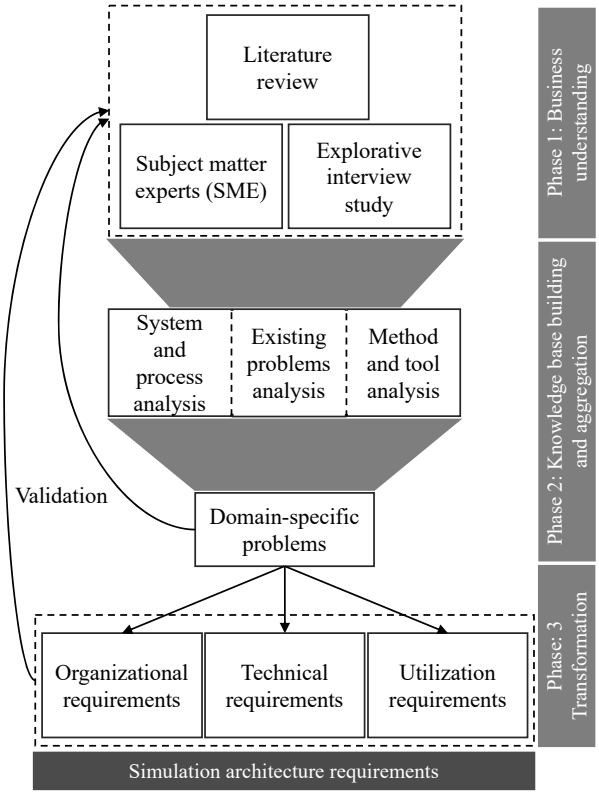


Figure 1: Determinational Approach for Architectural Requirements

Our simulation-focused approach (Figure 1) consists of business understanding, an initial phase that aims to identify sources for generating a comprehensive knowledge foundation. Starting with a literature review, an explorative interview study including all relevant partners along the supply chain (manufacturers, intermediate trade and retail, associational partners), aided by exchange with several subject matter experts (SME) within the team, we created a multi-layered and multi-perspective glimpse of general systematic and procedural conditions as well as challenges along the supply chain.

Based on insights gained in phase 1, we conducted further analysis in a second phase, knowledge base building and aggregation. We dissected the system's production, distribution, and circulation behavior and their systemic interactions. On a process level, we investigated necessary conditions and movement patterns of products and resources within the manufacturing company and also between partners of the supply chain. Problems regarding efficiency and sustainability were identified and specified regarding their origin in existing processes and, if possible, analyzed and concretized.

Simultaneously, methods and tools that might contribute to overcoming existing challenges were viewed to reduce identified black boxes in the relevant processes and simulate future scenarios. Based on the dissection, domain-specific problems could be described and validated by synchronizing them with the initially detected sources of business understanding.

Phase three, Transformation, mainly addressed characterizing specific requirements resulting from challenges. Given the standards and conditions, organizational requirements that refer to diverse demands along the supply chain were identified. At the same time, necessary organizational conditions must be ensured for the successful development of comprehensive DTs.

The DT being developed must meet a range of technical requirements and functionalities, including the ability to map relationships between machines, resources, and products, the utilization of domain knowledge, rules, and indicators, connectivity to various data sources and simulations, visualization capabilities, integration of data analysis apps, and forecasting capabilities.

Utilized, individual interests and aims regarding the usage of a DT were specified, including the essential

provision of sensible user interfaces and high data security standards. Furthermore, requirements that software developers need as a foundation for building were emphasized.

However, the availability of data to develop and use DTs is heavily restricted due to varying levels of digitalization and interests among supply chain partners, including limitations in transferability between different systems. Alongside efforts to access data and form a basis for their functionality, simulation models will initially generate data and test various scenarios for energy utilization, routing, and product launching [3].

2.3 Model and Simulation Approach

The simulation approach has been selected to address data black boxes, gain well-founded knowledge about processes and their reciprocal influence, and identify optimization potential within individual steps [18]. To capture the system's behavior, we have chosen a multi-method simulation approach [25].

A meta-model will be developed to understand system behavior and dynamics more thoroughly, as well as global dependencies between system inputs and outputs. Individual process chains will also be analyzed to create a foundation for developing discrete event simulation models [15]. Both methods need to be combined to achieve the overall goal of representing an entire supply chain. Given the high complexity of the system and the need to examine emergent behavior, agent-based methods will also be incorporated into the modeling process [16].

2.4 Data Collection and Analysis Methods

Data generation and connection are crucial aspects of the project. The team will work on establishing a robust data infrastructure that enables the integration of data from various sources, including sensors, historical records, and domain expertise. Advanced data analysis techniques, such as machine learning and data mining, can extract valuable insights from the collected data [23]. The resulting dataset can inform and validate the simulation models, ensuring their accuracy and reliability in representing the real-world system [19].

Our methodology combines state-of-the-art DT technologies, simulation paradigms, and data analysis techniques to develop a comprehensive and sustainable approach for optimizing supply chain operations and processes in the beverage industry. By addressing the challenges associated with data availability, model integration, and system complexity, the project aims to demonstrate the potential of DTs in driving progress towards the United Nations SDGs [2].

3 Use Case and Results

3.1 Project BeverGreen

BeverGreen focuses on leveraging digital technologies to address the specific challenges faced by the beverage industry, such as resource efficiency, reusable bottle circulation scheduling, and energy efficiency. Global issues, including rising energy and raw material costs, limited resources, climate protection initiatives, and new resilience requirements for supply chains exacerbate these challenges. To tackle these challenges, it aims to develop an assistance system that maps existing data structures to domain-specific ontologies containing relevant information, forming the basis for GDTs.

An assistance system is the starting point for linking internal and external datasets, such as life cycle assessment databases and CO2 equivalents (CO2e). By integrating these datasets, the project seeks to create a holistic view of the beverage industry's environmental impact and identify opportunities for optimization. The development of GDTs, in combination with machine learning methods, is a crucial objective of the BeverGreen project, aiming to identify and realize energy and resource savings in exemplary application scenarios.

3.2 Explanation of the Simulation-Based Digital Twin Architecture

Two primary areas are focused on: closed application circles in production and circular value creation networks. In brewing processes, DTs are being developed to optimize resource consumption, minimize waste, and improve overall efficiency. Creating simulation-based representations of brewing processes enables stakeholders to simulate various scenarios, test optimization strategies, and make data-driven decisions to reduce the environmental footprint of their products.

Within logistics, the project addresses challenges associated with reusable bottle circulation scheduling. DTs of this circuit aims to optimize the flow of reusable bottles, minimize transportation costs, and reduce the overall environmental impact of distribution. Co-opting real-time data and advanced analytics seeks to create a

more efficient and sustainable logistics network for the beverage industry.

BeverGreen's use cases demonstrate the potential of DTs and machine learning in driving the sustainable Transformation of the beverage industry. The project aims to serve as a beacon for other industries seeking to optimize their operations and reduce their environmental impact by addressing specific challenges related to resource efficiency, energy consumption, and circular value creation. The insights gained from this project will contribute to developing best practices and guidelines for implementing digital technologies in pursuit of sustainability goals.

3.3 Architecture

The proposed architecture for simulation-based representations is an integral part of this development and is illustrated in Figure 2. It aims to provide a comprehensive framework for integrating various data sources, enabling semantic linking, and facilitating holistic DTs.

3.3.1 Data Sources and Mapping

The first step in the architecture is to identify and connect all relevant data sources across the supply chain that provide the necessary data for the desired functionalities. As shown in Figure 2, this includes data from Enterprise Resource Planning (ERP), Manufacturing Execution Systems (MES), Supervisory Control and Data Acquisition (SCADA) systems, and other databases, which are crucial for depicting production and distribution processes at all relevant operational system levels within and across partners.

For BeverGreen, specific data sources may include brewery production data, inventory levels, energy consumption metrics, and transportation logistics information. Additionally, environmental data sources, such as emission data, weather data, traffic data, economic and political data, and state or institutional regulations, must be incorporated to increase sustainability along the entire supply chain. In the context of the beverage industry, this may involve integrating data on carbon footprints, water usage, and waste generation throughout the production and distribution processes.

To integrate these diverse datasets, a Federated Database Systems (FDBSs) will be established, which connects to the various data sources through Application Programming Interfacess (APIs) and serves as the essential basis for the simulation. The FDBS leverages technologies such as data lakes, data warehouses, and ETL (Extract, Transform, Load) processes to enable robust data integration from heterogeneous sources in a predefined ontology. A data pipeline will be set up to prepare data for use in simulation models and feed data analysis workflows. These analytical processes, such as machine learning algorithms for optimization and data mining techniques for identifying patterns and anomalies, can be embedded at this stage to enhance the data processing capabilities.

In the context of collaborative data warehousing initiatives. To facilitate simulation objectives and develop company-wide DTs, the industry strictly uses anonymized or aggregated data to protect sensitive information.

However, guaranteeing data security across DTs functions and layers utilized by various supply chain partners is a formidable challenge that must be addressed. Numerous studies have investigated security risks and proposed several approaches to tackle these issues, both from a technological standpoint; security protocols are paramount in terms of security management and procedures [26]. Our project team will evaluate different measures to identify and implement effective security solutions while establishing a robust user and access rights management system.

3.3.2 Simulation Models and Data Routes

The prepared simulation data is used to parameterize linear and circular production models via an API, enabling the initiation of multiple simulation runs and experiments, as depicted in Figure 2. In BeverGreen, linear production models may represent brewing processes, while circular production models may encompass reusable bottle circulation and recycling processes.

To validate and verify the initial results and, more importantly, the underlying simulation model, validation techniques such as historical data validation, face validity, and sensitivity analysis are applied. These techniques involve comparing simulation results with existing real-life data in the FDBS, ensuring that the model accurately depicts the real-world system. For example, the simulated energy consumption and resource utilization in the brewing process can be compared against historical data to validate the model's accuracy.

Furthermore, the resilient simulation results that emerge later must be imported back into the FDBS to continuously feed the knowledge base with

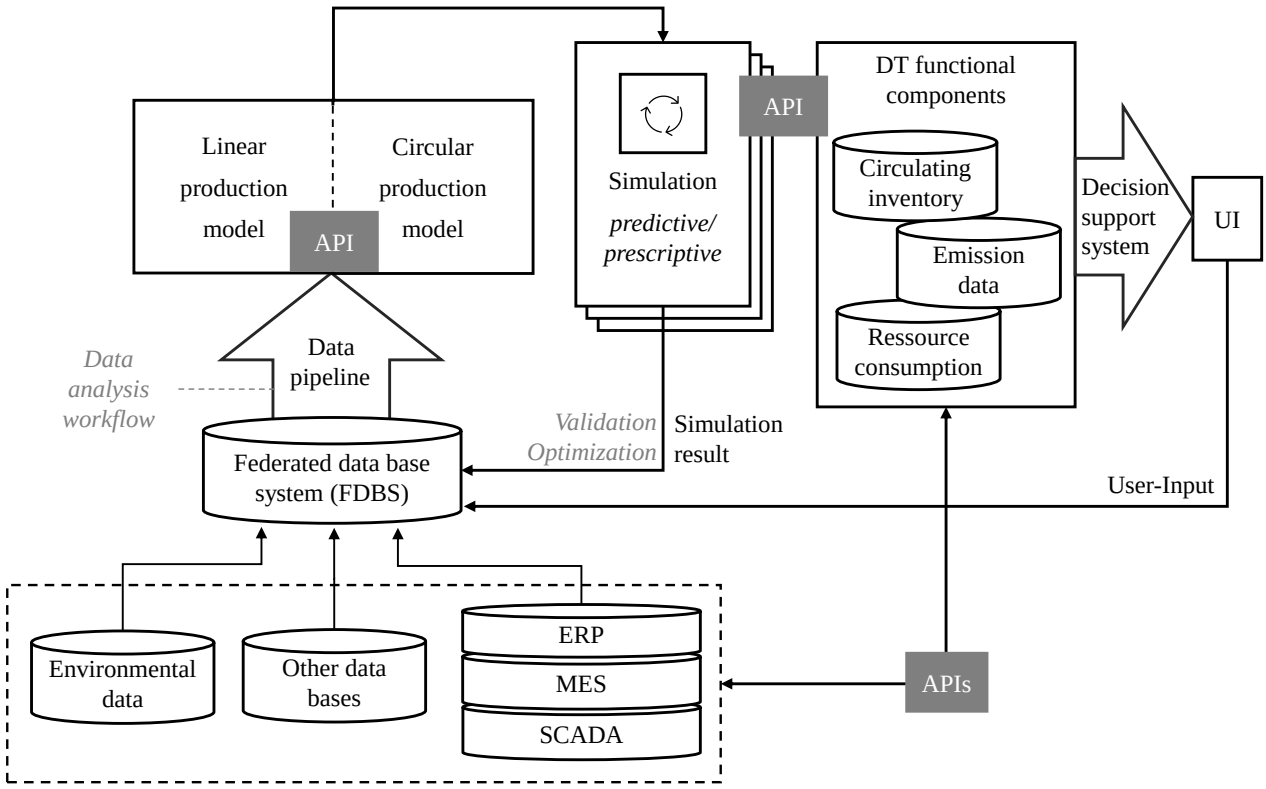


Figure 2: Architecture of a Simulation-based DT

new findings. The resulting feedback loop and simulation-assisted exploration of What-If analysis allows decision-makers to anticipate the impact of potential changes or disruptions on the supply chain, which guides iterative improvement of simulation models and hence DTs overall.

Another API is set up to incorporate the results of various simulation runs and experiments into the DTs functional components, e.g. circulating inventory, resource consumption, and emission data. Specifically, simulation results provide synthetic data to understand assumed relationships between machines, resources, and products, such as indicators and rules for circulating inventory, consumption, or emissions. For instance, the simulation may reveal correlations between production time, energy consumption, and peak electricity usage, which can be used to optimize the shedule of brewing processes for sustainability. All results are merged with data retrieved by the DTs using a multi-connective data mapping assistant, ensuring a comprehensive and up-to-date representation of the system.

3.3.3 Decision Support and User Interface

Enriched with synthetic data, DTs will provide a rich backend for decision support systems of various individual stages along supply chains. In order to offer options designed to detect patterns, predict future trends, and provide actionable insights to support data-driven decision-making. The decision support is realized via a User Interface (UI), as illustrated in Figure 2, and enables stakeholders to access decision-specific elements.

The emerging decision support capabilities of DTs are propelled by enabling synoptic comparison of results after processing the vast amounts of data generated by simulation models and integrated data sources. It also has to be able to visualize the wealth of information generated within a DT. Adapted to the use cases and needs of different applications or roles within the beverage industry, the UI serves as a hub for user access to DT functional components.

4 Discussion and Outlook

The proposed architecture offers a robust framework for integrating diverse data sources, ontologies, and supporting the creation of a comprehensive DT. By incorporating simulation models, data mapping, and user interfaces, this architecture supports the gradual development of DTs, promoting data-driven decision-making and process optimization in the BeverGreen project and the broader beverage industry.

The modular nature of the architecture, with its microservices-based data integration and flexible simulation modeling approach, allows for adaptation to different supply chain structures and sustainability challenges beyond the beverage sector. Adapting and enhancing the architecture to industry-specific data sources, simulation models, and key performance indicators (KPIs) provides a tailored solution that supports sustainable development across sectors.

The simulation component is pivotal in developing and continuously improving DTs by enabling organizations to explore various scenarios, test optimization strategies, and make informed decisions to enhance sustainability performance. The seamless integration of simulation models allows for the continuous refinement of the virtual representation based on real-world feedback, making it an increasingly powerful tool for driving sustainable innovation and decision-making as it evolves and incorporates new data sources and simulation results.

Based on the current state of understanding, the following procedure includes modeling production chains and logistics within the beverage industry at the meta and process level as a basis for running simulations. In this particular process, we address the connection between models to describe the supply chain comprehensively and to tackle overall aims in terms of emissions, circulating inventory, and resource consumption further.

However, the proposed architecture faces obstacles in ensuring data security when integrating sources from multiple partners along the supply chain. This requires careful consideration of specialized solutions and robust security management, where various techniques have been suggested to solve existing challenges with technological or security management measures and procedures to overcome these hurdles. Our project team will consider the framework's scalability and generalizability to other industries and supply chain structures. It may require further investigation and validation to ensure its effectiveness in supporting adequate data security.

As the BeverGreen project progresses and demonstrates the value of simulation-based GDTs in driving sustainable development, it will serve as a blueprint for other industries seeking to harness the power of digital technologies to optimize their operations and reduce environmental impact. By sharing best practices, lessons learned, and the technical architecture developed within the project, we aim to accelerate the adoption of DTs including GDTs across various sectors, moving closer to Industry 5.0 and sustainable production.

Acknowledgement

This research and development project is/was funded by the Federal Ministry of Economics and Climate Protection (BMWK) in the program "GreenTech Innovation Competition - Digital Technologies as the Key to the Ecological Transformation of the Economy" (funding code 01MN23008B) and supervised by the DLR project management organization.

References

[1] United Nations. Transforming our world: The 2030 agenda for sustainable development. United Nations, Department of Economic and Social Affairs. 2015.

[2] Vinuesa R, Azizpour H, Leite I, Balaam M, Dignum V, Domisch S, Felländer A, Langhans SD, Tegmark M, Fuso Nerini F. The role of artificial intelligence in achieving the Sustainable Development Goals. *Nature Communications*. 2020;11(1).
URL http://dx.doi.org/10.1038/s41467-019-14108-y

[3] Jiang F, Ma L, Broyd T, Chen K. Digital twin and its implementations in the civil engineering sector. *Automation in Construction*. 2021;130:103838.
URL http://dx.doi.org/10.1016/j.autcon.2021.103838

[4] Lu Y, Liu C, Wang KIK, Huang H, Xu X. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*. 2020;61:101837.
URL http://dx.doi.org/10.1016/j.rcim.2019.101837

[5] Nahavandi S. Industry 5.0—A human-centric solution. *Sustainability*. 2019;11(16):4371.

[6] Demir KA, Döven G, Sezen B. Industry 5.0 and human-robot co-working. *Procedia Computer Science*. 2019;158:688–695.

[7] Tao F, Sui F, Liu A, Qi Q, Zhang M, Song B, Guo Z, Lu SCY, Nee AYC. Digital twin-driven product design framework. *International Journal of Production Research*. 2018;57(12):3935–3953. URL http://dx.doi.org/10.1080/00207543.2018.1443229

[8] Barricelli BR, Casiraghi E, Fogli D. A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. *IEEE Access*. 2019;7:167653–167671. URL http://dx.doi.org/10.1109/ACCESS.2019.2953499

[9] Wagstyl D, et al. *Towards Carbon Neutrality Using Green Digital Twins for Industrial Energy Systems*. Institute of Electrical and Electronics Engineers (IEEE). 2024; ISBN: 979-8-3503-5229-0/24.

[10] BeverGreen. Grüner Digitaler Zwilling als Basis der nachhaltigen Transformation der Getränke- und Brauwirtschaft. *BMWK GreenTech Innovation Competition*. 2023;. URL https://bevergreen.de/

[11] Grieves M, Vickers J. Digital twin: Manufacturing excellence through virtual factory replication. *White paper*. 2014;1:1–7.

[12] Rasheed A, San O, Kvamsdal T. Digital Twin: Values, Challenges and Enablers From a Modeling Perspective. *IEEE Access*. 2020;8:21980–22012. URL http://dx.doi.org/10.1109/ACCESS.2020.2970143

[13] Fuller A, Fan Z, Day C, Barlow C. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access*. 2020;8:108952–108971. URL http://dx.doi.org/10.1109/ACCESS.2020.2998358

[14] Negri E, Fumagalli L, Macchi M. A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manufacturing*. 2017;11:939–948. URL http://dx.doi.org/10.1016/j.promfg.2017.07.198

[15] Law AM. *Simulation modeling and analysis*. McGraw-Hill New York, 5th ed. 2014.

[16] Macal CM, North MJ. Tutorial on agent-based modelling and simulation. *Journal of Simulation*. 2010;4(3):151–162. URL http://dx.doi.org/10.1057/jos.2010.3

[17] Sterman J. *Business Dynamics*. USA: McGraw-Hill, Inc., 1st ed. 2000.

[18] Brailsford SC, Eldabi T, Kunc M, Mustafee N, Osorio AF. Hybrid simulation modelling in operational research: A state-of-the-art review. *European Journal of Operational Research*. 2019;278(3):721–737. URL http://dx.doi.org/10.1016/j.ejor.2018.10.025

[19] Sargent RG. Verification and validation of simulation models. *Journal of Simulation*. 2013;7(1):12–24. URL http://dx.doi.org/10.1057/jos.2012.20

[20] Robinson S. Conceptual modelling for simulation Part I: definition and requirements. *Journal of the Operational Research Society*. 2008;59(3):278–290. URL http://dx.doi.org/10.1057/palgrave.jors.2602368

[21] Eldabi T, Paul RJ, Young T. Simulation modelling in healthcare: reviewing legacies and investigating futures. *Journal of the Operational Research Society*. 2007;58(2):262–270. URL http://dx.doi.org/10.1057/palgrave.jors.2602222

[22] Deuse J, Wagstyl D, Hernandez Moreno V, Polikarpov M, Wöstmann R, Hoffmann F. *Green Digital Twins in the Product Life Cycle*, p. 167–186. GITO mbH Verlag. 2023;. URL http://dx.doi.org/10.30844/wgab_2023_10

[23] Deuse J, Wöstmann R, Syberg M, West N, Wagstyl D, Moreno VH. *Establishing a Machine Learning and Internet of Things Learning Infrastructure by Operating Transnational Cyber-Physical Brewing Labs*, p. 171–178. Springer Nature Switzerland. 2024;. URL http://dx.doi.org/10.1007/978-3-031-65411-4_21

[24] Liu M, Fang S, Dong H, Xu C. A review of digital twin in product life cycle: Towards smart manufacturing. *Journal of Manufacturing Systems*. 2021;60:119–137.

[25] Roemer AC, Strassburger S. Hybrid System Modeling Approach for the Depiction of the Energy Consumption in Production Simulations. *2019 Winter Simulation Conference (WSC)*. 2019;pp. 1366–1377. URL https://api.semanticscholar.org/CorpusID:211243509

[26] Alcaraz C, Lopez J. Digital Twin: A Comprehensive Survey of Security Threats. *IEEE Communications Surveys and Tutorials*. 2022;24(3):1475–1503. URL https://doi.org/10.1109/COMST.2022.3171465

Container-Based Simulation: A Concept For Large-Scale Simulation Environments

Daniel Seufferth[1*], Falk Stefan Pappert[1], Heiderose Stein[1], Oliver Rose[1]

[1]Institute of applied computer sciences , University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany; [*]*daniel.seufferth@unibw.de*

Abstract. When simulation experiments grow in scale, a simulation environment providing appropriate computational resources is needed to produce results within a reasonable time. Containerization is a promising method for creating such a scalable environment. This method needs to be adapted to the simulation users' demands and should be easy to use for a project. Therefore, we provide a concept for a container-based environment, supporting various simulation projects, that dynamically scales and adapts the simulation workload to the available hardware. The concept encapsulates the environment technologies and provides an access service to the user for defining and editing simulation experiments and retrieving results. We discuss requirements for combining simulation and containerization to support the transition to such a container-based simulation environment. As a result, we see an opportunity to enhance large-scale simulation experiments using container methods and identify areas for further research.

Introduction

There are several fields of study in and adjacent to the broader simulation domain with a significant rise in demand for computational power. Prominent methods are Simulation Based Optimization (SBO), Machine Learning (ML), and data-farming.

SBO combines simulation models with optimization algorithms to find optimal or near-optimal solutions for problems [1]. Various approaches utilize SBO to assist in decision-making and finding ways to optimize real-world systems (e.g., see Lidberg et al. [2], Nikolopoulou and Ierapetritou [3], or Nguyen et al. [4]). As most optimization algorithms require multiple iterations over a system, increasing the number of simultaneously running versions of the model speeds up the optimization process significantly.

The importance of ML as a research field was shown in a recent study by Nature, which asked 1600 researchers about how ML and Artificial Intelligence (AI) in general will change in their future research. Among the main benefits are improvements to data processing and acquisition, and increased productivity [5]. Rai et al. [6] comprehensively review the importance and recent advances of ML for Industry 4.0 applications in manufacturing and production systems. A substantial amount of training data is necessary to create comprehensive ML-tools.

At the same time, the increasing complexity of simulation models and the number of simulation experiments challenge the simulation environment: As computational resources are constrained, the workload required for large-scale and complex simulations limits the feasibility of performing these experiments.

Efforts to speed up simulation experiments have been an active field of research since the 1970s [7]. The ongoing trend of cloud computing, virtualization, and containerization offers an exciting opportunity for creating a scalable simulation environment. The research in this field is diverse and growing. Król et al. [8] developed Scalarm, an infrastructure for distributing simulation workloads onto multiple computing nodes in heterogeneous environments. In more recent work, Anagnostou et al. [9] evaluated technological approaches with their work on simulation experimentation frameworks, applying a microservice-based auto-scaling approach utilizing MiCADO. MiCADO focuses on efficiently utilizing cloud resources. It extends Kubernetes' Application Programming Interface (API) objects with its Application Description Templates (see [10]), making it less universal.

Although these technologies have been around for some years, a significant uptick in adopting these methods is not yet visible in the simulation community, neither from the practitioner nor the software developer side. We attribute this to the perceived steep learning curve of using containers and container management in general. Especially packaging simulation workloads in containers and their convenient use is still an open issue. To approach this, we propose a container-based sim-

ulation architecture that supports different experiment setups, e.g., SBO, data-farming, or machine-learning-based approaches. Our goal is to provide a foundation for a heterogeneous environment that can dynamically support multiple projects to scale and adapt the distribution of computing resources.

This paper is structured as follows: First, we describe our concept for a container-based simulation environment from a simulationist's point of view. We briefly introduce the key technologies we need for this concept: containers and container orchestration. The following sections provide an overview of the key components of our concept. The third section discusses the requirements arising from combining simulation with containers in the context of large-scale experiments. Lastly, we summarize our research findings and give future research opportunities.

1 Introduction to containerization and container orchestration.

Container technology is a type of operating system virtualization, isolating a part from the underlying host Operating System (OS) for the processes of the software packaged in the container. Unlike a Virtual Machine (VM), containers utilize the OS and access the hardware of the host computer directly. This makes containers lightweight and flexible, as only the dependencies to run the containerized software are necessary. Thus, a container only brings its purpose-specific binaries and files and uses general libraries and binaries shared by the host OS. This is implemented by a software called container runtime, which enables containers to communicate with the host kernel and run processes, as stated by Hitchcock [11].

Containers are commonly created based on a container image. These are created using container engines, such as Docker, Podman, or Apptainer engines. All container engines follow a similar approach to image creation, utilizing a descriptive file that defines all steps necessary to containerize a software package. As a generalization, we call this file the containerization file in the following. Typically, a containerization file defines the base image and the more specific parts of the container built atop.

Containers are highly portable software packages, able to run on various hardware configurations, from multiple office Personal Computers (PCs) to full-fledged servers, as long as the underlying OS supports the specific container. Container orchestrators are used to manage large amounts of containers. Therefore, they are vital components for highly scalable simulation environments, where they handle large numbers of simultaneously running simulator instances.

There are several different orchestration software packages on the market, but one of the most prominent ones is Kubernetes, with which Google launches about 4 billion containers per week [12]. More information on containers and Kubernetes can be found in Huawei Technologies [13] and Poulton and Joglekar [14].

2 Container-based simulation environment

Modern software systems that need to be highly scalable are typically built using a service-based architecture, which is comprised of individually running services that offer functionality to both other services and the user. An approach to designing and scaling up these software systems is containerization, where each service runs within containers. Therefore, the number of container instances providing a service can be increased or decreased using container orchestration based on the service's demand and the system's current load. This approach efficiently addresses and uses large amounts of heterogeneous hardware and maximizes the utilization of existing hardware. Hence, enabling the dynamical adjustment of available computing resources to different simulation projects, even when these projects employ different simulation software packages.

Figure 1 provides an overview of the functional concept of the proposed container-based simulation environment and its main components. The *Design of Experiment Service* represents the core user strategy on what simulation settings need to be run; this can be an optimizer, a reinforcement learning agent, or another generator for the design of experiments. It generates scenarios and commissions the system with their evaluation. The *Scenario Manager* is responsible for managing the scenarios once they are in the system, scheduling and monitoring their execution. Within our architecture, we consider a scenario to be a complete data set representing a single instance of the modelled system. We recognize three types of scenario data depending on the viewpoint, with only the second one needed for a complete system understanding.

1. Abstract meta-data, stored in the *Scenario Status*

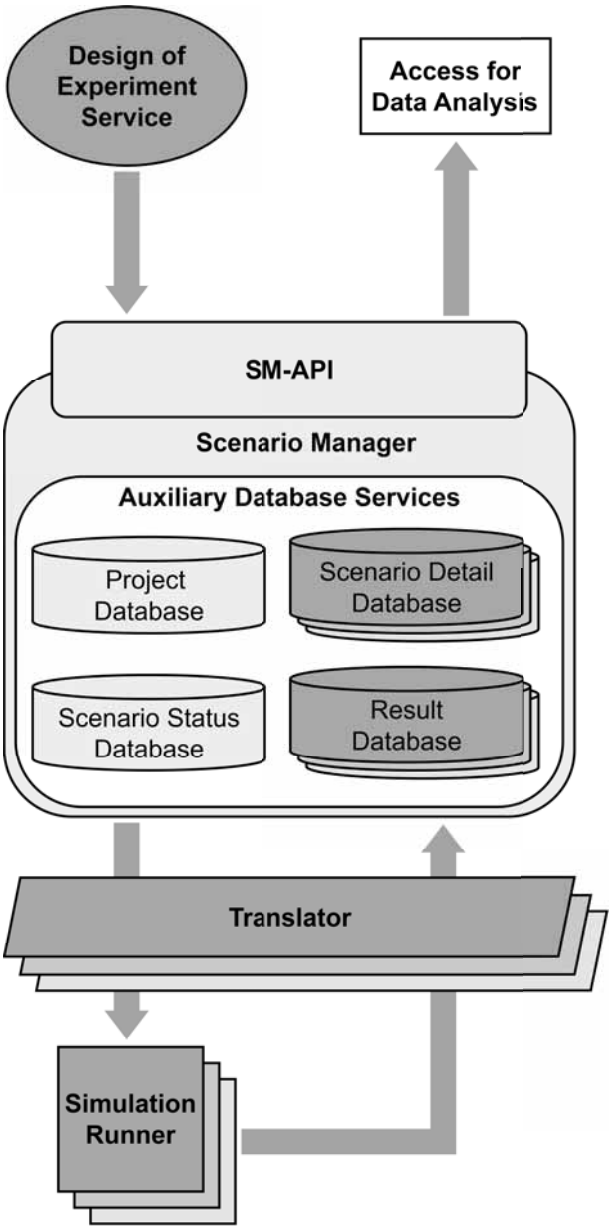


Figure 1: Concept of a container-based simulation environment, consisting of four main components: *Design of Experiment Service* (DoES), *Scenario Manager* (SM), *Translator* and *Simulation Runner* (SR).

Database, like a global scenario ID or the processing state of simulations desired for the scenario.

2. System description, stored in the *Scenario Detail Database*, provides all system information to create a valid model for the purpose.

3. Result data, stored in the *Result Database*, keeps records of all Key Performance Indicators (KPIs) and logs information desired from a scenario's simulation runs.

The *Translator* generates simulation-software-specific instances of a scenario and followingly makes the simulation executable by the simulator. These executable instances are run by the *Simulation Runners*, which report results to the *Scenario Manager* after execution.

Each of these components is implemented as a service within its container. This allows the available computing resources to be adjusted depending on the demand for these components and the system's general load. This setup, especially the *Simulation Runners*, enables the user to benefit from dynamically addressing large amounts of computing resources. Therefore, it allows for the adaptive provisioning of resources for different simulation projects using different software packages.

From our point of view, there are two general ways to implement our concept: focusing on project-specific components or aiming at reaching a universal system. In the universal approach, the semantic transformation from a project-specific to a universal model is done by the *Design of Experiment Service* (DoES). All scenarios handled and evaluated can, therefore, be stored in the same structure, allowing the *Scenario Database*, *Result Database*, and *Translator* to be universal as well. The downside of this approach is the need to transform and store scenarios, and therefore their model descriptions, in a generic format. This makes the creation of these three universal components very complex.

The more pragmatic approach is to design these components in a project-specific way, as indicated by the colouring scheme in Figure 1. This is, in our view, the more worthwhile option. However, this requires individually building database schemes and *Translators* for each project. The complexity of these components is significantly smaller, as a scenario entry only needs to hold a few key parameters, and the *Translator* can be specialized for the project. A further benefit of the project-specific approach is the improved possibilities for fine-tuning and adapting the model structure to the project-related domain.

2.1 Design of Experiment Service

As we mentioned earlier, the DoESs represent the scenario-creating elements of a simulation-based system: for example, the optimizer creating solution can-

didates and sending them off to be evaluated or the data-farming service implementing a given design of experiments. Therefore, the DoES represents the interface to the simulation environment and is necessary to enter simulation workloads. Via this interface, the setup of all scenarios is defined, which comprises the data required for generating the models, necessary computing restrictions, and other requirements.

2.2 Scenario Manager

The *Scenario Manager* (SM) is the management component of our concept and has various functions. Based on the project-specific simulation data provided by the DoES, it manages scenarios and coordinates their execution on the *Simulation Runner*s (SRs) using the *Translators* for model generation. Furthermore, it actively monitors the SRs, ensuring comprehensive oversight by consistently tracking the status of each scenario. Lastly, the SM also organizes the simulation results in databases and provides access to analyze the generated data.

The SM and its databases are implemented as containers. While the *Project Database* and *Scenario Status Databases* are universal, the *Scenario Detail Database* and *Scenario Result Databases* are project-specific deployments. The particular configuration of each project-specific database deployment depends on the scale of the experiment. Therefore, the configuration can range from one database container to complex deployments consisting of multiple database containers and their infrastructure, e.g., load balancers.

2.3 Translator

The *Translators* generate simulator-specific simulation models based on the scenarios passed to it by the SM. The data needed is pulled from the *Scenario Detail Database*. The specific implementation of the *Translator* highly depends on the respective project approach and the capabilities of the chosen modelling tool. A simple *Translator* would be used to forward parameters from the *Scenario Detail Database* to the SRs that already have a base model implemented. More sophisticated *Translators* may use the data from the *Scenario Detail Database* to automatically generate executable simulation models. The main technologies used to implement *Translators* come from the fields of model transformation and automated model generation. We refer to published research for the specific implementation of *Translators*. Thiers et al. [15] comprehensively introduce this research subject. They furthermore discuss the large variety of system description languages and propose a methodology for one system description language combined with a transformation step that loosely couples other languages to the back-end bridging abstraction language.

2.4 Simulation Runner

The SRs provide simulation execution as a service. These containers encapsulate the simulation engine and, when started, run the actual simulations. Different containers providing different simulation engines can be deployed, matching the projects run. After the scenarios are generated by the *Translator*, they get forwarded to the SR with the specified simulation engine required by the design.

The number of SRs is highly dynamic and is automatically scaled up and down based on the available computing resources. Figure 2 illustrates this breathing characteristic of the SRs. During idle time, only the template of a SR exists (1). When the *Translator* generates models that require an idle SR, the deployment increases the number of SRs to a size that fits the available hardware and the number of simultaneous instances defined for this project, visualized in (2) and (3). If demand decreases, the number of SRs decreases (4) and goes back to zero if this specific simulation engine is no longer needed.

3 Requirements to use containerization for simulation

To make simulation environments based on containers practicable for a user, some requirements must be met for simulation packages. Although containerization and running software as services from inside containers are generally not new concepts, they are rather unexplored in the simulation community. Therefore, it is necessary to illustrate how the specific use case of simulation is affected by containerization. We found that containerization, in combination with simulation, sets specific requirements that can be grouped into four general fields:

- Modeling
- Containerization

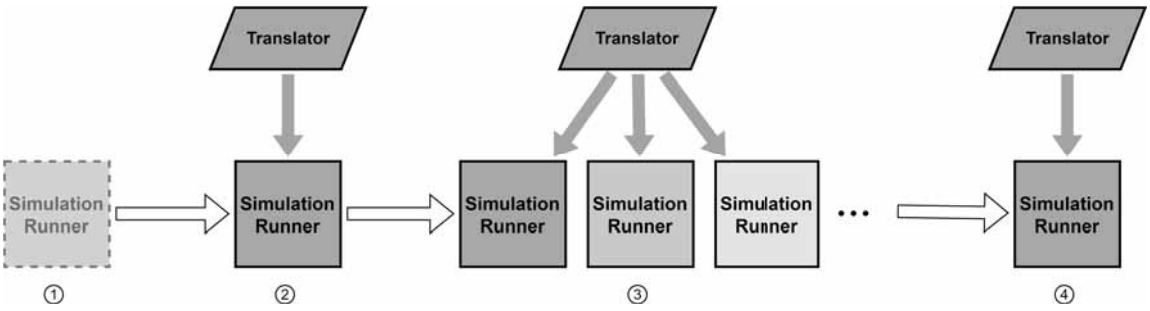


Figure 2: The *Simulation Runners* automatically scales up and down, based on the demand created by the scheduled scenarios of the *Scenario Manager*.

- Integration
- Licensing

We amend these requirements from [16] with the findings from our concept in the following.

3.1 Modeling requirements

The key modelling requirement on a simulation package for usage in large-scale systems is the model generation capability by external means. We found that simulation packages allow the automation of model building to different degrees, which we categorize into five levels:

1. No automation
2. Parameterization of a hand-made model
3. Bootstrapping models based on external data
4. External generation of model files
5. Online model generation using an API

In the proposed architecture for a container-based simulation environment, the *Translator* component handles automated model generation. To this end, some degree of external model generation needs to be supported by the simulation tool. Depending on the project, higher levels of model generation may be required, restricting the choice of simulators.

3.2 Containerization Requirements

The containerization requirements focus on what is needed to use simulation software efficiently in container-based, scalable environments. They are categorized into three topics: needs imposed by the OS, constraints due to the ephemeral nature of containerized applications, and requirements of the containerization process.

3.2.1 Operating System

An important requirement for the container-based simulation environment is providing the OS needed for the simulation software. As we mentioned in Section 1, containers are a form of OS virtualization, requiring a host OS that supports the container. The majority of currently used containers are Linux-based, requiring a host that runs Debian, RedHat, Ubuntu, or a similar OS. Consequently, most of the current container ecosystem, i.e., the applications and tools used for running containers in a scalable environment, is tuned for Linux containers.

In contrast, as shown in Table 1, a significant amount of software currently used by the simulation community for modelling and simulation is based on Windows. For simulation software that solely runs on Windows, Windows containers are required. Running Windows containers in orchestrator systems leads to hybrid environments, consisting of both Linux and Windows hosts, as the management layers of container orchestrators like Kubernetes require Linux hosts. Such hybrid environments can be considered less efficient than their Linux-only counterparts, as resources for both host systems need to be provided and managed. This results in worse utilization of the available hardware or increased efforts due to the need to dynamically adjust host system allocation depending on the observed demand. An ideal situation would be the availability of Linux options for all simulation software packages; as this is not the case, we see at least larger systems working on a hybrid basis for the foreseeable future.

Simulation tool	Supports Windows?	Supports Linux?	Supports macOS?
ANSYS	yes	yes	no
AnyLogic	yes	yes	yes
FlexSim	yes	no	no
MATLAB/Simulink	yes	yes	yes
OpenModelica	yes	yes	yes
Simio	yes	no	no

Table 1: List of simulation tools and the support of Linux-based OSs.

3.2.2 Ephemeral simulation containers

Containers may be killed at any time during their lifecycle. This can be caused by errors in their internal software or changes in maintenance or load control. This characteristic is called ephemeral. A system that depends on containers to provide functionality, therefore, needs to address this properly. The following are ideas to approach this problem:

Restarting failed simulation containers intuitively emerges as the simplest method to address this issue. Most container orchestrators are declarative, which means they try to achieve the desired state defined in the files posted to the API server of the orchestrator. Controllers continuously check the observed state of the cluster against the desired state and perform the tasks necessary to achieve the desired state if deviation is detected. To effectively utilize this feature of container orchestrators for large-scale simulation experiments, a controller that keeps track of the failed scenarios is essential. This functionality would be covered by the *Scenario Manager* in the previously proposed architecture.

If a simulation container is ended prematurely, the state of the model is lost, and the allocated resources for this process are wasted. Extending the restart of the containers by **keeping track and storing the progress of the simulation run** is a comprehensible next step. For this, continuous updates of the current simulation state, or at the least regular snapshots, must be kept in storage. This causes a significant amount of data to be transferred and stored. These resource-intensive tasks likely harm simulation speed: the theoretical performance gained through restarting the simulation container from a cached image of the last state will most likely be compromised by the processes necessary to create these images. From the authors' point of view, this approach is only feasible in an environment where successfully finishing simulation runs is otherwise unlikely, e.g., due to extensive simulation execution time or unstable computing hardware.

Most simulation experiments utilize replications to achieve sufficient confidence intervals. Therefore, **increasing the number of replications for each scenario** would be an efficient way to add redundancy and counteract the loss of simulation runs. Depending on stability, this method would not create a significant margin of load to the cluster, ensuring efficient utilization of the computing hardware available. However, increasing the number of replications for the scenarios does not ensure the successful execution of any scenario, as even a large number of replications could fail. Suppose the probability of failure of the container/model can be estimated. In that case, the required number of replicas needed to achieve a certain coverage can easily be calculated within the desired confidence.

3.2.3 Container creation

The last containerization requirement addresses constraints for containerizing simulation software.

To containerize a simulation model, it is necessary to either have an existing base image of your simulator or a base image that supports your simulator of choice. The latter option requires manually adding the simulation engine to the base image, which needs additional effort. This would usually be done with the help of a package manager or an installation method that does not require user interaction. If the simulation tool does not support these methods, a more tedious way of packaging the simulation engine must be utilized, e.g., emulating user inputs in the containerization file.

An optimized containerization process for simulation models would mean that simulation package developers and vendors provide usable base images of their

simulation software. With functioning base images, the SRs can easily be populated by the generated models from the *Translators*. Although some simulators already provide base images, e.g., MATLAB, and some container images for simulators created by the community, e.g. for SimPy, this is not the norm.

3.3 Integration requirements

The integration requirements describe how well a simulation package can be integrated into a container-based software system, e.g., how it can be started and executed and how results are managed. Containerized software usually runs headless, meaning that a Graphical User Interface (GUI) is typically unavailable during runtime. Therefore, in a containerized setup, it is strongly preferred to have a simulator that supports execution without a GUI. Suppose the simulation software is dependent on interactions with a graphical user interface. In that case, it is not well suited for a scaled-up environment as the manual interaction also needs to be scaled accordingly, which is typically not feasible.

Besides running, starting the simulation is also a concern for integration. Although an essential and everywhere available option in past days, starting the simulation run from the command line is no longer generally available. Similar to manual interactions with the model during runtime, using a GUI is also unfeasible for environments where many simulations are started in parallel. Although there are makeshift approaches, including the emulation of inputs from keyboard and mouse, that help automate these interactions, their implementation is typically less than ideal.

Most simulation execution environments used in simulation packages are not (yet) designed for distributed infrastructures. An experiment manager and an external way of triggering a simulation run are necessary for such systems. The most common way to do this is by using command-line interfaces. Starting simulations using an API is an even more convenient option.

Other important considerations are connectivity to databases and the possibilities to write results. If this is not possible, other options are required to parameterize the simulation model and gain access to results. At least the results must be exportable to an external file, which can then be evaluated and transferred by additional software in the container. This, of course, makes the container much more complex to create, but especially for older simulation packages, it may be the only available option.

3.4 Licensing Requirements

The last requirement for combining container technologies and simulation we see is the licensing of simulation software. Software vendors have different approaches to licensing their products, which may impact the cost structure of using this software in a containerized, highly scalable environment. Simulation packages, where costs only depend on the model development environment, allow very flexible scaling of the simulation to new or different projects. Licensing on a per-core/seat/user basis is also suitable for a containerized computing infrastructure but can get quite costly if sufficient hardware is available. Moreover, the licensing agreement should be thoroughly reviewed to determine whether the use in container environments is allowed.

Besides cost and permissibility, another noteworthy side of licensing is its technical enforcement. For a dynamic environment, concurrent licensing via licensing servers is ideal. In contrast, licensing schemes enforced by hardware restrictions put significant limitations on a containerized simulation platform. Examples of this approach would be licensing codes tailored to a specific computer or hardware keys provided as USB dongles.

4 Conclusion and further research

This paper proposes a high-level concept for a container-based simulation environment primed to meet the growing demand for large-scale simulation experiments. We introduced the four key components of our concept, consisting of the *Design of Experiment Service* (DoES) as a central service that allows access to the environment and defines all specifics of an experiment setup, the *Scenario Manager* (SM) as a management service for handling large numbers of scenarios, the *Translator* as a translation interface that generates simulation models from different description languages, and the *Simulation Runner*s (SRs) that provide containerized simulators which can be dynamically scaled up and down based on demand.

To support the transition to such a container-based simulation environment, we investigated what requirements arise from the combination of simulation and containerization. The described requirements cover various fields and must be considered when container-

izing simulation models. General modelling requirements describe the importance of populating the simulation containers with models.

The next step of our research will be to implement this concept, as we see a need to enhance large-scale simulation experiments and use container methods. This includes containerizing and testing different simulation software packages and assessing to what extent the simulation environment can support their dynamic management. Another future step is to evaluate the environment on different hardware setups. Other exciting areas of future research may look at different concepts, for example, a more universal DoES or strategies targeted to a specific hardware configuration.

Acknowledgement

We want to thank Uwe Langer and Alexandros Karagkasidis for their continuous support of the hardware infrastructure.

This research is funded by dtec.bw - Center for Digitization and Technology Research of the Bundeswehr. dtec.bw is funded by the European Union – NextGenerationEU.

References

[1] Fowler J, Sawah SE, Turan HH. Recent Advances in Simulation-Based Optimization for Operations Research Problems. *Annals of Operations Research*. 2023;320(2):545–546.

[2] Lidberg S, Aslam T, Pehrsson L, Ng AHC. Optimizing Real-World Factory Flows Using Aggregated Discrete Event Simulation Modelling. *Flexible Services and Manufacturing Journal*. 2020;32(4):888–912.

[3] Nikolopoulou A, Ierapetritou MG. Hybrid Simulation Based Optimization Approach for Supply Chain Management. *Computers & Chemical Engineering*. 2012;47:183–193.

[4] Nguyen AT, Reiter S, Rigo P. A Review on Simulation-Based Optimization Methods Applied to Building Performance Analysis. *Applied Energy*. 2014; 113:1043–1058.

[5] Van Noorden R, Perkel JM. AI and Science: What 1,600 Researchers Think. *Nature*. 2023; 621(7980):672–675.

[6] Rai R, Tiwari MK, Ivanov D, Dolgui A. Machine Learning in Manufacturing and Industry 4.0 Applications. *International Journal of Production Research*. 2021;59(16):4773–4778.

[7] Taylor SJ. Distributed Simulation: State-of-the-Art and Potential for Operational Research. *European Journal of Operational Research*. 2019;273(1):1–19.

[8] Król D, Wrzeszcz M, Kryza B, Dutka Ł, Kitowski J. Massively Scalable Platform for Data Farming Supporting Heterogeneous Infrastructure. In: *The 4th International Conference on Cloud Computing, Grids, and Virtualization: (Cloud Computing 2013) ; Valencia, Spain, 27 May - 1 June 2013 ; Held at ComputationWorld 2013*, edited by Zimmermann W, pp. 144–149. Wilmington: IARIA. 2013;.

[9] Anagnostou A, Taylor SJE, Abubakar NT, Kiss T, DesLauriers J, Gesmier G, Terstyanszky G, Kacsuk P, Kovacs J. Towards a Deadline-Based Simulation Experimentation Framework Using Micro-Services Auto-Scaling Approach. In: *2019 Winter Simulation Conference (WSC)*, edited by Navonil Mustafee, Ki-Hwan G Bae, Sanja Lazarova-Molnar, Markus Rabe, C Szabo, Peter Haas, and Young-Jun Son. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc. 2019; pp. 2749–2758.

[10] Project M. Application Description Template - MiCADO. https://micado-scale.github.io/adt/. 2023.

[11] Hitchcock K. Containers. In: *Linux System Administration for the 2020s*, pp. 155–200. Berkeley, CA: Apress. 2022;.

[12] Google. Why Choose GKE as Your Kubernetes Service. https://cloud.google.com/blog/products/containers-kubernetes/why-choose-gke-as-your-kubernetes-service. 2023.

[13] Huawei Technologies Co Ltd n. Container Technology. In: *Cloud Computing Technology*, pp. 295–342. Springer, Singapore. 2023;.

[14] Poulton N, Joglekar P. *The Kubernetes Book*. [United Kingdom]: Nigel Poulton, january 2022 ed. 2022.

[15] Thiers G, Sprock T, McGinnis L, Graunke A, Christian M. Automated Production System Simulations Using Commercial Off-the-Shelf Simulation Tools. In: *2016 Winter Simulation Conference (WSC)*. 2016; pp. 1036–1047.

[16] Seufferth D, Stein H, Pappert F, Rose O. On the Usage of Container and Container Orchestrators as a Computational Infrastructure for Simulation Experiments. *20 ASIM Fachtagung Simulation in Produktion und Logistik 2023*. 2023;pp. 393–401.

Towards Imaginative Robots: A Generative Pipeline for Simulated Environments

Christopher May[1*], Lorenz Suchy[1], Jörg Franke[1], Sebastian Reitelshöfer[1]

[1]Institute for Factory Automation and Production Systems (FAPS), Friedrich-Alexander-Universität Erlangen – Nürnberg, Egerlandstr. 7-9, 91058 Erlangen, Germany; *christopher.may@faps.fau.de*

Abstract. Autonomous Mobile Systems (AMS) offer significant advantages in industry and private sectors by adapting to diverse and dynamic environments. To train these systems, large amounts of data are required, typically obtained from simulated environments. However, the creation of these environments is often labor-intensive. Here, we propose a generative pipeline that provides a streamlined approach to virtual training and testing while allowing users to apply automated methods including generative AI. Our pipeline consists of four, partly iterative main steps. The pipeline spans from the creation of individual assets to the utilization of the simulated environments. The pipeline is then implemented for an exemplary scenario, utilizing multiple methods including generative AI. Furthermore, we propose a novel application of our pipeline to provide robots with the capabilities to "imagine" virtual experiences. The presented pipeline not only simplifies the process of generating simulated environments, but also resembles a scalable framework for developing increasingly complex AMS.

Introduction

Mobile robots and specifically Autonomous Mobile Systems (AMS) are changing the world. While transport robots are already well-established in industry, they have not yet reached their peak. In the coming years companies will expand their fleets and applications with new systems, increasingly powered by AI. [1]

Developing and training AI models requires massive amounts of data to ensure performance in the demanding, large, dynamic, and diverse operating environments of AMS. One solution to reduce the effort associated with collection and annotation of the required data is simulation. In simulated environments, the possibilities to generate synthetic data are virtually unlimited. However, generating data for all kinds of imaginable scenarios, is still related with large human efforts. Recent breakthroughs in generative AI could enable developers of AMS to reduce the needed effort while improving the quality of synthetic data from robotics simulation.

Most existing work on generating simulated robot operating environments focuses on reinforcement learning in small manipulation scenarios [2, 3]. Those do not lie within the scope of our work. One notable exception has been presented by Bonetto et al. Their approach focuses on "Generating Realistic Animated Dynamic Environments for Robotics Research", abbreviated "GRADE" [2]. GRADE requires an existing set of assets. Bonetto et al. have, amongst others proven that synthetic data from simulated environments can be sufficient to train and validate vision-based robots [3, 4]. Another related approach that utilizes generative AI has been presented in the position paper "Towards Generalist Robots: A Promising Paradigm via Generative Simulation" [5]. Xian et al. define the term "generative simulation". Their concept is supposed to generate scenes with accompanying robot-tasks and at the same time include training supervision. While the authors discuss multiple ideas and claim to be able to generate infinite data for various robots in diverse environments, at the time of writing this paper, the work of Xian et al. mainly remains a literature review without actual implementation [6]. Their related work "RoboGen" [7] focuses on motion planning for stationary robots.

In this paper, we first identify methods that are relevant in the context of simulated environments for AMS. We then present a generative pipeline for the creation of simulated environments for AMS. The pipeline consists of four main steps, which are partially iterative. In this modular approach, different methods can be employed in different steps of the pipeline. This applies to both conventional methods and generative AI-methods. We furthermore present an exemplary implementation of our pipeline that utilizes several of the methods discussed to create simulated environments, fully populated with AI-

generated assets. Finally, we introduce the concept of ‘imaginative robots’ and propose the application of our pipeline to enable robots to prepare for new and unknown situations autonomously.

1 Methods for the Creation of Simulated Environments

Before defining a pipeline for creating simulated environments, it is important to clarify the relevant methods. We identify four general methods that are relevant to simulated operating environments. These methods can incorporate existing models, databases, etc. Although these methods can be used in conjunction with each other, for the purpose of this discussion, we treat them as isolated from one another. We limit ourselves to a rather general evaluation, which is intended to provide general guidance. The presented methods may yield different results when specific approaches are evaluated. In this paper we focus on static, unarticulated environments. Customizability of assets and environments is still a relevant aspect for specific scenarios and with articulated models in mind for future work.

1.1 Manual Methods

The first and most obvious class of methods is manual methods. This classification includes all approaches where substantial work is done manually. Although manual methods can utilize tools, they do not involve automation. Users have control and may modify every aspect of their workpiece to fit within the requirements, as long as it is supported by the tools utilized.

While manual methods can produce high-quality handcrafted results, the trade-off is that they are largely time consuming. Therefore, they are not suitable for large-scale simulated environments.

1.2 Automated Reconstruction Methods

Due to the time-insensitivity of manual methods, the application of automated methods is attractive. A class of automated methods are methods for automated reconstruction. They are proven to be suitable for efficiently reconstructing larger scale outdoor but also indoor environments. [7]

Automated reconstruction approaches are often implemented as photogrammetric methods based on RGB data, but might also incorporate depth data. The gathered data is then combined into photorealistic 3D models that accurately represent their real-world counterpart. [7, 8]

A significant disadvantage of automated reconstruction methods is limited modifiability of the generated models. This hinders the application of photogrammetric methods in the context of generating new data for training and validation of AMS. Possible applications include the reconstruction of individual assets or the reconstruction of empty “base” environments that can be populated later on.

1.3 Procedural Methods

Methods for automated reconstruction cannot create new environments and therefore might be helpful in some aspects, but not to tackle the core problem of new and diverse data. Manual methods can build upon human imagination to create new content - however strongly impeded by the necessary manual labor. Hence, we will now shift towards methods that are able to create entirely new assets and environments with minimal human intervention.

Procedural methods generate content algorithmically within predefined constraints, without the need for manual input after an initial setup. These methods can produce a vast amount of diverse and complex data automatically, both in a deterministic manner but also by incorporating random elements. The absence of a manual input apart from the initial setup is a core feature of those procedural methods.

Procedural methods are well established in computer games, where they are used to generate expansive virtual worlds, such as in commonly known Minecraft. They also find application in robotics simulation: NVIDIA Omniverse includes a "Domain Randomizer", able to alter multiple parameters of a simulated scene randomly [9]. Further procedural approaches in robotics simulation include Cropcraft [10] for generating simulated crop fields or the already mentioned GRADE [2]. [11, 12]

1.4 Generative AI-based Methods

The next class of relevant methods is based on generative AI. Similar to procedural methods, generative AI-methods are able to computationally generate new content. Unlike procedural methods, they are generally not constrained to algorithmically pre-defined content. There are several popular approaches to implementing generative AI, such as Generative Adversarial Networks (GANs), Variational Autoencoder (VAEs) or Transformer Models [13–15]. The latter might be the most publicly known type of model for being the basis of LLMs like ChatGPT.

Another relevant approach involves diffusion models. Diffusion models start with random noise and iteratively refine it into a detailed output, guided by a prompt. Inspired by the physical diffusion process, these models reverse noise addition, leveraging conditioning information – like the provided prompt – to shape the noisy base towards the desired content. This approach enables the generation of high-quality outputs.[16, 17]

A further notable approach are Neural Radiance Fields (NeRFs). NeRFs synthesize 3D scenes from 2D images by using deep neural networks to gain a volumetric representation of a scene. They are able to generate high quality scenes, but at the cost of computational inefficiency. [18]

1.5 Summary of Relevant Methods

All of the methods discussed in this chapter are relevant and usable for creating simulated environments. However, each of them has specific advantages and disadvantages. Users have to choose a fitting method based on their specific needs. To summarize the findings of this chapter and to ease the decision-making process, Table 1 provides a generalized comparison of the methods mentioned. They are compared in five categories and rated from -- (worst) up to ++ (best):

- Human Effort involved, less is better
- Quality of results assets
- Customizability of assets for specific requirements, e.g. rigged objects
- Hardware requirements imposed by the method; lower requirements are rated better
- Originality, meaning the capability to generate new content

	Manual	Reconstruction	Procedural	Gen-AI
Effort	--	0	+	++
Quality	++	+	+	-
Customizability	++	-	-	0
Hardware requirements	0	-	-	--
Originality	++	--	0	+

Table 1: The four discussed methods for creating simulated environments are compared in regards to effort, quality, customizability, hardware requirements and originality.

2 Introduction of the Generative Pipeline

In the following we introduce a pipeline which enables its users to create, compose and harness simulated environments. All methods compared in the previous chapter can be applied throughout the pipeline. They may also be combined and different approaches might be used in different steps. The pipeline shown in Figure 1 consists of four steps, which are explained in a generic manner in this chapter. An exemplary implementation is discussed in the following chapter 4.

Figure 1: The proposed pipeline for the generation of simulated environments consists of four steps.

The foundation of every virtual environment are its individual components, which we refer to as assets. Hence the first step of the pipeline is the “Creation” step, where assets are generated. Those are 3D models of individual items, e.g., a machine or a table. They should be stored in a standardized and widely compatible format to ensure future usability.

The assets created in step one need to be classified and rated. This is done in step two, “Classification and Rating”. Depending on the method applied for creation of the assets, this step varies in complexity. The goal is to obtain a database of assets, classified at least by type and quality. An extensive, high quality model database is crucial for a successful implementation of later steps. Users might also incorporate existing and purchasable sets, needing to keep in mind the reduced control over the assets.

Building upon the assets created and classified in the previous steps, we can proceed to the third step of “Composition”. Here the simulated environments are composed from the models in the asset database. This step can vary greatly in complexity, depending on the size and complexity of the desired operating environment of the AMS in question.

The fourth step represents the application or actual use of the simulated environment and does not lie within the scope of our work. Typical applications include the generation of synthetic data, validation of the AMS software or reinforcement learning [3, 19].

Notably, the pipeline shown in Figure 1 does not end

here. Instead, an iterative process is started after the application step: The pipeline returns to the environment composition step. Here, a new simulated environment is created and then used for the desired application. This can be done over and over again. Compared to existing domain randomization approaches in robotics simulators, an entirely new environment can be created with minimal effort. The application can thus benefit from experiences in diverse and virtually unlimited environments. This is a core component of our approach and allows users to take full advantage of the work done in the first two steps.

3 Exemplary Implementation of the generative Pipeline

For the validation of the proposed pipeline, we chose a scenario of practical use for ourselves: An electronics production environment, which is to be used for the validation of an autonomous tow truck. In the following, we present an exemplary implementation of the pipeline using various methods.

We chose to focus the application of generative AI on the first step of the pipeline. The second step is conducted manually due to the nature of the results from the previous step. For step three we present and apply a highly adaptable procedural approach. In this publication the fourth step is limited to a qualitative evaluation of exemplary generated environments.

For implementation we chose – independently from [2] – to use the .usd-format and NVIDIA Isaac Sim (NIS) as simulation software. NIS offers significant benefits in regards to graphics and thus evaluation of vision-based algorithms over the established Gazebo simulator [2, 20].

3.1 Creation of Assets through Generative AI

In the first step of asset creation, we apply generative AI. After applying multiple AI-models and optimizing their settings, we settled on using MV Dream and Magic3D [21, 22]. Both were used through the threestudio framework [23].

The used models use two vastly different approaches. Magic3D is based on a text to image model with a huge training dataset. MV Dream is trained on a 3D-model database. This approach delivers results of higher quality, but for a smaller range of prompts. For very specific prompts like “a pick and place machine” Magic3D is the more promising approach. We also noticed that MV Dream delivers more consistent results than Magic3D. With the goal in mind of generating models that are as diverse as possible, Magic3D appears to be the better solution. Therefore, depending on the assets to be generated, one has to find a trade-off between higher quality or diverse assets. Generally, both approaches are able to generate 3D-models in usable quality as Figure 2 illustrates. The left section of the figure displays textured renderings, while the right section represents the normals of the meshes.

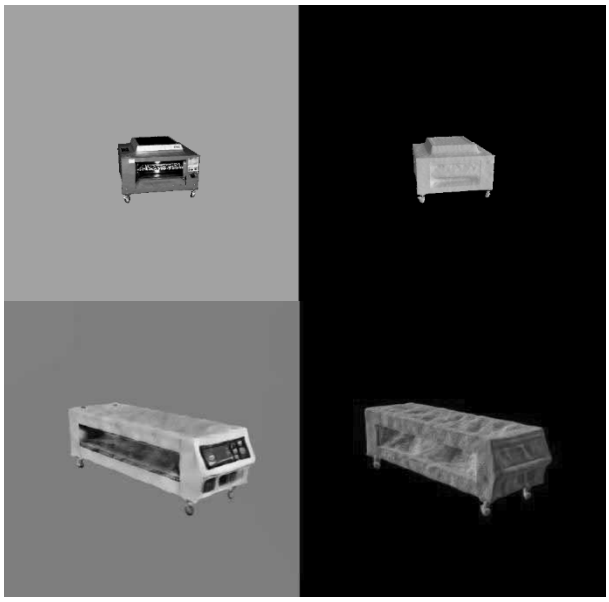


Figure 2: Both 3D models depicted are generated with the prompt “Industrial Reflow Oven”. The upper oven is created by Magic3D, the lower one by MVDream.

To ease the creation of a large number of assets, we use a script that automatically launches the AI model using a list of pre-defined prompts. The importance of using the right prompt when generating an asset is even more important than in 2D use cases. A prompt like “a pencil” likely won’t yield a usable result. A more promising prompt would be “an upright standing pencil”.

3.2 Manual Classification and Rating of Components

Due to the high hardware requirements of the AI models used in the first step, we were only able to generate a limited number of 300 assets over the course of multiple months. This low number of assets allows us to conduct the second step of the pipeline manually. It is simplified by the fact, that no classification of assets is needed due to the known prompts used for their creation.

However, the quality of the generated assets varies vastly, even within models generated with the same

prompt. The models are categorized into three different categories. “Good” are all useable models, “bad” are models where the mesh or texture have significant problems and “failed” for assets where the AI completely failed. Around 30% of the models are rated “good” and thus deemed usable. The models generated in the first step and rated “good” in this step form the basis for the next step of environment composition. Figure 3 shows a comparison of two models rated “bad” and “good”, created with the same prompt. Additional work is necessary for AI generated assets, since the AI-models we use are not aware of absolute scales. We thus have to scale and rotate the generated assets manually.

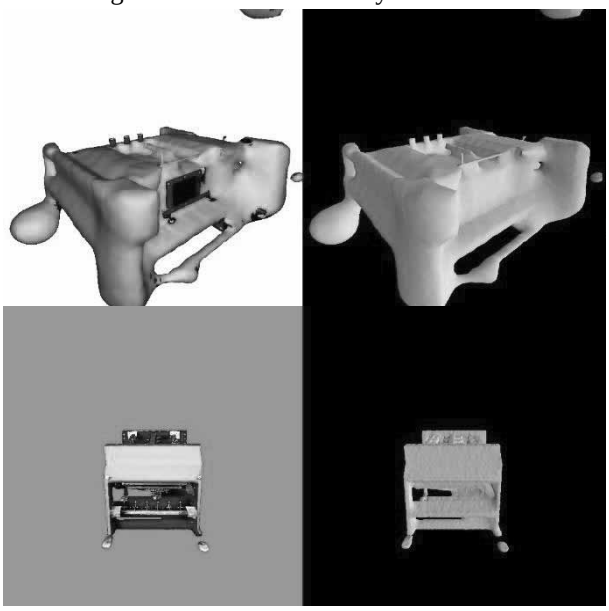


Figure 3: Even with the same prompt, the resulting assets can vary greatly in quality, as illustrated in this comparison of results from Magic3D with the prompt “Pick and Place Machine”. The upper model is rated as “bad”, the lower one as “good”.

3.3 Procedural Environment Composition

For environment composition, we present a procedural approach that uses environment subdivision and provides interfaces to the methods outlined above through a modular approach. For our implementation, we rely solely on our AI-generated model database. The environment composition can be divided into three substeps which are displayed in order in Figure 4.

Figure 4: The environment composition step can be broken down into the three substeps of layout generation, definition of bounding spaces and asset placement.

During layout generation, the available space is defined. A randomly sized rectangle is defined as the base for the layout. Next, the generated space is subdivided – also randomly – into the available classes of space. For our implementation, those are:

- Office space
- Storage space
- Production space

The latter is further divided into multiple production lines, depending on the size of the plant. An exemplary result of this process is shown in Figure 5.

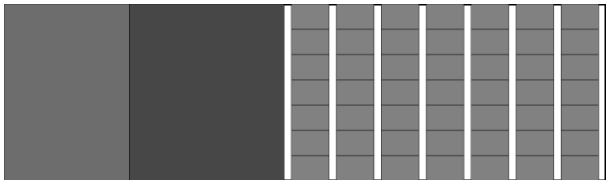


Figure 5: This exemplary procedurally generated floor layout consists of an office space (green), storage space (blue) and multiple production lines (red).

Subsequently, the defined spaces are further partitioned into bounding spaces. They are defined by their size, position and subtype. An iterative algorithm divides the spaces defined by the layout into smaller rectangular bounding spaces. Their size is chosen randomly within pre-defined bounds that are dependent on the class of the space. An exception is made for the production lines: To achieve a more realistic, uniform layout, their size is only generated once for each layout and thus identical.

Each bounding space is then equipped with a procedurally generated group of assets. For this purpose, a subfunction is called for each bounding space. This function generates a fitting group of assets within the given space. In our implementation, the function is defined among others for workplaces, storage racks, and production lines. An exemplary, randomly generated production line is shown in Figure 6.

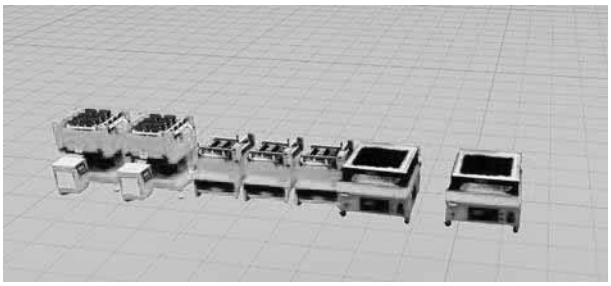


Figure 6: The depicted exemplary production line composed within step three consists of three different AI-generated machines, which are used two or three times.

For the placement of the production lines in our environments a modification has been made: While the sub-function generally generates a new group of assets for each bounding space, this is not fitting for the production lines. In practice, a production plant often operates several identical production lines. Therefore, a number of types of production lines is randomly chosen after space partitioning. The different lines – like the one in Figure 6 – are stored separately from the main .usd file. Instead of generating a new production line for each defined space, one is then randomly chosen from the pre-generated lines and placed within the available space including a randomized offset. By adding the modifications for the production lines to our implementation, we are both able to generate random environments, but also to obtain areas where a specific structure is necessary.

3.4 Assessment of Generated Environments

In this paper we restrict the application step to a qualitative assessment of environments generated by the pipeline. An advanced application is discussed in chapter 4.2. Figure 7, Figure 8, and Figure 9 represent examples of each kind of area defined in our implementation.

From the exemplary screenshots we can conclude that the presented pipeline and its implementation are suitable for the generation of simulated environments for AMS. The generated environments do not yet reach the same level of detail as handcrafted simulated environments. However, while composing an environment by hand would take hours or days, our pipeline is capable of composing environments in a few minutes, running on a standard desktop computer. We expect that advances in generative AI and further improvements to the pipeline will make it possible to generate environments and their assets with higher quality and more resource efficient in the near future.

Figure 7: This screenshot from an environment generated by our implementation of the pipeline depicts an office area composed with AI-generated workplaces. There are multiple different desks present, picked randomly from the asset database.

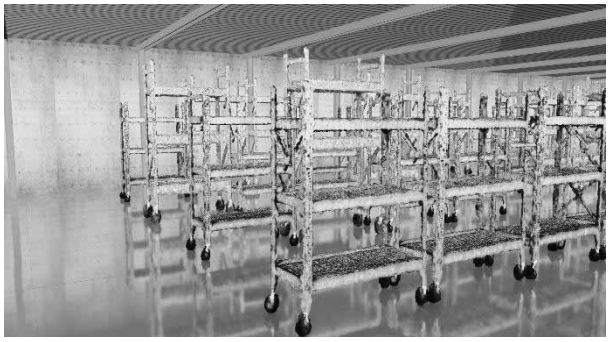


Figure 8: This screenshot from an environment generated by our implementation of the pipeline depicts a storage area with AI-generated storage racks.

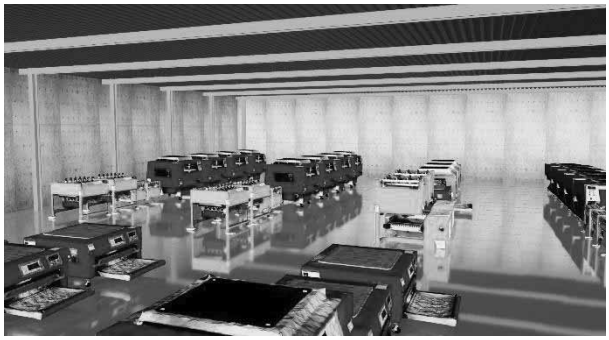


Figure 9: This screenshot from an environment generated by our implementation of the pipeline shows a production area consisting of multiple production lines with AI-generated machines. The lines on the left are identical and have been procedurally composed within step three.

4 Discussion and Outlook

In the following chapter the insights from this paper are discussed. We also give an outlook on a novel application of the pipeline.

4.1 Discussion

The validation of the generative pipeline presented in this paper underscores the pivotal role generative AI plays in the future development of AMS. With AI advancing at unprecedented speeds, this structured and modular approach is vital for future applications and allows users to update their pipeline as new and more advanced solutions emerge. Our approach has been demonstrated to be able to generate diverse and virtually unlimited environments with minimal human input. It is not yet capable of completely replacing human experts. Nevertheless, the pipeline offers a scalable solution to the data generation challenges discussed in the introduction.

Current key limitations of the pipeline include the generation of strictly rectangular layouts with continuous space classifications and the necessity for manual coding of asset placement functions. These constraints hinder the diversity and realism of the generated environments. Furthermore, manual evaluation of the assets in step two will not be a sustainable approach going forward. Another hindrance for large scale implementation is the computing power necessary. The employed generative AI models required around 40 GB of VRAM and took two to three hours per asset generated on a NVIDIA RTX 6000 ADA graphics card. At the state of the art, AI-generation of assets thus imposes significant costs for hardware acquisition and operation.

4.2 Imaginative Robots

Building upon the capabilities of the presented pipeline, we propose a novel application that could substantially improve the adaptability of AMS: Enabling robots to ‘imagine’. AMS could evaluate past experiences to identify potentials for improvement or to prepare for new tasks. The concept leverages the pipeline to enable robots to autonomously generate new, imagined experiences derived from past experiences or other inputs. Imaginative robots are able to generate and train on synthetic data tailored to unfamiliar environments, significantly enhancing their problem-solving capabilities and adaptability. This is key to advancing the flexibility and autonomy of AMS, enabling them to operate effectively in novel and unpredictable situations.

We believe that our pipeline holds promise not only for this imaginative approach but also for improving more established methods such as reinforcement learning.

5 Conclusion and Future Work

In this paper, we introduced a pipeline designed for generating simulated environments for AMS. This pipeline covers the entire spectrum from the creation of individual assets to the generation of complete simulated environments. It enables the rapid generation of large amounts of synthetic data, which is invaluable for robot training and validation.

Special attention was paid to advances in generative AI, which offer significant improvements over traditional methods. To validate our proposed pipeline, we implemented it and successfully generated a wide range of electronics manufacturing environments, populated by AI-generated assets. In addition, we introduced an innovative concept aimed at creating “imaginative” robots.

To exploit the full potential of our pipeline, we anticipate further developments in generative AI, which is advancing at a remarkable pace. Our ongoing efforts will focus on integrating newer AI models to improve the quality and efficiency of asset creation. An example could be LATTE3D, which has been released just at the time of writing this paper [24]. Additionally, we foresee the application of generative AI at various stages of the pipeline, including asset evaluation and layout generation, thereby broadening the range of scenarios our pipeline can address.

Building on these advancements, we aim to fully realize the concept of imaginative robots. Currently, this is achievable to some extent, but as our pipeline evolves to generate new assets and types of environments on the go, its full potential will be unlocked. Until then, the use of existing assets and predefined environment classifications provides a sufficient interim solution.

Acknowledgement

This publication was written as part of the research project “POV.OS – Hardware and Software Platform for Mobile Machinery”, funded by the German Federal Ministry for Economic Affairs and Climate Action on the basis of a resolution of the German Bundestag.

References

[1] Gartner. "Gartner Hype Cycle Shows Supply Chain Adoption of Mobile Robots Will Far Outpace Drones Over Next Three Years." Accessed: Mar. 25, 2024. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2023-08-17-gartner-hype-cycle-shows-supply-chain-adoption-of-mobile-robots-will-far-outpace-drones-over-next-three-years

[2] E. Bonetto, C. Xu, and A. Ahmad, "GRADE: Generating Realistic Animated Dynamic Environments for Robotics Research," 2023, doi: 10.48550/arXiv.2303.04466.

[3] E. Bonetto, C. Xu, and A. Ahmad, "Learning from synthetic data generated with GRADE," 2023, doi: 10.48550/arXiv.2305.04282.

[4] M. Metzner *et al.*, "Virtual training and commissioning of industrial bin picking systems using synthetic sensor data and simulation," *International Journal of Computer Integrated Manufacturing*, vol. 35, 4-5, pp. 483–492, 2022, doi: 10.1080/0951192X.2021.2004618.

[5] Z. Xian, T. Gervet, Z. Xu, Y.-L. Qiao, T.-H. Wang, and Y. Wang, "Towards Generalist Robots: A Promising Paradigm via Generative Simulation," May. 2023. [Online]. Available: http://arxiv.org/pdf/2305.10455v3

[6] Z. Xian and I. E. Ashimine. "Genesis-Embodied-AI/Genesis: A generative world for general-purpose robotics & embodied AI learning." Accessed: Mar. 4, 2024. [Online]. Available: https://github.com/Genesis-Embodied-AI/Genesis

[7] M. Lieret, V. Kogan, C. Hofmann, and J. Franke, "Automated Exploration, Capture And Photogrammetric Reconstruction Of Interiors Using An Autonomous Unmanned Aircraft," in *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, Takamatsu, Japan, 2021, pp. 301–306, doi: 10.1109/ICMA52036.2021.9512707.

[8] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015): Boston, Massachusetts, USA, 7-12 June 2015*, Boston, MA, USA, 2015, pp. 5556–5565, doi: 10.1109/CVPR.2015.7299195.

[9] NVIDIA. "Domain Randomization for RL." Accessed: Mar. 11, 2024. [Online]. Available: https://docs.omniverse.nvidia.com/isaacsim/latest/isaac_gym_tutorials/tutorial_gym_domain_randomization.html

[10] GitHub. "Romea/cropcraft: A Procedural World Generator for Robotics Simulation of Agricultural Tasks." Accessed: Mar. 11, 2024. [Online]. Available: https://github.com/Romea/cropcraft

[11] N. Brewer, "Computerized Dungeons and Randomly Generated Worlds: From Rogue to Minecraft [Scanning Our Past]," *Proc. IEEE*, vol. 105, no. 5, pp. 970–977, 2017. doi: 10.1109/JPROC.2017.2684358. [Online]. Available: https://ieeexplore.ieee.org/ielx7/5/7906639/07906675.pdf?tp=&arnumber=7906675&isnumber=7906639&ref=

[12] J. Togelius, N. Shaker, and M. J. Nelson, "Introduction," in *Procedural Content Generation in Games* (Computational synthesis and creative systems), N. Shaker, J. Togelius, and M. J. Nelson, Eds., Cham: Springer International Publishing, 2016, pp. 1–15.

[13] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," Jun. 2014. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/pdf/1406.2661.pdf

[14] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 2013. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/pdf/1312.6114

[15] A. Vaswani *et al.*, "Attention Is All You Need," Jun. 2017. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/pdf/1706.03762

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," Jun. 2020. [Online]. Available: http://arxiv.org/pdf/2006.11239

[17] C. Saharia *et al.*, "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding," May. 2022. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/pdf/2205.11487

[18] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," Mar. 2020. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/pdf/2003.08934

[19] B. Osinski *et al.*, "Simulation-Based Reinforcement Learning for Real-World Autonomous Driving," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 6411–6418, doi: 10.1109/ICRA40945.2020.9196730.

[20] M. Zwingel, C. May, M. Kalenberg, and J. Franke, "Robotics simulation – A comparison of two state-of-the-art solutions," in *ASIM SST 2022 Proceedings Langbeiträge*, F. Breitenecker, C. Deatcu, U. Durak, A. Körner, and T. Pawletta, Eds., Jul. 2022, pp. 171–178, doi: 10.11128/arep.20.a2033.

[21] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang, "MVDream: Multi-view Diffusion for 3D Generation," Aug. 2023. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/pdf/2308.16512.pdf

[22] C.-H. Lin *et al.*, "Magic3D: High-Resolution Text-to-3D Content Creation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] Y.-C. Guo *et al.* "threestudio: A unified framework for 3D content generation." Accessed: Mar. 28, 2024. [Online]. Available: https://github.com/threestudio-project/threestudio

[24] K. Xie *et al.*, "LATTE3D: Large-scale Amortized Text-To-Enhanced3D Synthesis," Mar. 2024. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/pdf/2403.15385

Stochastic Resource-Constrained Project Scheduling Problems with Continuous Random Variables

Rico Zöllner*, Mathias Kühn**, Konrad Handrich, Thorsten Schmidt

TU Dresden, Institute of Material Handling and Industrial Engineering, Dresden, Germany,
01062 Dresden; *rico.zoellner@tu-dresden.de, **mathias.kuehn@tu-dresden.de

Abstract. We present a simulation and optimization framework for stochastic resource-constrained project scheduling problems. The stochastic features are the job durations modeled by continuous random variables (without any time discretization) and the fluctuations are simulated by testing a sufficiently large number of realizations of an instance. The aim is to gain insights in the dependencies between the fluctuations of the input parameters and the objective function to enable a priori estimations. Such estimation methods developed by simulating small instances could be extrapolated to problems with a larger number of jobs or with more complicated features.

Introduction

Resource-constrained project scheduling problems (RCPSP) enjoy a widely spread relevance. From the theoretical perspective, they serve as research objects for combinatorial optimization, the development of heuristics and event-discrete simulation techniques. From the practical perspective, they arise in manufacturing contexts, worker allocation schemes, medical operation and surgery planning and even in packing problems – just to mention a few. In many real-world problems, process parameters and other quantities tend to fluctuate or are just known modulo some uncertainties. Consequently, improved modeling means incorporating such aspects. This can be done by considering parameters as random variables of a given distribution instead of fixed numbers.
A view in the literature underlines the importance of RCPSP: early attempts go back to [1, 2, 3] and recent surveys coping the variants of such problems are for example [4, 5]. In general, heuristics and the genetic algorithm play an important role [6, 7, 8], some further state-of-the-art algorithms can be found in [9, 10, 11, 12, 13]. The works [13, 14, 15, 16] pay attention to stochastic RCPSP mainly with discrete random variables. However, the NP-hardness of RCPSP impedes finding the exact optimum of larger problem instances within a reasonable time. Consequently, we are forced to optimize small enough problem instances. Then the findings obtained are extrapolated.
Our approach focuses on continuous random variables because many relevant processes or phenomena can be modeled by normally distributed, Gamma or Beta distributed quantities. A stochastic RCPSP is characterized firstly by the dependencies of its inner parts (called jobs), secondly by the duration of the jobs (continuous random variables) and thirdly by the number of available resources. The aim is to minimize the cycle time. Basically, we are interested in some aspects of the systematics with special attention to stochastic influences:

- How are the distributions of the input parameters and the distribution of the objective function related?
- Regarding the reliability, is it possible to estimate the variation of the objective function using the variation of the input parameters?

To attack these questions, we established a four-part simulation and analysis framework:

1. Random Number Generator: provides an appropriate set of random numbers of a prescribed distribution with expectation value and standard deviation as parameters to encode the duration of all jobs.
2. Structure Analyzer: investigates the dependencies of the jobs to find critical paths (dependencies are encoded in a directed network graph).
3. Optimizer: executes the optimization of the cycle time for all stochastic realizations of the RCPSP.
4. Output Analyzer: evaluates the resulting sample of realizations w. r. t. statistical characteristic values and distribution function fits.

Our paper is organized as follows: Section 1 provides the methodical basis in terms of a precise definition of the problem class scope and short notes on the Gamma

and the Beta distribution but of course not missing to describe the simulation procedure. Section 2 is dedicated to selected results. We summarize in Section 3.

1 Model Setup and Simulation Approach

In this section the employed methods are explained. First, the scope of a stochastic RCPSP is precisely defined. Second, the framework for solving such RCPSP is presented. These two main parts are supplemented by some short notes on the Gamma and the Beta distribution.

1.1 Stochastic RCPSP

To keep the view on the principle, we refrain here from features like multiple projects, multiple modi as well as transfer times or type representatives. A stochastic RCPSP is therefore described by:

- jobs $j = 1...J$ and their respective durations $d_j \in \mathbb{R}_{\geq 0}$ considered as continuous random variables; each job starts once and must not be interrupted
- successor matrix $S \in \{0,1\}^{JxJ}$ defined by

$$S_{j_1 j_2} = \begin{cases} 1 & \text{if job } j_2 \text{ follows after } j_1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

- number of renewable resources $R \in \mathbb{N} \cup \{\infty\}$, where each job occupies exactly one resource during its execution and $R = \infty$ refers to the unconstrained problem

In summary, a stochastic RCPSP is a tuple $(J, R, (d_j), (S_{j_1 j_2}))$ supplemented by the stochastic parameters entering the vector (d_j).

1.2 Simulation and Optimization Framework

The framework consists of the following four parts.

Random Number Generator:
Simulating stochastic RCPSP means generating a sufficient large sample of realizations. For each stochastic RCPSP and each job, the expectation value μ_j $(j = 1...J)$ of the duration d_j is kept fix. Furthermore, the coefficient of variation η is the same for all jobs, such that the standard deviation σ_j of the duration d_j follows from $\sigma_j = \eta \cdot \mu_j$. In addition, the type of distribution is chosen (Gamma or Bets distribution, we do not consider normally distributed random variables since they can obey negative values). The sample size varies from 10,000 up to one million.

Structure Analyzer:
Dealing with continuous random variables impedes simple, linear optimization models with binary decision variables (see [17, 18] for instance). Since we refrain here from a time discretization, we choose the following structural approach to incorporate the boundary of the number of resources. Let the successor matrix S be given. As a first step, the tool evaluates the potentially maximal number $R_{\max}(S)$ of resources needed such that there would be no queue (and the problem is unconstrained independently of d_j). Typically, $R_{\max}(S) > R$. As a second step, the tool collects all those successor matrices $\tilde{S}$ with $\tilde{S}_{j_1 j_2} \geq S_{j_1 j_2}$ for all j_1, j_2 and $R_{\max}(\tilde{S}) = R$. Considering $\tilde{S}$ instead of S, the RCPSP becomes unconstrained. In summary, the idea is to solve rapidly (many) unconstrained PSPs to find the optimum of the original RCPSP. That is the key point because the main issue lies in the efficient selection of all relevant $\tilde{S}$ matrices. To handle this challenge, we strongly employ the close connection between successor matrices and posets and the knowledge about such sets (generation, isomorphism classes, structural dependencies, [19, 20, 21, 22, 23]).

Optimizer:
The task of minimization the cycle time is equivalent to find the shortest path in a weighted directed graph (encoded in $\tilde{S}$). To this end, we employ both Gurobi and the Python package Networkx to go through the list of the relevant $\tilde{S}$ matrices. Such an evaluation is done for all stochastic realizations.

Output Analyzer:
The sample of all cycle times enters this analyzing tool which computes statistical quantities, for example mean value, standard deviation, coefficient of variation, skewness. In addition, it fits the data to a given type of distribution depending on multiple parameters. Particularly interesting is the relation in terms of appropriate parametrizations between input and output quantities.

All in all, the framework is completely automatized and can be combined with a simulation framework for heuristics [24] and with an AI tool for finding

promising fit parametrizations for the Output Analyzer. However, the results of the following section focus on stochastic fluctuations.

1.3 The Gamma and the Beta Distribution

We briefly collect some folklore about these two distributions both quite convenient for modeling process times or human working durations. Let X denote a random variable with its expectation value μ, its standard deviation σ and the dimensionless coefficient of variation $\eta = \frac{\sigma}{\mu}$. The probability density function of the Gamma distribution reads as

$$f^{(\Gamma)}_{\alpha,\beta}(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \tag{2}$$

with $x \geq 0$ (arbitrary large values are admissible), $\alpha = \frac{\mu^2}{\sigma^2}$ and $\beta = \frac{\mu}{\sigma^2}$; analogously, we have

$$f^{(B)}_{\alpha,\beta}(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{3}$$

for a Beta distribution with x ranging in [0,1] (linearly scalable to the interval $[x_{\min}, x_{\max}]$),

$$\alpha = \mu\left(\frac{\mu}{\sigma^2}(1-\mu)-1\right) \tag{4}$$

and

$$\beta = \frac{\mu}{\sigma^2}(1-\mu)^2 + \mu - 1. \tag{5}$$

The cumulative distribution functions are denoted by $F^{(\Gamma)}_{\alpha,\beta}$ and $F^{(B)}_{\alpha,\beta}$, respectively.

2 Selected Results

We picked three instances for our case study: 20 and 50 jobs for the unconstrained PSP and 10 jobs for investigating the RCPSP. As mentioned above, a universal coefficient of variation η is fixed (see below) and the expectation value μ_j of the duration d_j is chosen randomly in the interval [0.3, 0.7]. A main result of our investigation is that the distribution of the objective function can be described with high accuracy by a product ansatz having only two factors and four parameters even the problem class contains many parameters. In addition, this statement remains true of all values of the resource number R. Note that it is not sufficient to fit a simple Gamma (Beta) distribution if the input parameters are Gamma (Beta) distributed. Indeed, the second factor in the product is necessary.

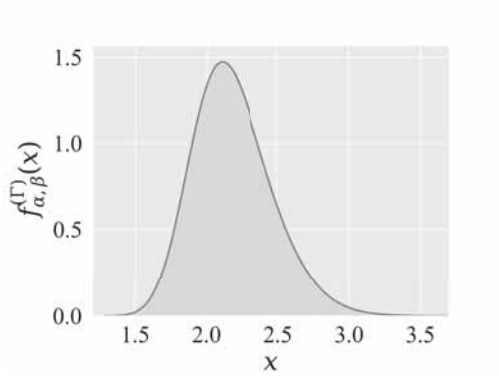


Figure 1: Histogram (blue) of the cycle time x and best fits (red) according to (6) for instance with $J = 20$, Gamma distribution and $\eta = 0.3$ (sample size one million).

Let us take a closer look at the product ansatz. We chose

$$F^{(D)}_{\alpha_1,\alpha_2,\beta_1,\beta_2}(x) = F^{(D)}_{\alpha_1,\beta_1}(x) \cdot F^{(D)}_{\alpha_2,\beta_2}(x) \tag{6}$$

with $D \in \{\Gamma, B\}$ which is inspired by the following facts.

- The sum of multiple Gamma distributed, independent random variables is Gamma distributed.
- The sum of multiple Beta distributed, independent random variables is approximately Beta distributed.
- The cumulative distribution function of the maximum of multiple, independent random variables is equivalent to the product of the cumulative distribution functions of the single random variables.

Hereby, the statements of the first two bullet points refer to the stochastical behavior along chains of jobs while the statement of the third bullet point refers to the comparison of such chains to select the critical path. However, this inspiration does not pay very much attention to the mathematical subtleties (especially the stochastical independency of the random variables).

2.1 Results of the Unconstrained PSP

We consider four samples of unconstrained problems: $J = 20$ with Gamma distribution (see Figure 1) and Beta distribution (see Figure 2), $J = 50$ with Gamma distribution (Figure 3) and Beta distribution (Figure 4). All

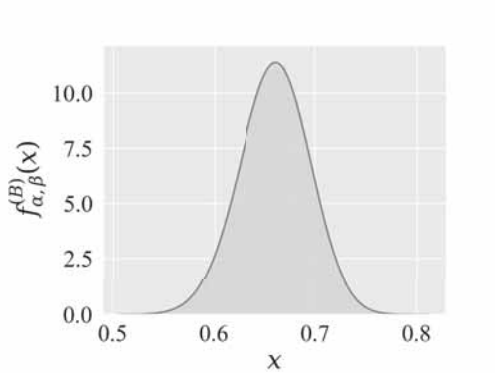


Figure 2: Histogram (blue) and best fits (red) according to (6) for instance with $J = 20$, Beta distribution, $\eta = 0.1$; the variable x is scaled to [0,1] by dividing the cycle time by the longest path (sample size one million).

figures display the histogram of the sample (size one million) and the best approximation according to equation (6). The four-parameter fit exhibits a sufficiently high quality for all considered problems even though the problem class depend on far more parameters. Regarding the 50-job instances, the distribution becomes more symmetric.

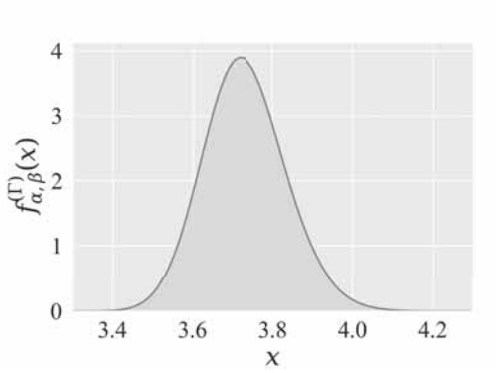


Figure 3: Histogram (blue) of the cycle time x and best fits (red) according to (6) for instance with $J = 50$, Gamma distribution, $\eta = 0.3$ (sample size one million).

As a next step, we compare dimensionless quantities. Figure 5 shows the coefficient of variation η_{out} of the objective function as a function of the coefficient of variation η_{in} of the input. Over a large number of instances, we see a stable parameter dependency. The behavior η_{out} vs. η_{in} is close to a power law

$$\eta_{\text{out}} = a \cdot \eta_{\text{in}}^{b} \tag{7}$$

with $a = 0.3227 \pm 7 \cdot 10^{-4}$ and $b = 0.7874 \pm 3 \cdot 10^{-3}$ (red curve in Figure 5). In addition, the objective function varies less than the input quantities. This stability in combination with the appropriate fit (6) allows an a priori estimation because the fit parameters of (6) can be estimated by a structural analysis (part of the Structure Analyzer) based on what is called the grade distribution [19, 20]. Such an estimation can be refined by a small sample. Therefore, after less runs the simulation gives already insights of the stochastic behavior of the objective function.

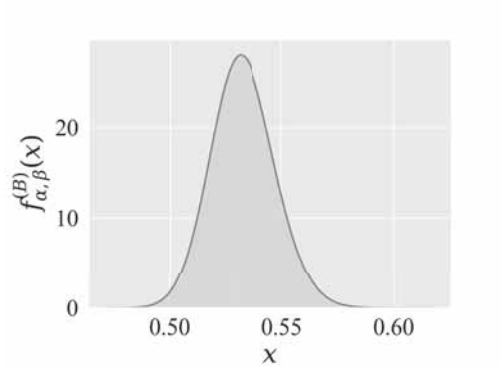


Figure 4: Histogram (blue) and best fits (red) according to (6) for instance with $J = 50$, Beta distribution, $\eta = 0.1$; x is scaled to [0,1] (sample size one million).

2.2 Results of the RCPSP

Let us now turn to the 10 job RCPSP. Mainly we are interested in two aspects: First, does the fit ansatz (6) still remain appropriate? Second, what kind of dependencies are between the resource bound R and the stochastic quantities μ_{out} and η_{out}?
Figure 6 shows the distribution (density function) of the objective function for a 10 job Gamma distributed sample for various values of R. Note that if $R \geq R_{\max}(S)$ the problem is unconstrained (independently of R). Regarding the first question, the answer is yes (such that the histograms are not shown in Figure 6), equation (6) covers all of the following cases: for the case $R = 1$, the execution of the jobs is equivalent to a single chain. So in this special case, the second factor in equation (6) becomes obsolete. For small values of R, the problem in-

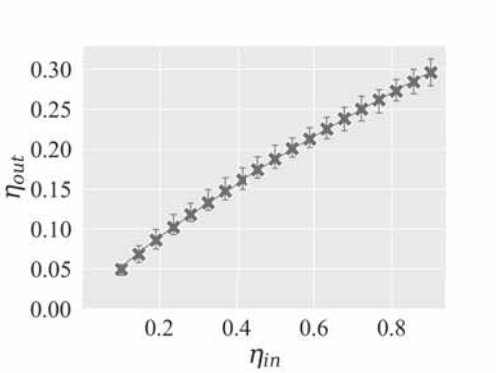


Figure 5: Coefficient of variation of the objective function η_{out} as a function of the coefficient of the variation η_{in} of the input distributions ($J = 20$, Gamma distribution, 100 instances per η_{in} value, sample size 10,000). The errorbars indicate minimum, mean (crosses) and maximum. The red curve depicts the power law regression.

stance remains constrained for all possible realizations. For further increasing R, the resultant distribution contains a mixture of realizations, where the durations are as such that the problem is de facto unconstrained, and realizations, where the resource constraint is indeed active. For $R \geq R_{\max}(S)$, the problem is unconstrained for all duration vectors (d_j). Typically, $R_{\max}(S) \leq J$.

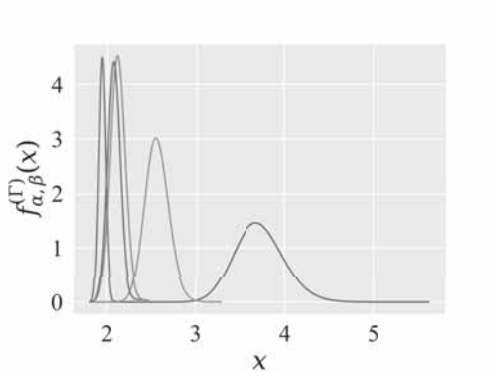


Figure 6: Distribution (density function) of the cycle time for varying resource number: $R = 1$ (blue), 2 (orange), 3 (green), 4 (red) and $5 = R_{\max}$ (purple). All cases match the fit (6). The sample size is 40,000 and $J = 10$ with Gamma distribution.

Clearly, a decreasing R causes an increasing μ_{out}. Figures 7 and 8 provide a more detailed and quantitative picture by showing μ_{out} and η_{out} as a function of R for

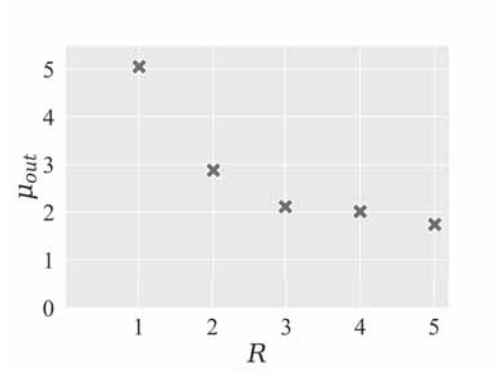


Figure 7: Dependecy of expectation value μ_{out} on the resource number for a pool of Gamma and Beta distributed instances with varying μ. We have $J = 10$ and $R_{\max} = 5$ fixed such that the problems feels no resource limitation for $R \geq 5$. The errorbars are not shown because they are negligible compering the size of the crosses.

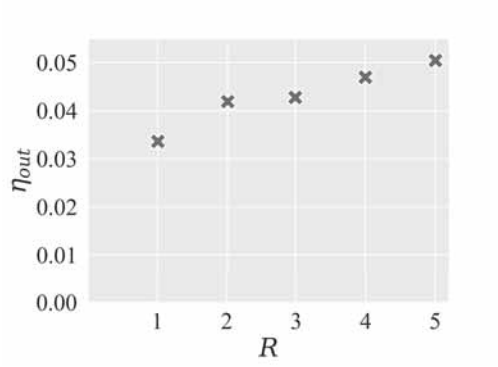


Figure 8: Dependecy of the coefficient of variation η_{out} on the resource number for a pool of Gamma and Beta distributed instances with varying μ. We have $J = 10$ and $R_{\max} = 5$ fixed such that the problems feels no resource limitation for $R \geq 5$. The errorbars are not shown because they are negligible compering the size of the crosses.

several instances. As it is the case in Figure 5, the dependency can be catched by a rather simple regression. Summarizing the previous examples, the stochatic characteristic values and the shape of the distribution of the cycle time as objective function are predictable both for unconstrained and constrained PSP.

3 Summary and Outlook

The purely continuous treatment reveals close connections between the input distributions and the distribution of the cycle time as objective function. Based on simulation studies, it is feasible to estimate the shape and all related stochastic properties of the resultant distribution since it turned out that the proposed product fit suffices for all practical purposes even though it seems hard to prove its validity by rigorous mathematical arguments. Although solving a single realization of an RCPSP remains challenging for a larger number of jobs, considering a population significantly relaxes the situation, smooths discontinuous aspects of the combinatorial optimization problem and eventually enables statements about confidence intervals of the cycle time having practical relevance.
Without wanting to look to far into the problem landscape, it seems natural to attack more sophisticated RCPSP (multi-mode and with transfer times and type representatives, for instance) with the continuous approach and to investigate the influence of these extended features on the shape of the objective function distribution. Hereby, questions of continuous dependency and stability are of particular interest. Innovative tools in event-discrete simulation and optimization will provide valuable contribution to this.

Acknowledgement

The work of Mathias Kühn and Rico Zöllner is supported by DFG project 418727532. The authors gratefully thank Ella Jannasch for her support.

References

[1] Pritsker AAB, Waiters LJ, Wolfe PM. Multiproject scheduling with limited resources: A zero-one programming approach. *Management science*. 1969; 16(1):93–108.

[2] Mohanty Ru, Siddiq M. Multiple projects-multiple resources-constrained scheduling: some studies. *The International Journal of Production Research*. 1989; 27(2):261–280.

[3] Kolisch R, Sprecher A, Drexl A. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management science*. 1995;41(10):1693–1703.

[4] Issa S, Tu Y. A survey in the resource-constrained project and multi-project scheduling problems. *Journal of Project Management*. 2020;5(2):117–138.

[5] Hartmann S, Briskorn D. An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of operational research*. 2022;297(1):1–14.

[6] Singh A. Resource constrained multi-project scheduling with priority rules & analytic hierarchy process. *Procedia engineering*. 2014;69:725–734.

[7] Gonçalves JF, Mendes JJ, Resende MG. A genetic algorithm for the resource constrained multi-project scheduling problem. *European journal of operational research*. 2008;189(3):1171–1190.

[8] Zhu L, Lin J, Li YY, Wang ZJ. A decomposition-based multi-objective genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. *Knowledge-based systems*. 2021; 225:107099.

[9] Liu Y, Jin S, Zhou J, Hu Q. A branch-and-bound algorithm for the unit-capacity resource constrained project scheduling problem with transfer times. *Computers & Operations Research*. 2023;151:106097.

[10] Liu Y, Zhou J, Lim A, Hu Q. A tree search heuristic for the resource constrained project scheduling problem with transfer times. *European Journal of Operational Research*. 2023;304(3):939–951.

[11] Ren Y, Lu Z, Liu X. A branch-and-bound embedded genetic algorithm for resource-constrained project scheduling problem with resource transfer time of aircraft moving assembly line. *Optimization Letters*. 2020;14(8):2161–2195.

[12] Poppenborg J, Knust S. A flow-based tabu search algorithm for the RCPSP with transfer times. *Or Spectrum*. 2016;38:305–334.

[13] Kadri RL, Boctor FF. An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*. 2018; 265(2):454–462.

[14] Hagstrom JN. Computational complexity of PERT problems. *Networks*. 1988;18(2):139–147.

[15] Deblaere F, Demeulemeester E, Herroelen W. Proactive policies for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*. 2011;214(2):308–316.

[16] Rostami S, Creemers S, Leus R. New strategies for stochastic resource-constrained project scheduling. *Journal of Scheduling*. 2018;21:349–365.

[17] Artigues C, Demassey S, Neron E. *Resource-constrained project scheduling: models, algorithms, extensions and applications*. John Wiley & Sons. 2013.

[18] Neumann K. Project scheduling with changeover times—Modelling and applications. In: *Proc. Int. Conf. Ind. Eng. Prod. Manage.*, vol. 1. 2003; pp. 30–36.

[19] Stanley RP. Enumerative Combinatorics Volume 1 second edition. *Cambridge studies in advanced mathematics*. 2011;.

[20] Stanley RP. Enumerative Combinatorics Volume 2. *Cambridge studies in advanced mathematics*. 2001;.

[21] Brinkmann G, McKay BD. Counting unlabelled topologies and transitive relations. *Journal of Integer Sequences*. 2005;8(2):3.

[22] Brinkmann G, McKay BD. Posets on up to 16 points. *Order*. 2002;19:147–179.

[23] Cordella LP, Foggia P, Sansone C, Vento M. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence*. 2004;26(10):1367–1372.

[24] Zöllner R, Kühn M, Handrich K, Schmidt T. Automated generation and simulation of hyper heuristics for stochastic multi-mode multi-project resource-constrained project scheduling problems with setup times. *20 ASIM Fachtagung Simulation in Produktion und Logistik 2023*;pp. 403–412.

Identifikation instabiler, unteraktuierter System mit nicht-linearem dynamischen Verhalten

Marian Göllner[1*], Sven Jacobitz[1], Roberto Ferrara[1], Xiaobo Liu-Henke[1]

[1]Institut für Mechatronik, Ostfalia Hochschule für angewandte Wissenschaften,
Salzdahlumer Str. 46/48, 38302 Wolfenbüttel; [*]*mar.goellner@ostfalia.de*

Abstract. Der vorliegende Beitrag befasst sich mit der Identifikation und Regelung instabiler, unteraktuierten Systeme mit nicht-linearem dynamischen Verhalten. Diese Systeme sind aufgrund ihrer Instabilität und nichtlinearen Reaktionen auf herkömmliche Regelungstechniken eine besondere Herausforderung für präzise Modellierung und effektive Steuerung. Zur Adressierung dieser Probleme entwickelten wir ein methodisches, modellbasiertes Vorgehen unter Einsatz von Rapid Control Prototyping (RCP), das auf physikalischen Modellen basiert und die Prozesse Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) und Hardware-in-the-Loop (HiL) integriert. Der methodische Rahmen umfasst die Identifikation der Systemdynamik durch messdatenbasierte Ansätze und die Verifikation der Modelle zur Sicherstellung ihrer Genauigkeit. Durch Anwendung dieser Modelle auf das spezifische Beispiel des S-Mobile, einem hochdynamischen Intralogistiksystem mit sphärischem Elektroantrieb, demonstrieren wir die Effektivität des Ansatzes. Die Ergebnisse zeigen eine verbesserte Modellgenauigkeit und eine robuste Steuerung des Systems, was dessen potenzielle Anwendbarkeit in ähnlich komplexen technischen Systemen unterstreicht.

Einleitung und Problemstellung

Die Modellierung und Regelung von intelligenten dynamischen Systemen stellt einen fundamentalen Aspekt der modernen Ingenieurswissenschaften dar. Besonders herausfordernd wird diese Aufgabe, wenn es um instabile, unteraktuierte Systeme mit nicht-linearem Verhalten geht [1]. Solche Systeme finden sich in einer Vielzahl von Anwendungen, von Robotik bis hin zu Energieübertragungssystemen, und erfordern präzise und zuverlässige Modelle für eine effektive Systemauslegung.

In der modernen Regelungstechnik sind die eingesetzten Modelle von entscheidender Bedeutung. Sie dienen nicht nur zur Systemauslegung, sondern sind integraler Bestandteil der Reglerfunktionen. Fehlende Genauigkeit in der Modellbildung kann zu suboptimaler Leistung und sogar zum Versagen des Systems führen. Daher ist die Identifikation der Systemdynamik, insbesondere bei nicht-linearen und unteraktuierter Systemen, eine zentrale Herausforderung. Diese Systeme zeichnen sich durch ihre Tendenz aus, auf konventionelle Regelungsmethoden mit unvorhersehbarem oder instabilem Verhalten zu reagieren.

Die folgende Abbildung 1 zeigt ein solches System, den Funktionsträger S-Mobile, der als hochdynamisches Intralogistiksystem mit sphärischem Elektroantrieb ausgelegt ist. Er besteht aus einem Aufbau der über rotationssymmetrisch angeordnete Aktuatoren mittels Allseitenrädern auf einer Kugel balanciert wird. Primäres Problem ist die Identifikation und Validierung des Streckenmodells, als integraler Bestandteil des Regelkonzeptes (vgl. [2]). Entsprechend hoch sind die Anforderungen an die Modellierungsgüte.

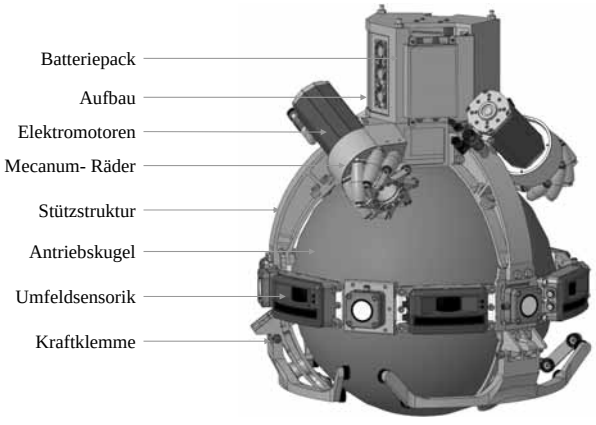


Abbildung 1: Funktionsträger S-Mobile als beispielhafte nicht-lineare, instabile Strecke.

1 Methodik

Für die Entwicklung komplexer cyber-physischer Systeme ist ein methodisches, modellbasiertes Vorgehen unerlässlich [3]. Hierbei hat sich das ganzheitliche, durchgängige, verifikationsorientierte Rapid Control Prototyping (RCP) durchgesetzt. Kern ist ein auf physikalischen Ansätzen basierendes White-Box-Modell des zu regelnden Systems sowie die Prozesse Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) und Hardware-in-the-Loop (HiL). Die Korrektheit des eingesetzten Modells ist demnach eine wesentliche Voraussetzung für eine effiziente Entwicklung und valide Entwicklungsergebnisse. Daher sind die Modellidentifikation und -verifikation Kernbestandteile des Modellbildungsprozesses [4].

Abbildung 2 illustriert den hierfür angewendeten allgemeinen messdatenbasierten Identifikationsprozess in Anlehnung an [5]. Anhand der Aufgabenstellung und den Anforderungen sowie den a-priori-Kenntnissen über das System erfolgt zunächst die Planung der Messungen sowie eine initiale Schätzung der Parameter des theoretischen Modells. Die Differenz aus Mess- und Simulationsergebnissen ε dient als Eingang für eine Optimierungsfunktion zur Parameterberechnung. Während der Modellverifikation erfolgt die Beurteilung, ob das reale Verhalten mit hinreichender Genauigkeit abgebildet wird. Ist dies der Fall, steht das endgültige Modell fest. Ansonsten erfolgt eine Anpassung des theoretischen Modells beispielsweise durch Erhöhung der Modellierungstiefe.

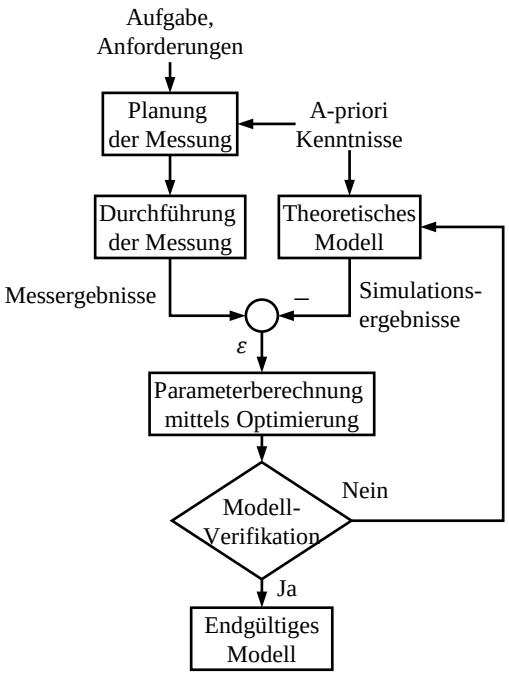


Abbildung 2: Messdatenbasierter Modellidentifikationsprozess.

2 Stand des Wissen

Identifikation in der Regelungstechnik bezieht sich auf die experimentelle Bestimmung des zeitlichen Verhaltens eines Prozesses oder Systems durch Analyse gemessener Signale. Ziel ist es, das Systemverhalten innerhalb einer definierten Klasse mathematischer Modelle so genau wie möglich abzubilden, wobei die Fehler zwischen dem realen Prozess bzw. System und seinen mathematischen Modellen minimiert werden sollen. Die Herausforderung besteht darin, die Diskrepanz zwischen dem tatsächlichen Systemverhalten und seinem mathematischen Modell so gering wie möglich zu halten. Dabei nutzen Forscher und Ingenieure gemessene Eingangs- und Ausgangssignale, um die Systemdynamik zu charakterisieren und zu modellieren. Dieser Prozess ist entscheidend für die Entwicklung präziser und effizienter Steuerungs- und Regelungssysteme, besonders in technischen Bereichen wie der Robotik, Luft- und Raumfahrt und der industriellen Automatisierung, wo exakte Modelle für eine optimale Leistung unerlässlich sind [6].

2.1 Klassifikation von Identifikationsmethoden

Identifikationsmethoden für dynamische Systeme lassen sich grundlegend nach ihrer Analysebasis in zwei Kategorien einteilen: Methoden im Zeitbereich und Methoden im Frequenzbereich [7]. Zeitbereichsmethoden nutzen Zeitreihendaten, um das dynamische Verhalten des Systems zu charakterisieren. Diese Methoden sind besonders nützlich, um nichtlineare Beziehungen zwischen Eingangs- und Ausgangssignalen zu modellieren [6].

Im Gegensatz dazu analysieren Frequenzbereichsmethoden das Systemverhalten durch Untersuchung der Reaktion des Systems auf sinusförmige Eingangssignale verschiedener Frequenzen. Diese Methoden sind effektiv in der Bestimmung der Systemcharakteristika durch die Analyse des Frequenzgangs. Frequenzbereichsmethoden eignen sich besonders gut zur Identi-

fikation von linearen Systemen, da sie eine klare und anschauliche Darstellung der Systemdynamik im Frequenzspektrum ermöglichen [8].

Darüber hinaus gibt es eine dritte Kategorie, die sogenannten gemischten Methoden, die Elemente beider Ansätze kombinieren, um die Vorteile von Zeit- und Frequenzbereichsanalysen zu nutzen. Diese hybriden Ansätze sind oft in der Lage, eine umfassendere Analyse zu bieten, indem sie sowohl die direkte Zeitantwort als auch die frequenzabhängigen Eigenschaften des Systems erfassen [1].

Die in der obigen Diskussion erwähnten Methoden setzen voraus, dass das Ein-/Ausgangsverhalten eines Systems direkt gemessen wird. Bei instabilen Systemen ist dieses Vorgehen jedoch nicht praktikabel. In solchen Fällen ist der Einsatz fortgeschrittener Techniken wie der Closed Loop Identification erforderlich [9]. Bei der Untersuchung von unteraktuierten Systemen ist es zudem entscheidend, sicherzustellen, dass alle Systemzustände vollständig angeregt und analysiert werden.

2.2 Ansätze zur Identifikation nichtlinearer, instabiler Systeme

Die gegenwärtige Forschung im Bereich der Systemidentifikation eröffnet vielfältige Perspektiven auf die Herausforderungen und Methoden zur Analyse dynamischer Systeme, insbesondere auch unter Einsatz künstlicher Intelligenz und Machine Learning [9]. Ein verbreiteter Ansatz ist das Schließen des Regelkreises zur Stabilisierung des Systems. Hier wird zwischen direkten (Auswertung des System-Eingangs und -Ausgangs) und indirekten (Auswertung der Führungsgröße und des System-Ausgangs) Identifikationsmethoden unterschieden. Diese Methoden ermöglichen es sowohl konsistente Modelle zu erstellen als auch mit nicht modellierten Dynamiken (approximative Modellierung) umzugehen [10]. Xavier et al. [11] bieten eine eingehende Übersicht.

In der spezifischen Anwendung auf unteraktuierte Systeme wenden Chawla und Singla [12] adaptive, neuralbasierte Fuzzy-Inferenzsysteme (ANFIS) an. Hierbei wird das Modell eines inversen Pendels aus Eingangs-/Ausgangsdaten der dynamischen Systemantwort generiert. Die Genauigkeit des ANFIS-Modells wird sowohl durch den mittleren quadratischen Fehler als auch durch experimentelle Vergleiche mit realen Systemmodellen bestätigt. Hierbei handelt es sich jedoch um ein nicht physikbasiertes Modell, welches sich nicht für die Behandlung mit typischen regelungstechnischen Methoden eignet.

Durch Chen et al. [13] wird die duale Eingangs-Ausgangs-Parametrisierung (Dual Input-Output Parametrization, dual IOP) eingeführt, ein neues Verfahren zur Identifikation von linearen zeitinvarianten Systemen unter Verwendung von Messdaten des geschlossenen Regelkreises. Die Methode stellt eine Erweiterung der bisherigen Ansätze zur Identifikation im geschlossenen Regelkreis dar und vereinfacht insbesondere die Auslegung des stabilisierenden Reglers. Die Einsetzbarkeit für nichtlineare Systeme wird nicht diskutiert.

Zuletzt befassen sich González et al. [14] mit der Identifikation von instabilen, kontinuierlichen Systemen unter Verwendung verfeinerter Instrumentalvariablenmethoden, insbesondere im geschlossenen Regelkreis. Es wird dargelegt, dass bestehende Ansätze wie die Simplified Refined Instrumental Variable Method (SRIVC) in ihrer herkömmlichen Form nicht zuverlässig sind, wenn es um die Modellierung instabiler Systeme geht. Als Lösung wird eine Anpassung dieser Methoden vorgeschlagen, die die Einführung eines speziell angepassten Allpass-Filters im Vorfilterungsschritt umfasst. Diese modifizierten Methoden erlauben die Identifikation instabiler Systeme und minimieren die Abweichung bei Konvergenz. Die vorgestellte Methode ist jedoch nur für die Anwendung auf lineare Systeme vorgesehen.

3 Konzeption des neuen Verfahrens

Um den etablierten modellbasierten Auslegungsprozess durchlaufen zu können und um insbesondere modellintegrierende Reglerverfahren parametrieren und nutzen zu können, bedarf es einer tiefgehenden physikalischen Analyse der Systemzustände und Parameter. Dazu ist es nötig ein physikalisches Modell mit vollständig definierten Parametern der Regelstrecke aufzustellen [15]. Hierzu ist eine Modellbildungsmethode auf Basis von Bilanzgleichungen unter Berücksichtigung von Erhaltungssätzen zu favorisieren. Nach dem Noether-Theorem hat jede kontinuierliche Symmetrie der Wirkung einen Erhaltungssatz zur Folge, und umgekehrt gehört zu jedem Erhaltungssatz eine kontinuierliche Symmetrie der Wirkung. Somit ist die zugrundeliegende Physik fundamental im Modell erfasst und nicht durch Approximation verschleiert. Somit entstehen nicht-lineare Modelle mit vollständig aus der Physik herleitbaren Parametern. Die Identifikation dieser

ist wie eingangs erwähnt jedoch aufgrund der Nicht-Linearität und ggf. Zeitvarianz nicht-trivial. Das nachfolgend beschriebene Verfahren, welches grundsätzlich für jede Art von System in jeder Domäne durchführbar ist, beschreibt sequentiell die gezielte Identifikation von Systemparametern anhand eines mechanischen Systems:

3.1 Allgemeine Modellbeschreibung

Auf Basis der Kinematik und Dynamik des jeweiligen zu identifizierenden mechanischen Systems müssen zunächst die Dynamikfunktionen in Form von verkoppelten, nicht-linearen Differentialgleichungen hergeleitet werden. Die nachfolgende Gleichung zeigt das darauf basierende, verallgemeinerte Dynamikmodell [2]:

$$\underline{\underline{M}}\left(\underline{q}\right)\cdot\ddot{\underline{q}}+\underline{\underline{C}}\left(\underline{q},\dot{\underline{q}}\right)\cdot\dot{\underline{q}}+\underline{G}\left(\underline{q}\right)=\underline{\underline{F}}\left(\underline{q}\right)\cdot\underline{u} \qquad (1)$$

Die symmetrische Massenmatrix $\underline{\underline{M}}\in\mathbb{R}^{m\times m}$ hängt von den einzelnen Massen der starren Körper des Systems und den verallgemeinerten Koordinaten $\underline{q}\in\mathbb{R}^m$ ab. Der Vektor $\underline{\underline{C}}\in\mathbb{R}^{m\times m}$ beschreibt die verallgemeinerten Kreiselfkräfte, die aus den Zentrifugal- und Corioliskräften bestehen. Der Vektor $\underline{G}\in\mathbb{R}^m$ beschreibt die potentielle Energie über die Gravitation. Die eingeprägten Stellgrößen werden mit dem zeitveränderlichen Vektor $\underline{u}\in\mathbb{R}^n$ gekennzeichnet. Dessen Multiplikation mit der Funktionalmatrix $\underline{\underline{F}}\in\mathbb{R}^{m\times n}$ führt zur Momentenmatrix.

3.2 Linearisierung des Modells

Charakteristiken der Systemdynamik aus Gleichung (1) lassen sich besonders gut im Frequenzbereich analysieren. Hierfür wird zunächst ein lineares Modell in verschiedenen Arbeitspunkten benötigt. Hierbei lässt sich beispielsweise eine Taylor-Reihenentwicklung unter Verwendung von Jacobi-Matrizen zu verschiedenen Zeitschritten i einsetzen. Das linearisierte Modell am aktuellen Arbeitspunkt ist entsprechend:

$$\Delta\ddot{\underline{q}}=\underbrace{\left(-\underline{\underline{J}}_{\ddot{q}}(i)^{-1}\cdot\underline{\underline{J}}_{q}(i)\right)\cdot\Delta\underline{q}+\left(-\underline{\underline{J}}_{\ddot{q}}(i)^{-1}\cdot\underline{\underline{J}}_{\dot{q}}(i)\right)\cdot\Delta\dot{\underline{q}}}_{\text{für }\underline{\underline{A}}_i\cdot\underline{x}}$$
$$+\underbrace{\left(-\underline{\underline{J}}_{\ddot{q}}(i)^{-1}\cdot\underline{\underline{J}}_{u}(i)\right)\cdot\Delta\underline{u}}_{\text{für }\underline{\underline{B}}_i\cdot\underline{u}}+\underbrace{\left(-\underline{\underline{J}}_{\ddot{q}}(i)^{-1}\right)\cdot f\Big|_{AP_i}}_{\text{für }\underline{\underline{E}}_i\cdot\underline{z}_i}$$

Ergebnis ist ein lineares Zustandsraummodell nach Gleichung (2).

$$\begin{aligned}\dot{\underline{x}}&=\underline{\underline{A}}_i\cdot\underline{x}+\underline{\underline{B}}_i\cdot\underline{u}+\underline{\underline{E}}_i\cdot\underline{z}_i\\ \underline{y}&=\underline{\underline{C}}\cdot\underline{x}+\underline{\underline{D}}\cdot\underline{u}\end{aligned} \qquad (2)$$

3.3 Stabilisierung des Systems

Auf Basis der für den Arbeitspunkt gültigen Zustandsraumdarstellung soll nun das System zumindest um einen engen Gültigkeitsbereich um den Arbeitspunkt i mithilfe einer Zustandsrückführung stabilisiert werden. Eine Reglerverstärkung $\underline{\underline{K}}$, die über die Zustandsrückführung den Stellvektor $\underline{u}$ über das einfache Stellgesetz $\underline{u}^T=-\left(\underline{\underline{K}}^T\cdot\underline{x}\right)^T=-\underline{x}^T\cdot\underline{\underline{K}}^T$ bildet, ergibt sich aus diversen Ansätzen der Zustandsregelung und muss dabei nicht optimal ausgelegt aber bekannt und konstant sein. Eine Normalisierung der Systemverstärkung sowie die Transformation des Führungsvektors erfolgt über ein Vorfilter $\underline{\underline{N}}$. Es ergibt sich so das Stellgesetz des Reglers zu:

$$\underline{u}=-\underline{\underline{K}}_i\cdot\underline{x}+\underline{\underline{N}}_i\cdot\underline{w} \qquad (3)$$

Aus Sicht des Sollwerteingangs $\underline{w}$ reagiert der geschlossene Regelkreis innerhalb der hauptsächlich durch Stellgrößenbeschränkungen definierten physikalischen Grenzen nun wie ein stabiles Mehrgrößensystem. Die Systemgrenzen sind genauso zu prüfen wie die BIBO Stabilität. Dies ist durch eine Betrachtung der Steuerbarkeit gegeben; ist $\left(\underline{\underline{A}},\underline{\underline{B}}\right)$ vollständig steuerbar, so kann die Eigendynamik beliebig eingestellt, ergo auch das System stabilisiert werden.

3.4 Übertragungsfunktion und Entkopplung des Systems

Das durch Regelung nun stabilisierte System soll zur Identifikation in den Koordinaten des zu untersuchenden Starrkörpers (BCS) entkoppelt und die unabhängigen Übertragungswege als Übertragungsfunktionen dargestellt werden. So wird ein Zugang zur Hauptdiagonale der Übertragungsmatrix des Gesamtsystems über die Übertragungsfunktionen $\frac{\vartheta_x}{\vartheta_{x\_soll}}, \frac{\vartheta_y}{\vartheta_{y\_soll}}, \frac{\vartheta_z}{\vartheta_{z\_soll}}$ in den jeweiligen Raumrichtungen (entweder rotatorisch oder translatorisch) geschaffen. Der lineare Ersatzzustandsraum des Systems hat die Stellgrößen τ_1,τ_2,τ_3 der Aktuatoren als Eingangsvektor $\underline{u}(t)$, da die Wirkung der Stellgrößen in dem Streckenmodel bereits in Bezug

auf das BCS berücksichtigt sind. Die Übertragungsmatrix $\underline{\underline{G}}_S(s)$ der Strecke lässt sich aus der Zustandsraumdarstellung gewinnen indem dieser in den Bildbereich transformiert wird.

$$\underline{\underline{G}}_S(s) = \frac{\underline{X}(s)}{\underline{U}(s)} = \left(s \cdot \mathbb{I} - \underline{\underline{A}}\right)^{-1} \cdot \underline{\underline{B}} \tag{4}$$

Diese Strecke lässt sich nun in die in BCS-Raumrichtung betrachtete Strecke $\underline{\underline{G}}_{S\_R}$ und eine Transformation der Stellgrößen ins BCS mittels der Transformationsmatrix $\underline{\underline{T}}$ unterteilen. In gleicher Weise kann mit dem geschlossenen Regelkreis durch einsetzten des Stellgesetzes $\underline{u} = -\underline{\underline{K}} \cdot \underline{x} + \underline{\underline{N}} \cdot \underline{w}$ in die Zustandsraumdarstellung umgegangen werden. Die Übertragungsmatrix des geschlossenen Regelkreises im Führungsfall wird so zu:

$$\underline{\underline{G}}_W(s) = \frac{\underline{X}(s)}{\underline{W}(s)} = \left(s \cdot \mathbb{I} - \underline{\underline{A}} + \underline{\underline{B}} \cdot \underline{\underline{K}}\right)^{-1} \cdot \underline{\underline{B}} \cdot \underline{\underline{N}} \tag{5}$$

Der Zustandsregler und der Vorfilter setzen sich ebenfalls aus einem in den Raumrichtungen wirkenden Anteil $\underline{\underline{K}}_R$ und $\underline{\underline{N}}_R$ und der Transformation der Stellgrößen auf die Aktorpositionen durch die inverse Transformationsmatrix $\underline{\underline{T}}^{-1}$ zusammen. Dies lässt sich im Bildbereich durch die Übertragungsmatrizen $\underline{\underline{K}}(s)$ und $\underline{\underline{N}}(s)$ bzw. $\underline{\underline{K}}_R(s) = \underline{\underline{T}} \cdot \underline{\underline{K}}(s)$ und $\underline{\underline{N}}_R(s) = \underline{\underline{T}} \cdot \underline{\underline{N}}(s)$ des Reglers und Vorfilters ausdrücken. Das geregelte System lässt sich zur Betrachtung in den Raumrichtungen umformen [16].

3.5 Identifikation der Systemdynamik im Frequenzbereich

Da das Systemverhalten bisher nur theoretisch aus physikalischer Modellbildung hergeleitet wurde, soll nun durch Frequenzgangmessungen am realen System das Übertragungsverhalten der Strecke identifiziert werden. Weil das zu identifizierende System instabil ist, wird zunächst die Führungsübertragungsfunktion des durch einen linearen Regler stabilisierten Systems identifiziert und daraus das Übertragungsverhalten der ungeregelten Strecke berechnet werden. Da die Berechnung eines Frequenzgangs nur bei linearen Systemen anwendbar ist, wird zur Identifikation das an einem passenden Arbeitspunkt linearisierte Model verwendet. Diese Model soll durch Variation der Parameter an den gemessenen Frequenzgang angepasst werden. Dabei sollen nur so wenig freie Parameter wie möglich verwendet werden. Das Übertragungsverhalten des Reglers und Vorfilters ist bekannt und ergibt sich aus den bei der Messung verwendeten Regelparametern. Die Übertragungsfunktion des Aktors soll zunächst separat durch eine Frequenzgangmessung identifiziert werden, damit sie bei der Identifikation der Führungsübertragungsfunktion als bekannt angenommen werden kann. Die Aktorübertragungsmatrix entspricht dann der Führungsübertragungsmatrix der Stellgrößenregelung im BCS (X- Y- und Z-Richtung):

$$\underline{\underline{G}}_{M\_R}(s) = \underline{\underline{T}} \cdot \begin{bmatrix} G_{m1}(s) & 0 & 0 \\ 0 & G_{m2}(s) & 0 \\ 0 & 0 & G_{m3}(s) \end{bmatrix} \cdot \underline{\underline{T}}^{-1} \tag{6}$$

Dies führt auf eine weitere Möglichkeit der Darstellung der Übertragungsmatrix im Führungsfall mit den separierten Übertragungsfunktionen im Bezug zum BCS:

$$\begin{aligned} \underline{\underline{G}}_W(s) = &\left(\mathbb{I} + \underline{\underline{G}}_{S\_R}(s) \cdot \underline{\underline{G}}_{M\_R}(s) \cdot \underline{\underline{K}}_R(s)\right)^{-1} \\ &\cdot \underline{\underline{G}}_{S\_R}(s) \cdot \underline{\underline{G}}_M(s) \cdot \underline{\underline{N}}_R(s) \end{aligned} \tag{7}$$

Setzt man die beiden Führungsübertragungsmatrizen gleich, so kann man also über die Identifikation dieser zunächst durch herausrechnen der bekannten Übertagungsmatrizen des Zustandsreglers und ggf. Vorfilters sowie der im Vorfeld gemessenen und transformierten Übertragungsmatrix des Aktors auf die Übertragungsmatrix der ungeregelten Strecke schließen. Durch Gleichsetzten der Gl. (5) mit Gl. (7) ist es so auch möglich Rückschluss auf die am Arbeitspunkt definierte Dynamikmatrix sowie Eingangsmatrix des Zustandsraum zu ziehen und innerhalb der durch die Systemparameter gebildeten Koeffizienten dieser auch einzelne, unbekannte Parameter der Strecke zu identifizieren.

Da das ungeregelte und somit auch das geregelte System nichtlinear ist, hat die Amplitude der Anregung einen Einfluss auf das Systemverhalten, da diese das Verstärkungs- und Superpositionsprinzip grundsätzlich nicht erfüllen [17]. Es werden daher Frequenzgangmessungen mit verschiedenen Anregungsamplituden durchgeführt, um diesen Einfluss zu untersuchen.

Hierbei sind typische nicht-linearitäten mechanischer Systeme z.B. Reibung in Form von Coulumbscher bzw. Stribeck- Reibung, Kraftübertragungsabriss, Abhängigkeiten von Systemzuständen z.B. vom Gravitationsvektor/Geschwindigkeitsvektor und variable Koeffizienten [18]. Da diese Effekte innerhalb des Frequenzspektrums sowohl von der Anregungsamplitude als auch von der Anregungsfrequenz abhängig sind,

muss die Messung in Teilfrequenzbändern durchgeführt und anschließend mithilfe der Bewertung der Kohärenz zu einer Messung zusammengeführt werden.

Grundsätzlich wird die Struktur des physikalischen Modells genutzt um einen Ansatz für den zu erwartenden Frequenzgang unter Nutzung bekannter Parameter simulativ zu generieren. Der simulierte Verlauf des Amplituden- und Phasengangs wird mit dem Verlauf der Messkurve verglichen. Die Differenz aus Simulation und Messung werden als quadratische Fehlerfläche ε sowohl der Amplituden- als auch Phasenmesspunkte dargestellt und ergibt sich unter Einbeziehung des kohärenzabhängigen Gewichtungsfaktors k_{eval}.Das eigentliche Identifizieren der unbekannten Modellparameter erfolgt durch eine über Schätzungen initialisierte Optimierung dieser Fehlerfläche mittels eines modifizierten Downhill-Simplex-Algorithmus nach Nelder und Mead [19].

4 Verwendete Prüfstandinfrastruktur

Zur Validierung des vorgestellten Konzepts wurde der in Abbildung 3 dargestellte Prüfstand aufgebaut in welchem der einleitend vorgestellte Funktionsträger S-Mobile unter sicheren und reproduzierbaren Bedingungen getestet werden kann.

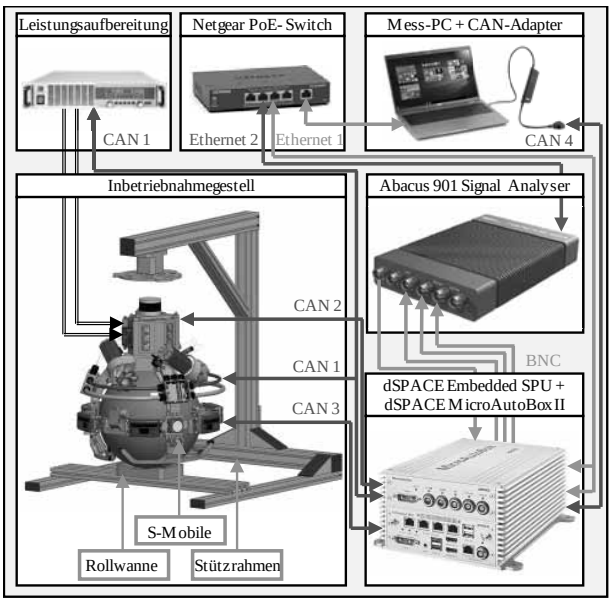


Abbildung 3: S-Mobile in Prüfstandsaufbau zur Identifikation.

Das S-Mobile ist in diesem Prüfstand in einem rekonfigurierbaren Inbetriebnahmegestell gefesselt, welches entweder eine Sperrung oder Begrenzung der Freiheitsgrade des Aufbaus ermöglicht, ohne dabei die Freiheitsgrade der Kugel zu hemmen. Der Versuchsträger ist durch Drahtseile mit einer definierten Länge so fixiert, dass eine Kippbewegung bis zu einem Maximalwinkel zugelassen wird. So kann die Stabilisierung des Aufbaus durch eine translatorische Bewegung des geometrische Zentrums der Kugel bei gleichzeitiger Kippsicherung erfolgen.

5 Beispielhafte Anwendung des Verfahrens am S-Mobile

Mittels des im vorherigen Kapitel vorgestellten Prüfstands wurden frequenzabhängige Analysen am Funktionsträger S-Mobile durchgeführt. Dazu wurden Soll-Zustände in Form von periodische Anregungen als Führungsgrößen an den Stabilisierungsregler gegeben. Anschließend wurde die Antwort des geschlossenen Regelkreises, bestehend aus Regler und Strecke sowie Aktuator und Sensorik ausgewertet. Die periodischen Anregungen wurden systemspezifisch ausgewählt um spezifische Nicht-Linearitäten zu berücksichtigen. So hat die Amplitude der Anregung einen Einfluss auf das Systemverhalten. Es werden daher Frequenzgangmessungen mit verschiedenen Anregungsamplituden in einem festen Frequenzbereich durchgeführt, um diesen Einfluss zu untersuchen. Zur Anregung werden Amplituden von $\hat{\vartheta}_{x\_\mathrm{soll}} = 1{,}5^\circ$ bis $\hat{\vartheta}_{x\_\mathrm{soll}} = 5^\circ$ als Chirp-Signale eingespeist.

Abbildung 4 zeigt die aufgenommen Antwortspektren als Frequenzgang im Bode-Diagramm. Hier wurden bereits Messreihen gewählt deren Kohärenz im gesamten Spektrum mindestens 0,7 beträgt. Es ist zu sehen, dass die Kohärenz der Messung ab ca. 20Hz beginnt signifikant schlechter zu werden. Dies ist auf die Datenübertragung mittels CAN-Bussystem sowohl vom RCP-System zu den Aktuatoren als auch von der Sensorik zurückzuführen. So wurden die Ist-Aufbauwinkel von einer Inertialen Messeinheit gemessen und mit einer Zykluszeit von 10ms per CAN übertragen. Dies entspricht einer Abtastfrequenz von 100Hz, wobei sich nach dem WKS-Abtasttheorem [20] theoretisch eine maximale abtastbare Frequenz von 50Hz ergibt. In der Praxis ist das rekonstruierte Signal allerdings schon bei geringeren Frequenzen verzerrt, was in einer geringeren Kohärenz der Signale resultiert. Bei höheren Amplituden konnte der Frequenzgang nur für geringe Fre-

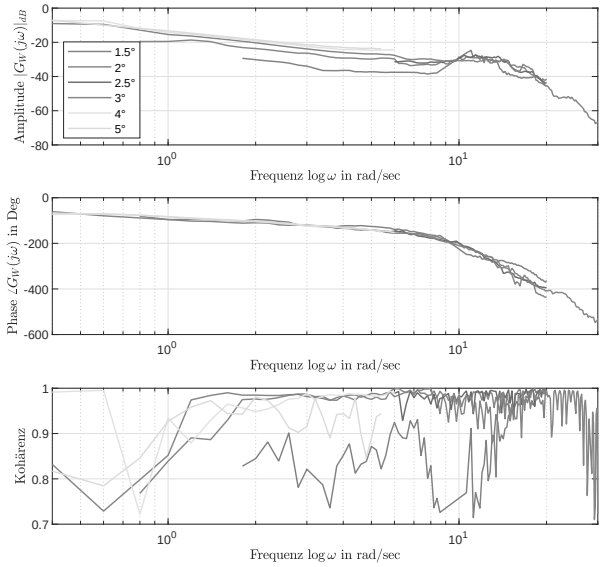


Abbildung 4: gemessene Frequenzgänge bei verschiedenen Amplitudenanregungen, gefiltert nach Kohärenz.

quenzen bestimmt werden, da bei einer Anregung mit einer hohen Frequenz und hohen Amplitude die Räder aufgrund der begrenzten Kraftübertragung durchdrehen und dabei die Kugeloberfläche beschädigen. Die Amplitudengänge liegen für große Anregungsamplituden dicht beieinander, während das System für kleine Anregungen bei niedrigen Frequenzen (<10Hz) eine geringe Verstärkung und zudem eine schlechtere Kohärenz aufweist. Dies ist auf den großen Einfluss der Haftreibungen im System bei so geringen Anregungsamplituden zurückzuführen. Der Phasengang zeigt hingegen für alle Messungen weniger Abweichungen. Die Identifikation der optimalen Parameter erfolgt über die Minimierung des quadratischen Fehlers zwischen einem angenommen Übertragungsglied als Führungsübertragungsfunktion des separierten Übertragungsweges innerhalb der Übertragungsmatrix zu dem gemessenen Frequenzgang. Da sowohl die Übertragungsfunktion des Aktuators als auch die Übertragungsfunktionen des Reglers (und Vorfilters) bekannt sind, kann anhand des berechneten Referenzmodells auf die Struktur des Übertragungsgliedes rückgeschlossen werden. Dies ist ebenso anhand der Beobachtung des gemessen Amplituden- und Phasenabfall abschätzbar.

Zur Identifikation der Parameter muss zunächst die Streckenübertragungsfunktion $G_{sx}(s) = \frac{\theta_x(s)}{M_x(s)}$ durch Linearisierung des nichtlinearen Models mit dem Massenträgheitsmoment als freien Parameter gebildet werden. Das Model soll an dem quasistabilen Punkt $(\vartheta_x = 0, \vartheta_y = 0)$ linearisiert werden, da das System bei der Frequenzgangmessung um diesen Punkt schwingt. Die Winkelgeschwindigkeit wird für die Linearisierung allerdings nicht mit $\dot{\vartheta}_x = 0$ angenommen, sondern soll der tatsächlichen Winkelgeschwindigkeit bei der Messung entsprechen. Diese ist bei der Messung allerdings nicht konstant, sondern hängt von der Frequenz und Amplitude der Schwingung ab. Bei einer harmonischen Eingangsschwingung $\vartheta_x(t)$ als Anregungssignal ergibt sich die Winkelgeschwindigkeit nach folgender Gleichung:

$$\vartheta_x(t) = A(\omega) \cdot \hat{\vartheta}_{x\_\text{soll}} \cdot \sin(\omega \cdot t + \varphi(\omega))$$
$$\dot{\vartheta}_x(t) = \frac{d\vartheta_x(t)}{dt} = A(\omega) \cdot \hat{\vartheta}_{x\_\text{soll}} \cdot \omega \cdot \cos(\omega \cdot t + \varphi(\omega))$$

Für den Nulldurchgang am quasistabilen Punkt um den das System schwingt ($\vartheta_x(t) = 0$), ergibt sich die maximale Winkelgeschwindigkeit:

$$\dot{\vartheta}_x(\omega) = A(\omega) \cdot \hat{\vartheta}_{x\_\text{soll}} \cdot \omega \tag{8}$$

Die folgende Abbildung 5 zeigt die jeweils zum gemessenen Frequenzgang zugehörigen maximalen Winkelgeschwindigkeiten. Diese werden für die Linearisierung des Referenzmodells genutzt, um über den in Kapitel 3 dargestellten Weg die Übertragungsmatrix herzuleiten. Aus dieser wiederum ist die Struktur und die Parameterabhängigkeit ersichtlich, die nun nach einem unbekannten Parameter aufgelöst werden kann. Durch

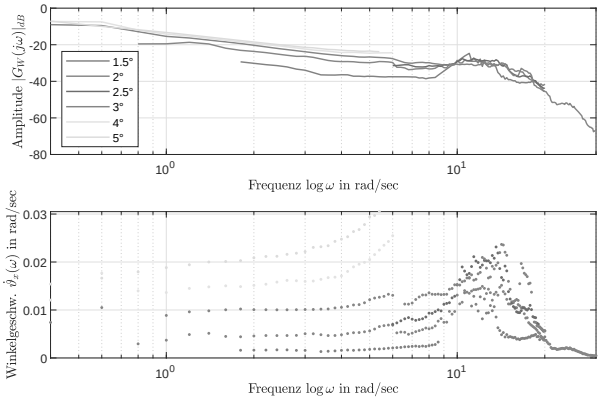


Abbildung 5: Winkelgeschwindigkeit im Nulldurchgang.

Linearisierung des nicht-linearen Models am betrachteten Arbeitspunkt kann z.B. eine vom Massenträgheitsmoment θ_x abhängige Streckenübertragungsfunktion hergeleitet werden um dieses als freien Parameter über die quadratische Optimierung zu identifizieren:

$$G_{sx}(s) = \frac{\vartheta_x(s)}{\tau_x(s)} = f(\sigma, \omega, \theta) \tag{9}$$

In Kombination mit dem bekannten/identifizierten Aktuator(-system) und den Übertragungsfunktionen des Reglers kann so die Führungsübertragungsfunktion für den gemessenen Freiheitsgrad gebildet werden. Die Übertragungsfunktion der Zustandsrückführung ergibt sich immer als ein PDn-Glied, da der Zustandsregler grundsätzlich alle Minimalkoordinaten und mindestens deren erste Ableitung rückführt [21]. Weiterhin wird dem Übertragungsglied eine Totzeit T_t beigestellt, die aus dem Datenübertragungsvorgang resultiert.

$$G_w(s) = \frac{\vartheta_x(s)}{\vartheta_{x\_\text{soll}}(s)} = \frac{N_{Rx} \cdot G_{sx}(s) \cdot G_m(s)}{1 + G_{sx}(s) \cdot G_m(s) \cdot K_{Rx}(s)} \cdot e^{-T_t \cdot s}$$

Der Frequenzgang des angesetzten Übertragungsglieds mit freien initialisierten Parametern ist in der folgenden Abbildung 6 in grün dargestellt. Durch die quadratische Optimierung resultiert der rote Verlauf nach Parameteranpassung. Durch Minimierung des Fehlers des Amplituden- und Phasengangs in Bezug auf die Messung wurden so unbekannte Werte für die freien Parameter des Beispielsystems identifiziert. Ein Abgleich mit den Konstruktionsdaten des Systems bestätigen die Richtigkeit des identifizierten Massenträgheitsmoments $\theta_x = 20,31$. Nach dieser Methode ist also eine Identifikation unbekannter Parameter an unteraktuierten, instabilen und nicht-linearen Systemen möglich.

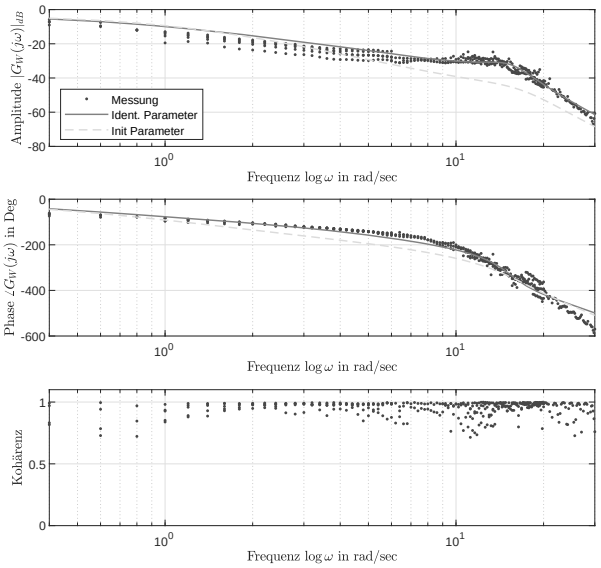


Abbildung 6: Identifiziertes Übertragungsglied für den Führungsfall.

6 Zusammenfassung und Ausblick

In diesem Beitrag wurde die Identifikation und Regelung von instabilen, unteraktuierten Systemen mit nicht-linearem dynamischen Verhalten diskutiert. Der Schwerpunkt lag auf der Entwicklung und Validierung eines neuen methodischen Vorgehens zur Systemidentifikation. Am Beispiel des S-Mobile, einem innovativen Intralogistiksystem, konnte erfolgreich demonstriert werden, wie durch die Anwendung des neuen Ansatzes die Systemdynamik präzise identifiziert und das System stabilisiert werden kann.

Für zukünftige Arbeiten bietet sich eine weitergehende Untersuchung des Vorgehens an, um eine noch größere Bandbreite an instabilen, unteraktuierten Systemen abdecken zu können. Insbesondere die Einbindung von fortschrittlichen, maschinellen Lernmethoden und Surrogate Modellen [22] könnte die Genauigkeit weiter verbessern und die Adaptivität der Regelungssysteme erhöhen. Zudem wäre die Untersuchung der Übertragbarkeit der entwickelten Methoden auf andere industrielle Anwendungen von großem Interesse, um deren universelle Einsetzbarkeit zu testen.

Acknowledgement

Gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur unter Fördernummer ZN3495 im Niedersächsischen Vorab der VolkswagenStiftung und betreut vom Zentrum für digitale Innovationen (ZDIN).

ZDIN Zentrum für digitale Innovationen Niedersachsen
VolkswagenStiftung
Niedersächsisches Ministerium für Wissenschaft und Kultur

Referenzen

[1] Schoukens J, Ljung L. Nonlinear System Identification: A User-Oriented Road Map. *IEEE Control Systems*. 2019;39(6):28–99.

[2] Liu-Henke X, Göllner M, Tao H. An intelligent control structure for highly dynamic driving of a spherical electrical drive. In: *2017 Twelfth International Conference on Ecological Vehicles and Renewable Energies (EVER)*. Piscataway, NJ: IEEE. 2017; pp. 1–10.

[3] Liu-Henke X, Jacobitz S, Göllner M, Scherler S, Yarom OA, Zhang J. A Holistic Methodology for Model-based Design of Mechatronic Systems in Digitized and Connected System Environments. In: *ICSOFT 2021*. [Setúbal, Portugal]: SCITEPRESS - Science and Technology Publications, Lda., 2021. 2021;.

[4] Gevers M. Identification for Control: From the Early Achievements to the Revival of Experiment Design*. *European Journal of Control*. 2005;11(4-5):335–352.

[5] Liu-Henke X. *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab: Zugl.: Paderborn, Univ., Diss., 2005*, vol. 589 of *Fortschritt-Berichte VDI Reihe 12, Verkehrstechnik/Fahrzeugtechnik*. Düsseldorf: VDI-Verl., als ms. gedr ed. 2005.

[6] Isermann R. *Identifikation dynamischer Systeme*. Springer-Lehrbuch. Berlin: Springer, 2nd ed. 1992.

[7] Ljung L. Frequency Domain Versus Time Domain Methods in System Identification – Revisited. In: *Control of uncertain systems: modelling, approximation, and design*, edited by Francis BA, vol. 329 of *Lecture notes in control and information sciences*, pp. 277–291. Berlin and Heidelberg: Springer. 2006;.

[8] Pintelon R, Schoukens J. *System Identification: A Frequency Domain Approach*. Wiley. 2012.

[9] Nelles O. *Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes*. Cham: Springer International Publishing, 2nd ed. 2020.

[10] van den Hof P. Closed-loop issues in system identification. *Annual Reviews in Control*. 1998; 22:173–186.

[11] Xavier J, Patnaik SK, Panda RC. Process Modeling, Identification Methods, and Control Schemes for Nonlinear Physical Systems – A Comprehensive Review. *ChemBioEng Reviews*. 2021;8(4):392–412.

[12] Chawla I, Singla A. ANFIS based system identification of underactuated systems. *International Journal of Nonlinear Sciences and Numerical Simulation*. 2020; 21(7-8):649–660.

[13] Chen R, Srivastava A, Yin M, Smith RS. Closed-Loop Identification of Stabilized Models Using Dual Input-Output Parameterization.

[14] González RA, Rojas CR, Pan S, Welsh JS. Refined instrumental variable methods for unstable continuous-time systems in closed-loop. *International Journal of Control*. 2023;96(10):2527–2541.

[15] Isermann R. *Mechatronische Systeme*. Berlin, Heidelberg: Springer Berlin Heidelberg. 2008.

[16] Liu-Henke X, Scherler S, Göllner M, Maisik J, Fritsch M. Simulationsgestützte Konzeption der Antriebstopologie eines fahrerlosen Transportfahrzeugs. In: *Tagungsband Workshop 2018 ASIM/GI-Fachgruppen*, edited by Loose T, ARGESIM Report, pp. 169–174. Wien: ARGESIM Verlag. 2018;.

[17] Wittenburg J. *Dynamics of multibody systems*. Berlin and New York: Springer, 2nd ed. 2008.

[18] Göllner M, Liu-Henke X, Frerichs L. Analyse und Simulation des Kraftübertragungsverhaltens von Mecanum-Rädern. In: *Proceedings ASIM SST 2020*, edited by Deatcu C, Lückerath D, Ullrich O, Durak U, ASIM Mitteilung, pp. 89–98. Wien: ARGESIM. 2020;.

[19] Isermann R, Münchhof M. *Identification of Dynamic Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg. 2011.

[20] Seibt P. *Algorithmic Information Theory: Mathematics of Digital Information Processing*. Signals and communication technology. s.l.: Springer-Verlag, 1st ed. 2006.

[21] Karrenberg U. *Signale – Prozesse – Systeme*. Berlin, Heidelberg: Springer Berlin Heidelberg. 2017.

[22] Göllner M, Yarom OA, Fritz J, Liu-Henke X. Rechenzeitoptimierte Modellierung nicht-linearer physikalischer Prozesse mit Surrogate Modellen zur Anwendung in echtzeitfähigen Optimierungsverfahren. In: *Virtueller ASIM Workshop 2021*, edited by Liu-Henke X, Durak U, ASIM Mitteilung. Vienna: ARGESIM/ASIM. 2021;.

Studie zum Einsatz der ereignisdiskreten Simulation in Produktion und Logistik

Robin Sutherland[1*], Felix Özkul[1], Sigrid Wenzel[1]

[1]Fachgebiet für Produktionsorganisation und Fabrikplanung, Universität Kassel, Kurt-Wolters-Straße 3, 34125 Kassel, Deutschland; **robin.sutherland@uni-kassel.de*

Abstract. Die Simulation hat sich in vielen Unternehmen als modellgestützte Analysemethode für den Bereich Produktion und Logistik etabliert. Um die aktuelle Marktdurchdringung, den Nutzen der Simulation für Unternehmen und den Entwicklungsstand von simulationsbezogenen Themen abzuschätzen, wurde 2013 eine empirische Studie durchgeführt. Dabei wurden neben vielen positiven Einflüssen auch Forschungs- und Entwicklungsbedarfe identifiziert. Um die Entwicklungen der letzten zehn Jahre und den aktuellen Stand dieser Themen zu untersuchen, wird eine Anschlussstudie durchgeführt. Das Studiendesign und die aktuellen Studienergebnisse werden in diesem Beitrag vorgestellt und in vergleichender Weise mit den Ergebnissen von 2013 diskutiert.

Einleitung

Die ereignisdiskrete Simulation wird seit vielen Jahren als Analysemethode in Produktion und Logistik eingesetzt [1] und hat sich branchenübergreifend bei Unternehmen bewährt [2]. Sie ermöglicht Unternehmen experimentelle Untersuchungen ihrer komplexen Produktions- und Logistiksysteme unter Berücksichtigung eines stochastischen Zeitverhaltens sowie Entscheidungen auf Basis nachvollziehbarer und abgesicherter Planungsergebnisse zu treffen [1]. So führt die Durchführung einer Simulationsstudie bei Unternehmen spätestens im Betrieb zu Kostenersparnissen [1] und damit zu einer verbesserten Wettbewerbsfähigkeit. Aber eine konkrete monetäre Bewertung des Nutzens und damit auch der Wettbewerbsfähigkeit ist für Unternehmen schwierig [2]. Um den Beitrag der Simulation zur Wettbewerbsfähigkeit von Unternehmen zu identifizieren und den aktuellen Stand der Entwicklung zur Simulation in Produktion und Logistik zu analysieren, haben Wenzel und Peter (2013) eine empirische Studie durchgeführt [2]. Die Studienergebnisse zeigen, dass die Simulation einen signifikanten Beitrag zur Wettbewerbsfähigkeit von Unternehmen leisten kann, aber es noch weitere Forschungs- und Entwicklungsbedarfe gibt, um die technische Unterstützung des Einsatzes von Simulation zu verbessern [2]. Deshalb wird im Rahmen dieses Beitrags überprüft, ob die damals aufgezeigten Defizite heute abgebaut sind und wie sich der aktuelle Stand des Einsatzes der ereignisdiskreten Simulation in Produktion und Logistik in den letzten zehn Jahren entwickelt hat. Dazu wird die empirische Studie, deren Aufbau und Ergebnisse nachfolgend vorgestellt werden, in leicht angepasster Form erneut durchgeführt. Neben einer Beurteilung der qualitativen Einflüsse der Simulation auf die Wettbewerbsfähigkeit von Unternehmen sowie der Diskussion des aktuellen Standes von simulationsbezogenen Themen werden auch die Einsatzbereiche und eingesetzten Werkzeuge erhoben. Abschließend werden die Ergebnisse und daraus geschlossene Erkenntnisse in einem Fazit reflektiert.

1 Studiendesign

Die im Juni 2023 durchgeführte Studie wendet sich an Simulationsanwender und -experten, um den Nutzen der ereignisdiskreten Simulation sowie den Stand der Entwicklung in Produktion und Logistik abzuschätzen. Zudem werden die aktuellen Ergebnisse mit den Ergebnissen der bereits im Jahre 2013 durchgeführten Studie [2] verglichen. Die Studie basiert auf einem Online-Fragebogen, der – wie 2013 – an die Mitglieder der Arbeitsgemeinschaft Simulation (ASIM), an die Mitglieder der Fachausschüsse 204 „Modellierung und Simulation“ und 205 „Digitale Fabrik“ (sowie allen zugeordneten Richtlinienausschüssen) der Gesellschaft für Produktion und Logistik des Vereins Deutscher Ingenieure (VDI-GPL) sowie an Vertreter aus der Automobilindustrie per E-Mail versendet wurde. So konnten ca. 250 – 300 Personen erreicht werden.

1.1 Aufbau des Online-Fragebogens

Der Online-Fragebogen ist ähnlich aufgebaut wie bei der bereits 2013 durchgeführten Studie [2]. Insgesamt umfasst der Fragebogen drei Fragengruppen:

1. Einsatz ereignisdiskreter Simulation in Unternehmen
2. Stand der Technik der ereignisdiskreten Simulation
3. Eigenschaften der befragten Institution

Die erste Fragengruppe umfasst allgemeine Fragen zum Einsatz der ereignisdiskreten Simulation in Unternehmen. Durch diese Fragen werden potenzielle Einsatzbereiche der Simulation im Bereich Produktion und Logistik (z. B. Produktionsplanung) ermittelt. Weiterhin werden die Teilnehmenden gefragt, seit wann sie die ereignisdiskrete Simulation nutzen, welche Simulationswerkzeuge zum Einsatz kommen und ob sie mit diesen Werkzeugen zufrieden sind.

Zusätzlich werden in der ersten Fragengruppe verschiedene durch die Simulation beeinflussbare Faktoren der Wettbewerbsfähigkeit (z. B. die absolute Planungsdauer) im Hinblick auf eine positive, keine oder gar eine negative Veränderung durch den Simulationseinsatz bewertet. Zudem erfolgt eine qualitative Einschätzung, ob durch den Simulationseinsatz Einsparungen oder Mehrkosten erzielt werden können.

In der zweiten Fragengruppe wird der Stand der Technik der ereignisdiskreten Simulation thematisiert, um die Eignung für verschiedene simulationsbezogene Themen (z. B. 3D-Visualisierung oder Digitale Fabrikzwillinge) zu bewerten. Die ausgewählten Themen orientieren sich an der Studie von 2013 [2] und werden um Inhalte aus den Thesen und Handlungsfeldern für die Simulation in Produktion und Logistik des VDI-Fachausschusses 204 „Modellierung und Simulation“ ergänzt [3]. Die Bewertung der Themen soll ihre Einordnung in einen Gartner Hype Cycle (zum Gartner Hype Cycle vgl. [4]) ermöglichen. Diese Bewertung wird durch eine zeitliche Einordnung (i. e. wie lange das Thema schon in der Community diskutiert wird) ergänzt. Die Zeitintervallklassen sind „weniger als 2 Jahre“, „2 – 5 Jahre“, „6 – 10 Jahre“ und „seit mehr als 10 Jahren“. Zusätzlich können Themen als „nicht einschätzbar“ oder als nicht relevant („wird nicht diskutiert“) bewertet werden.

Die letzte Fragengruppe beschäftigt sich mit den Eigenschaften der befragten Institution. Dafür werden die Branchenzugehörigkeit, die Anzahl an Mitarbeitenden, der Jahresumsatz sowie der eigene Tätigkeitsbereich der befragten Person abgefragt.

1.2 Teilnehmerrücklaufquote und Einordnung der Unternehmen

Von den erreichten Personen haben 50 Personen mindestens eine Frage beantwortet und 31 haben den Fragebogen vollständig beantwortet. Damit ist die Teilnehmerzahl im Vergleich zur Studie von 2013 (73 Teilnehmende) deutlich geringer. Da auch die unvollständigen Fragebögen in die Bewertung einbezogen werden und bei einzelnen Fragen auch Enthaltungen erlaubt sind, ist der variierende Stichprobenumfang bei den Auswertungen jeweils als "n" angegeben.

Tabelle 1 zeigt die prozentuale branchenbezogene Verteilung der Befragten.

Branche	2013	2023
Hochschulen	10 %	35 %
Automobilindustrie	13 %	25 %
Logistik-/Transportgewerbe	4 %	13 %
Ingenieurbüro	4 %	6 %
IT-Dienstleister	10 %	6 %
Maschinenbau	8 %	6 %
Sonstiges	5 %	3 %
Außeruniversitäre Forschungseinrichtung	27 %	3 %
Flugzeugbau	0 %	3 %
Elektronik	6 %	0 %
Metallerzeugung und -bearbeitung	5 %	0 %
Automobilzulieferer	3 %	0 %
Chemieindustrie	3 %	0 %
Baugewerbe	1 %	0 %
Sonstige verarbeitende Gewerbe	1 %	0 %

Tabelle 1: Vergleich der Aufteilung der Befragten nach ihrer Branchenzugehörigkeit (in Prozent; n=32)

Mit 34 % der Teilnehmenden bilden die Hochschulen die größte Gruppe, gefolgt von der Automobilindustrie (25 %) und dem Logistik-/Transportgewerbe (13 %). Im Vergleich zum Jahr 2013 ist in allen drei Branchen ein deutlicher Zuwachs zu verzeichnen.

Einen bemerkenswerten Rückgang gibt es hingegen bei Teilnehmenden aus der außeruniversitären Forschung (-24 %). Im Vergleich zu 2013 sind Teilnehmende aus der Elektronikbranche, Metallerzeugung und -bearbeitung, Chemieindustrie, dem Baugewerbe oder sonstigen verarbeitenden Gewerbe sowie Automobilzulieferer nicht mehr vertreten.

Neben der branchenbezogenen Verteilung der Teilnehmenden lässt sich auch eine deutliche Veränderung bei der Institutionsgröße (s. Tabelle 2) und beim Jahresumsatz (s. Tabelle 3) erkennen.

Anzahl Mitarbeitende	2013	2023
> 249	61 %	79 %
50 – 249	15 %	12 %
< 50	24 %	9 %

Tabelle 2: Vergleich der Aufteilung der Befragten nach ihrer Institutionsgröße (in Prozent; n=33)

79 % der Befragten gehören zu Großunternehmen, was einem Zuwachs von 18 % im Vergleich zum Jahr 2013 entspricht. Demgegenüber ist ein Rückgang bei den kleinen Unternehmen (< 50 Mitarbeitende) um 15 % zu verzeichnen.

Jahresumsatz [in €]	2013	2023
> 50 Mio. €	59 %	64 %
10 – 50 Mio. €	6 %	27 %
< 10 Mio. €	35 %	9 %

Tabelle 3: Vergleich der Aufteilung der Befragten nach ihrem Jahresumsatz (in Prozent; n=33)

Bei der Betrachtung des Jahresumsatzes bilden Unternehmen mit mehr als 50 Millionen Euro Umsatz die größte Gruppe (64 %), gefolgt von Unternehmen mit einem Umsatz zwischen 10 und 50 Millionen Euro (27 %). Der Anteil der Unternehmen mit weniger als 10 Millionen Euro Umsatz ist unter den Befragten von 35 % auf 9 % zurückgegangen.

Neben den Vertretern aus der Industrie- oder Dienstleistungsbranche sind auch Teilnehmende in der Forschung tätig (s. Tabelle 4). Diese stellen in der Studie 2023 den größten Teilnehmerkreis. Die beschriebenen Veränderungen in der Zusammensetzung des Teilnehmerkreises sind bei den Auswertungen der Studie zu berücksichtigen.

Tätigkeitsfelder	2013	2023
Forschung	22 %	46 %
Industrie	54 %	39 %
Dienstleister	24 %	15 %

Tabelle 4: Vergleich der Aufteilung auf die Tätigkeitsfelder der Institutionen (in Prozent; n=33)

2 Stand zum Einsatz der Simulation in Unternehmen

In diesem Kapitel werden die einzelnen Einsatzbereiche und die verwendeten Simulationswerkzeuge näher untersucht, um den aktuellen Stand zum Einsatz der ereignisdiskreten Simulation in Unternehmen zu beschreiben.

2.1 Einsatzbereiche der ereignisdiskreten Simulation

Die verschiedenen Einsatzbereiche der ereignisdiskreten Simulation sind absteigend entsprechend ihrer prozentualen Häufigkeit (Mehrfachnennungen möglich) in Tabelle 5 aufgelistet. Neben den aktuellen Ergebnissen sind auch die Studienergebnisse von 2013 im Vergleich dargestellt.

Am häufigsten wird die ereignisdiskrete Simulation in der Produktionsplanung (68 %) eingesetzt. Zudem werden die Layoutplanung (64 %), die Unternehmenslogistik (56 %) und die Systemplanung (50 %) häufig genannt. Diese Einsatzbereiche waren auch schon 2013 relevant, wobei die prozentuale Häufigkeit für diese Bereiche nochmal gestiegen ist. Insbesondere der Zuwachs bei der Systemplanung (+28 %) ist nennenswert.

Weitere relevante Einsatzbereiche sind die Fabrikstrukturplanung (38 %), die Verbesserung im laufenden Betrieb (30 %), die unternehmensübergreifende Logistik (28 %) sowie die Entwicklung und Testung von Steuerungssoftware (24 %).

Einsatzbereiche	2013	2023
Produktionsplanung	50 %	68 %
Layoutplanung	43 %	64 %
Unternehmenslogistik	49 %	56 %
Systemplanung	22 %	50 %
Fabrikstrukturplanung	30 %	38 %
Verbesserung im laufenden Betrieb	38 %	30 %
Unternehmensübergreifende Logistik	19 %	28 %
Entwicklung und Test von Steuerungssoftware	21 %	24 %
Emulation	17 %	18 %
Anlagenrealisierung	20 %	16 %
Inbetriebnahme	16 %	16 %
Störfallmanagement	11 %	14 %
Personalschulung	14 %	4 %
Keine Angabe	0 %	4 %
Sonstiges	18 %	0 %

Tabelle 5: Vergleich der Einsatzbereiche der ereignisdiskreten Simulation (in Prozent; n=49)

2.2 Simulationseinsatz und Werkzeugnutzung

Die ereignisdiskrete Simulation ist ein etabliertes Werkzeug in der Praxis. Über 80 % der Befragten setzen die Simulation bereits seit über 10 Jahren ein. Das zeigt, dass

die Methode eine gewisse Reife besitzt und sie als wertvoll für die Analyse von Produktions- und Logistiksystemen angesehen wird. Dementsprechend ist es auch nachvollziehbar, dass im Vergleich zum Jahr 2013 der Anteil der Befragten, die die ereignisdiskrete Simulation schon über 10 Jahre einsetzen, um 24 % gestiegen ist. Dies deutet auf eine zunehmende Akzeptanz und Verbreitung der Methode hin. Damit verbunden sind die Anteile in den anderen Kategorien leicht gesunken.

Bei der Wahl der Simulationswerkzeuge (Tabelle 6; Mehrfachnennungen möglich) zeigt sich ein klarer Trend: Plant Simulation (74 %) und AnyLogic (30 %) sind die am häufigsten verwendeten Simulationswerkzeuge. Im Vergleich zum Jahr 2013 ist der Anteil der Anwender von AnyLogic um über 20 % gestiegen. Somit ist im deutschsprachigen Raum eine Marktverschiebung zu beobachten.

Simulationswerkzeug	2013	2023
Plant Simulation	62 %	74 %
AnyLogic	7 %	30 %
DOSIMIS-3	9 %	11 %
AutoMod	7 %	11 %
Emulate3D	-	9 %
SimPy	-	6 %
SLX	4 %	-
FlexSim	3 %	-
Enterprise Dynamics	3 %	2 %
Arena	3 %	-
Eigenentwicklungen	11 %	2 %
Sonstige	33 %	40 %

Tabelle 6: Vergleich der Simulationswerkzeuge (in Prozent, Mehrfachnennungen möglich; n=43)

Der Anteil an Eigenentwicklungen hat am stärksten abgenommen (-9 %). Dies könnte mit den hohen Kosten sowie dem Zeitaufwand für die Entwicklung eigener Simulationswerkzeuge oder der zunehmenden Verfügbarkeit von kommerziellen Werkzeugen und dem damit verbunden Support der Softwareanbieter zusammenhängen. Zudem hat der Anteil an Open Source Software (z. B. SimPy) zugenommen.

Werkzeuge mit weniger als drei Einzelnennungen werden unter dem Punkt „Sonstige“ (z. B. Vensim, Matlab oder Witness) zusammengefasst. Die hohe Anzahl an weiteren Werkzeugen deutet darauf hin, dass es in der Praxis keine eindeutige Präferenz für ein einzelnes Werkzeug gibt, sondern dass die Auswahl von verschiedenen Faktoren (z. B. den vorhandenen Unternehmensressourcen oder den persönlichen Präferenzen der Anwender) abhängt. Auch die Gründe für den Einsatz eines neuen Simulationswerkzeugs sind sehr individuell. So nennen die Befragten bspw. die persönliche Weiterbildung oder die Reduzierung der Abhängigkeit von einem Hersteller als Gründe. Zudem beziehen sich die Gründe auf einen bestimmten Funktionsumfang oder die Spezialisierung auf eine Anwendung.

3 Beurteilung qualitativer Einflüsse der Simulation auf die Wettbewerbsfähigkeit von Unternehmen

Um den Nutzen des Simulationseinsatzes in den beschriebenen Einsatzbereichen zu bewerten, erfolgt in diesem Kapitel eine Bewertung von qualitativen Faktoren und eine Analyse, inwieweit diese die Wettbewerbsfähigkeit eines Unternehmens beeinflussen. Dafür sollen die Befragten beurteilen, ob die einzelnen Faktoren eine Verbesserung, Verschlechterung oder keine Veränderung der Wettbewerbsfähigkeit verursachen. Die abgefragten Faktoren sind in Abbildung 1 entsprechend ihrer prozentualen Häufigkeit der Nennungen „Verbesserung“ aufsteigend sortiert.

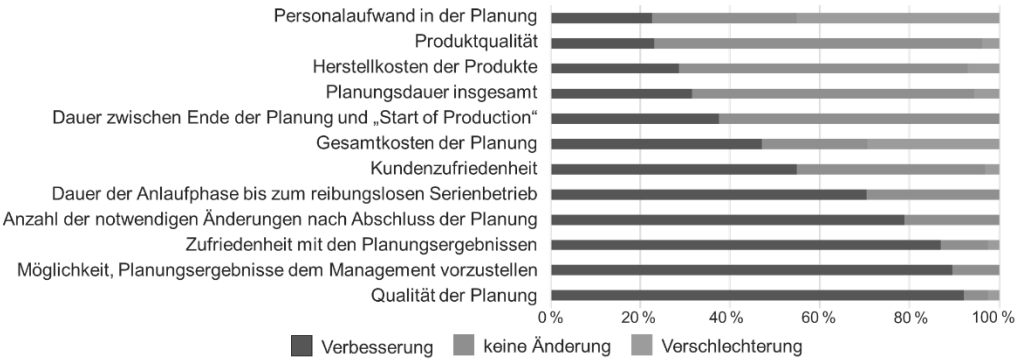


Abbildung 1: Vergleich der Faktoren in Bezug auf die Wettbewerbsfähigkeit von Unternehmen (in Prozent; n=45)

Bei der Betrachtung der Ergebnisse fällt auf, dass nur der Personalaufwand (45 %) und die Gesamtkosten der Planung (29 %) laut den Befragten einen sichtbaren negativen Einfluss auf die Wettbewerbsfähigkeit von Unternehmen aufgrund des Simulationseinsatzes aufweisen. Diese Faktoren wurden auch schon 2013 von den Befragten als negative Einflüsse eingestuft. Lediglich die Planungsdauer (6 %) wird von den Befragten 2023 nicht mehr größtenteils als negativer Faktor gesehen. Verbesserungen werden insbesondere hinsichtlich der Dauer der Anlaufphase bis zum reibungslosen Serienbetrieb (70 %), der Anzahl der notwendigen Änderungen nach Abschluss der Planung (79 %), der Möglichkeit, Planungsergebnisse dem Management vorzustellen (89 %), der Zufriedenheit mit den Planungsergebnissen (87 %) und der Qualität der Planung (92 %) gesehen. Auch diese Ergebnisse sind vergleichbar mit denen von 2013. Die Faktoren haben zwar keine direkten monetären Auswirkungen, führen aber zu einer transparenten und effizienteren Planung, wodurch langfristig die Wettbewerbsfähigkeit eines Unternehmens gesteigert werden kann. So schafft bspw. die Präsentation der Planungsergebnisse eine transparente Basis für fundierte Entscheidungen des Managements.

Keine Einflüsse auf die Wettbewerbsfähigkeit werden bei der Planungsdauer (63 %), der Dauer zwischen Ende der Planung und dem „Start of Production (SOP)“ (63 %), den Herstellkosten der Produkte (64 %) und der Produktqualität (73 %) gesehen. Bei der Kundenzufriedenheit sind sich die Befragten uneins, ob sie einen positiven Einfluss (52 %) oder keinen Einfluss (42 %) auf die Wettbewerbsfähigkeit von Unternehmen hat.

Allgemein ist im Vergleich zu den Studienergebnissen aus 2013 der prozentuale Anteil der Ausprägung „Verschlechterung“ bei jedem Faktor zurückgegangen. Daraus kann geschlossen werden, dass sich die ereignisdiskrete Simulation als Planungsunterstützung unabhängig von der Unternehmensgröße oder der Branche etabliert hat und ihr Nutzen positiv bewertet wird.

Eine einfache Möglichkeit, die Vorteilhaftigkeit des Simulationseinsatzes aufzuzeigen, besteht in der Bewertung des Aufwands und Nutzens. Demnach ergeben sich nach Angabe der Befragten Einsparungen durch den Simulationseinsatz (86 %); 56 % davon sehen sogar hohe oder sehr hohe Einsparungen. 11 % sehen keine Veränderung und 4 % sogar Mehrkosten. Der Vergleich zu den Ergebnissen von 2013 zeigt, dass insgesamt der prozentuale Anteil der Einsparungen leicht gestiegen (+12 %) ist (s. Abbildung 2).

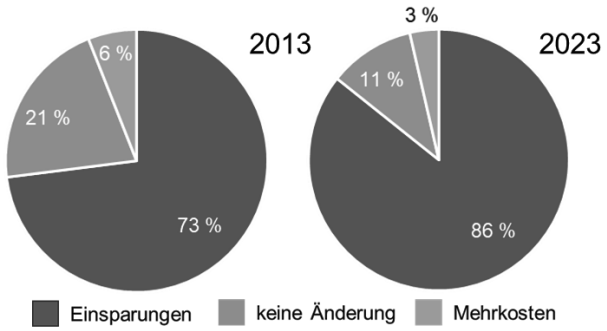


Abbildung 2: Vergleich Einsparungen oder Mehrkosten durch Simulation (in Prozent; n=34)

4 Aussagen zum Stand der Simulation im produktiven Einsatz und zu möglichen technischen Defiziten

Die zweite Fragengruppe beschäftigt sich mit der Relevanz und dem Umsetzungsgrad von aktuellen Themen der Simulations-Community. Durch diese Untersuchung soll festgestellt werden, ob die ereignisdiskrete Simulation für Aufgaben befähigt ist („im produktiven Einsatz“) oder ob es noch technische Defizite gibt. Die Auswahl der Themen orientiert sich an der Studie von 2013 [2] und Entwicklungsthemen (mit einem * markiert), die aus den VDI-Thesen und Handlungsfeldern [3] abgeleitet sind. Insgesamt werden 45 Themen (vgl. Tabelle A1 in [5]) abgefragt, die in den Gartner Hype Cycle eingeordnet werden. Die ersten drei Phasen (1) „erste Ideen“, (2) „überzogener Hype“ und (3) „kein Interesse (mehr)“ beschreiben den anfänglichen Hype und die einhergehende Ernüchterung bei der Einführung von neuen Technologien. Die beiden letzten Phasen (4) „Relevanz für den produktiven Einsatz erkannt“ und (5) „im produktiven Einsatz“ beschreiben die Etablierung einer Technologie und ihre Nutzung in der Praxis. Diese Phasen dienen auch im Fragebogen als Antwortmöglichkeiten.

Das Vorgehen für die Einordnung der Themen orientiert sich an dem Vorgehen von 2013 [2]. Dabei werden nur die Themen in den Gartner Hype Cycle eingeordnet, die von über 50 % der Befragten einschätzbar sind und mehrheitlich einer der sechs Antwortmöglichkeiten (bzw. zwei benachbarten Antwortmöglichkeiten) zugeordnet werden können. In Tabelle 7 sind alle Themen, die entweder 2013 oder 2023 als nicht einschätzbar bewertet werden, dargestellt. In der aktuellen Studie werden die

Themen „High Level Architecture (HLA)“, „Komponentenorientierung“, „Model Driven Architecture“, „Modellmanagementsysteme“ und „Modellreduktionsverfahren“, die für die Mehrheit der Befragten (≥ 50 %) nicht einschätzbar sind, nicht weiter betrachtet.

Thema	2013	2023
Agentenorientierte Simulation	42 %	17 %
Agile Modellentwicklung	51 %	20 %
Data Farming	62 %	21 %
High Level Architecture	54 %	59 %
Komponentenorientierung	41 %	56 %
Model Driven Architecture	63 %	79 %
Modellmanagementsysteme	30 %	50 %
Modellreduktionsverfahren*	-	62 %
Offene und skalierbare Simulationswerkzeuge	44 %	41 %
Serious Games	63 %	33 %

Tabelle 7: Themen, die 2013 oder 2023 als nicht einschätzbar bewertet werden (in Prozent; n=35)

Die geringe Einschätzbarkeit der HLA ist ggf. auf die geringen Anwendungspotenziale im Bereich Produktion und Logistik zurückzuführen.

Im Vergleich zu 2013 sind die agentenorientierte Simulation (17 %), die agile Modellentwicklung (20 %) und das Data Farming (21 %) von einer höheren Anzahl an beteiligten Personen einschätzbar. Die zunehmende Bedeutung der agentenorientierten Simulation zeigt sich auch in den letzten Tagungsbänden der ASIM [6, 7].

Für die verbleibenden 40 Themen wird die Häufigkeitsverteilung der Antworten über die sechs Ausprägungen untersucht (vgl. Tabelle A2 in [5] für die Häufigkeitstabelle). Für die Themen in der folgenden Aufzählung kann dementsprechend kein Konsens (Mehrheit) für eine Ausprägung bzw. zwei benachbarte Ausprägungen abgeleitet werden, wodurch eine eindeutige Positionierung auf dem Gartner Hype Cycle nicht möglich ist für:

- Agile Modellentwicklung
- Data Farming
- Einbindung in die Digitale Fabrik
- Ganzheitliche, interagierende Simulationsmodellelemente*
- Interoperabilität
- Offene und skalierbare Simulationswerkzeuge
- Reduktion des Rechenzeitbedarfs*
- Simulation als Service
- Simulation im Internet
- Simulation und Digitale Produktzwillinge*
- Simulation und Künstliche Intelligenz (KI)*
- Simulations- und Modellbildungskompetenz für Mitarbeiter:innen in kleinen und mittleren Unternehmen*
- Simulationsbasierte Assistenzsysteme
- Standardisierte, durchgängig und vollautomatisiert nutzbare Domänenmodelle*
- Vernetzung der Simulation mit dem realen System über entsprechende Auto-ID-Techniken zur Nutzung der aktuellen Produktions- und Logistikdaten*
- Verteilte Modellierung und Simulation
- Verzahnung unterschiedlicher Berechnungsverfahren*

Bei den aufgelisteten Themen fällt auf, dass einige neue (im Vergleich zur Studie von 2013) Themen aufgeführt sind. Dies zeigt, dass die Themen in der Community zwar diskutiert werden, aber noch keine Einigkeit über den aktuellen Entwicklungsstand besteht. So werden bspw. mit dem kombinierten Einsatz von Simulation und KI, dem Data Farming, den simulationsbasierten Assistenzsystemen oder der Interoperabilität sowohl der produktive Einsatz als auch lediglich erste Ideen assoziiert.

Die verbleibenden 23 Themen werden entsprechend ihrer mehrheitlich zugeordneten Ausprägung in den Gartner Hype Cycle eingeordnet (s. Abbildung 3). Falls zwei Ausprägungen gleich oft genannt werden, wird das Thema zwischen den beiden Ausprägungen positioniert (z. B. Virtual Reality). Die Sortierung innerhalb einer Ausprägung erfolgt aufsteigend anhand der prozentualen Häufigkeit (vgl. Tabelle A3 in [5]). Zusätzlich ist die Abbildung um eine Zeitspanne ergänzt. Dabei werden die meisten Themen bereits seit über 10 Jahren diskutiert.

Bis auf die domänenspezifischen Modellierungssprachen werden alle verbleibenden Themen als relevant betrachtet. Das Thema Serious Games wird zwar schon sehr lange in der Community diskutiert, bisher konnte aber die Relevanz für den produktiven Einsatz noch nicht erkannt werden.

Für das Thema „selbstlernende Simulationsmodelle“ sind erste Ideen vorhanden, die Aufmerksamkeit bezüglich datenbasierter Themen nimmt in der Community zu, sie werden aber noch nicht seit über 10 Jahren in der Community diskutiert. Der Einsatz von Virtual Reality wurde bereits 2013 als überzogener Hype eingeordnet. Technologische Weiterentwicklungen haben dazu geführt, dass die Technologie noch nicht ins Tal der Enttäuschung gefallen ist und von einigen Befragten immer noch als relevant beurteilt wird. Deshalb ist weitere For-

schung notwendig, damit die Relevanz für den produktiven Einsatz gewährleistet werden kann.

Für die meisten Themen in Abbildung 3 wird die Relevanz für den produktiven Einsatz erkannt oder sie sind bereits im produktiven Einsatz. So werden marktfähige Simulationsumgebungen mit domänenspezifischen Bausteinbibliotheken für den Simulationseinsatz angeboten. Für die erfolgreiche Anwendung der Simulation stehen zudem Vorgehensmodelle für die Durchführung einer Simulationsstudie sowie für die Verifikation und Validierung zur Verfügung. Um die Simulationsergebnisse darzustellen, wird die 3D-Visualisierung im produktiven Einsatz verwendet. Zusätzlich befindet sich der kombinierte Einsatz von Simulation und Datenanalysen (neues Thema) bereits im produktiven Einsatz. Im Vergleich zum Jahr 2013 können mittlerweile auch die Modelldokumentation und die automatische Experimentdurchführung dem produktiven Einsatz zugeordnet werden. Die meisten kommerziellen Simulationswerkzeuge verfügen über Funktionen, um automatisiert Experimente zu planen, Parameter zu variieren und die Ergebnisse zu analysieren. Zudem ist in der Forschung die Aufwandsreduktion bei der Experimentdurchführung weiterhin Gegenstand der Forschung (vgl. [8]), wodurch die Relevanz des Themas weiter bestehen bleibt.

Neben der automatischen Experimentdurchführung könnten auch bereits bei der Erstellung eines Simulationsmodells Aufwände reduziert werden, indem die manuelle und zeitintensive Modellierung automatisiert wird. Bereits in der Studie von 2013 wurde die Relevanz der automatischen Modellgenerierung für die Praxis erkannt. Auch in der Forschung beschäftigen sich seit Jahren einige Autoren mit diesem Thema [9, 10]. Dennoch hat sich noch keiner der entwickelten Ansätze für den produktiven Einsatz bewährt, da sie häufig auf einen Anwendungsfall zugeschnitten und nur mit einem hohen Aufwand übertragbar sind [9]. Zudem ist es schwierig, das komplexe Systemverhalten von Produktions- und Logistiksystemen zu generieren, weil die bisherigen Ansätze bereits Probleme bei der Umsetzung von einfachen Kontrollflussstrategien haben [9]. Deshalb hat sich weiterhin noch kein Ansatz oder Standard im produktiven Einsatz durchgesetzt.

Ähnlich verhält es sich mit den Referenzmodellen, der simulationsbasierten Steuerungsentwicklung und der Eingangsdatenqualität, welche sich – trotz der erkannten Relevanz vor zehn Jahren – noch nicht im produktiven Einsatz befinden.

Zusätzlich konnten neue relevante Themen für den produktiven Einsatz identifiziert werden. So werden die Emulation und die agentenorientierte Simulation, mittlerweile im produktiven Einsatz gesehen. Darüber hinaus hat insbesondere der Simulationseinsatz im Kontext des Digitalen Fabrikzwillings in der Forschung und Praxis an Aufmerksamkeit gewonnen. Trotz einer fehlenden einheitlichen Definition zeigen einige Forschungsarbeiten die Potenziale einer Anwendung auf [bspw. 11]. Für die Überführung der Technologie in den produktiven Einsatz ist aber noch Forschungsbedarf vorhanden.

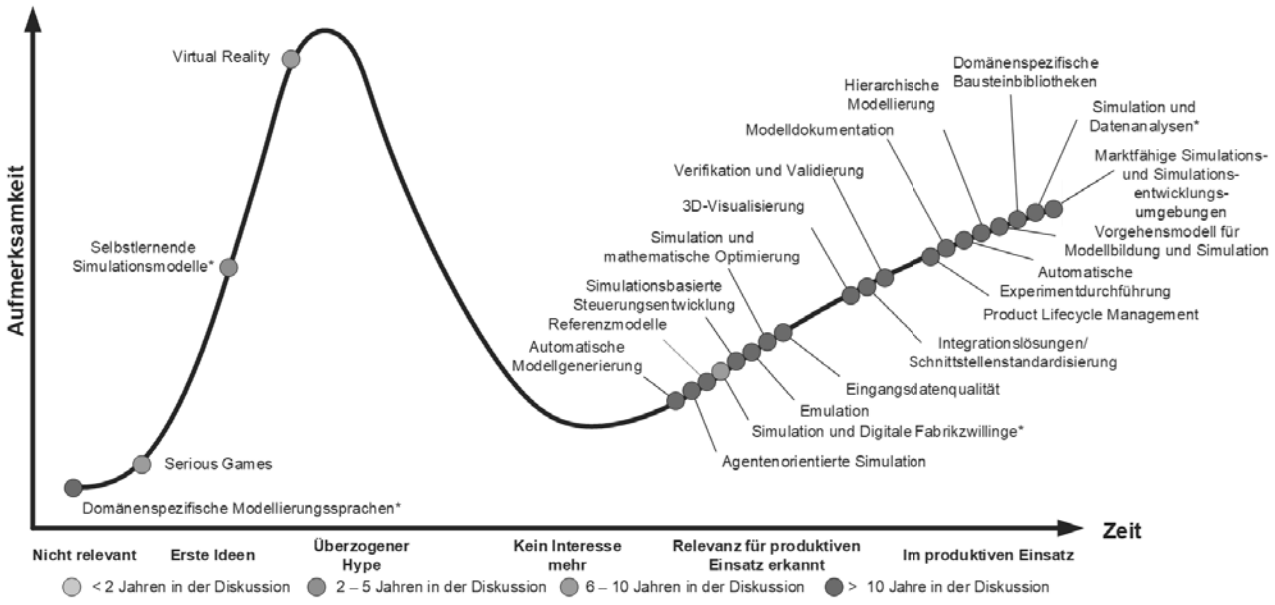


Abbildung 3: Gartner Hype Cycle für simulationsbezogene Themen in 2023 (n=35)

5 Fazit

Die Ergebnisse der durchgeführten empirischen Studie zeigen, dass Unternehmen bereits seit vielen Jahren die ereignisdiskrete Simulation für verschiedene Einsatzbereiche anwenden. Die Ergebnisse lassen zudem Aussagen zum Einfluss der Simulation auf die Wettbewerbsfähigkeit von Unternehmen sowie eine Untersuchung des aktuellen Standes der Simulation und dessen Veränderungen im Vergleich zum Jahr 2013 zu. Durch den Vergleich der beiden Studienergebnisse können Aussagen über die Entwicklung von simulationsbezogenen Themen getroffen werden. Einige der Themen haben in den letzten Jahren einen erheblichen Fortschritt erfahren, was durch die hohe Anzahl an Themen auf dem Gartner Hype Cycle, die bereits im produktiven Einsatz sind und einen hohen Nutzen für Unternehmen bieten, bestätigt werden kann.

Jedoch gibt es auch Themen, die von den Befragten nicht eindeutig zugeordnet werden können (z. B. Simulation im Internet) oder größtenteils nicht eingeschätzt werden können (z. B. Modellreduktionsverfahren). Die Potenziale dieser Themen müssen durch weitere Forschungen gehoben und ihre Entwicklung weiterverfolgt sowie durch Forschende publiziert werden. Darüber hinaus gibt es auch Themen, die sich im Vergleich zu den Studienergebnissen von 2013 nicht weiterentwickelt haben. Für diese Themen (z. B. automatische Modellgenerierung) müssen die themenspezifischen Gründe identifiziert werden, damit auch diese Themen in den produktiven Einsatz gelangen. Zudem können auch themenübergreifende Gründe – wie bspw. die fehlende Übertragbarkeit von Forschungsergebnissen auf andere Einsatzbereiche oder die fehlende Werkzeugunabhängigkeit von Unternehmen – eine Hürde für eine breite Anwendung mancher Themen in der Praxis sein. Für eine weiterführende Analyse des aktuellen Standes der simulationsbezogenen Themen könnten zukünftig auch die Antworten in Abhängigkeit der Unternehmensgröße oder Branchen ausgewertet werden.

Allgemein zeigen die Studienergebnisse aber wieder einen positiven Einfluss der ereignisdiskreten Simulation auf die Wettbewerbsfähigkeit von Unternehmen. Deshalb kann davon ausgegangen werden, dass sie auch in Zukunft als Analysemethode eingesetzt wird sowie die untersuchten Themen weiterentwickelt und neue Themen hinzukommen werden.

Danksagung

Wir bedanken uns ganz herzlich bei allen Befragten der Studie für die aufgebrachte Zeit und das bemerkenswerte Engagement.

Literatur

[1] Gutenschwager K, Rabe M, Spieckermann S, Wenzel S. *Simulation in Produktion und Logistik: Grundlagen und Anwendungen*. Berlin, Heidelberg: Springer Vieweg; 2017.

[2] Wenzel S, Peter T. Simulation zur Sicherstellung der Wettbewerbsfähigkeit – Ergebnisse einer Umfrage zur Simulation in Produktion und Logistik. In: Friedewald A, Lödding H (Hrsg.). *Produzieren in Deutschland – Wettbewerbsfähigkeit im 21. Jahrhundert*. Berlin: GITO mbH Verlag; 2013. S. 233-264.

[3] Furmans K, Hanschke T, Möller DPF, Rabe M, Wenzel S, Zabel A, Zisgen H. *Simulation für Produktion und Logistik*. Düsseldorf: VDI-Gesellschaft Produktion- und Logistik; 2021.

[4] Gartner Inc. Gartner *Hype Cycle – Wie man Technologie-Hype interpretiert*. https://www.gartner.de/de/methoden/hype-cycle (abgerufen: 19.07.2024).

[5] Sutherland R, Özkul F, Wenzel S. *Studie zum Einsatz der ereignisdiskreten Simulation in Produktion und Logistik (Forschungsdaten)*. DOI: 10.48662/daks-41

[6] Breitenecker F, Deatcu C, Durak U, Körner A, Pawletta T (Hrsg.). *ASIM SST 2022 – Proceedings Langbeiträge*. Wien: ARGESIM Verlag; 2022.

[7] Bergmann S, Feldkamp N, Souren R, Straßburger S (Hrsg.). *Simulation in Produktion und Logistik 2023*. Ilmenau: Universitätsverlag Ilmenau; 2023.

[8] Wilsdorf P, Heller J, Budde K, Zimmermann J, Warnke T, Haubelt C, Timmermann D, van Rienen U, Uhrmacher AD. A Model-Driven Approach for Conducting Simulation Experiments. *Applied Sciences*. 2022; 12(16): S. 7977. DOI: 10.3390/app12167977

[9] Wenzel S, Stolipin J, Rehof J, Winkels J. Trends in Automatic Composition of Structures for Simulation Models in Production and Logistics. In: Mustafee N, Bae K-HG, Lazarova-Molnar S, Rabe M, Szabo C, Haas P, Son Y-J (Hrsg.). *Proceedings of the 2019 Winter Simulation Conference*. 2019 Winter Simulation Conference; 2019 Dez; National Harbor, MD (USA). New York: IEEE. S. 2190-2200. DOI: 10.1109/WSC40007.2019.9004959

[10] Bergmann S, Straßburger S. Automatische Modellgenerierung – Stand, Klassifizierung und ein Anwendungsbeispiel. In: Mayer G, Pöge C, Spieckermann S, Wenzel S (Hrsg.). *Ablaufsimulation in der Automobilindustrie*. Berlin, Heidelberg: Springer Vieweg; 2020. S. 333-347.

[11] Spoor JM, Weber J. Schematic Categorization and Definition of Applied and Target-oriented Digital Twins. In: Bergmann S, Feldkamp N, Souren R, Straßburger S (Hrsg.). *Simulation in Produktion und Logistik 2023*. 20. ASIM-Fachtagung.; 2023 Sep; Ilmenau. Ilmenau: Universitätsverlag. S. 103-112. DOI: 10.22032/dbt.57795

Simulation-enhanced Action-oriented Process Mining in Production and Logistics

Felix Özkul[1*], Robin Sutherland[1], Sigrid Wenzel[1]

[1]Department Organization of Production and Factory Planning (pfp), University of Kassel, Kurt-Wolters-Str. 3, 34125 Kassel, Germany; **felix.oezkul@uni-kassel.de*

Abstract. Process mining is increasingly being used to gain insights into processes based on operational data. Recently, approaches have been researched as to how these findings can be automatically transferred into process-regulating actions during system operation to correct deviations between the actual and target process in real time. However, the implementation of such action-oriented process mining mechanisms requires sufficient testing of the implemented actions in the application to prevent undesirable side effects in the real system. This article explains how discrete-event simulation in production and logistics can be used to mitigate risks in the context of implementing action-oriented process mining through the use of an emulation model. For this purpose, we present simulation-enhanced action-oriented process mining as well as a proof-of-concept implementation based on a use case.

Introduction

Discrete-event simulation (DES) has proven itself across many industries as a planning tool for production and logistics systems (e.g., see [1][2]). Its application enables users to analyze "what-if" or "how-to-achieve" questions by executing simulation experiments during the planning phase of a production or logistics system. The utilization of simulation during commissioning or system operation is less mature, but the potential benefits of using simulation during these system life cycle phases are also apparent (see [3]). In the following, a distinction is made between the levels of the technical system and the control system with respect to systems in production and logistics. This distinction can also be made with appropriate modeling in the simulation application, so that there is a real technical system, a real control system, a simulated technical system, and a simulated control system. In the context of this paper, *emulation* describes the use of a real control system being used in a simulated technical system (other types of coupling are described in [4]). The simulated technical system thus receives the same inputs that the emulated real technical system would receive, and the real control system can thus be tested simulatively.

DES can be used in combination with other digital methods such as process mining, for example, to reduce the effort involved in creating simulation models, or to simulatively generate synthetic input data for the application of process mining techniques [5]. The added value of the combined use of both methods has been presented in the pertinent scientific literature – also for the area of production and logistics [6].

Process mining is a research area at the intersection of data science and process science [7]. Process mining is mainly used to analyze processes based on past operational data (see [8]) and its use is favored by the increasing prevalence of information technology systems in production and logistics [6]. The classic types of process mining are process discovery, conformance checking (see Section 1), process enhancement, and more recently, performance analysis, comparative process mining, predictive process mining (machine learning-supported process mining), and action-oriented process mining (see Section 1; [7]). These types of process mining have some overlaps in terms of both methodology and application.

In combination with DES, the combined use of both methods can be used to gain insights into both the past and the future. Not part of this given time continuum is the present and the associated consideration of the combined use of methods during ongoing system operation. Input data for process mining techniques is conventionally stored in so-called event logs (see Section 1 for more details). The process information stored in these event logs can be used, for example, to extract models describing process behavior (process discovery), and to retrospectively identify deviations between the target and actual process (conformance checking). However, for process mining to be used in an action-oriented manner in

the current system operation, the process-related insight must be gained in *real time*, i.e., possibly before a process instance is completed and thus before the complete event log is generated. Streaming process mining methods (alternatively online process mining) are suitable for this purpose (see [9]). Streaming process mining analyzes so-called event streams (see Section 1) instead of conventional event logs [9], thereby gaining knowledge about process instances even before they are completed and, if necessary, enabling actions that regulate process instances.

Action-oriented process mining aims to generate these diagnostic-based actions (see [7]), thereby bridging the gap between insight and action that conventional process mining cannot. However, the implementation of such action-oriented process mining-based process control mechanisms has not yet been researched in detail in the area of production and logistics, and the risks associated with the implementation of action-oriented process mining are obvious. For example, in case of failure, testing on the real system may affect its operational performance (e.g., due to unforeseen system failures caused by actions of the action-oriented process mining mechanism). DES could, on the one hand, help mitigate these risks by generating event streams for different process scenarios, which are used as input resources for action-oriented process mining to analyze the different response actions (e.g., an online adjustment of machine parameters or job scheduling) as well as their effects. On the other hand, as an element of decision support, DES can help to determine the impact of non-conformity with regard to process instances to gain predictive insights into the necessary process control actions. This enables users to mitigate the risks associated with the implementation of action-oriented process mining and helps increasing its effectiveness in practice. Our approach uses simulation as an emulation model of an underlying technical system to test an action-oriented process mining mechanism.

The paper is structured as follows: Section 1 explains the key process mining terms. Section 2 specifies theoretical scenarios for *simulation-enhanced action-oriented process mining*. Section 3 presents a use case based on a proof-of-concept implementation. The paper concludes with a summary and a research outlook.

1 Process Mining Terms and Context

A *process* in the context of process mining refers to "a coherent series of changes that unfold over time and occur at multiple levels" ([10], p. 3). These changes are triggered by *events*. Event logs record the execution of processes based on events that start or end *activities*. Process and activity executions represent so-called instances. A process instance is also referred to as a *case*. Each case can be described by a sequence of events or activities, which is logged as a trace in the event log. In other words, event logs are a multiset of traces, with each trace being a sequence of events ([7], def. 3).

Events are assigned to their respective cases using a corresponding case identifier. Events and traces can be described in more detail using additional information (such as resources performing activities). In practical use, event logs log a finite number of events. Different types of process mining can then be applied on the basis of these logged events. Processes (as defined above) can be mapped and analyzed using process models (e.g., models in Petri net notation [11]).

Conformance checking can be used to check how well a process model is able to replay the process behavior observed in an event log. The scientific literature mentions various conformance checking methods, such as footprints, token-based replay (for Petri nets), and alignments [12]. The ability of a process model to reproduce recorded process behavior is referred to as its *fitness* [8][12] and is the most important indicator for describing process model quality (the other process model quality indicators are *precision*, *generalization*, and *simplicity*, see [12]). On the one hand, conformance checking can be used to quantify the quality of a modelled or extracted process model in relation to an event log (i.e., recorded process behavior). On the other hand, it is also possible to analyze the conformity of the logged cases in relation to a normative process model (i.e., a binding model specifying the target process). This approach of conformance checking is central to the implementation of simulation-enhanced action-oriented process mining.

1.1 Streaming Process Mining

The examination of cases using event logs enables an *a posteriori* determination of conformity in relation to a normative process model. However, for a process instance to be corrected during its execution in a production

and logistics system the conformity assessment must be carried out *a priori* with regard to the completion of the process instance. This requirement can be met by analyzing event streams.

Van Zelst et al. ([13], p. 7) define an *event stream* as "a continuous stream of events executed in context of an underlying business process". An event stream comprises a – potentially infinite – number of events (as defined above). The investigation of event streams is the subject of streaming process mining and includes streaming process discovery and online conformance checking [9]. Event stream analysis allows for the processing of very large event logs (like event logs too large to be stored in memory) or for continuous monitoring of processes. The latter is important for our contribution in the context of production and logistics. In principle, event streams can be analyzed using conventional process mining algorithms by combining events into batches and then processing them like event logs. However, there are also dedicated algorithms that enable online analyses (e.g., prefix alignments [14] or the analysis of behavioral patterns ([15], see Section 2) for conformance checking). Van Zelst et al. [13] present an architecture (S-BAR) for the application of common process discovery algorithms in online settings using abstract representations. These techniques focus on the investigation of activity relationships and control flow. In addition, Stertz et al. [16] present an approach that analyzes the temporal relationship between activities using a temporal profile. A temporal profile contains information about the (stochastically) expected durations of activities as well as their temporal distance and a normative process model can be infused with it to allow the application of temporal conformance checking (see Figure 1 "temporal items"). A temporal profile can, for example, be extracted from a normative process execution log (i.e., a log containing valid traces of a target process) which contains information about the start and end times of activities (i.e., start and end events) or on the basis of expert domain knowledge. New events are compared with a temporal profile and time-related outliers are detected by calculating their Z-score [17]. Temporal conformance can also be checked in the event stream in addition to the control flow view. The approach in this paper checks event streams based on behavioral conformance (see [15]) and temporal conformance [16] to monitor process execution online.

1.2 Action-oriented Process Mining

Streaming conformance checking based on event streams enables online monitoring of processes. Action-oriented process mining builds on this and uses streaming process mining to automatically generate actions based on detected deviations from target processes (i.e., non-conformities) that imply risks associated with the violation of process *constraints*. In the context of production and logistics, constraints can be, for example, case-specific production deadlines that are about to be missed (violation of temporal conformance) or the prescribed production sequences that are not being followed in the ongoing process (violation of behavioral conformance).

Park and Van der Aalst [18] provide a comprehensive framework for implementing action-oriented process mining for business processes with a focus on operational support (more in [8]). During the execution of business processes events are logged in a real information system and an event stream is generated which is then analyzed by the *constraint monitor* [18]. The constraint monitor analyzes the event stream given the abovementioned constraints (which, in our approach, are temporal and behavioral constraints), that are formalized in a constraint formula (e.g., in a production setting, one such formula could evaluate if a product is manufactured given a prescribed production sequence). The analysis of events given the constraint formulae yields constraint instances which are then sent as a constraint instance stream to the *action controller* [18]. The action controller evaluates the incoming constraint instances and generates an *action* based on the evaluation result and given an action formula [18]. The action formula takes in the constraint instance stream and a time window to produce a set of *transactions* (i.e., operations which the underlying information system can perform). An example of such an action in a production setting would be to increase an order's priority if production delays (i.e., temporal nonconformance) are diagnosed.

This paper builds on the framework of Park and Van der Aalst [18] and extends it with simulation-related components focusing on the implementation of action-oriented process mining in production and logistics.

Drieschner et al. [19] present an approach that focusses on simulation as a learning tool for action-oriented process mining. The similarity between this approach and our contribution is that both approaches use simulation as a data generator for process mining (see Section 2). However, the focus of Drieschner et al. [19] is pedagogical

and focuses on user interaction. Our approach focuses on the automated testing of an implemented action-oriented process mining mechanism. For this purpose, the simulation model emulates the real system and simulation runs are conducted.

2 Simulation-enhanced Action-oriented Process Mining

Based on the framework of Park and Van der Aalst [18], the following Figure 1 shows a possible architecture for simulation-enhanced action-oriented process mining. During the execution of a simulation model, a simulation-generated event stream is created whose events are recorded by an event monitor and evaluated with regard to their temporal and behavioral conformance. For this purpose, the approach of Stertz et al. [16] is applied on the basis of a normative temporal profile and the formulation of temporal constraints to identify temporal outliers within the event stream. Temporal constraints can be formulated, for example, by target dates that are set for the completion of an order. If, for example, a processing activity occurs too late or takes too long, this deviation is recognized via temporal conformance checking. In addition, the event monitor monitors the behavioral conformance of incoming events. The behavioral conformance is assessed by analyzing *behavioral patterns* (i.e., a set of activities and possible control flow relations [15]) and a reference behavioral model (expressed as the set of *prescribed* behavioral patterns which are expected for the underlying process (see [15]). This is shown as the reference behavioral model in Figure 1. This granular perspective on the control-flow allows an in-vivo analysis of singular observable behavioral patterns (which can be inferred from events but at a higher level of abstraction) during process execution and allows for the calculation of three distinct key indicators (see [15]): C*onformance* (indicating the correctness of the observed behavior), *completeness* (indicating case progression), and *confidence* (indicating the expected stability of the conformance).

Checked events are appended to the checked event stream and relayed to the simulation action controller. Depending on the event status (i.e., whether an event conforms or not), a suitable event routine is selected from the event routine space and communicated to the simulation model. Since actions in the (discrete-event) simulation model are generated by executing event routines, we use the term *event routine space* instead of the action space (cf. [18]). We refer to the *simulation action controller* as the component that fulfils the tasks of the action controller (see [18]). If specific behavioral or temporal non-conformity is detected for an event, a routine for handling these deviating events is automatically selected (if an event conforms, the default event routine is executed). The described workflow is shown in Figure 2.

The effects of the decisions by the simulation action controller can then be observed during the remainder of the simulation runs. Based on the simulation model configuration, event streams containing non-conforming events, i.e., *temporal or behavioral imperfections*, can be

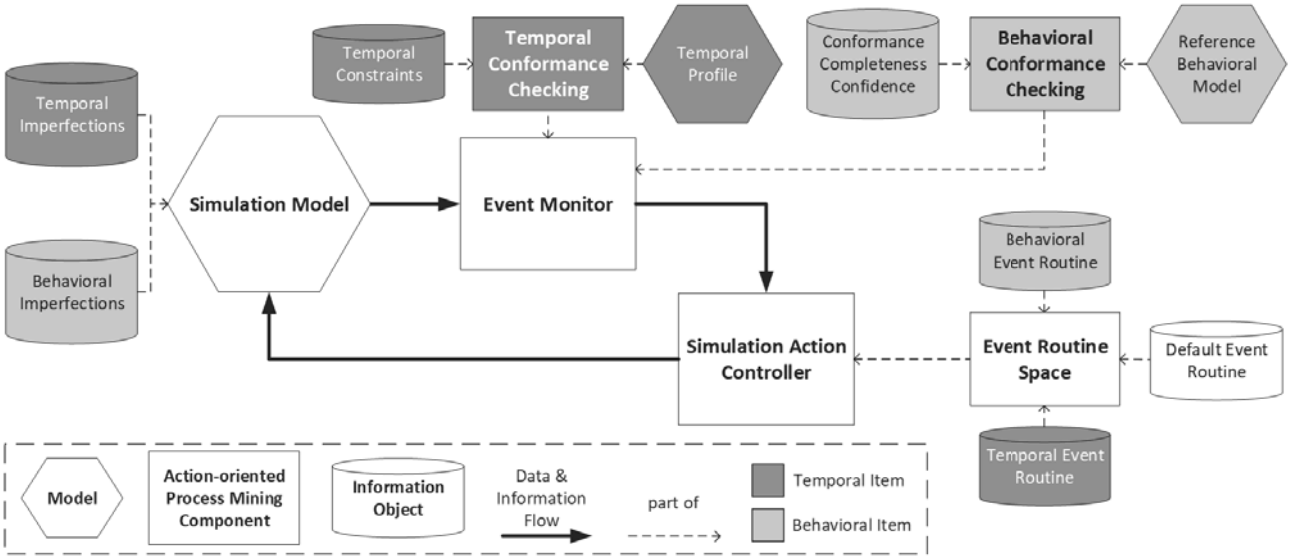


Figure 1: Architecture of the presented approach.

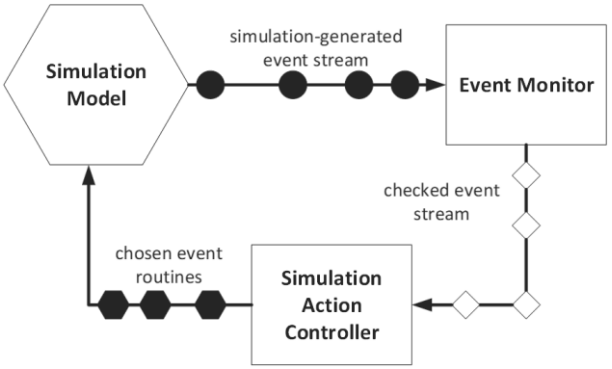


Figure 2: Information flow of the outlined approach.

generated during a simulation run to invoke automatic actions of the simulation action controller, the effects of which are assessed in the simulation.

This allows the action controller to be thoroughly tested simulatively before it is implemented in the real

system. Risks relating to the impairment of process sequences in the real system can be mitigated in this way. The results of the simulation application can be statistically validated through systematic experimental design and, in the context of the implementation of action-oriented process mining, should also capture rare events and faults that are difficult to observe in real system operation. The specific number of simulation runs depends on the specific use case and the number of different scenarios which have to be addressed by the simulation action controller. This ensures that the system behavior for these edge cases is also considered in the action controller of the real system. This is the qualitative added value of the proposed approach: Based on the execution of simulation experiments that cover the range of expected cases of process deviations in the real system, action-oriented process mining can be implemented without having to make any changes to the real system. Instead, the simulation is used as an emulation model for the real control system and risks associated with the action-oriented process mining implementation are mitigated.

3 Use Case

The following use case demonstrates the idea behind simulation-enhanced action-oriented process mining (see Section 1) using a practical case study and implements the architecture in Figure 1.

3.1 Application System

The application system is a conveyor system on a university laboratory scale, Figure 3 shows the corresponding simulation model.

Starting from the source conveyor, load units are fed onto the main conveyor. Load units carry objects that have an object type that determines the target production sequence on the machines M1-M6 and one load unit corresponds to one case. In addition, there are various stopping points (H1-H7) on the main conveyor, which can read and write to RFID tags attached to the load units. In addition to the conveyed object type, other load unit-related information (attributes) is also stored on the RFID tags, such as the priority of the order (high/low), the target time for completion of the order and the next machine in the object type-specific processing sequence. The ma-

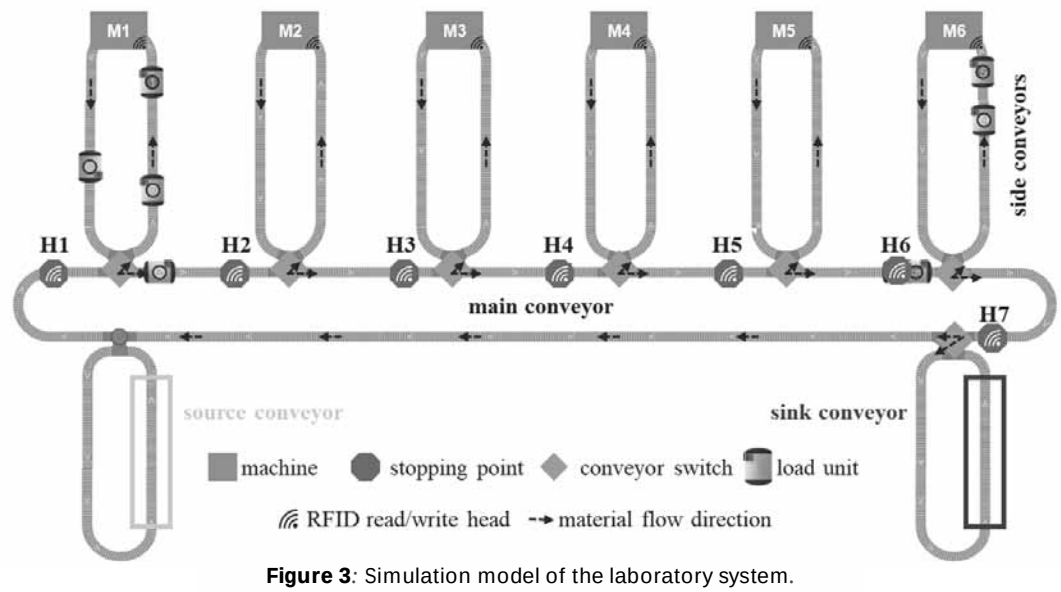


Figure 3: Simulation model of the laboratory system.

chines are located on side conveyors, which are connected to the main conveyor via conveyor switches. Side conveyors have a certain load unit capacity and load units can only be discharged onto a side conveyor if it can accommodate further load units. At the conveyor switches, load units with high priority have right of way; for load units with the same priority, first in first out (FIFO) applies.

We formalize a workflow net (a Petri net with certain properties, see [8]), which is a simplified process model for the behavior of the load units (cases) in the application system (Figure 4). The normative workflow net exhibits maximum fitness in relation to the underlying conveyor system. However, it enables additional behavior that is not possible in the real system due to the object type-specific processing sequences (i.e., its precision is low). To increase the precision of the model, the formalization of color sets and the introduction of transition guards would be appropriate. For simplicity, however, we decided not to include a colored Petri net of the load unit process. Starting from the source ('exit_source'), the various stopping points on the main conveyor are controlled and logging events are fired ('log_at_H*').

Depending on the conveyed object type on the load unit, its processing progress and the availability of side conveyor capacity, the load units are processed at the machines ('processing_on_M*') or conveyed in a waiting loop. Skips are modelled as silent transitions; load units that cannot be processed at their destination machine will still pass the subsequent stopping points on the main conveyor (i.e., it is not possible to skip logging activities, which is coherent with respect to the layout in Figure 3). Furthermore, the process model is infused with a temporal profile, which is based upon domain knowledge about the controls of the reference system. For clarity, we graphically omit the temporal distances between all activities and activity (transition) durations (examples for both are highlighted in blue in Figure 4).

3.2 Use Case

The goal of the use case is to illustrate how the approach can handle incorrect processing sequences (behavioral non-conformance) and processing delays (temporal non-conformance) and how automated actions can address them.

Due to the stochastic order loading of different object types, congestion of load units can occur in the system, which delays the processing of orders. Furthermore, it is assumed that the reading of the RFID tags – which are placed on the load units – is not always error-free, and that read and write errors can occur. A case is considered compliant if its processing sequence corresponds to the target processing sequence of its object type and the order is completed in time. If the target processing sequence is violated, the event monitor detects this deviation as behavioral non-conformance. Delays related to the processing of load units may lead to the detection of temporal non-conformance (see Section 2). In case of behavioral non-conformance, rerouting to the original target machine is defined as an automated action in the action space of the action controller.

In case of temporal non-conformance, the priority of

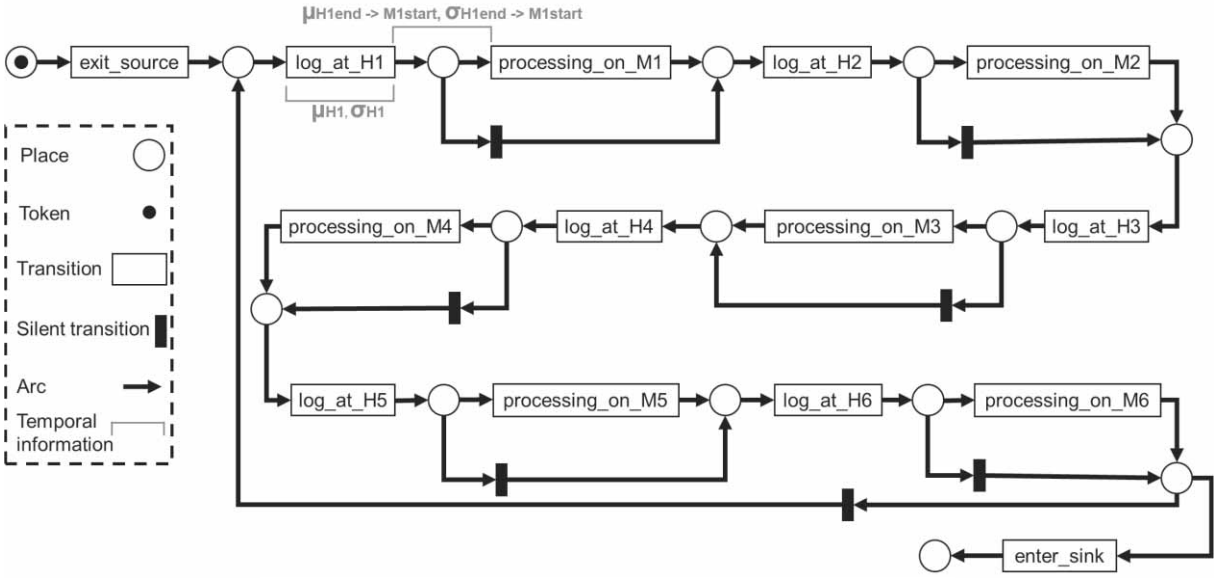


Figure 4: Workflow-net of the load unit process.

a load unit must be increased if there is a delay so that it can be routed to the conveyor switches more quickly if necessary. In the event of temporal non-conformance of a processing activity, it must also be assumed that a machine requires maintenance. A maintenance worker should therefore be sent out in this case to carry out the maintenance as quickly as possible.

3.3 Technical Implementation

We implement the architecture from Figure 1 using AnyLogic 8 and open-source process mining software systems. Figure 5 shows the proof-of-concept implementation.

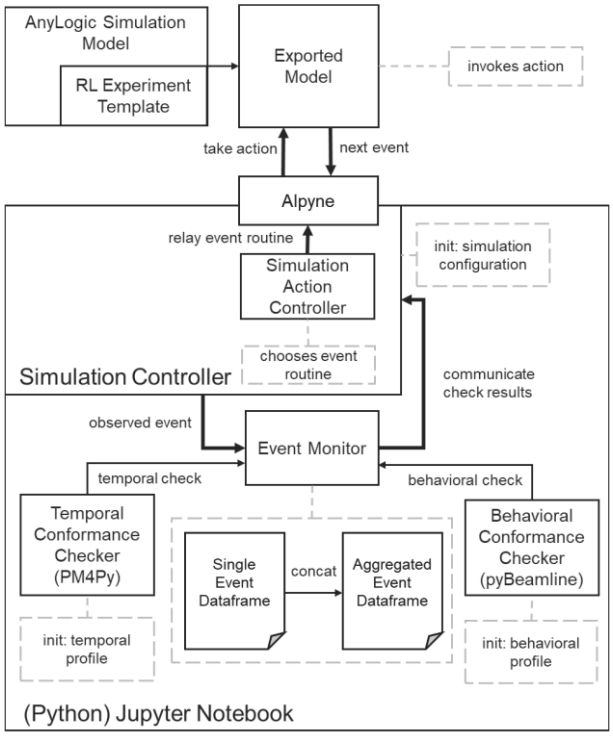


Figure 5: Structure of the proof-of-concept implementation.

First, a simulation model is built and an experiment template is implemented. The current AnyLogic software ecosystem provides a template for Reinforcement Learning (RL) (i.e., the possibility to define observations, actions and configurations) which can be adapted because – similarly to RL – our approach utilizes the observation and action capabilities in the simulation context.

Therefore, we repurpose the RL pipeline to implement simulation-enhanced action-oriented process mining. The simulation model containing the experiment template is exported to a standalone (Java) model, which is bi-directionally linked to a Python Jupyter Notebook using the Alpyne software library [20]. The exported model is configured to invoke actions based on certain key events, those being the logging events shown in Figure 4 at the stopping points (H1-H6). Whenever a logging event occurs, the simulation engine pauses the simulation run and passes the events to the Python simulation controller, which implements Alpyne as its simulation interface. The observed event is then relayed to the event monitor which transforms and logs the observed event in a Pandas (https://pandas.pydata.org/docs/) Dataframe object (‘Single Event Dataframe’) to perform temporal and behavioral conformance checking. Software systems for the implementation of streaming conformance checking (the central component of the event monitor) are currently only available to a limited extent (see [21]). The comprehensive Python-based open-source framework PM4Py [22] implements the temporal conformance checking approach of Stertz et al. [16]. Burattin [21] provides an open-source streaming process mining software named Beamline, which is available for Java and Python (pyBeamline). Our work implements pyBeamline’s behavioral conformance checking [15][21] and provides a compact implementation. Based on a reference event log (which can be generated, for example, by simulating the workflow net shown in Figure 4 or by exporting traces from a valid simulation model after the warm-up phase) a normative behavioral model is initially mined with pyBeamline and new event instances are checked against it. In order to assess the behavioral conformance of a running case, we implement a second Dataframe object to which each event is appended (technically, at each iteration the ‘Single Event Dataframe’ and the ‘Aggregated Event Dataframe’ are concatenated). This step is computationally costly but necessary to contextualize events with regard to the previous events of their running case. Furthermore, each event is checked against the temporal profile of the normative process which is created using PM4Py (the computation of a temporal profile requires start and end timestamps for activities). The PM4Py implementation additionally requires floating point numbers as timestamps whereas other algorithms often work based on datetime formatted timestamps. After conformance checking, the results of the check are relayed to the simulation action controller which then chooses an appropriate event routine (action). If the event monitor does not register non-conformance, the default routing logic is applied to the load unit (see Section 3.1). If, for example, the event monitor detects that a case is being processed late (temporal non-conformance), the simulation action controller increases its priority to enable faster transport

to the processing machines. After selecting a suitable action, it is communicated to the simulation model via Alpyne and the simulation engine continues the simulation run. The control logic of the simulation can be modified by the simulation controller (with appropriate implementation in the simulation model) to specifically generate non-conforming events, for example by generating misreads at the stopping points with a predetermined probability, which ensures that load units are incorrectly ejected or not ejected. In this way, the components of action-oriented process mining can be specifically checked within the framework of simulation experiments with regard to their ability to detect and recommend and thus their suitability as operational support before they are implemented in the real system (see Section 2). This avoids risks associated with undesirable side effects in reality.

4 Summary and Outlook

This article presents a combination approach of discrete-event simulation and process mining, in which a simulation model is used as an emulation model for the implementation of action-oriented process mining. The qualitative added value lies in the mitigation of risks associated with the automated process-regulating actions in production and logistics systems. Furthermore, the use of simulation allows the introduction of targeted imperfections to improve the testing of action-oriented process mining mechanisms. However, the current results can only be seen as a starting point for subsequent research challenges. These concern, among other things, the incorporation of a more dynamic action space and the explicit consideration of data and information quality. The current implementation uses a rules-based mechanism for generating event routines because the underlying use case is of lower complexity than industrial systems. Herein lies a limitation of the proposed approach. Possible research directions in this context include the exploration of action spaces using RL (since the current software implementation is already predestined for it). The consideration of data and information quality is crucial for the successful combined use of simulation and process mining and is therefore a subject for future research as well. Furthermore, the quantitative added value of the presented approach has yet to be shown in more complex production and logistics settings in subsequent studies.

Literature

[1] Gutenschwager K, Rabe M, Spieckermann S, Wenzel S. *Simulation in Produktion und Logistik*. 1st Edition. Berlin, Heidelberg: Springer; 2017. 290 p.

[2] Verein Deutscher Ingenieure. VDI 3633 Blatt 1 – Simulation von Logistik-, Materialfluss- und Produktionssystemen – Grundlagen. 2014, Berlin: DIN Media.

[3] Bicalho-Hoch A L, Özkul F, Wittine N, Wenzel S. A Tool-Based Approach to Assess Simulation Worthiness and Specify Sponsor Needs for SMEs. In Feng B, Pedrielli G, Peng Y, Shashaani S, Song E, Corlu C G, Lee L H, Chew E P, Roeder T, Lendermann P, editors. Proceedings of the 2022 Winter Simulation Conference. *2022 Winter Simulation Conference (WSC)*; 2022 Dec; Singapore. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey. 1818-1829. doi: https://doi.org/10.1109/WSC57314.2022.10015373.

[4] Follert G, Trautmann A. Emulation intralogistischer Systeme. In Wenzel S, editor. Simulation in Produktion und Logistik. 12. ASIM-Fachtagung; 2023 Sep; Kassel. San Diego, Erlangen: Society for Modeling and Simulation International SCS Publishing House e.V. 521-530.

[5] Özkul F, Sutherland R, Wenzel S, Jessen U, Spieckermann S. Verknüpfung von ereignisdiskreter Simulation und Process Mining in Produktion und Logistik. In Breitenecker F, Deatcu C, Durak U, Körner A, Pawletta T, editors. ASIM SST 2022 Proceedings Langbeiträge. 26. ASIM Symposium Simulationstechnik; 2022 Jul; Vienna. ARGESIM Publisher: Vienna. 39-48. doi: https://doi.org/10.11128/arep.20.a2035.

[6] Spieckermann S, Stöhr N, Mayer G, Özkul F, Wenzel S. Fallbeispiele aus Produktion und Logistik für die Verknüpfung von ereignisdiskreter Simulation und Process-Mining. In Bergmann S, Feldkamp N, Souren R, Straßburger S, editors. Simulation in Produktion und Logistik. 20. ASIM-Fachtagung; 2023 Sep; Ilmenau. Ilmenau: Universitätsverlag Ilmenau. 155-165, doi: https://doi.org/10.22032/dbt.57785.

[7] Van der Aalst W M P. Process Mining: A 360 Degree Overview. In: Van der Aalst W M P, Carmona J, editors. *Process Mining Handbook*. 1st edition. Cham: Springer International Publishing; 2022. p 3-34.

[8] Van der Aalst W M P. *Process mining. Data science in action*. 2nd Edition. Berlin, Heidelberg: Springer Publishing; 2016. 467 p.

[9] Burattin A. Streaming Process Mining. In: Van der Aalst W M P, Carmona J, editors. *Process Mining Handbook*. 1st edition. Cham: Springer Publishing; 2022. p 349-372.

[10] vom Brocke J, Van der Aalst W M P, Grisold T, Kremser W, Mendling J, Pentland B, Recker J, Roeglinger M, Rosemann M, Weber B. Process Science: The Interdisciplinary Study of Continuous Change. *SSRN Electronic Journal*. 2021; 1-9. doi: https://dx.doi.org/10.2139/ssrn.3916817.

[11] Reisig W. *Understanding Petri Nets*. 1st Edition, Berlin, Heidelberg: Springer Publishing; 2013. 236 p.

[12] Carmona J, Van Dongen B, Solti A, Weidlich M. *Conformance checking. Relating processes and models*. 1st Edition, Cham: Springer Nature; 2018. 285 p.

[13] Van Zelst S J, Van Dongen B F, Van der Aalst W M P. Event stream-based process discovery using abstract representations. *Knowledge and Information Systems*. 2017; 54: 407-435. doi: https://doi.org/10.1007/s10115-017-1060-2.

[14] Van Zelst S J, Bolt A, Hassani M, Van Dongen B F, Van der Aalst W M P. Online conformance checking: relating event streams to process models using prefix-alignments. *International Journal of Data Science and Analytics.* 2019; 8: 269-284. doi: https://doi.org/10.1007/s41060-017-0078-6.

[15] Burattin A, Van Zelst S J, Armas-Cervantes A, Van Dongen B F, Carmona J. Online Conformance Checking Using Behavioural Patterns. In Weske M, Montali M, Weber I, vom Brocke J, editors. Business Process Management. *16th International Conference on Business Process Management*; 2018 Sep; Sydney, Australia. Cham: Springer Publishing. 250-267. doi: https://doi.org/10.1007/978-3-319-98648-7_15.

[16] Stertz F, Mangler J, Rinderle-Ma S. Temporal Conformance Checking at Runtime based on Time-infused Process Models. 2020; doi: https://doi.org/10.48550/arXiv.2008.07262.

[17] Crocker L, Algina J. *Introduction to classical and modern test theory*. 2nd Edition, New York: Holt, Rinehart and Winston; 1986. 527 p.

[18] Park G, Van der Aalst W M P. Action-oriented process mining: bridging the gap between insights and actions. *Progress in Artificial Intelligence*. 2022; 1-22. doi: https://doi.org/10.1007/s13748-022-00281-7.

[19] Drieschner C, Heckl A, Krcmar H. Simulation Tool for Learning Action-Oriented Process Mining. In Castro M, El-Abd M, Zeid A, editors. EDUCON 2023 Conference Proceedings. *IEEE Global Engineering Education Conference 2023*, 2023 May; Kuwait. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey. 1-8. doi: https://doi.org/10.1109/EDUCON54358.2023.10125248.

[20] Wolfe-Adam T, Alpyne - A Python library for executing AnyLogic models, Anylogic. https://github.com/t-wolfeadam/Alpyne.

[21] Burattin A. Beamline: A comprehensive toolkit for research and development of streaming process mining. *Software Impacts.* 2023; 17(100551):1-3. doi: https://doi.org/10.1016/j.simpa.2023.100551.

[22] Berti A, Van Zelst S, Schuster D. PM4Py: A process mining library for Python. *Software Impacts*. 2023 17(100556):1-7. doi: https://doi.org/10.1016/j.simpa.2023.100556.

AGV Traffic Management: Simulation-Based Deadlock Resolution and Collision Reduction via Deep RL with PPO

Mustafa Jelibaghu[1], Michael Eley[1], Oliver Rose[2], Alexander Palatnik[1], Leon Roth[1], Tim Thorwart[1]

[1]University of Applied Sciences Aschaffenburg, Würzburger Straße 45, 63743 Aschaffenburg, Germany; {mustafa.jelibaghu, michael.eley, alexander.palatnik, s200991, s201070}@th-ab.de

[2] University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany; oliver.rose@unibw.de

Abstract. This paper explores the application of multi-agent reinforcement learning using the Proximal Policy Optimization (PPO) algorithm for resolving deadlocks in material flow systems with Automated Guided Vehicles (AGVs). A multi-agent strategy that optimizes the dynamics and interactions of multiple AGVs in real-time is implemented. The integration of the Population Based Training (PBT) algorithm from Ray enables continuous adaptation and improvement of learning processes. Subsequent modifications to the reward system have also been implemented to enhance the model's efficiency and effectiveness. The efficacy of the proposed approach is evaluated using a material flow simulation for a real industrial use case. The results demonstrate significant improvements in reducing collisions and increasing throughput within the system. This study highlights the potential of multi-agent reinforcement learning and specifically the PPO algorithm, to enhance the performance and efficiency of material flow systems with AGVs.

Introduction

Industries increasingly rely on automated systems for production and logistics, which brings about complexity in management, particularly with deadlocks where automated guided vehicles (AGVs) block each other, halting operations. Traditional methods to resolve these deadlocks (Xu et al. 2014; Hussain et al. 2015), such as waiting or rerouting AGVs, are inadequate for larger systems due to scalability and adaptability issues. This paper proposes that Reinforcement Learning (RL), especially Multi-Agent Reinforcement Learning (MARL), can effectively address these challenges. It focuses on the Proximal Policy Optimization (PPO) algorithm, combined with Population Based Training (PBT) and targeted reward adjustments, to develop a robust and safe system for multi-agent environments.

The paper is organized as follows: Following a comprehensive literature review in Chapter 1, Chapter 2 describes the used simulation environment, the specific challenges faced throughout the project and how they were addressed. Finally, Chapter 3 presents and discusses the obtained results, while lastly chapter 4 outlines perspectives for future research directions and the further development of the model.

1 Literature

1.1 Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning in which an agent learns to make decisions by interacting with its environment. This learning paradigm has evolved considerably since the seminal work of Sutton and Barto (1998). At its core, RL is about developing a strategy or policy that maximizes the agent's cumulative long-term reward based on the actions it takes in different states.

The application of RL in complex decision-making tasks, has shown promise in the past. The ability of RL to learn from experience and adapt to dynamic environments makes it ideal for dealing with the uncertainties and complexity associated with such systems (Zhang et al. 2021; Choi et al. 2022).

The difference between Multi-Agent Reinforcement Learning (MARL) and Single-Agent Reinforcement

Learning (SARL) lies in the number of agents that learn and make decisions in the environment. While SARL focuses on scenarios in which a single agent controls all AGVs, MARL deals with situations in which multiple agents act simultaneously. MARL is particularly relevant to the problem of deadlocks in logistics systems, as it considers the interaction of multiple AGVs and can develop strategies for cooperative behavior to effectively avoid or resolve conflicts and deadlocks [Stone & Veloso, 2000].

1.2 Proximal Policy Optimization (PPO)

The Proximal Policy Optimization (PPO) algorithm, presented by Schulman et al. (2017), is a further development in the family of policy gradient methods in RL. PPO aims to improve the stability and efficiency of the learning process by finding a balance between the agent's exploration ability and the utilization of what has already been learned. In contrast to its predecessors, such as the Trust Region Policy Optimization (TRPO) algorithm, PPO offers a simplified yet effective method for policy optimization, making it particularly suitable for use in dynamic and complex environments such as those found in automated logistics systems.

1.3 Population Based Training (PBT)

Population Based Training (PBT) is an approach for hyperparameter tuning that combines the advantages of genetic algorithms and hand-guided optimization. Developed by Jaderberg et al. (2017), PBT enables adaptive and time-efficient optimization of the learning processes of AI models. PBT dynamically adjusts hyperparameters as the model is trained, leading to a continuous improvement in model performance. The main benefit of PBT lies in its ability to adapt to changing conditions within the training environment. Compared to conventional hyperparameter tuning methods, optimal configurations are achieved more effectively and quickly.

By combining these advanced techniques - MARL with PPO and dynamic adaptation through PBT - this project aims to develop a robust system for automated logistics that can avoid deadlocks while increasing the efficiency and reliability of the overall system.

2 Simulation Model

There has been a growing interest in using RL for deadlock resolution in intralogistics systems. In (Müller et al. 2022; Jelibaghu et al. 2023), the authors proposed a deadlock resolution method using a single RL agent. The agent was trained to learn how to resolve deadlocks by observing the states of the system and taking actions that lead to desired outcomes. The agent was able to achieve high levels of deadlock resolution and collisions avoidance in (Müller et al. 2022).

As part of the research project, a real-world application for an AGV system was considered and modelled in Plant Simulation (cf. Figure 1). The application is a high-bay warehouse with five aisles that the AGVs can only enter and exit from one side (dead ends). There are three AGVs available that have the task of moving pallets from the goods receipt, where the orders are created automatically and assigned to the AGVs (well known as dispatching), to the high rack. At the beginning the AGVs are located at the park station. A deadlock situation is shown in Figure 1. The deadlock occurs because the AGV01 currently located on the STR02 wants to enter aisle02. At the same time, the AGV02 wants to leave aisle02. The simulation model is a digital twin of the logistics system, enabling scenario testing and performance optimization. It tracks material movement and component performance to identify bottlenecks and improvement areas.

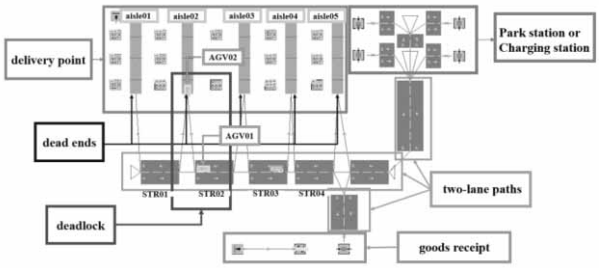


Figure 1:Illustration of a deadlock with three AGVs at the beginning of dead ends

The AGVs are controlled by an artificial neural network, which is implemented using Python code and connected to the simulation model and the AGV agents via the COM interface.

We decided to reduce the complexity of the environment, by reducing the number of agents to two. This adjustment aimed to provide a clearer view of how well our agent could learn and adapt, allowing for a more detailed observation.

To track the hyperparameters and the progression of rewards throughout the training, we utilized Weights & Biases (wandb). This platform enabled us to gain insights into the performance of our model and observe the

development of hyperparameters over the course of training. These adjustments resulted in clearer and more informative training progressions, allowing for more thorough analyses of the model's behavior and effectiveness.

In Table 1 we present the initial setup of training configurations, action space and reward system, to show the baseline from which we began observing and adjusting the models performance and learning abilities.

As Table 1 shows, at first the trials of the trainings were carried out serially and not in parallel. This had a considerable influence on the hyperparameters.

Training configurations	
Episodes per training	300
Steps per epsiode	2000
Trials per training	5
Concurrent trials	1
Perturbation interval	5
Observation Space	
General	ID of the considered AGVs
For each agent *i*	*x* and *y* position of agent *i* Current speed of agent *i* *x* and *y* position of the target destination of agent *i*
Action Space	
Possible actions	MoveForward, MoveBackward, Stop
Reward-System	
Perform action	-1
Pick up order	+100
Complete order	+5000
Not-loaded agent exits aisle	+10
Agent enters occupied aisle	-10
Loaded agent exits aisle	-10

Table 1: Summary of initial action space and reward-system

3 Experiments and Results

Our work builds on the work of (Müller et al. 2022; Jelibaghu et al. 2023). We propose a deadlock resolution method using a team of RL agents that is based on a real-world intralogistics system. We aim to evaluate our method on a number of different scenarios and demonstrate that it is able to achieve high levels of deadlock resolution and performance enhancement. In Figure 2 we show the MARL functionality. Compared to Single Agent Reinforcement Learning (SARL), the MARL approach is a decentralized approach. This means that each AGV is considered as an independent decision maker.

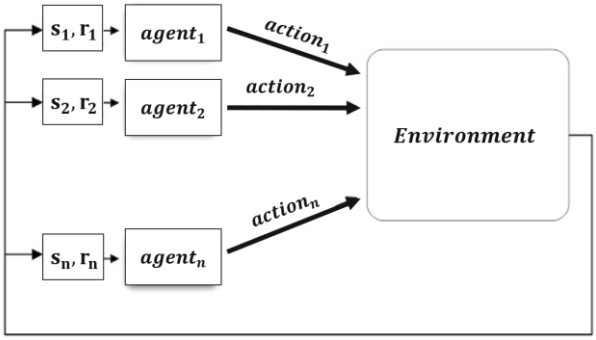


Figure 2: Agents in a multi-agent reinforcement learning (MARL)

In the initial phase of our study, we set out to intentionally provoke a deadlock scenario by initializing two or five agents and assigning the same drop-off location within the warehouse for the first ten orders. This approach quickly proved to be too aggressive or challenging, as it led to the AGVs consistently getting stuck. The high number of AGVs and task concentration offered little room for exploration. The resulting constant number of deadlocks highlighted the inadequacy of such a complex setup. Making it difficult to observe and understand the behavior of the reward system, the hyperparameters, and the learning efficacy of our agents.

The analysis is divided into five distinct sections to reflect the evolution of our model through various stages of adjustments.

Initially, the investigation begins with an examination of results obtained from the baseline setup, which represents our model's performance under initial conditions without any alterations to the reward system or hyperparameters tuning strategy. This foundational setup, shown in Table 2, is crucial for establishing a benchmark against which the impact of subsequent modifications can be measured.

Hyperparameter	Search space function	Search space
Minibatch size	Randint	[4; 4000]
SGD-iterations	Randint	[3; 30]
Clip-parameter	Uniform	[0.1; 0.3]
Learning rate α	Uniform	[0.000005; 0.003]
KL-Coefficient	Uniform	[0.3; 1]
KL-target range	Uniform	[0.003; 0.03]
Discount factor γ	Uniform	[0.8; 0.9997]
GAE parameter λ	Uniform	[0.9; 1]
Value function coefficient	Uniform	[0.5; 1]
Entropy coefficient	Uniform	[0; 0.01]

Table 2: Hyperparameter search spaces and functions

After establishing a baseline understanding, the focus shifts to the outcomes following strategic adjustments to the reward system. This part dives into how modifying the reward parameters influences the learning trajectory and decision-making processes of the agents.

The following part of this chapter explores the effects of refining the tuning mechanism. It examines how the implementation of a more dynamic and responsive tuning approach impacts the model's performance.

Lastly, we show the ability and the potential of our model to reduce the number of collisions while increasing the number of completed orders. This gives an outlook on the capability of the model to successfully resolve deadlocks.

Baseline Setup

We begin by examining the performance of one agent (SARL) before expanding our analysis to a scenario involving two agents (MARL). To be clear, we built the following scenario. First we ran and analyzed 1 agent in the environment and then we took the same model and increased it to 2 agents. Figure 3 shows the result of the analysis. We use the result of the single agent as a benchmark. We analyze the maximum rewards achieved during the training. This metric is indicative of the agent's ability to optimize its behavior within the environment, with the highest reward in a series of trials representing the peak efficiency reached by the agent. Out of five trials conducted during this phase, only the best-performing trial is considered for detailed analysis. This selection criteria ensures that we capture the agent's optimal performance potential under the baseline settings (cf. Table 1).

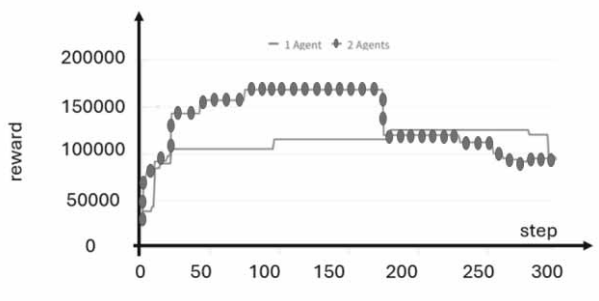


Figure 3: Max reward during training with multi agents

In the initial graphical analysis, contrasting performance outcomes between single-agent and dual-agent setups provide insightful revelations about their operational efficiency within the simulation. The Figure 3 refers to 1 agent and 2 agents results. 2 agent refers precisely to the accumulated reward of 2 agents. Initially, it might be assumed that the single agent would outperform due to its sole occupancy of the environment, facing no competition or interference. However, our findings show a different picture; the dual-agent setup achieved a higher maximum reward (cf. Figure 3). It demonstrates the cumulative advantage of having double the number of steps available compared to a single agent. This increased action potential, combined with effective learning behavior, indeed resulted in higher rewards for the dual-agent training scenario. It emphasized the significance of collaborative efforts in navigating complex environments.

Despite having twice as many steps at their disposal, the fact that the dual agents did not achieve double the reward of a single agent can be directly attributed to the escalated complexity introduced by their interaction. This complexity presents both challenges and opportunities for learning and optimization, highlighting the delicate balance between cooperation and competition in a shared environment.

Upon closer examination across all five trials for each training setup, it becomes evident that the two-agent model not only outperforms the single-agent framework in terms of maximum rewards but also in cumulative mean reward. This shows that the enhanced learning outcomes facilitated by multi-agent interactions are significant. The shared policy learning dynamic allows the system to benefit from a richer set of information for policy adjustments. This increases the likelihood of achieving successful, stable training outcomes.

Reward System

Table 3 shows the updated reward system, detailing the strategic modifications made to enhance navigation in the simulated environment. These changes were strategically implemented to swiftly move agents out of the aisles, thereby reducing potential deadlocks and easing traffic congestion.

Reward-System	
Perform action	-1
Unloaded AGV performs *MoveForward* on aisle	-5
Unloaded AGV performs *MoveForward* on aisl	-5
Unloaded AGV performs *MoveBackward* on aisle	+2
Pick up order	+100
Complete order	+5000
Unloaded AGV exits aisle	+25
Loaded AGV exits aisle	-10
AGV enters occupied aisle	-10
Unloaded AGV enters aisle	-50

Table 3: Adjusted reward-system

This refinement entailed modifying the existing movement incentives for "*MoveForward*", "*MoveBackward*", and "*Stop*" actions, which are integral to the agents' navigation through the simulation environment.

The standard penalty assigned to any step taken remained set at -1, preserving the agents' motivation to minimize unnecessary movements. However, to promote more efficient behavior post-order completion, executing a "*MoveForward*" or a "*Stop*" action was assigned a larger penalty of -5, while a reward of +2 was allocated for a "*MoveBackward*" action. This careful calibration of rewards and penalties was designed to maintain a balance where agents are discouraged from counterproductive movements without overshadowing the adverse implications of such actions, thereby sustaining the learning impact.

To further encourage the desired behavior, the action of successfully exiting an aisle was assigned a higher reward of +25. Conversely, to deter agents from exploiting this system by repeatedly entering and exiting the aisle without carrying a load, a hefty penalty of -50 was imposed on unloaded travel on the street. This modified incentive system was implemented to guide the agents towards behaviors that enhance overall efficiency in the logistics environment.

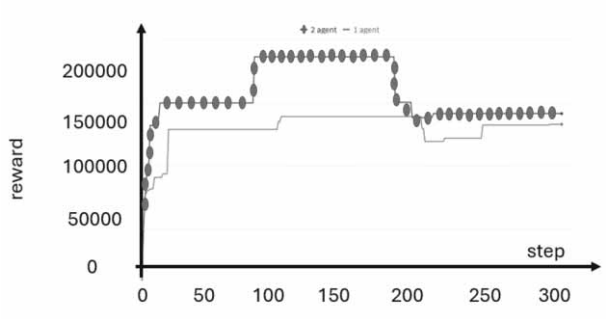


Figure 4: Max reward of training with new reward setup.

Figure 4 illustrates an improvement in the maximum reward output compared to Figure 3, which represents the results from the initial setup. This enhancement confirms that the modifications to the reward system have been effective, showing a positive impact on the overall performance.

Hyperparameters

In our exploration of hyperparameters and the implementation of the tuning process, the utilization of the Population Based Training (PBT) algorithm played a pivotal role. Despite careful inclusion of all relevant parameters within the PBT tuner, along with the definition of search spaces and corresponding search functions, it was noted that several values remained at their initial setting throughout the training period. This phenomenon of no adaption applied to all hyperparameters except for the clip coefficient. Unexpectedly, the clip coefficient exhibited variations that extended beyond the predefined search space boundaries. At the start of each trial, a random value for lambda was selected from within the search space, establishing the initial conditions under which each trial would proceed.

To leverage the full capabilities of the PBT mechanisms, we adjusted the number of concurrent trials to five, allowing for simultaneous training of all trials. This setup positioned each trial as an integral component of the PBT's population, fostering a dynamic learning environment. The perturbation interval, set at every five episodes, served as a benchmark for evaluating and comparing trials. This interval determined the feasibility of continuing a trial based on its performance. This necessitates adjustments to align with more successful trials or terminating trials that failed to show promise. It should be mentioned here that training the trials in parallel leads to significantly longer training times and to a partially

incomplete recording of the courses in wandb.

This strategic approach aimed to optimize the learning process by facilitating real-time adaptations and fostering a competitive yet collaborative environment among the trials. The performance outcomes, captured in Figure 5, illustrate the impact of these adjustments on the model's learning efficiency.

Due to gaps in data collection for these parallel trials, the data of some trials was not fully tracked by wandb. Follow several reasons can be attributed: Technical Issues, System Overload, Network Interruptions, Human Error and Software Bugs. Nevertheless, the results in Figure 5 showcase clear improvement in the collected max-reward, compared to the baseline setup with serial trials (Figure 3). These results suggest that parallel training, despite requiring longer training durations, has a positive impact on the agents' performance. Such high-reward instances demonstrate the potential benefits of conducting training in parallel. It demonstrates that this approach can significantly enhance the learning outcomes and operational efficiency.

The primary advantage of conducting parallel trials in PBT lies in the dynamic adjustment of hyperparameters in real time. While most parameters remained constant during serial trials, the training involving parallel trials saw dynamic adaptations. This leads and enhances learning behaviour, making the training process more effective. By terminating less promising trials, the algorithm, ensures that no resources are wasted on prolonging the training time unnecessarily. Thereby, it puts the focus on the most promising configurations.

training methodologies.

Figure 6 provides a detailed visualization of the hyperparameter values across all trials. It is evident that many trials exhibit similar or identical hyperparameter values, which is consistent with the observation from serial trainings that hyperparameters tend to remain constant. Nevertheless, the trials that underwent dynamic adjustments generally show improved performance, reinforcing the advantage of real-time hyperparameter tuning. However, this does not negate the possibility of achieving good performance in serial trials, as previously demonstrated.

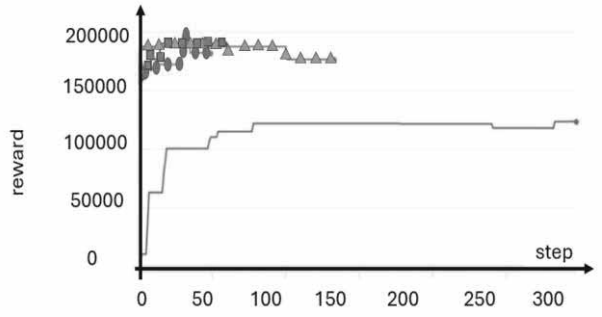


Figure 5:Max reward of training with 2 agents and concurrent trials

When analysing the hyperparameters of the parallel training trials, it is essential to recognize that the values presented represent only the average due to the dynamic adjustments made during the process. From these averages, it is observable that a higher entropy coefficient, which introduces more randomness into the decision-

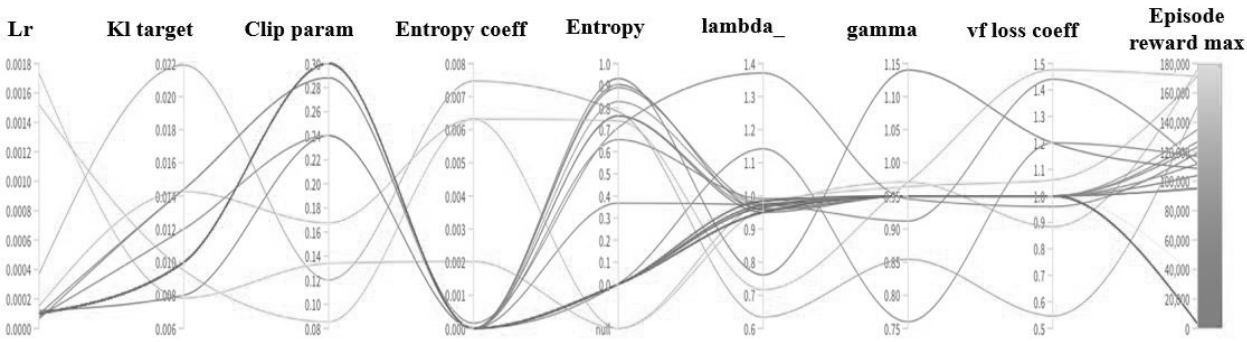


Figure 6: Hyperparameter ranges and their impact on the max reward

This approach allows for the rapid identification of superior hyperparameter configurations compared to manual tuning, The capability to dynamically adjusting and refining the training process in real time represents a pivotal shift towards more agile and responsive agent making process, can be beneficial to the results. An increase in entropy generally encourages exploration, which can prevent the agents from getting stuck in suboptimal behaviors.

Furthermore, a slight uptick in the learning rate,

paired with a reduced clip parameter, appears to be advantageous. The learning rate determines how quickly the model adapts to new information, while the clip parameter helps in stabilizing the policy update. Adjusting these parameters could lead to a more efficient learning process by balancing the rate of adaptation with the stability of learning. However, pinpointing the exact values that yield the best performance is challenging, as a wide array of configurations have resulted in favorable outcomes. Therefore, while certain trends in hyperparameter adjustments can be identified as generally positive, the diversity in effective configurations emphasizes the complexity and adaptive nature of the learning environment. This variability shows the need for a comprehensive hyperparameter search when tailoring the model to maximize performance.

Collision avoidance

In the final section of the analysis, we demonstrate the capability of our model to not only avoid collisions but also to reduce their occurrence over the course of training, while simultaneously enhancing the throughput of completed orders. We present three Figures to illustrate these developments: Figure 7 displays the reduction in the number of collisions throughout the training sessions. Figure 8 shows the corresponding increase in the number of successfully completed orders. These completed orders lead to the reward shown in Figure 9.

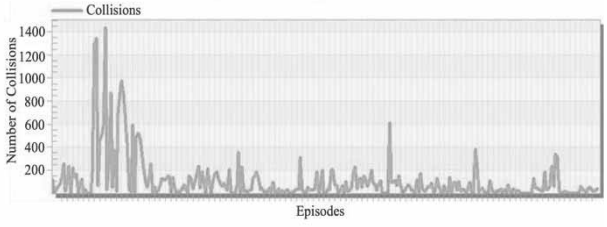


Figure 7: Number of cumulative collisions of 2 agents during training process

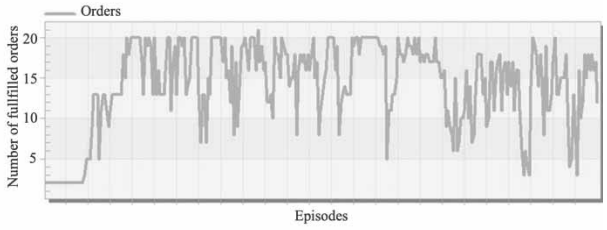


Figure 8: Number of cumulative completed orders of 2 agents during training process

These figures clarify that the MARL approach effectively enhances operational efficiency by increasing throughput and minimizing disruptions caused by collisions. Furthermore, these results indicate that the model possesses significant potential to avoid and resolve deadlocks, as collisions and deadlocks present very similar scenarios within our simulations. This demonstrates substantial improvements in both safety and efficiency of the AGV-fleet.

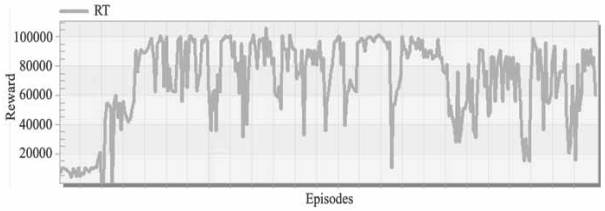


Figure 9: Accumulated reward of 2 agents during training process

Testing phase

Eventually, we wanted to verify the policy our agents have learned over the period of training. Therefore, we established checkpoints to capture and store the policy and hyperparameter configurations learned by the agents at specific training stages. By reloading these checkpoints into a model, we were able to verify that the agents were indeed learning. During the 100-episode learning phase the agents were still allowed to continue exploring, with the data loaded from the checkpoints serving as starting points for further learning and adaptation.

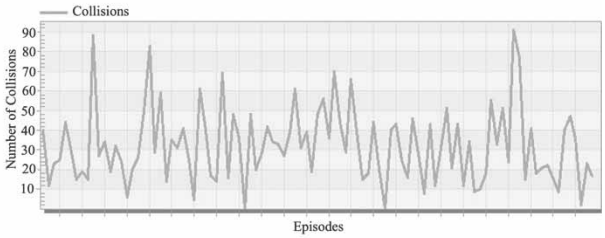


Figure 10: Number of cumulative collisions of 2 agents during testing phase

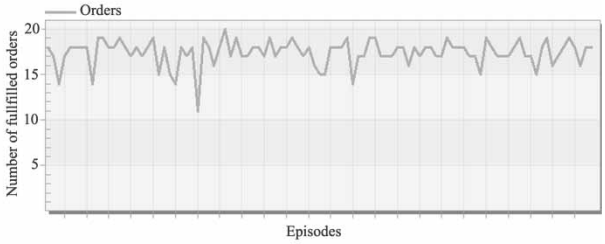


Figure 11: Number of cumulative completed orders of 2 agents during testing phase

Figures 10 to 12 offer a clear visualization of the outcomes from the checkpoints applied during our testing phase. Figure 10 illustrates the cumulative number of collisions, which have remained consistently low from the beginning of the testing phase, suggesting that the collisions are well-managed throughout the training. Figure 11 shows the cumulative number of completed orders, where it is evident that the orders are completed at a high rate, consistently ranking in the upper performance tiers of our training data. Lastly, Figure 12 shows the cumulative rewards occrued, which are also high from the outset of the testing phase. These visuals collectively demonstrate that the checkpoints contain effective policy and hyperparameter configurations, and that the agents did not require a reset or additional exploration phases to achieve these results. This underscores the robustness and efficiency of the learning mechanisms we have implemented.

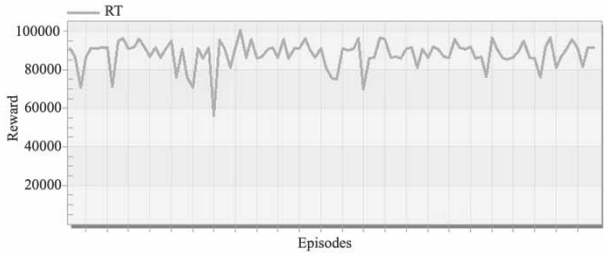


Figure 12: Sum of collected reward of 2 agents during testing phase

3 Conclusion and Outlook

In summary, this study has demonstrated that the Proximal Policy Optimization (PPO) approach coupled with Population Based Training (PBT) for hyperparameter tuning is suitable for efficiently coordinating AGV fleets to reduce collisions and deadlocks and thus throughput to increase. Possible extensions of the approach address the question of how to increase the number of agents. This will allow us to better understand the scalability of our approach and to refine the interplay between agents. Furthermore, we plan to experiment with different configurations of the perturbation interval, a critical mechanism in PBT that can significantly influence learning outcomes. In parallel, further adjustments to the reward system are on the horizon. These modifications aim to enhance the learning process even more, and we intend to evaluate the impact of these changes, especially when combined with parallel trials. With these adjustments, we anticipate uncovering even richer insights into the model's performance. While our findings have shown promise, the capabilities of our model and the process of refining its performance is ongoing. There are still many opportunities to further improve the model and gain deeper understanding, driving the evolution of AGV coordination in logistics.

References

[1] Xu, J.; Zheng, Z.; Lyu, M.: CGA-based deadlock solving strategies towards vehicle sensing systems. EURASIP Journal on Wireless Communications and Networking 2014 (2014). https://doi.org/10.1186/1687-1499-2014-214.

[2] Hussain, S.; Kumar, S.; Janardh, G.: Deadlock avoidance and re-routing of automated guided vehicles in flexible manufacturing systems. International Journal of Advances in Production and Mechanical Engineering (IJAPME) 1 (2015) 4.

[3] Sutton, R. S., & Barto, A. G. (1998). Reinforcement Learning: An Introduction. MIT Press.

[4] Tong Zhou, Dunbing Tang, Haihua Zhu, Zequn Zhang, Multi-agent reinforcement learning for online scheduling in smart factories, Robotics and Computer-Integrated Manufacturing, Volume 72, 2021, https://www.sciencedirect.com/science/article/pii/S0736584521000855

[5] H. -B. Choi et al., "MARL-based Optimal Route Control in Multi-AGV Warehouses," 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Korea, Republic of, 2022, pp. 333-338, doi: 10.1109/ICAIIC54071.2022.9722643.

[6] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2017). Proximal Policy Optimization Algorithms. arXiv:1707.06347.

[7] Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., & Kavukcuoglu, K. (2017). Population Based Training of Neural Networks. arXiv:1711.09846.

[8] Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. Autonomous Robots, 8(3), 345-383.

[9] Müller, M.; Reggelin, T.; Kutsenko, I.; Zadek, H.; Reyes-Ruiano, L.: Towards deadlock handling with machine learning in a simulation based learning environment. Proceedings of the 2022 Winter Simul tion Conference 11-14 December 2022 Singapore, 2022, pp. 1485-1496.

[10] Jelibaghu, M., M. Eley, and A. Palatnik. 2023. “Simulation-Based Resolution of Deadlocks in Automated Guided Vehicles using Deep Reinforcement Learning” . ASIM Fachtagungen Simulation in Production and Logistics, September 2023 Technische Universität Ilmenau, Germany.

Using Component-based Discrete-event Modeling with NSA-DEVS – an Invitation

Peter Junglas[1*], David Jammer[2], Thorsten Pawletta[2], Sven Pawletta[2]

[1]PHWT-Institut, PHWT Vechta/Diepholz, Am Campus 2, 49356 Diepholz, Germany; [*]*peter@peter-junglas.de*

[2]Research Group Computational Engineering and Automation, University of Applied Sciences Wismar, Philipp-Müller-Straße 14, 23966 Wismar, Germany;

Abstract. The quest for a simulation scheme that combines the preciseness of the PDEVS formalism with the ease of use of standard simulation environments has lead to the definition of NSA-DEVS, which has meanwhile been shown to provide a useful basis for real-world applications. A set of modeling and simulation tools for NSA-DEVS is freely available that uses Matlab as programming language and the graphical editor of Simulink for the construction of complex models from simple atomic components.

To demonstrate that these tools are ready for general discrete-event based applications, the implementation of a textbook example is presented in some detail. It is shown that components that are necessary for a transaction-oriented style can be modeled easily, leading to a comprehensible model with a solid mathematical basis.

Introduction

For modeling and simulation using the discrete event approach, practitioners can choose between a lot of commercial simulation environments, which provide users with a wide range of components and helpful tools [1]. However, the behaviour of complex models can sometimes be different than expected, especially because the documentation often does not provide all necessary details. In such situations users generally build a toolset of workarounds to make things work. But this often leads to conceptual problems and does not deepen the understanding of the precise behaviour of a model [2].

On the other hand, one could start instead with a precise description of the model and its components, using the well-established PDEVS formalism [3]. There are even a few free tools that provide a user-extensible set of DEVS-based components and a graphical user interface for combining components to build complex models [4]. But Preyser et al. have shown in [5] that the way, how PDEVS uses transitory states (i. e. states with lifetime 0), makes it hard to define some simple reusable components, especially when they show Mealy behaviour. Therefore they proposed a revised version of the PDEVS formalism [6] that allows for the direct description of Mealy components.

However, this formalism still has problems with chains of concurrent events [7]. Therefore it has been extended to NSA-DEVS (*Non-Standard Analysis DEVS*), which solves these difficulties by formally introducing infinitesimal delays. This idea has been analyzed thoroughly in [8, 9] and used to implement a large real-world example [10]. A corresponding simulation environment has been built, which contains graphical tools and a growing library of components. It is based on Matlab and the graphical editor of Simulink and is freely available from [11].

Since this article is primarily an invitation to use these new tools for modeling and simulation in discrete-event based studies, it concentrates on practical aspects, not on the underlying mathematical formalism. After a short definition of NSA-DEVS and a recapitulation of previous results, the structure of the tools and the basic workflow for concrete studies will be presented in some detail. A basic example from Law's textbook [12] will illustrate how to implement and apply components for standard entity-based applications.

1 Definition of NSA-DEVS

Like the PDEVS specification, the NSA-DEVS formalism describes two types of models: *atomic models*, which are the basic components, and *coupled models*, which combine atomic and coupled models in a hierarchical structure.

An atomic model (cf. Fig. 1) is given by a set X of input ports, each with a name, a similar set Y of output ports, a set S of internal states and an input delay time τ. Each state s has a lifetime, given by the time

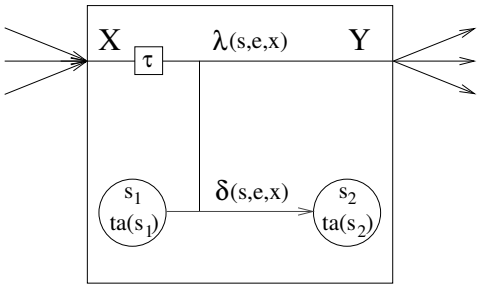


Figure 1: Dynamics of an atomic component.

advance function $ta(s)$. The behaviour is given by the output function λ, which defines output values y, and the transition function δ, which computes the next internal state. Both functions depend on three values: the current state s, the elapsed time e since the last transition and the input values x. When an external event, i. e. a set x of input values, occurs at time t, λ is called at time $t+\tau$, followed by an immediate call of δ. An internal event, i.e. a state change after a waiting time $ta(s)$, leads to a direct (undelayed) call of λ and δ.

The essential modification of NSA-DEVS is the introduction of the input delay together with extended time values: Introducing an infinitesimal value $\varepsilon > 0$, times are defined as values of the form $a+b\varepsilon$. The delay time is usually defined as $\tau = \varepsilon$ and only changed at rare occasions to guarantee a given order of events. Furthermore, the lifetime of states can never be 0, but an "immediate" (transitory) state change needs at least an infinitesimal time τ_D.

A coupled model is basically a set of several lists that describe the submodels used (atomic or coupled), the input and output ports of the coupled model and all connections between the submodels and from or to the ports of the coupled model. Inputs of a coupled model are immediate, i. e. they have no additional input delays.

2 Current Status of NSA-DEVS

The fundamental insight of [5] was that while the basic PDEVS formalism is sufficient to model any discrete-event based system, this is generally not possible with every reusable component. After the definition of NSA-DEVS one had therefore to show that it was up to this task.

The first step was the definition of a corresponding abstract simulator [8]. According to the DEVS philosophy, the simulator is actually part of the definition of the formalism. It adds a semantic layer to the static definition of models by defining their exact behaviour. In a next step [9], a set of standard examples was defined formally and implemented using simple reusable atomic components. A special point of interest was, how one can define the infinitesimal parameters introduced by NSA-DEVS in a simple and systematic way. This was studied further with a large real-world example in [10] consisting of 391 atomic and 88 coupled models in 5 hierarchical levels. It contained 391 input delay times τ and 12 additional delays τ_D for transitory states. Its construction has been simplified by the introduction of a graphical model builder, which uses Simulink for the definition of coupled models.

In the course of these investigations, a basic set of atomic models has been defined and implemented:

- sources (constant, several generators),
- math operations (add, gain, multiply, divide, compare),
- logic operations and flipflops based on IEEE 1164,
- routing components (combine, distribute),
- a QSS-based integrator,
- a sink (toworkspace) for logging simulation results,
- common logistics components (queue, server, batch, unbatch, terminator),
- statistical computations (getmax, utilization).

For all atomic models, corresponding NSA-DEVS blocks are provided for the Simulink editor and assembled in libraries. They make it possible to construct coupled models using Simulink's graphical capabilities for positioning and connecting blocks and ports. In addition, block parameters can be set, among them the values for the input delay τ and, where necessary, the transition delay τ_D.

All delay values are predefined and usually set to the default value $\tau_{def} = \varepsilon$. An exception are components that emit trains of output values with infinitesimal time distances, such as queue and combine atomics. They need larger values for τ_D, which are predefined in the library to $\tau_D = 2\varepsilon$. This often works, but has to be enlarged in special cases. If a special ordering is requested for loops that contain several sequences of components, one has to increase a few input delays to slow down some paths [10].

3 Implementation in Matlab

For the implementation of the example models, a set of tools and a component library have been constructed that are based on Matlab and the Simulink editor. Using a simple example, we will describe the basic workflow necessary to implement own models and running simulations.

Listing 1: Atomic model am_add2.

```
classdef am_add2 < handle
  properties
    s
    in1
    in2
    name
    tau
    debug
  end
  methods
    function obj = am_add2(name, tau,
        debug)
      obj.s = "running";
      obj.in1 = 0;
      obj.in2 = 0;
      obj.name = name;
      obj.debug = debug;
      obj.tau = tau;
    end
    function delta(obj,e,x)
      if isfield(x, "in1")
        obj.in1 = x.in1;
      end
      if isfield(x, "in2")
        obj.in2 = x.in2;
      end
    end
    function y = lambda(obj,e,x)
      s1 = obj.in1;
      s2 = obj.in2;
      if isfield(x, "in1")
        s1 = x.in1;
      end
      if isfield(x, "in2")
        s2 = x.in2;
      end
      y.out = s1 + s2;
    end
    function t = ta(obj)
      t = [inf, 0];
    end
  end
end
```

The atomic models are the basic building blocks. They are implemented in Matlab as classes that contain a constructor and the methods `delta`, `lambda` and `ta`. A simple example is the class `am_add2` for the addition of two input values (cf. Listing 1) and shows how to implement a Mealy component. On first sight, it looks similar to its counterpart in a continuous environment, but the discrete-event nature leads to a very typical change: Since input values are defined only, when an input event arrives, these values have to be stored internally using properties `in1` and `in2`. The property `s` is used throughout the whole library to denote "macroscopic" states, which is useful in more complex atomics. Here it is simply set to a constant value. The remaining properties store the values of external parameters, in this case the name of the component, the input delay and a debug flag.

The constructor provides initial values for all properties, its parameter list defines the set of external parameters of the component. The `delta` method just stores incoming values. The `lambda` method computes the output value, which is given here by the sum of the incoming or stored values. Finally the `ta` method returns the lifetime of the state, which is always given as a two-dimensional vector `[a, b]`, denoting the time $t = a + b\varepsilon$. In this case it is always infinite.

As a graphical representation of `am_add2` a Simulink subsystem with the name `am_add2` is stored in an NSA-DEVS library using the Simulink editor. Internally it just consists of unconnected input and output ports, which have the names of the ports that are used inside the atomic model. Additionally, it has a mask that defines the order and values of all parameters – except `name`, which is set to the name of the actual component – and short description and help texts (cf. Fig. 2).

Figure 2: Mask of the `am_add2` atomic.

A coupled model, such as the simple example model `demo1` shown in Fig. 3, is defined by a Matlab function that creates all its atomic and – using recursion – its coupled components and all connections. To simplify this tedious and error prone programming task, the user instead builds the model from the Simulink representation by copying the components from the library and con-

necting them in the standard way. The `toworkspace` models are used here to collect the simulation results. In the example, their parameter `varname` is set to `"input"` and `"result"` respectively.

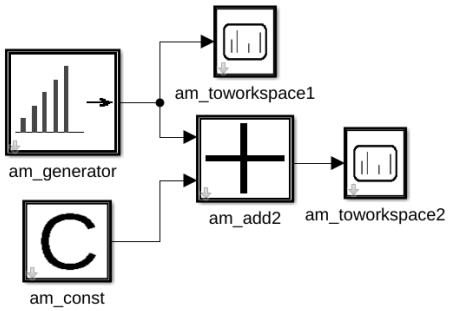


Figure 3: Coupled model `demo1`.

The very simple script shown in Listing 2 can now be used to create and run the model and plot the simulation results.

Listing 2: Run script for model `demo1`.

```
function testDemo()
  tEnd = 6;

  model_generator("demo1");
  out = model_simulator("demo1", tEnd);
  plot_results(out, tEnd);
end
```

From the Simulink representation, the `model_generator` creates the Matlab scripts for all coupled models. Next the `model_simulator` runs the model and collects all results in the struct variable `out`. In the example model it contains the two fields `out.input` and `out.results`, which each have subfields `t` and `y` for the time and result values. They can now be plotted easily with Matlab's standard functions.

A typical result is shown in Fig. 4. The `am_generator` creates increasing numbers, starting at $t = 1$, which are shifted by the constant value 3. The output event at $t = 0$ probably comes unexpected. It is due to the `am_const` atomic that sends its value only once at the beginning. It is then stored in `am_add2` and added to the initial value 0 stored for the other input port. One should bear in mind that the coupled models look like Simulink models, but the inner workings of discrete-event models are still very different.

Discrete-event models can easily contain very hard to find errors. To support the debugging process, the

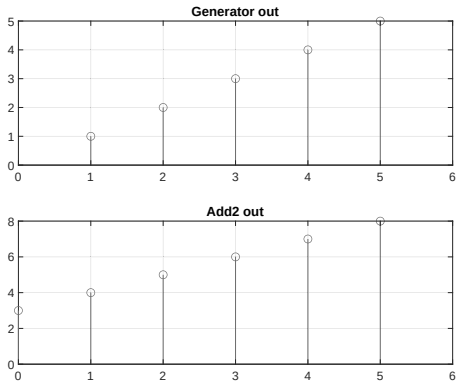


Figure 4: Plot of simulation results for the model `demo1`.

toolset supports three different levels, from simple time stamps over debug outputs from individually chosen atomic components to a complete output of internal simulator messages. The last level creates a graphical representation of all messages with a huge amount of information and is usually only useful for simple test models.

4 Example Model

In order to show that the methods and tools presented can be used directly for transaction-based modeling, we will implement a standard textbook example [12]. The model describes a time-shared computer with N terminals, which submit jobs of varying computing time demands. These jobs are processed on a single CPU in time-slices of length q using a round-robin scheduler with a switching time t_{swap}. When a job completes, a new job is created after a waiting time. The waiting and processing times are exponentially distributed random variables with mean values t_W and t_S. After the completion of N_J jobs, the average response time and queue length and the CPU utilization are computed.

A common method to implement such a model uses entities describing the jobs, which contain attributes such as the service time, i. e. the remaining processing time, or the start time. Entity attributes are implemented using struct variables. Four atomics have been created to handle such entity attributes (cf. Fig. 5): `am_adddata` adds a set of fields denoting new attributes to each incoming entity. If the input is not already an entity (i. e. of type struct), an entity is created with an additional attribute that stores the input value. `am_writedata` changes the value of an entity

attribute using values from other attributes. The changing function is defined as string parameter describing an arbitrary Matlab command. `am_readdata` outputs the value of an attribute from the input entity and `am_deletedata` deletes a set of attributes.

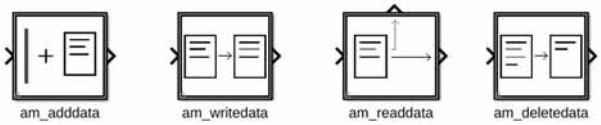


Figure 5: Atomic models for entity handling.

A few existing atomics have been modified to optionally read an attribute from an incoming entity instead of using a parameter or an input, among them the `server` and `distribute` components. With these atomics, one can create a coupled model for a server with exponentially distributed service times t_S (cf. Fig. 6): The `addTS` component adds an attribute to store the value of t_S, `setTS` sets the value of this attribute using the Matlab command string

```
"out = -" + tS + "*log(rand());"
```

where the variable `tS` is the mean service time, given by a mask parameter. The `am_server` component uses the attribute of incoming entities to set the current service time. Finally, `deleteTS` deletes the attribute for the sake of better encapsulation. Using a different formula for the computation of t_S, which uses several attributes, one can implement complex strategies for calculating the service time.

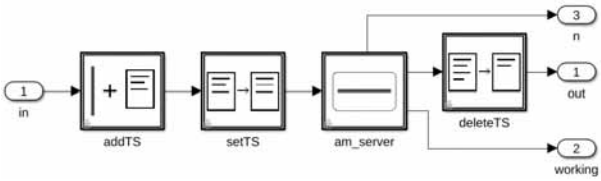


Figure 6: Coupled model of a server with entity-dependent service times.

A standard component in a transaction-based environment is the N-server, which can serve up to N incoming entities, each with its own service time. Its exact behaviour can be quite complicated and often is not transparent to the user of a commercial program. The NSA-DEVS description eliminates all ambiguities that arise e. g. with several incoming and outgoing entities at the same time. The diagram in Fig. 7 describes the basic behaviour of the `am_nserver` atomic, where monitoring compliance with the maximum server capacity has been omitted for better readability. Among its properties are a list E of entities in the server, a corresponding list σ of remaining service times and a list *qOut* of outgoing entities. An interesting difference to the simple server is the possibility that several entities can be ready at the same time. To handle this, the N-server moves all finished entities to *qOut*, changes to the state *emitting* and outputs them with a time delay of t_D. According to the rules stated above, t_D is predefined as 2ε, but may need to be enlarged in special applications. All details defining the exact behaviour can be found in the open source code of `am_nserver.m` [11].

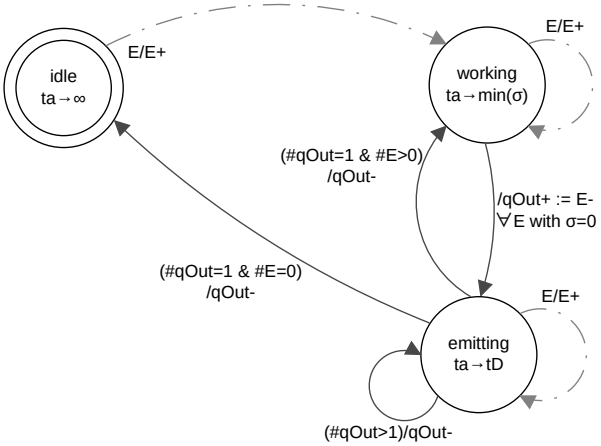


Figure 7: State diagram of the N-server component.

$\longrightarrow$	state transition due to internal event
$\dashrightarrow$	state transition due to external event
ta	lifetime of the state
E	external event (entity at input)
E+	insert entity in list E
E-	remove entity from list E
#E	number of entities in list E

With these atomics the three coupled models `Terminals`, `CPU` and the complete model `timeShared` can be assembled easily. The `Terminals` model (cf. Fig. 8) starts with a generator `initialJobs` that creates N entities at time 0 (more precisely at times $n \cdot t_D$) with consecutive IDs. They get the attributes `startTime`, `outPort` and `remainingServiceTime`, which is set to the individual service times. An N-server implements the individual waiting times. The `startTime` is set to

the current simulation time, using the utility function *get_time*(), and the entities proceed to the CPU. When a job is fully processed by the CPU, a corresponding ID is sent as external event to the input of coupled model `Terminals`, where it is promoted to a new job by the atomic model `constAdder`.

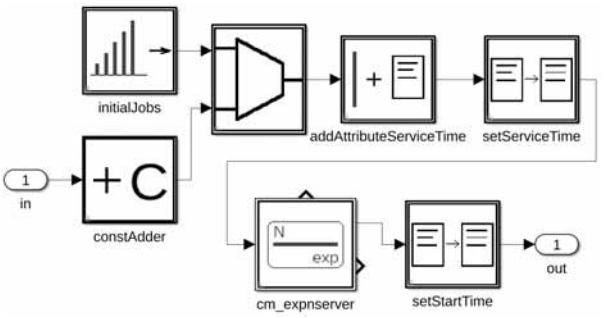


Figure 8: Coupled model `Terminals`.

The `CPU` model (cf. Fig. 9) starts with a queue for the waiting jobs, followed by a server representing the CPU proper. Its service time is set to $q + t_{swap}$ or less, if the job is almost ready. After processing, the entity attributes are updated: The time slice is subtracted from the `remainingServiceTime` and `outPort` is set to 1, if the job is ready, or 2 otherwise. The internal feedback from the server to the queue signals to the queue whether the server is available.

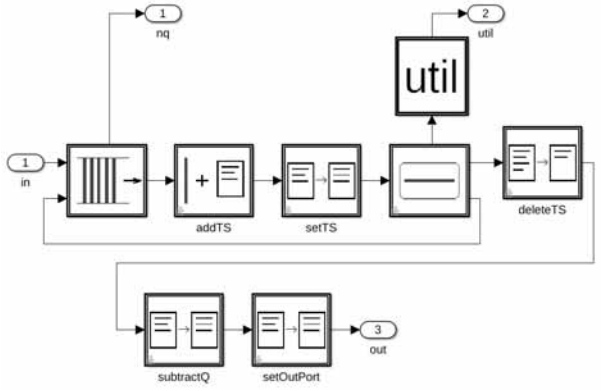


Figure 9: Coupled model `CPU`.

The complete model `timeShared` (cf. Fig. 10) shows the loop around the CPU that jobs are sent, until they have got their complete service time. The `distribute` atomic uses the `outPort` attribute to route finished jobs through the upper port. Finally the response time is computed by subtracting the value of attribute `startTime` from the current time and the entity is terminated. The `terminator` counts the outgoing jobs and sends this value back to the coupled model `Terminals`, where new jobs are created. A `stop` atomic halts the simulation, after N_J jobs have been processed.

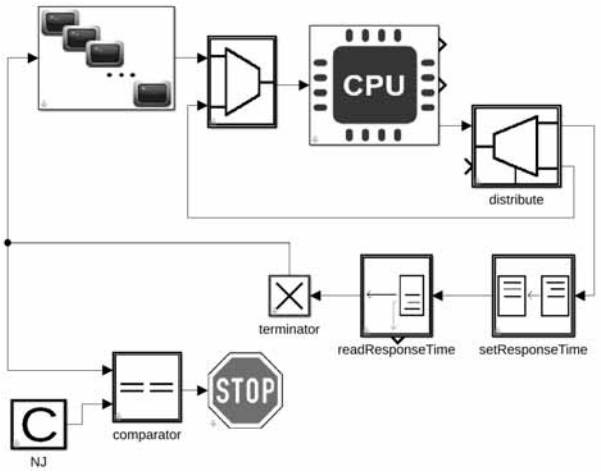


Figure 10: Complete coupled model `timeShared`.

Adding a few output blocks, one can gather enough information to get a complete picture of the model behaviour. The information displayed in Fig. 11 is sufficient to read off the waiting and service times of the jobs and follow each job individually through the model. This allows to thoroughly check the system behaviour. Especially intuitive is the plot of the remaining time after the CPU, which nicely displays the loops of the jobs around the CPU and the interaction of several jobs.

The component library contains the atomic `am_getmean` that calculates the running mean value of its input values, and the atomic `am_utilization` for the computation of the CPU utilization. Adding them to the `timeShared` model, one easily gets all requested statistical data. A typical example with $N = 20$ and $N_J = 1000$ is shown in Fig. 12. The results are similar to those of the SimEvents version of this model that had been used in [2], and consistent with the results shown in [12].

5 Conclusions

The implementation of the `timeShared` model has once again shown that NSA-DEVS is a conveniant basis for component-based modeling of discrete-event sys-

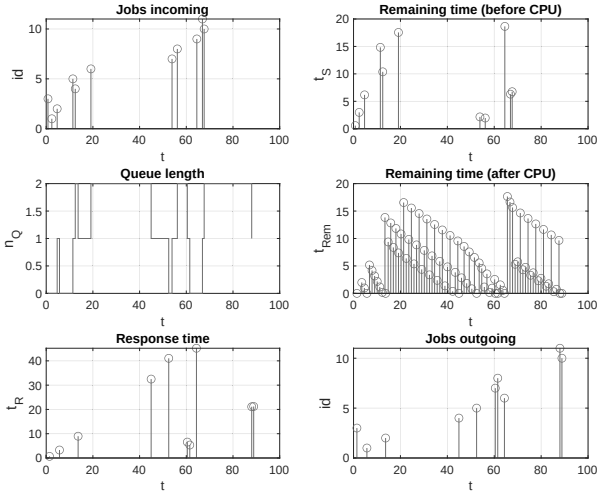


Figure 11: Detailed results of `timeShared`.

tems with a sound mathematical foundation. Especially, not one of the delay parameters had to be changed from its default value. This should be the typical case for systems with stochastical elements, where the probability of concurrent events is quite small.

Furthermore, the toolset available freely from [11] has proven its versatility: After finding and implementing a suitable set of atomics for handling entities with variable attributes and adding some standard atomics to the library, the construction of a transaction-oriented application proceeded by standard graphical methods. We invite all modelers interested in discrete-event modeling to try out these tools, ask for enhancements or even provide useful new atomics to the library.

During the design of the free NSA-DEVS simulator and the library, the focus has mainly been on correctness and simplicity. This shows, when measuring its performance: The simulation of `timeShared` with $N = 40$ and $N_J = 1000$ has a runtime of around 45 seconds on a recent PC platform, while the corresponding SimEvents version needs less than 2 s. Of course, the comparison is not quite fair, since SimEvents compiles the Matlab code before a run. Nevertheless, there is definitely large potential for improvement by trading elegance of construction for runtime performance and by generally reducing the number of messages sent.

With the presented tools and methods, one can finally tackle the questions that have been raised in [2]:

> What are the shortcomings of current implementations? Which concepts or components are missing? How could a reasonable set of components be defined?

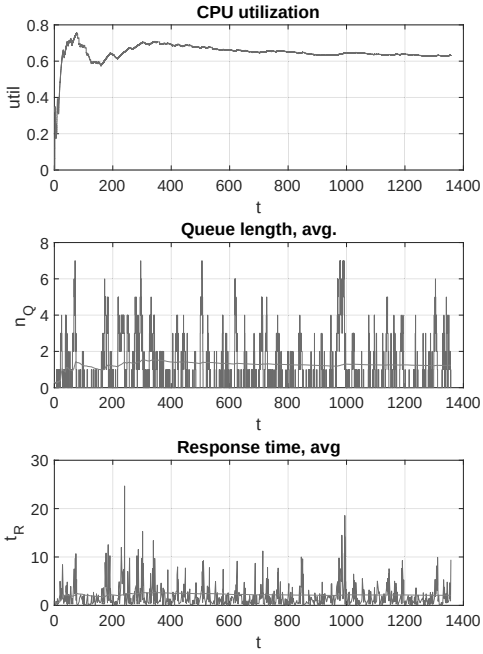


Figure 12: Statistical results of `timeShared`.

The atomic models for entity handling and the N-server introduced above are practical examples, how to precisely define fundamental building blocks due to their underlying NSA-DEVS based formulation. Another step along these lines would be the introduction of versatile queue models that are capable of supporting all the applications denoted in the ARGESIM benchmark C22 [13]. To cite [2] once again:

> For the advancement of transaction-based modeling it is vital that it is based on a thorough theoretical analysis to reveal the fundamental abstractions and basic components that are necessary.

This is true more generally for all discrete-event based modeling. NSA-DEVS and corresponding tools seem to be a promising path to promote such a program.

References

[1] Dias LMS, Vieira AAC, Pereira GAB, Oliveira JA. Discrete simulation software ranking—A top list of the worldwide most popular and used tools. In: *2016 Winter Simulation Conference (WSC)*. IEEE. 2016; pp. 1060–1071.

[2] Austermann L, Junglas P, Schmidt J, Tiekmann C. Conceptional problems of transaction-based modeling and its implementation in SimEvents 4.4. *SNE Simulation Notes Europe*. 2017;27(3):137–142. doi: 10.11128/sne.27.tn.10383.

[3] Zeigler BP, Muzy A, Kofman E. *Theory of Modeling and Simulation*. San Diego: Academic Press, 3rd ed. 2019.

[4] Franceschini R, Bisgambiglia PA, Touraille L, Bisgambiglia P, Hill D. A survey of modelling and simulation software frameworks using Discrete Event System Specification. In: *Proc. of 2014 Imperial College Computing Student Workshop*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2014; pp. 40–49.

[5] Preyser FJ, Heinzl B, Raich P, Kastner W. Towards Extending the Parallel-DEVS Formalism to Improve Component Modularity. In: *Proc. of ASIM-Workshop STS/GMMS*. Lippstadt. 2016; pp. 83–89.

[6] Preyser FJ, Heinzl B, Kastner W. RPDEVS: Revising the Parallel Discrete Event System Specification. In: *9th Vienna Int. Conf. Mathematical Modelling*. Wien. 2018; pp. 242–247.

[7] Junglas P. NSA-DEVS: Combining Mealy Behaviour and Causality. *SNE Simulation Notes Europe*. 2021; 31(2):73–80. doi: 10.11128/sne.31.tn.10564.

[8] Jammer D, Junglas P, Pawletta T, Pawletta S. A Simulator for NSA-DEVS in Matlab. *SNE Simulation Notes Europe*. 2023;33(4):141–148. doi: 10.11128/sne.33.sw.10661.

[9] Jammer D, Junglas P, Pawletta T, Pawletta S. Implementing Standard Examples with NSA-DEVS. *SNE Simulation Notes Europe*. 2022;32(4):195–202. doi: 10.11128/sne.32.tn.10623.

[10] Jammer D, Junglas P, Pawletta T, Pawletta S. Modeling and Simulation of a Real-world Application using NSA-DEVS. *SNE Simulation Notes Europe*. 2023; 33(4):149–156. doi: 10.11128/sne.33.tn.10652.

[11] CEA Wismar. *NSA-DEVS on GitHub*. `https://github.com/cea-wismar/NSA-DEVSforMATLAB`.

[12] Law AM. *Simulation Modeling and Analysis*. New York: McGraw-Hill, 5th ed. 2014.

[13] Junglas P, Pawletta T. Non-standard Queuing Policies: Definition of ARGESIM Benchmark C22. *SNE Simulation Notes Europe*. 2019;29(3):111–115. doi: 10.11128/sne.29.bn22.10481.

Visual NSA-DEVS Modeling Using an Adapted DEVS Diagram

Thorsten Pawletta[1*], David Jammer[1], Peter Junglas[2], Sven Pawletta[1]

[1]Research Group Computational Engineering and Automation, University of Applied Sciences Wismar, Philipp-Müller-Straße 14, 23966 Wismar, Germany;

[*]*thorsten.pawletta@hs-wismar.de*

[2]PHWT-Institut, PHWT Vechta/Diepholz, Am Campus 2, 49356 Diepholz, Germany;

Abstract. NSA-DEVS (Non-Standard Analysis Discrete Event System Specification) is an advancement of the DEVS formalism for modeling and simulating discrete-event and hybrid systems. DEVS supports modular-hierarchical modeling and clearly separates model from simulator. The primary objective of NSA-DEVS is to simplify the modeling of components with Mealy behavior while maintaining a simple simulator structure and the exact set-theoretic definitions of DEVS. With the implementation of a modeling and simulation environment, a comprehensive library of generic components, and real-world applications, the use of NSA-DEVS has been evaluated. However, the implementation of complex applications revealed that set-theoretic modeling should be complemented by visual techniques to facilitate system design and documentation. This paper explores how a visual representation equivalent to NSA-DEVS can be developed based on the known DEVS diagram and Harel's Statecharts.

Introduction

The set-theoretic DEVS formalism [1, 2] and its popular version PDEVS [3] are widely used for studying discrete event systems. However, Preyser et al. [4] have shown that it is difficult to define some basic reusable components, especially when they have Mealy behaviour. They introduced a revised version called RPDEVS [5]. However, RPDEVS struggles with handling chains of concurrent events [6].

Junglas argues in [7] that mathematical problems often are due to oversimplification in modeling. The macroscopic abstraction of phenomena by events results in simultaneous events, while in the underlying microscopic dynamics the corresponding processes are separated by small time delays. NSA-DEVS (*Non-Standard Analysis DEVS*) [8, 9] uses hyperreal numbers to represent infinitesimal time delays, thereby solving the problem of simultaneous event cascades. In contrast to PDEVS, Mealy behavior can be modeled without transitory states, which simplifies the specification of generally reusable model components.

A modeling and simulation (M&S) environment based on NSA-DEVS has been developed using MATLAB and Simulink's graphical editor [10, 11]. A few things became clear when implementing a real application [12], which consists of several layers and hierarchies with a large number of model components. First, designing and testing atomic model components specified in purely textual form using set theory requires a lot of time when the specification involves a large number of events and states. Second, using the visual support of the Simulink editor has proven to be extremely beneficial for modeling networked systems and hierarchical structures. This insight is not new, and several approaches to represent DEVS models graphically have already been developed. Based on preliminary work, Song and Kim designed the *Revised DEVS Diagram* [13] for Classic DEVS. In the authors' opinion, the Revised DEVS Diagram provides the best compliant representation of atomic DEVS models, even taking into account newer UML-based approaches, such as in [14].

The paper explores how the Revised DEVS Diagram can be adapted to depict NSA-DEVS models. It starts by giving a brief overview of the NSA-DEVS formalism and the Revised DEVS Diagram, and then examines the key differences between modeling with Classic DEVS and NSA-DEVS. Next, it discusses the basics of the modified diagram proposed here. Finally, a case study is used to demonstrate how NSA-DEVS models can be specified using the customized diagram.

1 Background

This section gives a short review of the NSA-DEVS formalism and the Revised DEVS Diagram for Classic DEVS.

1.1 The NSA-DEVS Formalism

Analogous to the DEVS formalism, NSA-DEVS defines a set-theoretic model specification and an operational semantics, called abstract simulator, to execute the specification. As with all DEVS formalisms, a distinction is made in the model specification between *atomic* and *coupled* models. An abstract simulator is defined for each model type, which is referred to as the *simulator* for atomic models and the *coordinator* for coupled models. During the execution phase, an abstract simulator is assigned to each model. A *root coordinator* manages the hierarchy of abstract simulators.

An *atomic model* describes the dynamic behavior of an arbitrarily complex system and is defined by a 7-tuple $< X, S, Y, \tau, ta, \delta, \lambda >$ with:

X	set of input ports and values,
S	set of states,
Y	set of output ports and values,
$\tau \in {}^*\mathbb{R}_{\text{fin}}^{\geq 0}$	input delay time,
$ta : S \to {}^*\mathbb{R}_{\text{fin}}^{\geq 0} \cup \{\omega\}$	time advance function,
$\delta : Q \times X^+ \to S$	state transition function,
$\lambda : Q \times X^+ \to Y^+$	output function.

The input and output sets X, Y contain *pairs of ports and values*, where ports are given by names.

$$\begin{aligned} X &= \{(p,v) | p \in P_{in}, v \in X_p\} \\ Y &= \{(p,v) | p \in P_{out}, v \in Y_p\} \end{aligned}$$

The sets X^+ , Y^+ consist of sets of pairs from X, Y to describe the simultaneous appearance of input or output values at different ports. Simultaneous input events at the same port are not allowed. The definition of the transition function δ and the output function λ contain the set $Q = \{(s,e) | s \in S, 0 < e \leq ta(s)\}$ that combines a state and the elapsed time e since the last transition.

According to the model specification, the *simulator* must process external, internal and confluent events. All three event types lead to a call of λ followed by a change to a new state according to δ. The time advance function ta may be infinitesimal or infinite (using $\omega := 1/\varepsilon$, with ε as infinitesimal value), but it is always > 0, thereby excluding proper transitory states. The delay time τ between the arrival of a set of inputs and the call of λ and δ is generally an infinitesimal, often given by a default value $\tau_{def} = \varepsilon$. Generally, all hyperreals used in models and the simulator have the form $a + b\varepsilon$ for real a,b. A more detailed description of the operational semantics of the simulator can be found in [10] and the full algorithm in [8].

A *coupled* NSA-DEVS model is defined as in *PDEVS with ports* [2]. It consists of input and output ports and a set of atomic or coupled models, which are connected among themselves and to the external ports. Outputs are transported as usual and a coupled model has no additional input delays. Since the operational semantics of the *coordinator* and the *root coordinator* are not essential for understanding a DEVS diagram for atomic models, they are not discussed in detail and reference is made to [8].

1.2 The Revised Diagram for Classic DEVS

To represent the dynamics of atomic models visually, Song and Kim formulated the Revised DEVS Diagram [13]. It is based on a series of preliminary works, whose origin probably goes back to Prähofer [15]. Based on the idea of Harel's *Statecharts* [16], Prähofer introduces a DEVS diagram which already contains descriptive elements adapted for atomic models.

The Revised DEVS Diagram in [13] is specifically designed for *Classic DEVS with Ports* [2]. According to the DEVS formalism, two types of diagrams are distinguished, one for atomic and another for coupled models. The latter is similar to the UML *Composition Structure Diagram*, which is extended by a field for specifying the *select function* specific to Classic DEVS.

The *diagram for atomic models* is based on a structuring of events and states. Events are grouped by categories into classes. Thus, events have a type and a value and are called a *message*. The set of *Ports* defines the interface of an atomic model. A port can only process messages of the same type, and only one message at a time, and is defined by *portName:messageType*. While an input port can only be connected to one source, an output port can be connected to several targets. The following syntax applies to input and output messages: *inportName?message* and *outportName!message*.

States are structured in *phases*, as in Harel's Statecharts [16]. A *phase* is a representative value of a set of states which produce the same output event and/or have the same time advance (remaining lifetime) at the

states. A phase is represented by a rounded box with the value of the phase variable and its remaining lifetime $@T$ until the next internal event. In addition to the phase variable, there is the set of *ordinary state* variables S_v, so that the current system state $s \in S$ results from the values of the pairs $(phase, s_v)$ with $s_v \in S_v$.

Phase transitions are triggered by external or internal events. Transitions caused by external events are represented by a *solid* line, transitions caused by internal events by a *dashed* line. In Classic DEVS, there are no simultaneous internal and external (*confluent*) events in atomic models. These are resolved at the level of the coupled models using a *select function*. Furthermore, there is no Mealy behavior, i.e. outputs can only be generated by internal events. Both types of events lead to state changes, i.e. to a phase transition and/or to changes in the ordinary state variables, and to a recalculation of the lifetime $@T$ of the current phase. This DEVS-based dynamics can be described with a triple (*event, guard, action*) at the phase transitions using the following notation:

$$inportName?message@[guard]/\{actions\}$$
$$outportName!message@[guard]/\{actions\}$$

The first notation defines state changes triggered by external events and the second one outputs and state changes caused by internal events. The $@[guard]$ defines a logical expression that depends on the *message* or the ordinary state variables. A phase transition, the execution of actions, and the sending of output events will only become active if the guard is true.

Figure 1 shows the specification of the dynamics of a single server as DEVS diagram. The server receives entities E of type $\textcircled{E}$ to be served via the port $in:\textcircled{E}$ and sends processed ones via the port $out:\textcircled{E}$. Furthermore, it sends the current status via the port $working:\{0,1\}$, where 0 stands for the IDLE phase and 1 for BUSY. Additionally, the initial phase is denoted with a bold box. The types and initial values of the phase variable and the ordinary state variables are defined in the lower box. The transition annotation $in?E/\{job = E, \sigma = 0\}$ specifies the transition from IDLE to BUSY due to a message E on the port *in* without defining a guard. The two actions describe the allocation of the server with the entity $(job = E)$ and the scheduling of an immediately internal event $(\sigma = 0)$ in order to send the server's new status as a message on the port *working* (no Mealy behavior!). The behavior resulting from the internal event

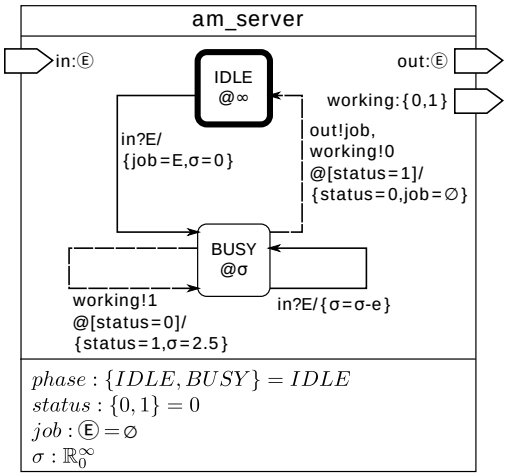


Figure 1: DEVS diagramm of a single server with discrete-event output of the server status.

specifies the dashed phase transition with the annotation $working!1@[status = 0]/\{status = 1, \sigma = 2.5\}$. However, the guard defines the condition for activating exactly this transition and the action $\sigma = 2.5$ sets the lifetime of the phase (fixed service time). Entities arriving in BUSY $(in?E/\{\sigma = \sigma - e\})$ are discarded, but the remaining time advance is recalculated using the internal variable e for the elapsed time since the last state change.

The transition from BUSY to IDLE with the annotation $out!job, working!0@[status = 1]/\{status = 0, job = \emptyset\}$ describes the output of the processed entity (job) on the port *out* when the service time has expired, and the output of the new server status at port *working*. Moreover, the ordinary state variables *status* and *job* are updated. The example shows how the guards control which of the two transitions takes place when an internal event occurs.

In addition to the techniques described, the Revised DEVS Diagram supports other mechanisms such as hierarchical phase compositions and parallel phases [13].

2 A Diagram for NSA-DEVS

This section begins with a short discussion of the key differences in modeling with Classic DEVS and NSA-DEVS. Subsequently, we introduce some adapted and new modeling elements for a DEVS diagram to represent atomic NSA-DEVS models.

2.1 Key Differences in Modeling with Classic DEVS and NSA-DEVS

The roots of NSA-DEVS are based on PDEVS [3] and RPDEVS [4]. Similar to these formalisms, concurrent events are not resolved at the level of coupled models. In contrast to Classic DEVS, there exists no *Select function* at the level of coupled models.

In NSA-DEVS, the handling of simultaneous internal and external events, termed *confluent events*, must be specified at the atomic model level. However, as outlined in Section 1.1, the approach differs significantly from that of PDEVS.

Analogous to Classic DEVS, NSA-DEVS prohibits multiple simultaneous external events on an input port. However, it does support simultaneous input messages on different ports.

With direct support for Mealy behavior in NSA-DEVS, the segmentation of state changes into event type-specific transition functions has been merged into a single state transition function δ. Moreover, all three event types (external, internal and confluent) prompt a call to the output function λ, followed by a transition to a new state using δ, and the re-scheduling of the next internal event via the *ta* function.

As detailed in Section 1.1, NSA-DEVS excludes proper transitory states with a lifetime of zero. Thus, internal events are always scheduled by the function *ta* with a value greater than zero, even if infinitesimal.

2.2 Adapted and New Modeling Elements

The differences in modeling and processing an atomic model between Classic DEVS and NSA-DEVS require the adaption and incorporation of some new modeling elements into the DEVS diagram. Figure 2 provides a summary of the diagram elements for NSA-DEVS.

The basic structure of the DEVS diagram shown in Figure 1 with the division into three parts: (i) name field, (ii) phase diagram, and (iii) variable definition, has been retained. The representation of ports and phases is also unchanged. The case study in Section 3 shows that it is sometimes useful to split the phase diagram into sub-diagrams. In order to always be able to clearly identify the initial phase, the initial transition has been adopted from Harel's statecharts. In NSA-DEVS, phase transitions can occur due to external, internal and confluent events. Although there are formally three event types in NSA-DEVS, only two line types are used for phase transitions in the adapted DEVS diagram. Phase transitions due to internal and confluent events are displayed with a dot-dash line. Due to the extended annotations at the phase transitions, explicitly internal events can still be clearly distinguished from confluent events in the diagram. In addition, the element *condition junction with priorities* already introduced by Freymann [17] for DEVS diagrams was adopted. This allows phase transitions to be summarized and case distinctions to be defined, which improves the clarity of the diagram.

Item	Description
	**Input** or **output port** with name and permitted message type
	**State phase** with value of the phase variable and it's remaining lifetime until the next internal event;
	*Initial state phase*
	**Phase transition** due to an external (⟶) or internal/confluent (⇢) event
	**Condition junction** and **Priorities** for modeling case distinctions in phase transitions
inportName?msg	**Phase transition annotations**: *External event* as message with *type* and value at input port *Name*
inports?	*Read all input ports*
@[guard]	Definition of a *transition guard* with guard result true or false
/{outportName!msg,..., stateVar1=value,...}	*Transition actions*, defining output messages with: message *msg* at output port *Name*, and updates of ordinary state vaiables
%	Comment

Figure 2: Summary of modeling elements of the NSA-DEVS Diagram for atomic models.

Due to the different semantics of NSA-DEVS and Classic DEVS, the annotations at phase transitions have

been partially changed. The annotation of an *external event* on a single input port is unchanged with *inportName*?*msg*. However, NSA-DEVS also supports simultaneous events on different input ports. For this purpose, the notation *inports*? is introduced. It specifies reading messages from all input ports simultaneously where ports without a message are set to *empty message* ($\emptyset$). The rules for defining *guards* have not been changed. The transition actions ($/\{actions\}$) specification has been extended by defining *output messages* and *updates of ordinary state variables* as actions. The change arises from the semantics of NSA-DEVS: Unlike Classic DEVS, Mealy behavior is modeled without transitory states, meaning that external events can immediately trigger output events. The general relocation of output messages to the action part ensures uniform indication of cause and effect for all event types.

Comments marked with % can be inserted in all three subfields of the DEVS diagram.

3 Case Study: Simple Queuing System

This section demonstrates the specification of a simple queuing system (SQS) using generic atomic models, usually organized in a model base. Figure 3 shows the structure of the SQS as a coupled model, developed within the NSA-DEVS M&S environment [11]. The associated model base provides an atomic model for recording statistical variables provided by ports such as *nq* (number of entities in queue) and *ns* (number of occupied servers).

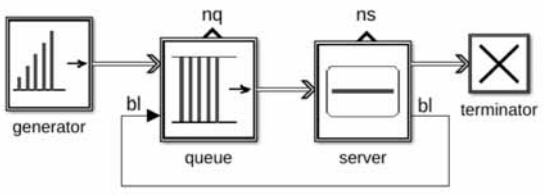


Figure 3: Coupled model `SimpleQueuingSystem`.

Figure 4 shows the DEVS diagram specifying the *generator*. The top box defines the name of the atomic model class, and the configurable variables, such as the interarrival time *tG* of the entities. The bottom box defines all variables. The state variable E is of type *Entity*, which defines a field variable *id*. This is initialized with the start ID, set by the configurable variable *n*0. The phase diagram in the middle shows the two phases with the initial phase PROD. After a time advance of *tG* time units, an internal event $@tG$ is triggered. This results in two actions: (i) sending an output message with the current entity E at port *out* and (ii) incrementing the state variable $E.id$. The guard $@[E.id - n0 < nG]$ decides whether a phase transition occurs back to PROD or to FINISH. In FINISH, the generator becomes inactive due to the defined time advance $@\infty$. The variable τ is a default variable and should be defined for each atomic model according to the NSA-DEVS specification. But it is irrelevant for the generator and could be omitted here as it does not define input ports.

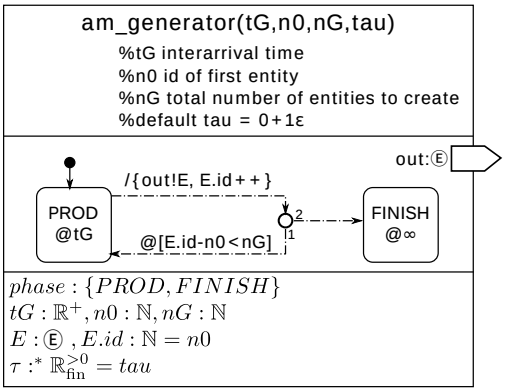


Figure 4: Atomic model class `am_generator`.

Figure 5 shows the DEVS diagram specifying the atomic model class for the *queue*. The queue works according to the push principle, i.e. it releases entities until it receives a blocking event. The advantage over a pull queue policy is a reduction in the number of events when the queue stores entities for an N-server, i.e. a server with a capacity greater than one.

The diagram defines an input and output port for the entities E, an input port $bl : \{0,1\}$ for blocking and unblocking events and a port for outputting the current number of entities in the queue ($nq : \mathbb{N}$). The *phase* variable defines four phases, and the three ordinary state variables are defined: (i) a list q for storing entities E, (ii) the input delay time τ and (iii) the internal delay time τ_D. The comment in the top box notes that the configurable variables are set to their default values, making $\tau_D > \tau$. This means that simultaneous input events are processed with an infinitesimal time interval before internal events. The type *List* for storing entities defines typical list operations, which are noted in the diagram

as follows:

- #q . . . number of elements in queue
- q+(E) . . . insert element E into queue
- q–(1) . . . remove first element from queue
- q(1) . . . read first element from queue without delete

The phase diagram is divided into sub-diagrams, which are separated from each other by a solid line. The *first* diagram shows the four phases with the initial phase *EmptyFree* and all possible transitions from *EmptyFree* (marked with a bold box) to other phases. The *inports*? transition annotation indicates that this transition occurs, if a message is received at one of the input ports (in, bl) or at both ports simultaneously. If the first guard $@[in = E]$ is not satisfied, only a transition to *EmptyBlocked* occurs. Otherwise, an output message with the number of entities in the queue is sent to port *nq* and the new entity E is inserted into list q. The second guard $@[bl = 1]$ checks whether a blocking event is present at the same time and regulates the transition to *QueuingBlocked* or to *QueuingFree*. A transition to phase *QueuingFree* always involves the scheduling of an internal event with an infinitesimal time advance τ_D.

The *second* diagram describes all possible transitions originating from *EmptyBlocked*. This phase is also only left by an external message. If only an unblocked message $bl = 0$ is received, a switch to *EmptyFree* occurs. In the case of an incoming entity, the guard $@[in = E]$ is satisfied and the same actions are performed as described in the first diagram. If an unblocked message $bl = 0$ is also present, then the guard $@[bl = 0]$ is satisfied and a transition to *QueuingFree* occurs, otherwise to *QueingBlocked*.

The *third* diagram specifies the transitions that originate from *QueuingBlocked*. In case of an incoming entity $in = E$, the same actions are performed as in the two previous diagrams. The second guard $@[bl = 0]$ checks for an unblocking event and decides whether to switch to *QueuingFree* or to return to *QueuingBlocked*.

The most complex case are the transitions originating from *QueingFree*. For a better overview, the possible transitions are shown in two sub-diagrams. As mentioned above, *QueuingFree* schedules an internal event with infinitesimal time advance τ_D to send an entity stored in the list q on port *out*. However, the system can receive a blocking message $bl = 1$ at the same time.

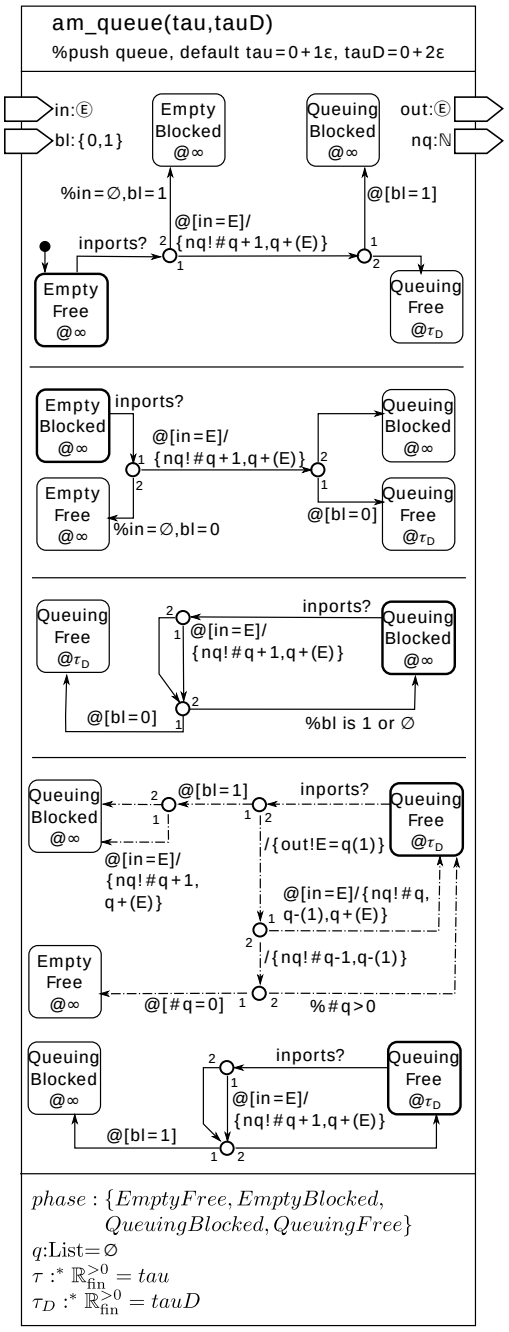


Figure 5: Atomic model class `am_queue`.

The simultaneous occurrence of both events is resolved by the hyperreal state variables τ and τ_D. The setting $\tau_D > \tau$ ensures that no entity is sent when a simultaneous blocking event is received. The specification of this confluent event situation is shown in the first sub-

diagram of phase *QueingFree*.

The *inports*? transition annotation indicates that all input ports are checked for messages before the internal event is triggered. If a blocking message $bl = 1$ is present ($@[bl = 1]$ is satisfied), a transition to *QueingBlocked* occurs. The guard $@[in = E]$ checks whether an entity E is present at the input port *in* at the same time. If so, the current queue length is sent as an output message to port *nq* and the new E is inserted into the list q. If there is no message at port *bl* or $bl = 0$ the transition action $/\{out!E = q(1)\}$ is executed. The first entity is read from the list q and sent to port *out*. It is then checked whether an entity is present at the input port *in* at the same time. If so, an output event with the queue length is sent to port *nq*, the list q is updated (delete sent E and insert arrived E) and the system returns to *QueingFree* to output another entity. If no new entity has arrived, the current queue length is sent to port *nq* and the list q is updated (delete sent E). The last guard $@[\#q = 0]$ checks whether the list q is empty and regulates the transition to *EmptyFree* or the return to *QueuingFree*.

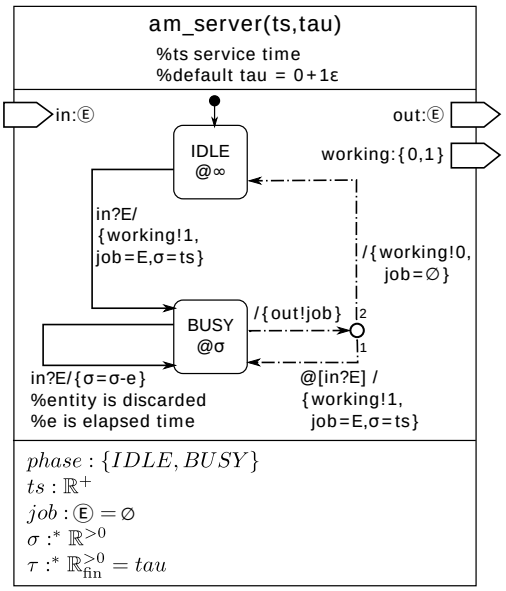


Figure 6: Atomic model class `am_server`.

Even though the next internal event is scheduled in *QueingFree* with infinitesimal time advance τ_D, external input messages may be present during this time span. The specification of this case is shown in the second sub-diagram for *QueingFree*. All input ports are checked by *inports*? for external messages. If the first guard $@[in = E]$ is satisfied, the new queue length is sent as output message and the new entity is inserted into the queue list q. The second guard $@[bl = 1]$ checks if a blocking message has been received and decides whether a transition to *QueingBlocked* or a return to *QueingFree* takes place.

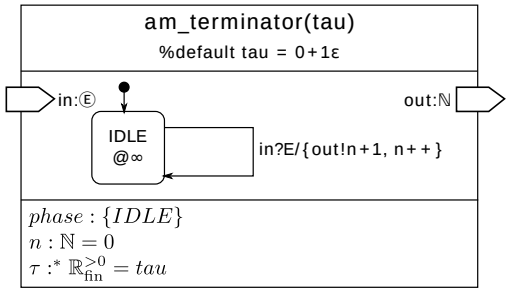


Figure 7: Atomic model class `am_terminator`.

Although the atomic model class `am_queue` supports N-server, only the specification of a *single server* is discussed below, shown as a DEVS diagram in Figure 6. The specification does not take into account the output port *ns* shown in Figure 3, which outputs the number of occupied servers, as this information is not of interest for a single server. The server is configurable with a service time *ts*. The infinitesimal input delay time τ can be set to the default value. The initial phase is IDLE. When an entity E arrives at port *in*, then: (i) a server occupancy event is sent to the port *working*, (ii) the entity and the service time are stored in ordinary state variables, and (iii) the server switches to BUSY, where an internal event $@\sigma$ is scheduled according to the service time. The server checks for external events in the BUSY phase, but does not register incoming entities (you could count them). The timeless return to BUSY, however, requires a rescheduling of the internal event using the action $\sigma = \sigma - e$. The internal variable *e* stores the elapsed time since the last event. If the internal event is triggered after the service time has expired, an external event may occur at the same time (confluent event). In any case, an output message is sent with the served entity *job* via port *out*. The guard $@[in?E]$ checks if an external event occurs at the same time. In this case a server busy output message ($working = 1$) is sent, the new entity E is scheduled as the current job in service, and the server returns to BUSY. Otherwise, a transition to IDLE occurs, a server is free output mes-

sage ($working = 0$) is sent, and the state variable *job* is updated.

Figure 7 shows the specification of the *terminator*. In addition to the model structure in Figure 3, the model class `am_terminator` defines an output port $out : \mathbb{N}$. It counts the number of incoming entities and sends changes of the counter as an output message to port *out*.

4 Conclusion

At first glance, the specification of atomic systems with a DEVS diagram may seem confusing, especially the complexity of the notations at phase transitions. However, it must be emphasized that the adjusted DEVS diagram can represent the complete dynamic specification of atomic models and can be converted one-to-one into program code for the NSA-DEVS simulation environment, where atomic NSA-DEVS models are implemented as a MATLAB class and organized in a model base.

The developers' experience is that the specification with DEVS diagrams is very helpful after a short training period, especially in the system design and system documentation phase. This becomes especially clear, when implementing complex atomics like an N-server [10]. Its graphical description has also been very helpful during the implementation and debugging phases, and documents the complex code in a clear-cut way, that highlights the basic underlying ideas.

References

[1] Zeigler BP. *Theory of Modeling and Simulation*. New York: Wiley-Interscience, 1st ed. 1976.

[2] Zeigler BP, Muzy A, Kofman E. *Theory of Modeling and Simulation*. San Diego: Academic Press, 3rd ed. 2019.

[3] Chow ACH. Parallel DEVS: A Parallel, Hierarchical, Modular Modeling Formalism and its Distributed Simulators. *Transactions of The Society for Computer Simulation International*. 1996;13(2):55–67.

[4] Preyser FJ, Heinzl B, Raich P, Kastner W. Towards Extending the Parallel-DEVS Formalism to Improve Component Modularity. In: *Proc. of ASIM-Workshop STS/GMMS*. Lippstadt. 2016; pp. 83–89.

[5] Preyser FJ, Heinzl B, Kastner W. RPDEVS: Revising the Parallel Discrete Event System Specification. In: *9th Vienna Int. Conf. Mathematical Modelling*. Wien. 2018; pp. 242–247.

[6] Junglas P. NSA-DEVS: Combining Mealy Behaviour and Causality. *SNE Simulation News Europe*. 2021; 31(2):73–80. doi: 10.11128/sne.31.tn.10564.

[7] Junglas P. Mathematical Problems due to Oversimplication. In: *4th Northern-Light Symposium on Mathematical Education in Engineering*. Hamburg-Bergedorf. 2024; pp. 27–41.

[8] Jammer D, Junglas P, Pawletta T, Pawletta S. A Simulator for NSA-DEVS in Matlab. *SNE Simulation Notes Europe*. 2023;33(4):141–148. doi: 10.11128/sne.33.sw.10661.

[9] Jammer D, Junglas P, Pawletta T, Pawletta S. Implementing Standard Examples with NSA-DEVS. *SNE Simulation Notes Europe*. 2022;32(4):195–202. doi: 10.11128/sne.32.tn.10623.

[10] Junglas P, Jammer D, Pawletta T, Pawletta S. Using component-based discrete-event modeling with NSA-DEVS – an invitation. In: *Proc. of ASIM 2024 – 27. Symposium Simulationstechnik*. Munich, Germany. 2024; .

[11] CEA Wismar. *NSA-DEVS on GitHub*. `https://github.com/cea-wismar/NSA-DEVSforMATLAB`.

[12] Jammer D, Junglas P, Pawletta T, Pawletta S. Modeling and Simulation of a Real-world Application using NSA-DEVS. *SNE Simulation Notes Europe*. 2023; 33(4):149–156. doi: 10.11128/sne.33.tn.10652.

[13] Song HS, Kim TG. DEVS Diagram Revised: A Structured Approach for DEVS Modeling. In: *Proc. Eur. Simulation Conf.* Eurosis, Belgium. 2010; pp. 94–101.

[14] Özmen Ö, Nutaro J. Activity Diagrams for DEVS models: A Case Study Modeling Health Care Behavior (WIP). In: *TMS/DEVS, SCS Spring Simulation Symposium*. Alexandria. 2015; .

[15] Prähofer H, Pree D. Visual Modeling of DEVS-Based Multiformalism Systems Based on Higraphs. In: *Proc. of the Winter Simulation Conference*. Los Angeles, CA USA. 1993; pp. 595–603.

[16] Harel D. STATECHARTS: A Visual Formalism for Complex Systems. *Science of Computer Programming*. 1987;pp. 231–274.

[17] Freymann B. Task-Based Multi-Robot Controls Based on the SBC Framework and DEVS [dissertation]. Ph.D. thesis, Technical Univ. Clausthal in coop. with Univ. of Appl. Sciences Wismar, Wismar, Germany. 2022. doi: 10.11128/fbs.40.

Entwicklung einer Funktion zur spurgenauen Lokalisierung basierend auf visuellen Informationen

Taihao Li[1*], Xinhai Xu[1], Marian Göllner[1], Sven Jacobitz[1],Xiaobo Liu-Henke[1]

[1]Fachgruppe Regelungstechnik und Fahrzeugmechatronik, Hochscule Ostfalia, Salzdalumer 46/48, 38302 Wolfenbüttel, Deutschland; *ta.li@ostfalia.de*

Kurzfassung. Die Einführung neuer digitaler Technologien treibt den gesellschaftlichen und wirtschaftlichen Wandel voran, insbesondere im Bereich der Mobilität, die durch autonomes Fahren in cyber-physischen Systemen (CPS) ermöglicht wird. Die Realisierung autonomer Fahrsysteme erfordert Fahrzeuge in der Lage, in komplexen Straßenumgebungen eine spurgenaue Lokalisierung und eine spurgenaue Lokalisierung anderer Verkehrsteilnehmerdurchzuführen. In dieser Arbeit wird eine Funktionalität zur spurgenaueren Selbstlokalisierung und zur genaueren Lokalisierung anderer Verkehrsteilnehmer entwickelt, die auf visuellen Informationen basiert. Diese ermöglicht es dem Fahrzeug, Straßeninformationen und andere Verkehrsteilnehmer zu identifizieren. Die Funktionen werden in einem virtuellen Testszenario validiert.

Einleitung

Derzeit befinden sich Gesellschaft und Wirtschaft in einem Transformationsprozess aufgrund des Einsatzes neuer digitaler Technologien. Eine bedeutende Technologie für zukünftige Mobilität ist das autonome Fahren in der vernetzten CPS-Umgebung [1]. Die Society of Autonomous Engineers (SAE) hat sechs verschiedene Stufen des autonomen Fahrens (L0-L5) spezifiziert [2]. Das Ziel des autonomen Fahrens (L5) besteht darin, dass Fahrzeuge ohne Eingriff des Fahrers vollständig autonom die Wahrnehmung, Entscheidungsfindung und Ausführung übernehmen, die ursprünglich dem Fahrer oblag. Aus [3] ist bewusst, dass Hochpräzise Navigations und Lokalisierung eine der Schlüsseltechnologien für autonome Fahrzeuge sind, wobei die spurgenaue Lokalisierung eine Grundvoraussetzung für diese Lokalisierung darstellt. Eine weit verbreitete Methode zur spurgenauen Positionierung ist die Verwendung von GNSS Echtzeit-Kinematik (RTK), die jedoch häufig aufgrund schlechter Satellitengeometrie ausfällt und zu erheblichen Fehlern führt. Darüber hinaus führen Unterbrechungen des Funksignals durch Wolkenkratzer in städtischen Gebieten zu Problemen [4]. Eine Alternative für die genaue Lokalisierung ist das Matching mit einer Karte, der auf Sensordaten und früheren Karten basiert. Durch das Abgleichen von Lidar-Daten mit Punktwolkenkarten kann eine hochpräzise Ortung erreicht werden [5]. Aber mit dieser Methode nimmt oft viel Speicherplatz in Anspruch und erfordert häufige Aktualisierungen der Karte.

Ein Ansatz ist die Bestimmung der Fahrzeugposition durch visuelle Erkennung der Fahrspur und der benachbarten Fahrspuren. Darüber hinaus ist es für eine effiziente Routenplanung auf Fahrstreifenebene erforderlich, dass die Fahrzeuge Hindernisse und Verkehrsteilnehmer auf anderen Fahrstreifen wahrnehmen. Die Objektlokalisierung sollte spurgenau erfolgen, um die Routeplanung zu optimieren. Daher wird in dieser Arbeit die Bilder aus der Kamera verwendet, um Fahrspur zu erkennen. Dadurch wird eine Spurgenaue Lokalisierung des Ego-Fahrzeugs sowie eine Spurgenaue Lokalisierung anderer Verkehrsteilnehmer ermöglicht.

Der weitere Beitrag ist wie folgt gegliedert: zunächst wird in Abschnitt eins der strukturierte, modellbasierte und verifikationsorientierte Rapid Control Prototyping (RCP)-Entwicklungsprozess ausführlich erläutert. Anschließend werden in Abschnitt zwei die aktuellen Fortschritte und Herausforderungen bei der erforderliche Technologie dargestellt. Der dritte Abschnitt umfasst die Konzeption für die Lokalisierungsfunktion. In Abschnitt vier wird schließlich den Entwurf der Lokalisierung vorgestellt. Die Ergebnisse der spurgenauen Lokalisierung werden in Abschnitt fünf dargestellt. Abschließend erfolgt eine Zusammenfassung der Arbeit und ein Ausblick auf zukünftige For-

schungsarbeiten.

1 Methodik

Der durchgängig modellbasierte und verifikationsorientierte Funktionsentwurf und die Absicherung vernetzter mechatronischer Systeme nach [6] hat sich in zahlreichen Anwendungen in Forschung und Industrie als zeit- und kosteneffizient erwiesen. Abbildung 1 zeigt den modellbasierten mechatronischen Entwicklungskreislauf, der zur Absicherung Model-in-the-Loop (MiL)-, Software-in-the-Loop (SiL)- und Hardware-in-the-Loop (HiL)-Simulationen sowie die Echtzeitrealisierung durch Prototypen umfasst.

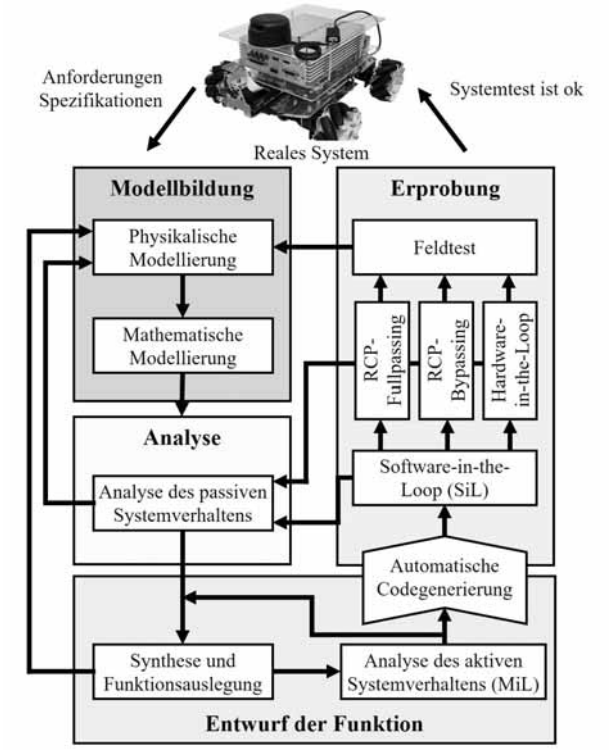


Abbildung 1: Mechatronischer Entwicklungskreislauf.

Der Entwurfsprozess beginnt mit der Modellbildung basierend auf dem realen System, welches gemäß der Anforderungen reduziert bzw. vereinfacht wird, sodass sich zunächst ein physikalisches Modell ergibt. Dieses wird mithilfe physikalischer Gesetzmäßigkeiten in ein mathematisches Modell überführt, welches wiederum bspw. in Form von Signalflussplänen im Rechner abgebildet und mithilfe von CAE-Werkzeugen und entsprechender Numerik simuliert werden kann. Der Modellbildungsprozess umfasst Messungen am realen System. Dabei werden die Parameter des mathematischen Modells identifiziert und die Simulation validiert. Eine anschließende Analyse der Simulationsergebnisse lässt Schlussfolgerungen über die grundlegenden statischen und dynamischen Eigenschaften des realen Systems zu, auf dessen Grundlage die Konzeption der Funktionen erfolgt. Es werden sowohl die Funktionsarchitektur als auch Ansätze zu dessen Auslegung und Optimierung festgelegt. Komplexe Funktionen werden zudem nach dem verallgemeinerten Kaskadenprinzip in hierarchische Teilfunktionen zerlegt um die Funktionskomplexität zu reduzieren und so den Auslegeprozess handhabbar zu machen. Dazu werden bereits in einem frühen Entwicklungsstadium Hard- und Softwareanforderungen berücksichtigt und Schnittstellen für die funktionsübergreifende Kommunikation definiert. Der Erprobungsprozess wird parallel zur Entwicklung durchgeführt. Wenn eine Teilfunktion entwickelt worden ist, wird diese getestet und analysiert.

In dieser Arbeit wird mit Hilfe dieser Methode zunächst die Anforderungen werden bestimmt. Durch die Analyse des physikalischen Modells, sowie der Mathematik, wird die Funktion modelliert. Jeder Teil der Funktion wird entsprechend getestet.

2 Stand des Wissens

Mit der Entwicklung des Deep Learning wurden verschiedene auf Deep Learning basierende Algorithmen für die Aufgabe der Fahrspurerkennung vorgeschlagen [7], von denen die meisten gefaltete Neuronales Netze verwenden [8]. In [9] wird eine Technik vorgeschlagen, die neuronale Netze mit dem langes Kurzzeitgedächtnis (convLSTM) kombiniert, um die zukünftige Trajektorie eines Fahrzeugs vorherzusagen. In [10]wird ein auf semantischer Segmentierung basierendes Kodierungs-/Dekodierungsnetzwerk vorgeschlagen. Nach dem Downsampling des Bildes wird ResNet-50 verwendet, um die gewünschten Merkmale zu extrahieren, die dann mit dem Unet-Modell decodiert und upgesamplet werden. In [11]wird ein Netzwerk mit semantischer Segmentierung und optischer Flussschätzung kombiniert. Die semantische Segmentierung wird für Schlüsselbilder verwendet. Unkritische Frames werden mit Hilfe eines Netzwerks zur Schätzung des optischen Flusses verfolgt und schließlich werden die Fahrspuren mit Hilfe von Clustering (Density-Based Spatial Clustering of Applications with

Noise) identifiziert. Das Modell LanetNet [8] zeichnet sich durch seine hohe Leistungsfähigkeit aus. Es ist ein End-to-End-Deep-Learning-Modell, das die gleichzeitige Erkennung und Segmentierung von Fahrspurlinien, ein Merkmalsextraktionsnetzwerk und ein integriertes Clusternetzwerk umfasst. Das Merkmalsextraktionsnetz verwendet eingefaltete Neuronales Netz, um Merkmale aus dem Bild zu extrahieren. Das Clusternetzwerk hingegen gruppiert die extrahierten Merkmale, um die verschiedenen Fahrspurlinien zu segmentieren. Zu den Vorteilen gehören eine höhere Echtzeitleistung und Robustheit. Im LaneNet-Modell werden häufig vortrainierte Modelle als Kodierer verwendet. Der Vorteil von Pre-Trainingsmodellen besteht darin, dass sie die Merkmalsextraktion verbessern und den Trainingsprozess bis zu einem gewissen Grad beschleunigen können. VGG16 [12], ResNet [13] und BiSeNet V2 [14] sind häufig verwendete Pre-Trainingsmode.

Die Technik der Objekterkennung löst das Problem der Klassifizierung und Lokalisierung von Objekten in einem Bild [15]. Die Anwendung dieser Technik legt den Grundstein für die automatische Analyse und das Verständnis komplexer Szenario. Ihre Effektivität wirkt sich direkt auf die Leistung verschiedener KI-Anwendungen aus, z. B. auf die Objektverfolgung [16, 17].

Die klassische Objektierkennungsalgorithmus waren hauptsächlich vor dem Jahr 2008 verbreitet. Sie basierten auf manuell extrahierten Merkmalen und folgten in etwa folgendem Ablauf: Markierung von Kandidatenbereichen, Merkmalsextraktion, Erkennung und Klassifizierung. Repräsentative Detektoren sind der VJ (Viola-Jones) Detektor [18] und der DPM (Deformable Parts Model) Detektor [19]. Zu den Nachteilen traditioneller Objekterkennungsalgorithmen zählen geringe Genauigkeit, schwache Robustheit und langsame Verarbeitungsgeschwindigkeit. Mit der Entwicklung der künstlichen Intelligenz sind allmählich Objekterkennungsalgorithmen auf Basis von Deep Learning entstanden

Die auf Deep Learning basierenden Objekterkennungsalgorithmen lassen sich hauptsächlich in zwei Arten unterteilen: One-Stage- und Two-Stage-Verfahren. Das Two-Stage-Verfahren R-CNN [20], Fast R-CNN [21] und Cascade R-CNN [22] lösen dieses Problem, in dem sie die Bildregionen in kleine Regionen aufteilen und separat klassifizieren. Zuerst werden den Erkennungsbereich abgegrenzt, dann erfolgt die Merkmalsextraktion und Klassifizierung. R-CNN zeichnet sich durch die Verwendung von Faltungsnetzen zur Merkmalsextraktion aus, wodurch die Schwächen herkömmlicher Erkennungsmethoden ausgeglichen werden, und verwendet eine selektive Suche zur Abgrenzung vom Bereich von Interesse (ROI), wodurch der Rechenaufwand reduziert wird. Fast R-CNN baut auf R-CNN auf und verbessert die Erkennungsgeschwindigkeit durch die Einführung einer neuen Verlustfunktion und erreicht eine erste Implementierung der End-to-End-Erkennung [23]. Two-Stage-Verfahren zeichnen sich durch eine hohe Erkennungsgenauigkeit aus, sind jedoch langsamer und für Echtzeitanwendungen weniger geeignet. One Stage Verfahren, vertreten durch die YOLO-Serie [24], zeichnen sich dadurch aus, dass sie keine Begrenzung des Erkennungsbereichs benötigen und direkt die Wahrscheinlichkeit der Objektkategorie und die Positionsdaten liefern, was die Erkennungsgeschwindigkeit deutlich erhöht.

Für Algorithmen zur Zielverfolgung lassen sich die derzeit verwendeten Verfolgungsschritte in zwei Hauptteile unterteilen: 1. Bewegungsmodell und Zustandsschätzung, die hauptsächlich dazu dienen, die Position von Objekten in nachfolgenden Frames vorherzusagen; 2. die Assoziation neuer Frame-Erkennungsergebnisse mit der aktuellen Trajektorie. Die derzeit führenden Tracking-Algorithmen sind DeepSORT [25] und Bytetrack [26]. Ersterer ist eine Weiterentwicklung des SORT-Algorithmus und verwendet die Mahalanobis-Distanz und Erscheinungsinformationen, um die Tracking-Frames des aktuellen Frames mit den Erkennungs-Frames abzugleichen. Allerdings sind der Rechenaufwand und die Verzögerung relativ hoch, so dass es für Anwendungen, die eine sofortige Reaktion erfordern, ungeeignet sein kann. Bytetrack verfügt über eine starke Assoziationsstrategie, die durch ein Sekundärer-Matching von Trajektorien mit hohen und niedrigen Vertrauensrahmen sowohl die Genauigkeit beibehält als auch die Geschwindigkeit erhöht und eine gute Robustheit bietet.

3 Konzeption

Im vorliegenden Abschnitt wird eine Konzeption für die spurgane Lokalisierung dargestellt. Abbildung 2 zeigt die Konzeption des der Spurgenaue Lokalisierungsfunktion. Um eine spurgenaue Lokalisierung zu ermöglichen, muss die Funktion folgende Anforderungen erfüllen: Das Fahrzeug kann Fahrspuren erkennen und die Anzahl der Fahrspuren bestimmen, auf denen es

sich befindet. Das Fahrzeug kann andere Verkehrsteilnehmer erkennen und die Fahrspuren der anderen Verkehrsteilnehmer bestimmen.

In dieser Arbeit kann ein Video V als eine chronologisch geordnete Sammlung von Einzelbildern betrachtet werden, wobei jedes Einzelbild als Matrix dargestellt und mit $\underline{\underline{B}}(t)$ bezeichnet wird. Außerdem wird die Bildrate des Videos als r bezeichnet wird.

- Erkennung der Fahrspur: Der Hauptzweck dieses Teils besteht darin, die Fahrspur-Informationen in den Daten zu extrahieren. Die jede Spurtrennlinie $l_i(x)$ können durch ein Polynom 1 dargestellt. i bedeutet die Nummer der Spurtrennlinie und x bedeutet den und der Koordinatenwert der Fahrspur in der Fahrtrichtung

$$l_i(x) = a_n x^n + \ldots + a_1 x + a_0 \tag{1}$$

  Der Gesamtrahmen dieses verwendeten Modells ist als Kodierungs-/Dekodierungsstruktur konzipiert, die für die binäre Segmentierung und die Instanzsegmentierung zuständig ist. Der Kodierer ist für die Merkmalsextraktion und der Dekodierer für die Verlustberechnung bei der binären Segmentierung und der Instanzsegmentierung zuständig. Der binäre Segmentierungsbild $\underline{\underline{B}}_b(t)$ und der Instanzsegmentierungsbild $\underline{\underline{B}}_i(t)$ wird durch das Modell erstellt. Bei der binären Segmentierung handelt es sich um eine semantische Segmentierung in zwei Kategorien (Fahrspur/Hintergrund); bei der Instanzsegmentierung wird die Cluster-Konzept verwendet, um jede Fahrspurlinie zu klassifizieren. Um die Genauigkeit der Anpassung der Fahrspurlinien zu sichern, muss das Rauschen im Binärbild verarbeitet werden. Das verarbeitete binäre Segmentierungsbild und das Instanzsegmentierungsbild werden als Eingaben verwendet, um ihre Fahrspurlinienmerkmale zu extrahieren. Die erhaltene Ausgabe sind die Koordinaten des Fahrspurlinienpunktsatzes $C_i(t)$. Schließlich werden die Fahrspurpolynome durch Kombination der Clustering-Ergebnisse angepasst, um ein Polynom für jede Fahrspurlinie zu erhalten.
- Lokalisierung: Das Hauptziel dieses Teils besteht darin, Verkehrsinformationen aus den Daten zu extrahieren, die zur Identifizierung von Verkehrsteilnehmerkategorien wie Fahrzeugen, Fußgängern, Fahrrädern usw. erforderlich sind. Sobald die Kategorie erkannt ist, muss das erkannte Objekt verfolgt werden. Es ist auch erforderlich, für jedes erkannte Fahrzeug eine eindeutige Identifikationsnummer id festzulegen und Informationen über die Position $\underline{P}_{id}(t)$ des Fahrzeugs sowie Informationen über die Fahrspur $l_{id}(t)$, auf der das Fahrzeug fährt, aufzuzeichnen.

4 Entwurf

In diesem Kapitel wird der Entwurf der Lokalisierung vorgestellt. In dieser Arbeit wird das Design der Lokalisierung beschrieben. In diesem Kapitel wird ein Video als Eingabe verwendet. Das Video zeigt die Verkehrsinformationen der Fahrspuren aus der Ego-Perspektive. Wie bereits erwähnt, umfasst die Informationsauswertung sowohl die Identifizierung von Fahrspurlinien als auch die Lokalisierung.

4.1 Erkennung der Fahrspur

In diesem Modell wird das Kodiermodell BiSeNetV2 verwendet. Die Aufgabe dieses Teils ist die Merkmalsextraktion, die aufgrund der parallelen Verzweigungsstruktur von BiSeNetV2 rechnerisch effizient ist. Nach der Dekodierung werden die binäre Abbildung und die Instanzabbildung erhalten. Dann wird die morphologische Schließungsoperation an der binären Abbildung durchgeführt, um die kleinen Hohlräume im Bild zu beseitigen und die Bildgrenzen zu glätten, um das Ergebnis der binären Segmentierung zu optimieren und zu bereinigen und ein klareres und vollständigeres Ergebnis der Fahrspursegmentierung zu erhalten. Danach wird mit der Verarbeitung der Segmentierungskarte begonnen: Zunächst werden die verarbeitete binäre Karte und die Instanzabbildung als Eingaben verwendet, um ihre Fahrspurmerkmale zu extrahieren, und die erhaltene Ausgabe sind die Koordinaten der Fahrspurpunktsätze.Das DBSCAN-Clustering wird für die Koordinaten der Fahrspurlinienpunkte durchgeführt, um die Anzahl der Cluster, die Clusterbeschriftungen und die Clusterzentren zu ermitteln. Um die Genauigkeit der Fahrspurberechnung zu gewährleisten, müssen Fahrspurpolynome durch Kombination der Clustering-Ergebnisse und Fahrspurkoordinaten angepasst werden, um die spezifischen Punktkoordinaten jeder Fahrspur zu erhalten. Das Verfahren wird durch Prozedur 1 dargestellt.

4.2 Lokalisierung

Die Lokalisierung erfordert zunächst eine genaue Erkennung der Fahrzeuge im Video. Da diese Aufgabe ei-

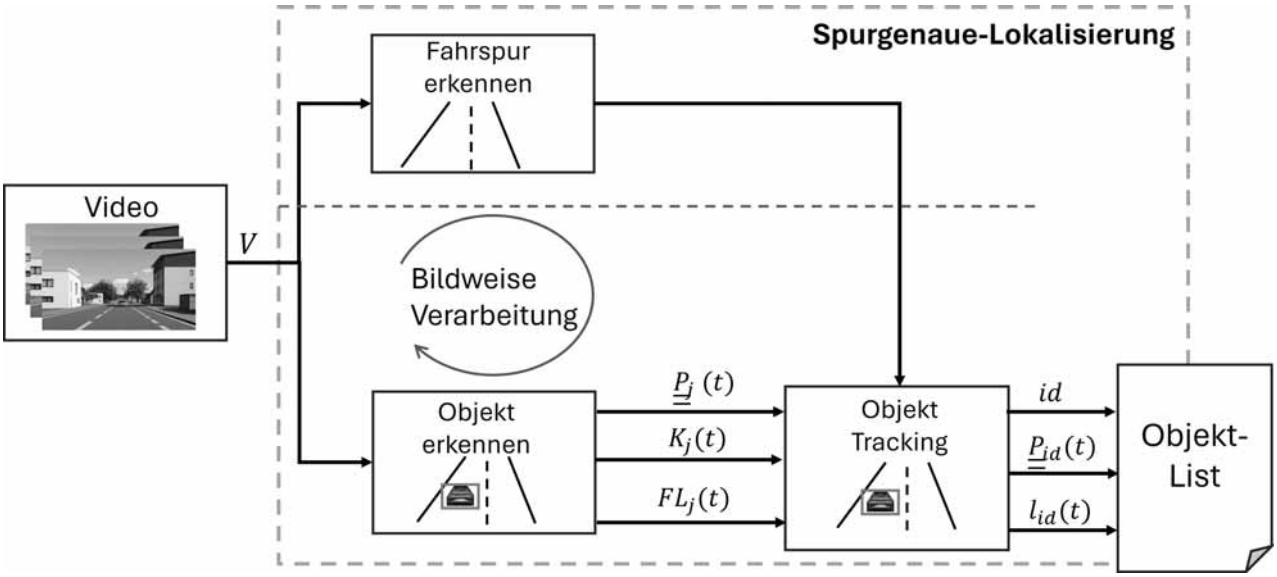


Abbildung 2: Konzeption der spurgenauen Lokalisierung

Prozedur 1 Erkennung der Fahrspur

1: INPUT:V

2: OUTPUT: $l_i(x)$

3: **for** $\underline{\underline{B}}(t)$ in V **do**

4: Kodieren: $\underline{\underline{F}}(t) = BiSeNet(\underline{\underline{B}}(t))$

5: Dekodieren: $\underline{\underline{B}}^b(t), \underline{\underline{B}}^I(t) = Decode(\underline{\underline{F}}(t))$

6: Filter: $\underline{\underline{B}}^f(t) = Filter(\underline{\underline{B}}^b(t))$

7: Cluster: $C_i(t) = DBSCAN(\underline{\underline{B}}^b(t), \underline{\underline{B}}^I(t))$

8: Fahrspur: $l_i(x) = Ausgelichung(C_i(t))$

9: **end for**

ne Effizient der Objekte im Video erfordert, sind die Anforderungen an die Verarbeitungszeit und die Genauigkeit des Erkennungsalgorithmus sehr hoch. Nach einem Vergleich verschiedener Objekterkennungsalgorithmen wurde der YOLO Algorithmus ausgewählt, da er in Bezug auf Geschwindigkeit und Genauigkeit überlegen ist. Nach der Erkennung eines Fahrzeugs erstellt der Algorithmus automatisch eine Box $\underline{\underline{P}}_{j(t)}$ um das Fahrzeug, der durch vier Koordinatenpunkte enthält. j ist die Nummer des erkannten Fahrzeugs. Diese Box befindet sich in der Nähe der umgebenden Kanten des Fahrzeugs, so dass der Mittelpunkt der unteren Seite der Box als Position des Fahrzeugs betrachtet werden kann. Zusätzlich wird im Erkennungsprozess ein Konfidenzwert $K_j(t)$ erzeugt, der die Wahrscheinlichkeit angibt, dass das erkannte Ergebnis zur Fahrzeugkategorie $FL_j(t)$ gehört. Dieser Konfidenzwert bestätigt nicht nur die Erkennungsgenauigkeit, sondern ist auch ein wichtiger Matchingparameter, um das erkannte Fahrzeug dem Objekttracker zuzuführen, damit wird sichergestellt, dass der Tracker die Fahrzeuge genau verfolgen kann.

Nach der erfolgreichen Erkennung von Fahrzeugen wird in dieser Arbeit der Byte-Track-Algorithmus verwendet, um die erkannten Fahrzeuge zu verfolgen. Die Hauptfunktion von Byte-Track besteht darin, jedem erkannten Fahrzeug eine eindeutige Identifikationsnummer zuzuweisen, die nach Erhalt der Position des erkannten Fahrzeugs und der Konfidenzwerte vergeben wird. Anhand dieser ID können die Positionen desselben Fahrzeugs in aufeinander folgenden Videobildern verfolgt. Durch die Analyse der Positionsdaten des Fahrzeugs in Bezug auf die Fahrspur kann außerdem festgestellt werden, dass sich das Fahrzeug auf einer bestimmten Fahrspur befindet. Byte-Track zeichnet auch die Position jedes Fahrzeugs in jedem Bild auf, um den Fahrweg des Fahrzeugs zu rekonstruieren. Diese Funktionen verbessern nicht nur die Kontinuität und Genauigkeit der Verfolgung, sondern liefern auch wichtige Daten über die Dynamische Information der Fahrzeuge. Das Verfahren wird durch Prozedur 2 dargestellt, wobei s angibt, wie vielen Metern jedes Pixel entspricht.

Prozedur 2 Lokalisierung

INPUT: V, $l_i(x)$, s,r

OUTPUT: $t, id, \underline{P}_{id}(t), v_{id}(t), l_{id}(t))$

for $\underline{\underline{B}}(t)$ in V **do**

$f = f + 1$

$\underline{\underline{B}}(t)$ = $addline(\underline{\underline{B}}(t), l_i(x))$

$\underline{P}_j, K_j(t), FL_j(t) = YOLO(\underline{\underline{B}}(t))$

$\underline{P}(t)^{j,B}$, $ID = Track(\underline{P}_j, K_j(t), FL_j(t))$

$\underline{P}_{id}(t) = Box\_Mittelpunktder\_Unterseite(\underline{P}(t)^{j,B}, id)$

if $\underline{P}_{id}(t)[x] > l_i^{-1}(\underline{P}_{id}(t)[y])$ **then**

$l_{id}(t)$ ist i

end if

$Save(t, id, \underline{P}_{id}(t), l_{id}(t))$

end for

5 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Spurgenaue Lokalisierung dargestellt. Diese Funktion wird in einer virtuellen Simulationsumgebung validiert.

Abbildung 3 zeigt die Validierung dieser Objekterkennungsfunktion in einem virtuellen Simulationsumgebung. Die Fahrzeuge in dieser Umgebung werden erkannt. Das Fahrzeug in dem rechteckigen Box stellt das erkannte Fahrzeug dar. Auf dem Kasten befindet sich eine Nummer, die der Identifikationsnummer des erkannten Fahrzeugs bedeutet.

Abbildung 3: Identifizierte Fahrzeuge

Abbildung 4 zeigt die erkannten Fahrspuren in der Simulationsumgebung. Die Linien stellen die angepassten Fahrspuren dar, die mit den tatsächlichen Fahrspuren im Bild übereinstimmen.

Abbildung 4: Erkannte Fahrspuren

Abbildung 5 zeigt die befahrbaren Bereiche der Fahrspuren und ihre Fahrbahnnummern. Die Straße auf der linken Seite ist mit 1 gekennzeichnet, die Straße in der Mitte mit 2, und der Fahrradweg auf der rechten Seite ist nicht erkannt. Zu diesem Zeitpunkt befinden sich sowohl das Ego-Fahrzeug als auch das erkannte Fahrzeug auf Fahrspur 2.

Abbildung 5: Numerierte Fahrspur

6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden sowohl die Selbstlokalisierung auf Fahrspurebene als auch die Lokalisierung anderer Fahrzeuge auf Fahrspurebene entwickelt. Diese Arbeit besteht aus zwei Aspekten: Erstens wird ein Modell auf der Grundlage einer Kodierungs- und Dekodierungsstruktur angewandt, um die Fahrspuren zu identifizieren ununnd zu nummerieren. Zum anderen wird der Algorithmus YoLo verwendet, um andere Fahrzeuge auf der Straße zu identifizieren und deren Fahrspur zu berechnen.Die Funktion wurde in einem

szenariobasierten virtuellen Simulationsumgebung validiert.

Zukünftige Arbeiten werden sich auf die Optimierung der Straßenerkennungsfunktion zur Anpassung an andere Umgebungen konzentrieren. Darüber hinaus werden weitere Sensoren wie LiDAR fusioniert und die erkannten Informationen über V2X-Paare an andere Verkehrsteilnehmer übermittelt werden.

Danksagung

Gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur unter Fördernummer ZN4172 im Niedersächsischen Vorab der VolkswagenStiftung.

Literatur

[1] Francini M, Chieffallo L, Palermo A, Viapiana MF. Systematic Literature review on smart mobility: A framework for future “quantitative” developments. *Journal of Planning Literature*. 2021;36(3):283–296.

[2] SAE International. Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems. 2021;SAE Standard J3016_202104.

[3] Rabe J, Necker M, Stiller C. Ego-lane estimation for lane-level navigation in urban scenarios. 2016; pp. 896–901.

[4] Qian C, Zhang H, Li W, Tang J, Liu H, Li B. Cooperative GNSS-RTK ambiguity resolution with GNSS, INS, and LiDAR data for connected vehicles. *Remote Sensing*. 2020;12(6):949.

[5] Ding W, Hou S, Gao H, Wan G, Song S. LiDAR Inertial Odometry Aided Robust LiDAR Localization System in Changing City Scenes. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020; pp. 4322–4328.

[6] Liu-Henke X. *Mechatronische entwicklung der aktiven Feder-, Neigetechnik für das schienenfahrzeug RailCab*. VDI Verlag. 2005.

[7] Li J, Jiang F, Yang J, Kong B, Gogate M, Dashtipour K, Hussain A. Lane-DeepLab: Lane semantic segmentation in automatic driving scenarios for high-definition maps. *Neurocomputing*. 2021; 465:15–25.

[8] Wang Z, Ren W, Qiu Q. Lanenet: Real-time lane detection networks for autonomous driving. *arXiv preprint arXiv:180701726*;.

[9] Wu Z, Qiu K, Yuan T, Chen H. A method to keep autonomous vehicles steadily drive based on lane detection. *International Journal of Advanced Robotic Systems*. 2021;18(2):17298814211002974.

[10] Dewangan DK, Sahu SP, Sairam B, Agrawal A. VLDNet: Vision-based lane region detection network for intelligent vehicle system using semantic segmentation. *Computing*. 2021;103(12):2867–2892.

[11] Lu S, Luo Z, Gao F, Liu M, Chang K, Piao C. A fast and robust lane detection method based on semantic segmentation and optical flow estimation. *Sensors*. 2021;21(2):400.

[12] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:14091556*;.

[13] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. 2015.

[14] Yu C, Gao C, Wang J, Yu G, Shen C, Sang N. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation. 2020.

[15] Xiangbing LI, Lian C. Face Detection in Natural Scene Based on Improved Faster-RCNN[J]. *Computer Engineering*. 2021;47(1):210–216.

[16] Huang KQ, Ren WQ, Tan TN, et al. A review on image object classification and detection. *Chinese Journal of Computers*. 2014;37(6):1225–1240.

[17] Liu H, Ma J, Zhang G. Review of studies on deep learning-based content recommendation algorithms [J]. *Computer Engineering*. 2021;47(07):1–12.

[18] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. Ieee. 2001; pp. I–I.

[19] Felzenszwalb P, McAllester D, Ramanan D. A discriminatively trained, multiscale, deformable part model. In: *2008 IEEE conference on computer vision and pattern recognition*. Ieee. 2008; pp. 1–8.

[20] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014; pp. 580–587.

[21] Girshick R. Fast R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015; .

[22] Cai Z, Vasconcelos N. Cascade R-CNN: Delving Into High Quality Object Detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018; .

[23] Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017; pp. 2117–2125.

[24] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016; pp. 779–788.

[25] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: *2017 IEEE international conference on image processing (ICIP)*. IEEE. 2017; pp. 3645–3649.

[26] Zhang Y, Sun P, Jiang Y, Yu D, Weng F, Yuan Z, Luo P, Liu W, Wang X. Bytetrack: Multi-object tracking by associating every detection box. In: *European conference on computer vision*. Springer. 2022; pp. 1–21.

SNE SIMULATION NOTES EUROPE

Simulation Notes Europe (**SNE**) provides an international, high-quality forum for presentation of new ideas and approaches in simulation - from modelling to experiment analysis, from implementation to verification, from validation to identification, from numerics to visualisation - in context of the simulation process.

SNE seeks to serve scientists, researchers, developers and users of the simulation process across a variety of theoretical and applied fields in pursuit of novel ideas in simulation and to enable the exchange of experience and knowledge through descriptions of specific applications. **SNE** puts special emphasis on the overall view in simulation, and on comparative investigations, as benchmarks and comparisons in methodology and application. Additionally, **SNE** welcomes also contributions in education in / for / with simulation.

SNE is the official membership journal of **EUROSIM**, the Federation of European simulation societies and simulation groups, and the scientific membership journal of **ASIM**, the German simulation society. **SNE** is open for post-conference publications and for special issues organized by **EUROSIM** societies, e.g. **ASIM** thematic special issues or **ASIM** post-conference special issues.

SNE is primarily an electronic journal and follows an open access strategy, with free download in basic layout. Members of EUROSIM societies, as **ASIM**, **SIMS**, e.g. are entitled to download **SNE** in an elaborate and extended layout. Print **SNE** is available for specific groups of **EUROSIM** societies.

www.sne-journal.org

ISBN ebook 978-3-903347-65-6
ARGESIM Report 47
www.argesim.org

DOI 10.11128/arep.47
ASIM Mitteilung 190
www.asim-gi.org