

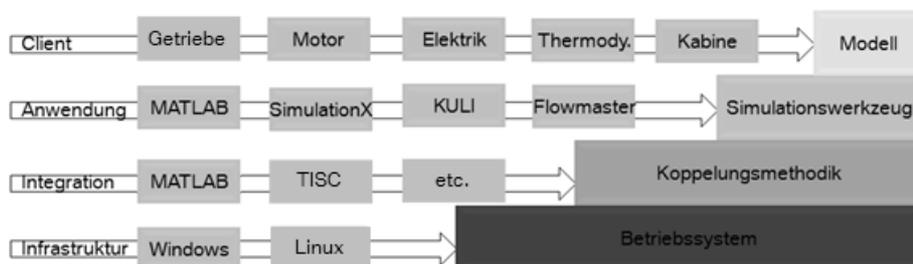
# Analyse des Einflusses der Co-Simulation bei der Modellintegration

Andreas Maier  
Andreas-Maier@online.de  
Maier\_A@rz.fh-augsburg.de  
HS-Augsburg / Elektrotechnik  
Hochschule für angewandte Wissenschaften Augsburg,  
An der Hochschule, 1 / 86161 Augsburg

Dr. Stefan-Alexander Schneider  
Stefan-Alexander.Schneider@bmw.de  
BMW AG, Forschungs- und Innovationszentrum,  
Knorrstraße 146 / 80788 München

## Kurzfassung

Dieser Beitrag fasst die Ergebnisse der Diplomarbeit „Bewertung des Simulationsverhaltens von Co-Simulations-Werkzeugen für einen Fahrdynamikregelverbund“ von Herrn Andreas Maier an der Fachhochschule Augsburg bei der BMW AG zusammen. Das Ziel der Diplomarbeit war die Evaluierung von auf dem Markt erhältlichen Integrationswerkzeugen für die gekoppelte Simulation um anschließend mit geeigneten Kandidaten den Einfluss der Co-Simulation mittels eines BMW internen Fahrdynamikregelmodells zu untersuchen. Aus der Diplomarbeit wird der Schwerpunkt Analyse präsentiert. Wesentliches Ergebnis dieser Analyse ist der Nachweis für bitidentisches Verhalten einer Co-Simulation mittels Integrationsplattform und eines Co-Simulation reproduzierenden Modells. Als Referenzverhalten wird das Verhalten angesehen, bei dem das originale Gesamtmodell in unabhängige Teilsysteme zerlegt wurde, deren externe Verbindungssignale durch Verzögerungsböcke ergänzt wurden. Durch diese Maßnahmen kann belegt werden, dass der Einfluss der Integrationsschicht keine unerwarteten Effekte in die Anwendungsschicht einschleppt.



**Bild 1:** Beispiel für die Applikationsbebauung einer heterogenen Simulationslandschaft

# 1 Stellenwert der Modellintegration im Entwicklungsprozess

Die Komplexität moderner Produkte erzwingt manchmal unübersichtliche Entwicklungsprozesse mit entsprechend vielen Entwicklungsphasen. Es gibt mannigfache Möglichkeiten diese Entwicklungsprozesse zu gestalten, siehe [1], und oftmals existiert kein Königsweg. Aus Sicht der Autoren zeichnen sich erfolgreiche Entwicklungsprozesse u.a. dadurch aus, dass eine einfache Bewertung der angestrebten funktionalen Eigenschaften prozesslokal, zeitnah und kostenoptimiert zu den Architekturentscheidungen des Systems und der Funktionen stattfinden können. Das ist sicherlich nicht für alle Aspekte eines Produktes durchführbar, dennoch ermöglicht die Verwendung von ausführbaren Eigenschaftsmodellen in der Funktionsentwurfsphase genau diese erwünschten Antworten zu (pre-)konfigurierten Systemen und Funktionen. Die wesentliche Voraussetzung ist dabei, dass die Modelle die Wirklichkeit in den betrachteten Eigenschaften hinreichend genau abbilden um die gewünschten Eigenschaften frühzeitig berechnen und somit im Sinne des Entwurfs bzgl. der Ziele bewerten zu können.

Virtuelle Ansätze ermöglichen dies auf der Basis von Software. Mit sogenannten Modellen resp. Programmen können Systeme, Funktionen und Eigenschaften simuliert werden. Die Autoren beobachten aktuell, dass graphische Modellierung konsequenterweise einen immer höheren Stellenwert einnimmt. Die Auswahl der Modellierungssprache sollte u.a. auch auf folgende Aspekte der Modellierungssprache eingehen:

1. Wie gut bildet die Modellierungssprache den Modellierungsgegenstand ab?
2. Wie gut ist die Semantik der Modellierungssprache beschrieben?
3. Wie gut sind die wesentlichen Eigenschaften der Modelle intuitiv von einem Modellierer umsetzbar resp. erfassbar?
4. Wie können Änderungen in den Entwurfsentscheidungen einfach umgesetzt werden?

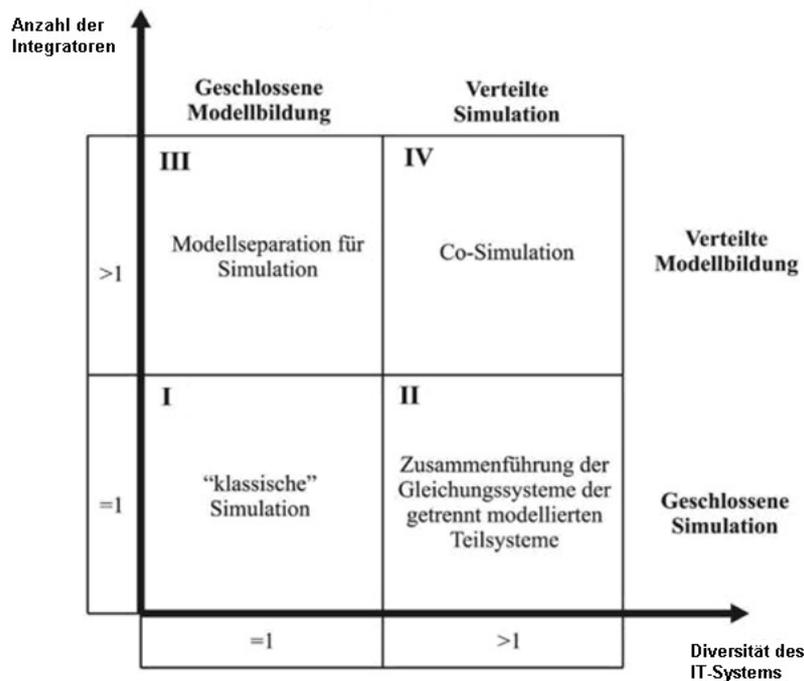
Die Güte der Simulationen und damit der Aussagen zu den Eigenschaften hängt einerseits von der Güte der Abbildung des Modells ab. Typischerweise werden die Modelle daher im Laufe der Entwicklung immer mehr verfeinert um die quantitativen Aspekte hinreichend genau berechnen zu können. Andererseits findet die Simulation auch bei der virtuellen Integration immer mehr Beobachtung um frühzeitig qualitative Effekte insbesondere bereichsübergreifender Aufgabenstellungen identifizieren zu können[2].

Im Folgenden schränken wir uns auf die Betrachtung diskreter Simulationen ein, wie sie z.B. bei der Entwicklung von Steuergeräte-Software vorliegt.

Sowohl die Modellverfeinerung als auch die Modellintegration führt zu immer umfangreicheren Modellen, Simulationen und Ergebnisstrukturen. Die klassische Simulation geht dabei als Spezialfall der sogenannten Co-Simulation auf.

Eine gekoppelte Simulation von Teilsystemen ist mit unterschiedlichen Strategien und verschiedenen Ansätzen durchführbar. Einen Vorschlag zur Begriffsabgrenzung zeigt beispielsweise Geimer in [3].

Dieser Ansichtswise nach lässt sich die Simulation von Teilsystemen wie in Bild 2 ersichtlich in vier Bereiche, abhängig von der Anzahl der Integratoren und der Diversität des IT-Systems einteilen.



**Bild 2:** Klassifikation der Co-Simulation nach Geimer [3]

Die Methode Co-Simulation führt zusätzliche Abstraktion in die Modellierung ein und kann daher Fehler in die berechneten Ergebnisse einschleppen. Es empfiehlt sich diesen Einfluss zu analysieren um keine unnötigen funktionalen Fehler in Simulationen insbesondere durch die verwendeten IT-Werkzeuge einzuführen.

## 2 Aufgaben der Co-Simulation

Eine zentrale Eigenschaft der Co-Simulation ist, dass durch die Koppelung der Teilsysteme mehrere Integratoren unabhängig Lösungen innerhalb der Teilsysteme berechnen und zu diskreten Zeitpunkten Informationen austauschen. Jedes Teilsystem kann durch eine mathematische Iterationsvorschrift repräsentiert werden.

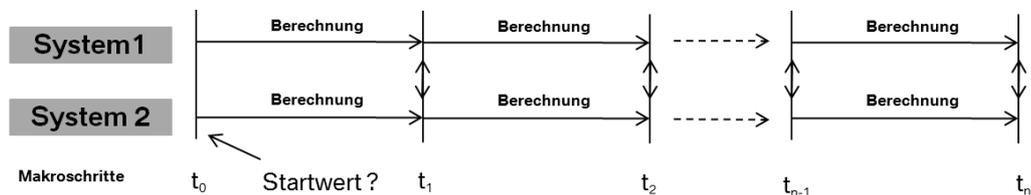
Die Integration der Modelle führt somit zu einem übergeordneten Iterationsverfahren. Die Co-Simulation muss daher den Informationsaustausch und die Synchronisation der einzelnen Iterationsverfahren übernehmen. Der Austausch der Teilergebnisse findet dabei

über eine sogenannte Integrationsschicht statt. Allgemein ist dies entweder per Modellexport inklusive der Löser des verwendeten Modellierungswerkzeuges oder durch Koppelung der Ausführungsumgebungen möglich.

Es stellt sich nun grundsätzlich die Frage, wie die Abarbeitungsreihenfolge der Teilsysteme innerhalb des Gesamtsystems erfolgen soll [4].

Die Möglichkeit der externen Festlegung einer Abarbeitungsreihenfolge wird hier nicht weiter betrachtet, da sie i.a. detailliertes Wissen der zu simulierenden Teilmodelle erfordert und das von einer universell einsetzbaren Entwicklungsmethode nicht einfach gewährleistet werden kann.

Daher konzentrieren wir uns im Folgenden auf eine parallele Abarbeitungsreihenfolge der Teilsysteme, wohlwissend einen Performanceverlust in Kauf nehmen zu müssen. Dabei tauschen die Teilsysteme zu diskreten Zeitpunkten sämtliche Ergebnisse aus, siehe Abbildung 3.



**Bild 3:** Parallele Ablaufreihenfolge der Teilsysteme

Die parallele Abarbeitungsreihenfolge führt dazu, dass sich die Teilsysteme gegenseitig nur verzögert beeinflussen können. Das steht dem Wunsch bereichsübergreifender Effekte identifizieren zu können zunächst scheinbar im Wege und ist eine potenzielle Fehlerquelle vergleichbar mit einer Totzeit, die in die Beschreibung des Gesamtsystems eingebracht wird. Diese Fehlerquelle kann allerdings durch die Verkleinerung der Makroschrittweite<sup>1</sup> hinreichend kontrolliert werden, die insbesondere bei Echtzeitregelsystemen nach unten durch technische Vorgaben bestimmt ist.

Abschließend soll noch auf eine weitere wesentliche Fragestellung hingewiesen werden, die insbesondere bei der parallelen Abarbeitungsreihenfolge aufgeworfen wird: Die Vorgabe der Initialwerte des Gesamtsystems zum Startzeitpunkt  $t_0$  auf der Basis der Initialwerte der Teilsysteme. Die Initialwerte des Gesamtsystems müssen so bestimmt werden, dass die gewünschten Eigenschaften des Gesamtsystems garantiert berechnet werden können.

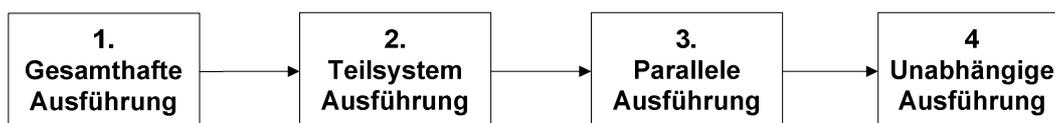
<sup>1</sup> Die **Makroschrittweite** ist der Zeitbereich, in dem die Berechnung der Teilsysteme erfolgt. Am Ende jedes Makroschrittes findet ein Informationsaustausch aller Teilsysteme statt. Die Rechen-schrittweite in den Teilsystemen wird dagegen als **Mikroschrittweite** bezeichnet.

### 3 Analyse-Methode

Im Folgenden sollen die erforderlichen Grundannahmen der Co-Simulation aufgezeigt um mittels einer systematischen Analyse zugänglich zu werden. Die erste bindende Grundannahme ist die parallele Abarbeitungsreihenfolge. Weitere bestimmende Faktoren bezüglich des Simulationsverhaltens sind:

- Koppelung der Teilsysteme über verschiedene Ausführungsumgebungen,
- die Wahl der Integrationsschicht und
- die Aufteilung der Anwendungsschicht in mehrere einzelne Instanzen.

Wie eingangs schon angekündigt, ist die bedeutsamste Anforderung an ein eingesetztes IT-System, dass keine weiteren bzw. unbekannt Fehler in die Simulation eingeschleppt werden. Das dargestellte Analyseraster in Bild 4 dient zur Erklärung der Vorgehensweise. Ziel ist hierbei den Einfluss der einzelnen Transformationen zu isolieren um schrittweise Vertrauen in das Simulationsverhalten einer Co-Simulation zu erlangen, siehe [5].



**Bild 4:** Analyseraster zur Verifikation von Simulationsbeeinflussenden Effekten

#### **Szenario 1: Gesamthafte Ausführung**

Als Referenz stellt dies die Ausgangsbasis für weitere Szenarien dar. Es handelt sich hierbei um ein Gesamtsystem eines physikalischen Fahrdynamikregelmodelles, welches in einem Modellierungswerkzeug ohne zusätzliche Veränderungen simuliert wird.

#### **Szenario 2: Teilsystemausführung**

Die für die spätere Co-Simulation verwendeten Modellteile des Gesamtsystems werden zu unabhängigen Subsystemen deklariert. Das Definieren eines Systems als unabhängig bedeutet für die Festlegung der Abarbeitungsreihenfolge, dieses als eine Einheit zu interpretieren und sobald alle Eingangssignale vollständig vorliegen in einem Makro-Rechenschritt abzuarbeiten, bevor mit den restlichen Systemen weiter gerechnet werden kann.

Durch diese Einstellung werden somit u.a. die Regler von den Regelstrecken getrennt und aus Sicht der Abarbeitungsreihenfolge als separate Modelle betrachtet.

Diese Festlegung stellt in Bezug auf die Architektur der Co-Simulation die erste Näherung dar, da bei Verwendung von zwei getrennten Simulations-Instanzen die Abarbeitungsreihenfolge nur für die eigene Modellstruktur erfasst und über die Subsystemgrenzen hinaus nicht mehr frei gewählt werden kann.

### **Szenario 3: Parallele Ausführung**

An den Schnittstellen der unabhängigen Teilsysteme werden zwischen den Ein- und Ausgängen zusätzlich  $1/z$ -Blöcke geschaltet. Diese  $1/z$ -Blöcke realisieren die verzögerte Signalübertragung von einem Sende- zu einem Empfangssystem durch die Kommunikation, bedingt durch die parallele Ablaufreihenfolge der Co-Simulationsplattform. Die  $1/z$ -Blöcke ermöglichen insbesondere Initialwerte zum Zeitpunkt  $t_0$  festzulegen. Die Abtastrate in den  $1/z$ -Blöcken entspricht hierbei dem Kommunikations-Intervall der Co-Simulationsschnittstellen und der Makrorechenschrittweite.

### **Szenario 4: Unabhängige Ausführung**

Die zuvor in einem Simulationswerkzeug als unabhängig deklarierten Teilsysteme werden nun aus diesem Verbund herausgetrennt und jeweils auf weitere Simulationswerkzeuge des gleichen Typs verteilt. Jedes der Teilsysteme wird nun über implementierte Schnittstellen mit der Co-Simulationsplattform gekoppelt und simuliert.

## **4 Ergebnisse**

Die in Abschnitt 3 beschriebene Analyse-Methode wurde auf ein BMW internes Fahrdynamikregel- mit einem DSC-Modell über das Simulationswerkzeug MATLAB angewendet. Als Co-Simulationsplattform wurde TISC von der TLK-Thermo GmbH [6] eingesetzt.

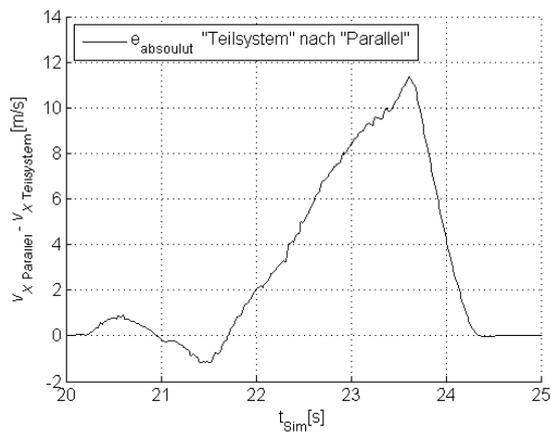
### **Einfluss Übergang „Gesamt“ nach „Teilsystem“**

Die Deklaration der Teilsysteme als unabhängig führte bei den ausgangseitigen Schnittstellensignalen des DSC-Modells zum Zeitpunkt  $t_0$  zu unterschiedlichen Werten  $x_0$ . Bis auf diese Abweichung waren die Simulationsdaten bitidentisch.

Hier zeigt sich bereits, dass für die Verwendung einer modularisierten Co-Simulation in einer heterogenen Simulationslandschaft bei der Entwicklung der Teilsysteme auf die Zuweisung als unabhängig geachtet werden muss bzw. diese daraufhin validiert sein müssen um anschließend im gekoppelten Gesamtverbund gültige Aussagen zu erzeugen.

## Einfluss Übergang „Teilsystem“ nach „Parallel“

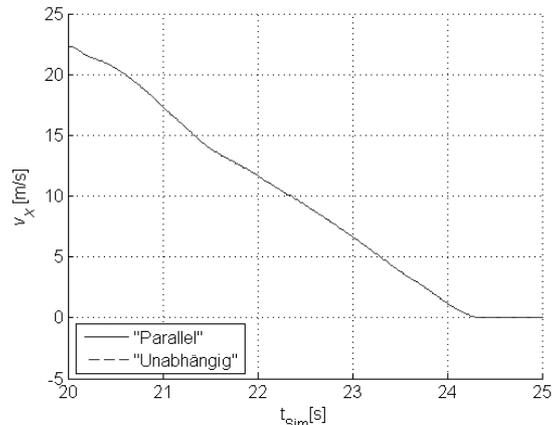
Das Einfügen der 1/z-Blöcke zur Nachbildung der parallelen Ablaufreihenfolgen hat wie erwartet große Auswirkungen auf das Simulationsverhalten. Am gewählten Fahrmanöver „Bremsen auf einer  $\mu$ Split-Strecke“ (Bild 5) ist der instabilere Zustand des Fahrzeuges während des Bremsvorganges erkennbar. Das Fahrzeug ist beim Szenario „Parallel“ zur Dauer des Bremsvorganges gegenüber dem „Teilsystem“ in der Spitze rund 11.35 m/s schneller in x-Richtung unterwegs. Hervorgerufen wird die Geschwindigkeitsdifferenz durch die über das DSC ausgetauschten Bremsignale aufgrund der Schnittstellenanpassung hinsichtlich der Co-Simulation.



**Bild 5:** Geschwindigkeitsdifferenz beider Szenarien

## Einfluss Übergang „Parallel“ nach „Unabhängig“

Der Vergleich beider Simulationsergebnisse bestätigt die Annahme und liefert die gewünschte Erkenntnis, dass bei einer Co-Simulation **keine** weitere **Abweichung** durch das **Austauschen** der **Integrations**schiicht erfolgt. Die Ergebnisstrukturen beider Simulationsergebnisse sind **bitidentisch**. Bild 6 zeigt dies am gewählten Beispiel der deckungsgleichen  $v_x(t_{sim})$ -Kennlinien des Fahrzeuges während des Bremsvorganges.



**Bild 6:** Szenarien „Parallel“ und „Unabhängig“ sind bitidentisch

## 5 Ausblick

Die Analyse des Co-Simulationsverhaltens anhand eines industrierelevanten Beispiels in den vorgestellten Szenarien zeigt, dass der Übergang von Gesamtsystem zu paralleler Ausführung aufgeteilt werden kann in wohlverstandene Teilübergänge. Dies ermöglicht die Verhaltens-invariante Umstrukturierung eines Gesamtsystems, das in einer Co-

Simulation mit paralleler Ausführung simuliert werden kann. Um das gewährleisten zu können, sind entsprechend der Teilübergänge sogenannte Modellierungsrichtlinien einzuhalten:

- Bereitstellung von unabhängigen, validierten Teilsystemen,
- Modulkonzept des Gesamtmodells zur Schnittstellenbeschreibung,
- wohldefinierte Schnittstellen, u.a. einheitliche Namenskonvention, Verwendung von SI-Basis-Einheiten, einheitliche Datentypen, einheitliche Bereitstellung von Modellinformationen,
- globale Variablen dürfen nur von einem Sender bereitgestellt werden,
- Initialwerte für sämtliche Teilsysteme im Rahmen des Gesamtsystems,
- das integrierte Gesamtmodell berücksichtigt die parallele Ausführung bereits durch 1/z-Blöcke sowie
- Berücksichtigung von Abbruchkriterien der Simulation innerhalb der Teilsysteme.

Die Ergebnisse zeigen, dass der Einfluss der Co-Simulation bei der Modellintegration verstanden und kontrolliert werden kann. Die Einhaltung gewisser Modellierungsrichtlinien für die Simulation garantiert darüber hinaus ein von der Integrationsschicht unabhängiges Gesamtverhalten. Die vorliegende Diplomarbeit ist ein erster Schritt in Richtung einer systematischen Analyse. Die Autoren sind daher für weiterführende Literaturhinweise bzgl. Einflüsse der Co-Simulation auf das Gesamtsystemverhalten dankbar.

## 6 Literatur

- [1] *International Council on Systems Engineering (IncoSe); American Institute of Aeronautics and Astronautics (AIAA)*:  
[http://www.incoSe.org/ProductsPubs/pdf/SEPrimerAIAA-INCOSE\\_1997-08.pdf](http://www.incoSe.org/ProductsPubs/pdf/SEPrimerAIAA-INCOSE_1997-08.pdf), 1997.
- [2] *Lund, C; Maister, W; Lange, C; Beyer, B.*: Innovation durch Co-Simulation!, Expert Verlag, Haus der Technik Fachbuch, Wärmemanagement des Kraftfahrzeuges VI, 2008.
- [3] *Geimer, M; Krüger, T; Linsel, P.*: Co-Simulation, gekoppelte Simulation oder Simulatorkopplung?, O + P Zeitschrift für Fluidtechnik, Band 50, 2006.
- [4] *Dronka, S.*: Die Simulation gekoppelter Mehrkörper- und Hydraulik-Modelle mit Erweiterung für Echtzeitsimulation, Shaker Verlag, Aachen, 2004.
- [5] *Maier, A.*: Diplomarbeit: Bewertung des Simulationsverhaltens von Co-Simulationsplattformen für einen Fahrdynamikregelverbund, HS-Augsburg / BMW Group München, 2011.
- [6] Software TISC, TLK-Thermo GmbH, <http://www.tlk-thermo.com/>
- [7] *Huckle, T., Schneider, S.-A.*: Numerische Methode, Springer, 2006.