

Comparison of Hybrid Modelling Approaches: ARGESIM-Comparison BCP - Bouncing Ball, Circuit with Diodes, Rotating Pendulum

Andreas Körner¹, Bernhard Heinzl¹, Matthias Rößler¹, Günther Zauner²,
Felix Breitenecker¹, Horst Ecker³
akoerner@asc.tuwien.ac.at

¹Institute for Analysis and Scientific Computing, Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

²dwh Simulation Services
Neustiftgasse 57-59, 1070 Vienna, Austria

³Institute for Mechanic and Mechatronics, Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

Abstract

Modelling and simulation of hybrid systems is getting more and more important in advanced modelling theory and application. Therefore, the requirements regarding flexibility on modern simulators are getting higher and higher. The necessity of fast and stable algorithms is increasing considering more complex systems of interest. In the last ten years, physical modelling started to support simulation experts in building their complex structures. Thereby the needs for simulators are going up a second time. To see how modelling and simulation environments deal with state events of different order, three classical examples are discussed. These parts offer a spectrum of questions for testing basic features and they represent minimum requirements to hybrid simulators regarding state events.

At first, the bouncing ball with some extensions is discussed. The following section deals with an electrical circuit - it can be used to compare the simulation results of different implementation techniques (e.g. physical modelling vs. classical DAE form) in one simulator. The third example is a definition of a pendulum on a rope with free flight phase, representing a simple system with changes in the state space dimension depending on the state variables. The main focus lies on different implementation techniques and the comparison of solution quality, calculation time and/or usability of the implementation strategy.

1 Introduction

Since 2000 the relatively old CSSL standard for simulation systems has become obsolete, and new standards and techniques for system simulation are arising.

At modelling level, Physical Modelling or Component-based Modelling has introduced a new era for multidomain modelling and system simulation. The ‘components’ may be part of textual or graphical libraries in various domains. With Modelica and with competitive VHDL-AMS, modelling languages with a certain standard have emerged, using component-based modelling with a-causal relations; the figure at right shows graphical physical modelling in mechanics and electrical engineering.

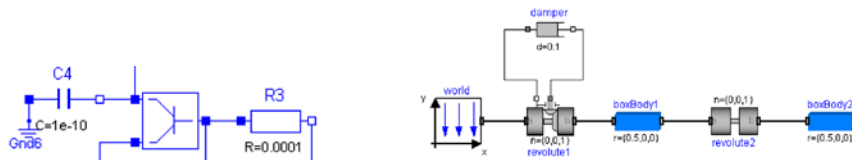


Figure 1: Modelica physical modelling for Analog Electrical Domain and Mechanics Multibody Domain.

Thereby, the components may be part of textual or graphical libraries in various domains. Of course it makes sense to couple this component based a-causal graphical modelling with the classical signal-oriented graphical modelling.

From mathematics’ viewpoint, instead of explicit state models now implicit ‘law-oriented’ model descriptions have become basis for subsequent simulation, resulting in implicit differential-algebraic systems (DAEs).

The additional algebraic equations have emerged a new problem, which relates to simulation level. In principle, the simulator must translate the model description into a DAE system with proper structure of differential and algebraic equations for the extended state space (consisting of differential states and algebraic states), so that a ‘modern’ DAE solver can handle the implicit state space model with sufficient accuracy and sufficient convergence. These conditions are given, if roughly speaking, dependencies of differential states and algebraic states are not too ‘weak’; unfortunately this is not the case in many e.g. mechatronic systems. Here either the simulation system is capable of automatic symbolical ‘extension’ of the systems (introduction of new differential states by differentiating the system symbolically – so-called index reduction), or the model description is split into different models, whereby the algebraic conditions only control switching between the models (handling a Structural-dynamic System). Consequently modern DAE solvers offer also features for state event handling: the solver checks the algebraic condition.

The algebraic equations, e.g. constraints, are coming along with another new challenge, with structural dynamic systems, which on the other hand are also coming along with parallel and serial model coupling (co-simulation). Constraints are very often coupled with state-dependent conditions for their validity – like loss of freedom, etc., requiring a conditional change of the model description controlled by thresholds etc. – and resulting in structural-dynamic systems; consequently the problem of state event description and state event handling becomes much more complex than in classic ODE models and raises new questions for proper hybrid model description. Hybrid systems and structural-dynamic Systems are more or less efficiently supported at modelling level, but another novelty in system simulation can help: modelling with statecharts.

At experiment level / experimental frame, new approaches have been driven by computer science. While object oriented programming became the quasi standard of software development, the object-oriented modelling paradigm, as a method for modelling of systems, has emerged as very useful for modelling of simulation models and object-oriented techniques have been introduced in simulation modelling. Especially statechart diagrams as part of the Unified Modelling Language (UML), a set of graphical modelling techniques and the de facto modelling standard in object-oriented programming, have shown to be really useful for discrete event simulation modelling.

Statechart diagrams were introduced by David Harel of the Weizmann Institute of Science in the 1980s. He extended the conventional state-transition-diagram formalism by the concepts of hierarchy, concurrency and communication. The basic principles of state-transition modelling include only a handful of concepts, basically drawing states a system can be in and transitions from one state to another. But not until the adding of the principles invented by Harel, statechart modelling became a powerful and at the same time easy to apply method to model arbitrary systems. The first version of statecharts was applicable for just discrete systems, but in the meantime a real time compliant version as part of the UML for real time (UML-RT) - standard, was developed, allowing also ‘dynamic states’ governed by ODEs/DAEs. With hybrid statecharts – the figure at left shows an example for the pilot ejection model – the era of Dynamic Statechart Modelling started which allow for transparent modelling of structural-dynamic systems and bridging between model level and experiment level.

The Physical Modelling approach sets new challenges for the translators of simulation systems, with some side effects to the environments (simulation level/runtime level/experiment level) of the simulation systems. The Dynamic Statechart Modelling approach not only sets challenges for the modelling level, but it also sets challenges for the experiment level and the approach constitutes a generic alliance between model frame and experimental frame – in contrary to Zeigler’s early postulation of ‘separation of model frame and experimental frame’.

The new ARGESIM Benchmark C20 – Hybrid and Structural-dynamic Systems intends to investigate new approaches to hybrid modelling and to handling of structural dynamic systems, with emphasis on physical modelling techniques and state event modelling and state event handling. Three case studies allow investigation of different modelling techniques and different implementations for hybrid systems or structural-dynamic systems, resp.: the Bouncing Ball, a Switching RLC Circuit, and the Rotating Pendulum – the initials of the case studies set up the abbreviation for this benchmark: the ARGESIM BCP – Benchmark.

2 Case Study 1: Bouncing Ball

Event Contact Phase.

Free Falling/Flying Phase. The motion of a free falling mass in a gravitational field is given by the following two differential equations for position x and velocity v :

$$\begin{aligned}\dot{x} &= v \\ \dot{v} &= -g - \beta v^2 \text{sign}(v)\end{aligned}$$

with g acceleration of gravity ($9,81 \frac{m}{s^2}$ on Earth) and β as coefficient for the air resistance. β is a cumulative constant combining the drag coefficient C_d (0.47 for a sphere), the density of the surrounding medium ρ , the cross section area A and the mass m :

$$\beta = \frac{1}{2} C_d \cdot \rho \cdot \frac{A}{m}$$

In reality the density would be a function of the height, but for small heights it can be assumed as a constant, thereby getting a simpler model. The state vector is $\vec{x}(t) = (x(t), v(t))^T$ for the falling/flying phase.

Setting β to zero can either represent a very small, but heavy ball, or indicate that the experiment is made in a vacuum, or simply neglecting the air resistance, resulting in a very simple linear model of linear type:

$$\frac{d}{dt} \vec{x}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \vec{x}(t) - g \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Event Contact Model. Bounce and contact can be implemented using different models. The event contact model is quite a simple approach, using Newton's 3rd law, and a coefficient μ to describe the loss of energy in case of bounce or impact, resp., and neglecting the deformation [3]. The velocity $v_{prev}(\hat{t}^I)$ of the ball right 'before' the impact with the ground 'jumps' to the velocity $v_{new}(\hat{t}^I)$ due to 'reflection' and energy loss.

Mathematical analysis. The linear model (41) allows an analytical calculation of the impact time instants $t_{A,m}$. Assuming initial position $x_0 > 0$ and initial velocity $v_0 = 0$, the first impact can be calculated using $x(t) = -\frac{g}{2}t^2 + bt + c$ the analytic solution :

$$t_{A,1} = \sqrt{\frac{2x_0}{g}}$$

Using the analytic solution, starting flight at $t_{A,1}$ with

$$x(t_{A,1}) = 0, \quad v(t_{A,1}) = -\mu v(t_{A,1})_{prev} = \mu \cdot g \cdot t_{A,1}$$

gives the next impact time

$$t_{A,2} = t_{A,1}(2\mu + 1),$$

$$\Delta t_{A,2} = t_{A,2} - t_{A,1} = 2\mu \cdot t_{A,1},$$

implying the general recursion for the impact instants $t_{A,m}$ and for the time distances $\Delta t_{A,m}$ between the bounces:

$$\Delta t_{A-m} = t_{A-m} - t_{A-m-1} = \mu(t_{A-m-1} - t_{A-m-2})$$

$$t_{A-m} = t_{A-m-1}(\mu + 1) - \mu t_{A-m-2}$$

This recursion allows calculating the time instant $t_{A,m}$ of the m -th bounce by means of the geometric series

$$t_{A,m} = \sqrt{\frac{2x_0}{g}} \cdot \left(-1 + 2 \sum_{i=0}^{m-1} \mu^i \right)$$

As $|\mu| < 1$, the above series converges and gives the limit for the series of bouncing time instants t_∞ :

$$t_{A,\infty} = \sqrt{\frac{2x_0}{g}} \cdot \frac{1 + \mu}{1 - \mu}$$

- being a finite number! These analytical considerations proof, that in finite time infinite many bounces take place, or that there is a final time, where the last impact takes place. Furthermore, numerical solutions using an event algorithm can be validated with the analytical results.

Dynamic Contact Phase.

In case of a significant contact phase, a more realistic model is necessary, which takes into account the elasticity in the contact region. The deformation can be modelled in first approximation by a spring-damper-element in parallel to the flying phase

Free falling/flying phase. Ignoring for simplicity the air resistance, the equations can be used for position and velocity, and the deformation is a damping of first order – during flight not coupled with position and velocity, but active:

$$\dot{x} = v, \quad \dot{v} = -g, \quad \dot{w} = -\frac{k}{d}w$$

The linear state space is given by

$$\frac{d}{dt} \vec{x}(t) = A_F \vec{x}(t) + B_F u(t) \quad \text{where} \quad \frac{d}{dt} \vec{x}(t) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{k}{d} \end{pmatrix} \vec{x}(t) - g \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

The variable v is still the velocity, x is the distance from the ground to the virtual bottom point of the not deformed ‘ball’ (may become negative), and w represents the deformation, so that the actual bottom position of the deflected ‘ball’, the distance $y(t)$ from the bottom of the deflected ‘ball’ to ground is given by

$$y(t) = x(t) + w(t)$$

whereby the deflection/deformation is measured in positive direction.

Contact phase. In the contact phase, the normalized contact force f_c determines the dynamic behaviour:

$$f_c = -kx - dv$$

Due to the unilateral contact between the ball and the ground, the contact force f_c cannot be negative, so that f_c is in fact given by $\hat{f}_c = \max(-kx - dv, 0)$.

In case of contact, the dynamic equation (38) for velocity v gets the contact force f_c added as counterforce to gravity, and \dot{w} , the change of deflection must equilibrate to velocity v . The dynamic equations in the continuous contact phase are consequently:

$$\dot{x} = v, \quad \dot{v} = -g + f_c = -g - kx - dv, \quad \dot{w} = -v$$

The linear state space is given by

$$\frac{d}{dt} \vec{x}(t) = A_C \vec{x}(t) + B_C u(t) \quad \text{where} \quad \frac{d}{dt} \vec{x}(t) = \begin{pmatrix} 0 & 1 & 0 \\ -k & -d & 0 \\ 0 & -1 & 0 \end{pmatrix} \vec{x}(t) - g \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

3 Case Study 2: Switching RLC Circuit

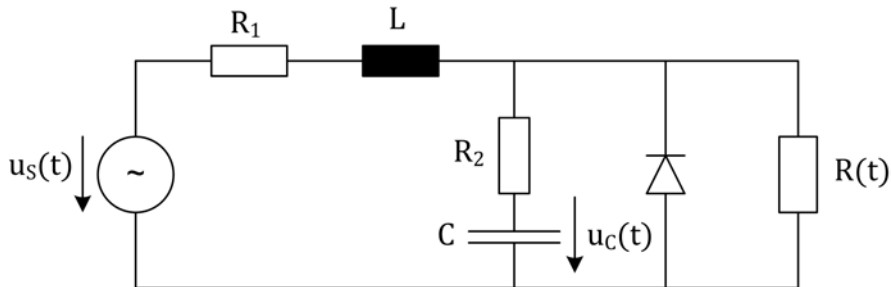


Figure 2: RLC Circuit for Case Study 2

Basic RLC Modelling – RLC-B model. Kirchhoff's laws and node equation allow setup of state equations for the basic RLC model (RLC-B model), which are also used by simulation systems with physical modelling features. Manual derivation of the system equations circuit will usually choose a state vector $\vec{x}(t)$ consisting of current $i(t)$ and capacitor voltage $u_c(t)$ and result in a linear state space:

$$\begin{aligned} x(t) &= (i(t), u_c(t))^T, \vec{u}(t) = (u_s(t)) \\ \frac{d}{dt} \vec{x}(t) &= A\vec{x}(t) + B\vec{u}(t) \\ \vec{x}(t_0) &= (i_0, u_{c,0})^T \end{aligned}$$

Automatic model generation e.g. by means of physical modelling will probably make a similar state space choice, but in general the principle choice is a general nonlinear system:

$$\begin{aligned} x(t) &= (i(t), u_c(t))^T, \vec{u}(t) = (u_s(t)) \\ \frac{d}{dt} \vec{x}(t) &= \vec{f}(\vec{x}(t), \vec{u}(t)) \\ \vec{x}(t_0) &= (i_0, u_{c,0})^T \end{aligned}$$

RLC model with switching Resistor – RLC-R. Switching in a circuit may be designed by a resistor, which switches from low to high resistance values, and vice versa, in parallel to the circuit.

Kirchhoff's law and node equations give a model description, extended by resolution of the additional parallel resistor, whereby time-dependent resistance values $R(t)$ can be seen as time-dependent input:

$$\vec{x}(t) = (i(t), u_C(t))^T, \vec{u}(t) = (u_S(t), R(t))^T$$

$$\frac{d}{dt} \vec{x}(t) = \vec{f}^R(x(t), u_C(t))^T, \vec{x}(t_0) = (i_0, u_{C,0})^T$$

The switching of values is not a real jump in the resistance values, it is imagined as fast transient from on level to off level, and back. Figure 3 shows the typical characteristic time-dependent curve for a time-dependent resistor switching, whereby the transient phases $\Delta t_{on} = \Delta t_{ff}$

$$\Delta t_{on} = |t_{on,e} - t_{on,s}| \quad \Delta t_{off} = |t_{off,e} - t_{off,s}|$$

are usually very short.

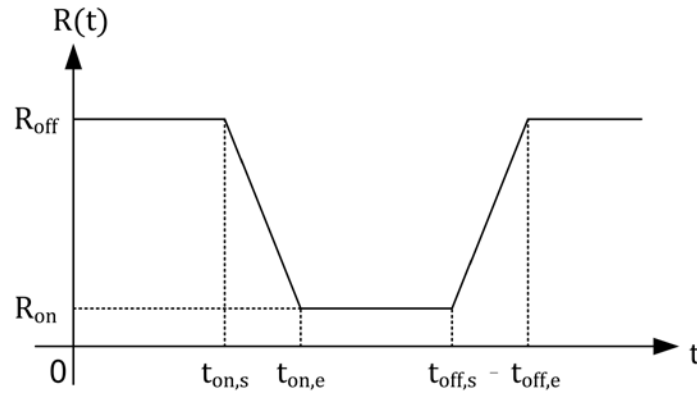


Figure 3: Characteristic of the time-dependent resistor

For proper modelling the question arises, how accurate the time-dependent resistance must be traced. For accurate and comparable results it is necessary to synchronize $t_{on,s}, t_{on,e}, t_{off,s}, t_{off,e}$, the switching time instants of the resistor, with the time instants t_k of the ODE solver. Especially the transient phases must be traced sufficiently accurate; the stepsize of the solver must be less than the solver's stepsize, i.e.

$$|t_k - t_{k-1}| < \Delta t_{on}, \quad |t_{i+1} - t_i| < \Delta t_{off}$$

For this synchronisation is necessary to model state events of type SE-I - input change event.

RLC model with witching Diode – RLC-D.

Simple diode models mimicry the dynamic behaviour as an ideal switch for the current $i(t)$ depending on diode voltage u_D limited as given in Figure 3, which switches from low to high resistance values, and vice versa, in parallel to the circuit.

$$i(u_D) = \begin{cases} 0, & u_D \leq u_{D,0} \\ i(t), & u_{D,0} < u_D, i(t) \leq I_S \\ I_S, & u_{D,0} < u_D, i(t) > I_S \end{cases}$$

This mathematical formulation describes the modes of the diode:

- Diode Backward Mode: for *input voltage less forward voltage* ($u_D < u_{D,0}$), the current through the diode is zero.
- Diode Forward Mode: for input voltage beyond forward voltage ($u_D > u_{D,0}$), the diode is assumed to be an ideal conductor and feeds the current $i(t)$ forward, but limited up to a maximal current $i(t) \leq I_S$.

Figure 4 shows summarizes the ideal switching behaviour of a diode, at left with ‘very’ ideal forward voltage $u_{D,0} = 0$, at right with more realistic forward voltage $u_{D,0} > 0$.

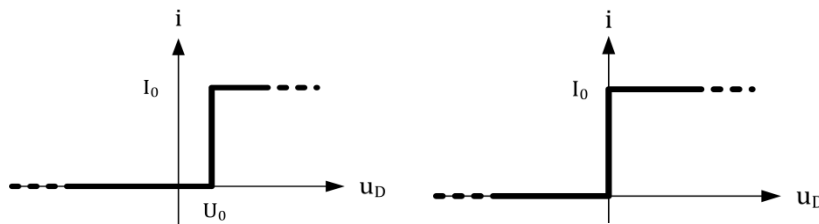


Figure 4: Characteristics of ideal diode

A diode is a nonlinear element, and indeed the switching dynamics evolve nonlinear dynamics. One nonlinear model is known as Shockley diode model.

The mathematical description is given by an exponential-like function for the switching transient behaviour:

$$i(u_D) = \begin{cases} 0, & u_D < 0 \\ I_S \left(e^{\frac{u_D}{U_T}} - 1 \right) & u_D \geq 0 \\ I_S & u_{D,0} < u_D, i(t) > I_S \end{cases}$$

with the saturation current $I_S = 10^{-12}$ A and the temperature voltage $U_T = 25$ mV, assuming ambient temperature.

Figure 5 shows the continuous characteristic curve for the Shockley diode model due to the formula above.

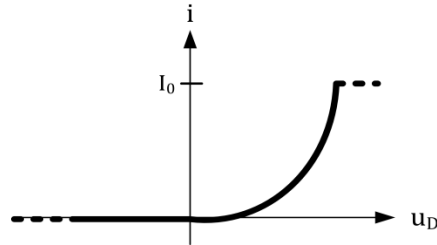


Figure 5: Diode model with Shockley equation

This nonlinear formula adds an algebraic equation to the explicit state space, if the node equation is resolved in case of forward mode of the diode:

$$I_S R_2 \left(e^{\frac{u_D}{U_T}} - 1 \right) + u_C + u_D = 0$$

Following the previous notation $u_D(t)$ is an algebraic state $z(t) = u_D(t)$ and is the corresponding algebraic equation, so that in feedforward mode the system becomes a real DAE system – instead of the ODE :

$$\begin{aligned} \frac{d}{dt} \vec{x}(t) &= \vec{f}^{DS,F}(\vec{x}(t), z(t), \vec{u}(t)) = \vec{f}^{DS,F}(i(t), u_C(t); u_D(t); u_S(t)) \\ g^{DS,F}(\vec{x}(t), z(t), \vec{u}(t)) &= g^{DS,F}(u_C(t); u_D(t)) = 0 \end{aligned}$$

or in state coordinates in more detail

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} i \\ u_C \end{pmatrix} &= \begin{pmatrix} f_1^{DS,F}(i, u_C; u_D; u_S) \\ f_2^{DS,F}(i, u_C; u_D; u_S) \end{pmatrix} \\ g^{DS,F}(u_C; u_D) &= 0 \end{aligned}$$

4 Case Study 3: Rotating Pendulum with free flight phase

This example describes a classical idealized pendulum on a rope with damping. The body, which is considered a point mass, is connected to a fixed point in a room via a rope of given length. The rope is assumed to be non-elastic and without mass. As a simplification, it is presumed that the body can move freely only in the plane, i.e. the area of a circle with a radius equal to the length of the rope.

Using these assumptions, the behavior of the pendulum can be described with two states:

- If the rope is tight, the body is moving along a circular path (state "bound"). Since the radius is fixed, the movement in this state can be described by one variable, i.e. the angle between the rope and the upper perpendicular.
- If the rope is loose, the body is free falling (state "flying") until the rope is tight again (back in state "bound"). This movement is defined by two state variables, i.e. the Cartesian x – and y – coordinates of the body.

The different states are illustrated in Figure 6, where you can also see that the two states have different numbers of degrees of freedom, which is the reason why the state space dimension changes as the pendulum alters its state.

Note that the angle φ is measured from the upper perpendicular in the range $[0, 2\pi]$ in order to have a continuous parameter when the pendulum is oscillating around the lower rest position.

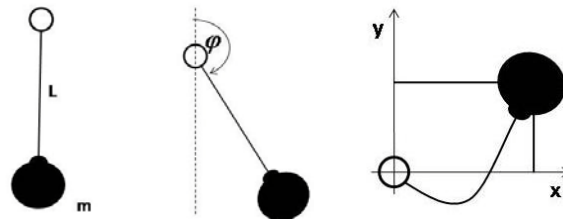


Figure 6: Left: Pendulum with mass m and length L ; middle: Classical oscillating pendulum (state "bound") with angle φ as degree of freedom; right: Free falling pendulum mass (state "flying") and Cartesian coordinates as degrees of freedom.

How the various forces are affecting the mass during the circular movement can be seen in Figure 7.

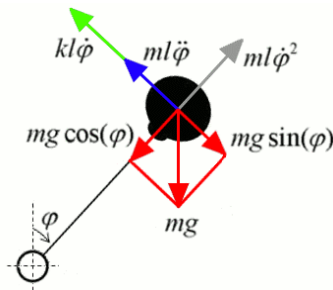


Figure 7: Forces affecting the mass in the "bound" state

The state "bound" can be described by using the angular momentum balance, which, with regard to Figure 7, leads to

$$\ddot{\varphi} + \frac{k}{m} \dot{\varphi} - \frac{g}{l} \sin \varphi = 0,$$

with the damping factor k , the mass m , the length of the rope l and the earth acceleration g . Furthermore, the force on the rope

$$F = +gm \cos \varphi + ml\dot{\varphi}^2$$

provides a criteria for the case when a state change from "bound" to "flying" occurs. If this force becomes lower than zero, the gravitational force outweighs the centrifugal force and the pendulum is entering the "flying" state.

Being in the "flying" state, the body has one more degree of freedom compared to the state "bound". The motion of the mass in this state is described by conservation of momentum in x – and y – direction:

$$m\ddot{x} = -k\dot{x}, \quad m\ddot{y} = -mg - k\dot{y}$$

Using an additional geometric equation,

$$r^2 = x^2 + y^2$$

it is tested if the rope is loose or completely tight. If r fulfills $r \geq 1$, the rope has become tight again, forcing the body back on the circular path and the pendulum switches to the "bound" state again.

Summing up, Figure 8 shows an overview of the two different models with the criteria for all state changes.

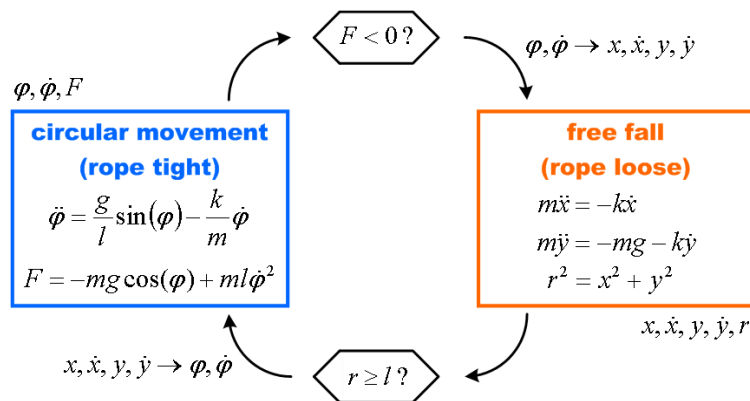


Figure 8: Summary of the two state models with criteria for the state changes representing the rotating pendulum with free flight phase

This example shows that different states of such a hybrid model may have to be described by different physical equations and that the number of necessary equations can differ among the states.

The occurring state event is therefore of type four (SE-D): the dimension of the state space changes discontinuously. These events are essential state events. They have to be located, transformed into a time event and modeled in discrete model parts.

To figure out the event functions h , Figure 7 can be used easily. It has to be taken into account that, additionally to the classical state event handling, a coordinate transformation has to be performed at each event. In case of changing from "bound" to "flying", the polar coordinates have to be transformed into Cartesian coordinates, and vice versa in the other case.

Basic simulation. Calculate and visualize a basic simulation run with the following parameters $m = 1,5kg$ $k = 0,9 \frac{kg}{s}$ $l = 1m$ $g = 9,81 \frac{m}{s^2}$ and the initial conditions

$$\varphi_0 = \varphi(0) = \frac{\pi}{4} \quad \dot{\varphi}_0 = \dot{\varphi}(0) = 15 \frac{1}{s}$$

Simulate beginning at $t = 0$ until the maximal oscillation does not exceed $\frac{\pi}{10}$ any longer. Identify the corresponding time.

Plot the coordinate $x(t)$ over the time, the trajectory $(x(t), y(t))$ and find the passages where the pendulum is in the "bound" state and the "flying" state, respectively.

Inconstant damping. Modify the model from the first task in a way such that a region with different damping factor is implemented. This region can be explained for example by the body diving in a medium with different density.

Assume that the different damping factor is affecting the pendulum only in the "bound" state. The "flying" state remains unchanged. An illustration of this situation is shown in Figure 9.

Using this assumption, a new event is created, i.e. an event of type one (SE-P): a parameter changes discontinuously. In the defined case, this event cannot be implemented without triggering the time point, when the angle φ reaches the region with different damping.

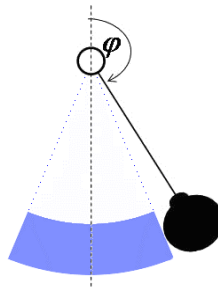


Figure 9: Region with different damping factor in the "bound" state

Test the modified model by calculating and visualizing a simulation run with a damping factor of $k_2 = 12,9 \frac{kg}{s}$ in the region $\varphi \in [\frac{6\pi}{7}, \frac{8\pi}{7}]$. For the other parameters you can use the same values as above. The event functions h detecting the state events for changing the damping factor are given as follows:

$$h_1(\varphi) = \varphi - \frac{6\pi}{7}, \quad h_2(\varphi) = \varphi - \frac{8\pi}{7}$$

Plot again the coordinate $x(t)$ over time and the trajectory $(x(t), y(t))$ during the entire simulation period. Identify the time periods of the "bound" state and the "flying" state and compare these results to the corresponding findings from the basic simulation.

5 References

- [1] *Breitenecker F., Zauner G., Popper N., Judex F. and Troch I.:* Development and New Concept for External and Internal State Events. SNE Simulation News Europe 17/2, 2007.
- [2] *Ecker H.:* The bouncing ball problem – modeling and simulation aspects. SNE Simulation News Europe, 34:9 – 14, 2002.
- [3] *Fritzson P.:* Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. John Wiley and Sons, 2004.
- [4] *Sokal N.O. and Sokal A.D.:* Class E - a newclass of high-efficiency tuned single-ended switching power amplifiers. IEEE Journal of Solid-State Circuits, SC-10:168–176, 1975.
- [5] *Viertel N. and Breitenecker F.:* Comparison 3 - analysis of a generalized class-e amplifier. SNE Simulation News Europe, 27:40, 1999.