V International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2013
S. Idelsohn, M. Papadrakakis and B. Schrefler (Eds)

# ON THE DIFFICULITIES OF REAL-TIME CO-SIMULATION

**G. STETTINGER*, M. BENEDIKT†, N. THEK† AND J. ZEHETNER†**

*Alpen-Adria Universität Klagenfurt
Institute of Smart System Technologies
Universitätsstrasse 65-67, 9020 Klagenfurt, Austria
e-mail: georg.stettinger@aau.at
web page: http://www.uni-klu.ac.at/tewi/ict/sst/me/index.html

†Virtual Vehicle Research Center
Area E - Electrics/Electronics and Software
Inffeldgasse 21/A/2, 8010 Graz, Austria
e-mail: martin.benedikt@v2c2.at, norbert.thek@v2c2.at, josef.zehetner@v2c2.at
web page: http://www.v2c2.at

**Key words:** Co-Simulation, Real-Time, Extrapolation, Coupling Types, Dead-Time

**Abstract.** In a co-simulation, subsystems are coupled via their in- and outputs to simulate the overall system behaviour. The subsystems are modelled in their domain specific simulation tools. The task changes if one coupled subsystem represents a real-time system. A real-time system which has to guarantees hard-real-time conditions influences the co-simulation concept: now the co-simulation also has to fulfill hard-real-time conditions. This type of co-simulation is called *real-time co-simulation.* The most important difference to a non-real-time co-simulation is the time correct overall simulation speed with respect to the involved real-time systems. To achieve this, all subsystems in form of non-real-time systems have to be synchronised to the involved real-time systems. The focus of this work lies on the problems that occur in a real-time co-simulation environment compared to a classical one. A concept to handle the additional problems is outlined and tested on an example real-time co-simulation.

## 1 INTRODUCTION

The co-simulation concept is often used in a modern vehicle development process in the field of the automotive industry [3, 5, 6]. Different subsystems are modelled in their domain specific simulation tools. To get an overall simulation result all subsystems have to be combined via their specific in- and outputs. The correct coupling of these signals is the main task of a so called *co-simulation platform.* The co-simulation concept has the big

advantage that all subsystems can use their domain specific solvers and no adaptations in the existing simulations are necessary [1]. Often the classical co-simulation concept is not enough to solve all occuring problems in the vehicle development process. One extension is the integration of hard-real-time systems in the co-simulation. This is the case when at least one modelled subsystem gets replaced by real hardware, e.g. HiL (Hardware in the Loop) systems or test benches. In this case the co-simulation is called *real-time co-simulation* and now the typical hard real-time conditions must hold during the simulation [3, 8]. The main difference compared to a classical co-simulation is the correct coupling with respect to the *wall-clock-time*. The involved real-time systems which fulfill the hard-real-time conditions run with the so called *wall-clock-time*.
The paper is organised as follows: Chapter 2 discusses the difficulties of a classical co-simulation. Chapter 3 describes the additional problems occuring in a real-time co-simulation. Chapter 4 shortly illustrates the structure of a real-time co-simulation. In chapter 5 an example real-time co-simulation is presented. The difficulties are shown and possible approaches to solve these problems are outlined.

The presented work is part of the ongoing research project ACoRTA (Advanced Co-Simulation Methods for Real-Time Applications[1]) at the Virtual Vehicle Competence Center in Graz, Austria. All techniques to handle the real-time co-simulation problem are implemented in the ICOS Real-Time Framework (ICOS RT).

## 2 DIFFICULITIES OF A CO-SIMULATION

Typically co-simulation problems are handled with the help of so called *co-simulation platforms*, which offer specific interfacing capabilities of the simulation tools and perform the data exchange between involved subsystems. The main requirements of such *classical* co-simulation problems are [7, 9, 10]:

- Proper choice of the coupling time instants
- Selection of an extrapolation method
- Definition of the subsystem scheduling

### 2.1 Choice of the coupling time instants (macro-time-step)

To distinguish between the step sizes of domain specific numerical solvers of the involved subsystems and the step sizes between two coupling instants, the micro- ($\delta T$) and macro-time-steps ($\Delta T$) are introduced (see Fig. 1). Every domain specific simulation tool typically has it's own micro-time-step which doesn't get influenced during the co-simulation. For a co-simulation only the macro-time-step has to be defined by the user. The involved subsystems get synchronised at every macro-time-step (which may be fixed or adaptive) [2, 4]. The coupling data exchange is done with the help of a co-simulation

[1] http://acorta.info

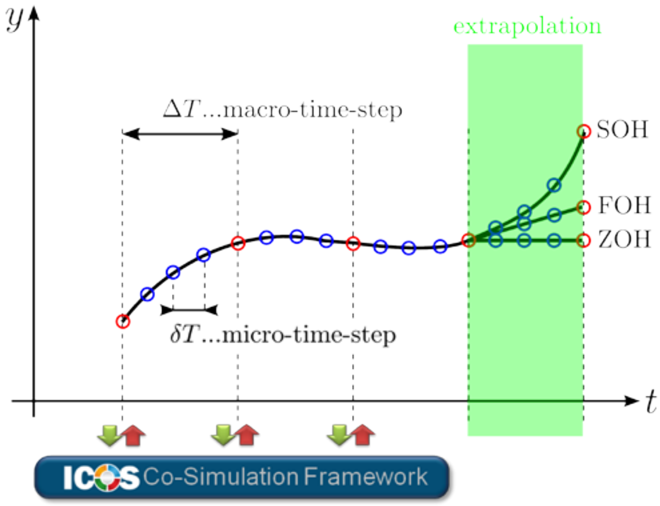


**Figure 1**: Micro- and macro-time-steps, polynomial extrapoltion techniques [9]

platform (e.g. the ICOS[2], Independent Co-Simulation Framework). The choice of the macro-time-step is very important for the quality of the overall simulation result because often a trade-off between the overall simulation time and the desired accuracy of the simulation results. In conclusion, smaller macro-time-steps lead to more accurate simulation results but with the drawback of longer simulation times [7, 10].

### 2.2 Signal-based extrapolation techniques

In the case of two interdependent subsystems, which are connected via their in- and outputs, at least one input has to be extrapolated to solve the coupled system [7]. In a classical co-simulation typically polynomial extrapolation techniques of low order are used to solve this problem [4, 5]. Fig. 1 shows the most commonly used polynomial extrapolation techniques such as zero-order extrapolation (zero-order hold, ZOH), first-order extrapolation (first-order hold, FOH) and second-order extrapolation (second-order hold, SOH). Due to the fact that every extrapolation technique is a prediction of the future coupling signal an error gets introduced [7]. This coupling error influences the overall simulation result and thus, this error must be as small as possible. Another important problem is the introduction of dead-times due to the used polynomial extrapolation. This dead-times can lead to instability in closed loop systems. The introduced dead-times mainly depend on the chosen macro-time-step (see Fig. 2) [7, 9].

### 2.3 Simulation tool scheduling

In principle, there are two different ways of subsystem scheduling: parallel and sequential. In the parallel case all inputs of the coupled subsystems has to be extrapolated

[2]http://www.v2c2.at/icos

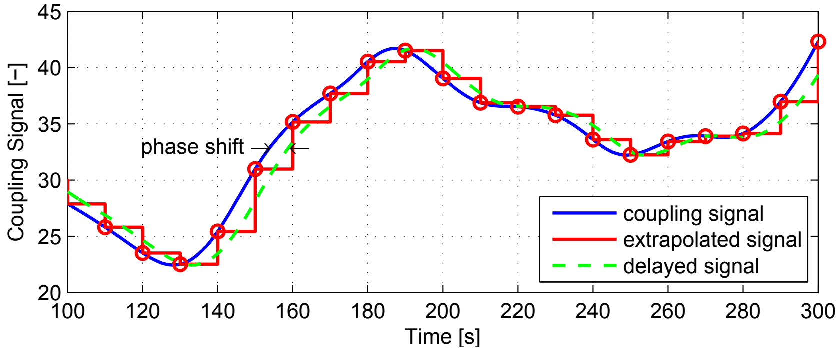


**Figure 2**: Introduced dead-time due to ZOH-extrapolation [9]

which lead to shorter simulation times, with the drawback of more extrapolation errors. Sequential scheduling requires less extrapolation effort with the the disadvantageous fact of longer simulation times. Additionally the execution sequence influences the entire system behavior [6, 7].

## 3 COMMENTS ON REAL-TIME CO-SIMULATION

For real-time co-simulation basically the same problems as for classical non-real-time co-simulation occur. One important aspect for a real-time co-simulation are hard real-time conditions which have to be fullfilled during a real-time co-simulation which is responsible for the following problems [9]:

1. Requirements for coupleable subsystems
2. Synchronisation of the involved subsystems
3. Handling of dead-times
4. Extrapolation of noisy coupling signals

These additional difficulties are discussed in detail in the following sections.

### 3.1 Requirements for coupleable subsystems

To fulfill the required hard-real-time conditions a time correct exchange of coupling data with respect to the *wall-clock-time* is necessary. This is only possible if the involved non-real-time systems, which do not guarantee hard-real-time conditions, have a simulation time which is faster than the *wall-clock-time*. Otherwise no synchronisation between the involved real-time and non-real-time systems is possible [9].

### 3.2 Synchronisation of the involved subsystems

The main task of the real-time co-simulation problem is the time correct exchange of coupling data with respect to the *wall-clock-time*. The synchronisation is typically realised via a slow down mechanism for the non-real-time systems. The simulation speed of these systems is faster than the *wall-clock* time and so these systems have to slow down until both system types run with the *wall-clock* time to perform a time correct synchronisation. So the involved non-real-time systems get paused to realise this slow down effect [9].

### 3.3 Handling of dead-times

To guarantee hard-real-time conditions a time correct coupling is mandatory as discussed in 3.2. So the communication delays due to the communication medium between coupled systems play an important role. These dead-times influence the stability of the closed loop system and can violate the hard-real-time conditions when the synchronisation fails. The coupling mechanism between the coupled systems split the communciation dead-times into sending and receiving dead-times (see Fig. 3). To guarantee the required hard-real-time conditions an approximate compensation of the sending and receiving dead-times is required [9].

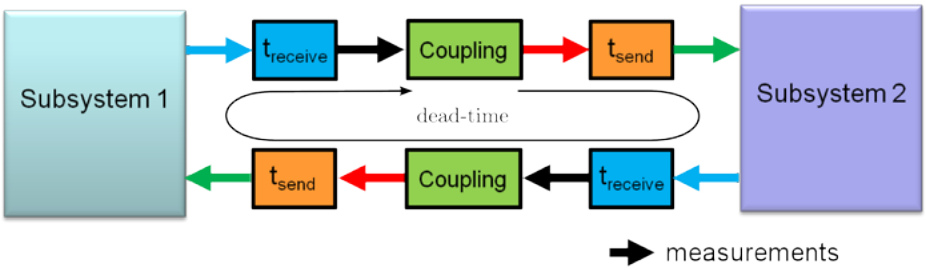


**Figure 3**: Sending and receiving dead-time due to the data transfer [9]

### 3.4 Extrapolation of noisy coupling signals

As discussed before, extrapolation is necessary to solve the co-simulation problem [9]. If real-time systems enter the co-simulation several coupling signals are typically measured with sensors and are often corrupted by noise. Signal-based extrapolation techniques, which are typically used for extrapolation, amplify the noisy measurements and so they cannot provide reliable coupling signals. So extrapolation methods which are robust against noisy measurements are required. To satisfy this requirement e.g. model-based extrapolation methods can be used [9].

## 4 STRUCTURE OF A REAL-TIME CO-SIMULATION

The focus of this work lies on the coupling of real-time and non-real-time systems. The synchronisation of the involved real-time and non-real-time sytems is done via ICOS RT (see Fig. 4). ICOS RT is responsible for the time correct communication between the involved subsystems. So the slow down mechanism is the most important task of ICOS

RT to perform a correct synchronisation between the real-time and non-real-time systems. Two additional coupling types which are not part of this work are handled by ICOS RT:

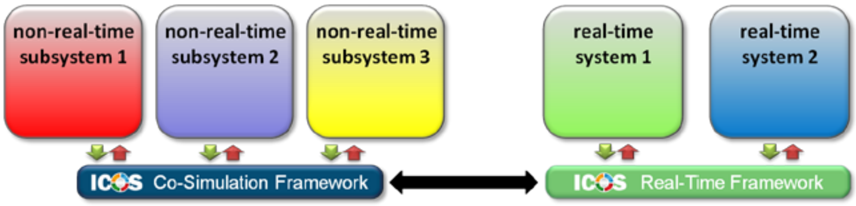


**Figure 4**: Coupling of real-time and non real-time systems [9]

The coupling of real-time systems and the coupling of tasks on one real-time system [9].

## 5 EXAMPLE: MILD HYBRID

The following example demonstrates the synchronisation mechanism of a real-time co-simulation.

### 5.1 Model description

The used example represents a hybrid vehicle which consists of seven subsystems (see Fig. 5). These involved subsystems are modelled in different simulations tools (*MATLAB/SIMULINK*[3], *KULI*[4] and *AVL CRUISE*[5]):

- **Vehicle (*AVL CRUISE*):**
  The vehicle model includes the power train of a parallel hybrid vehicle. This subsystem interacts with 17 inputs and 8 outputs with the other involved subsystems. The micro- and macro-time-steps are set to $\delta T = 0.001s$ and $\Delta T = 0.05s$, respectively.

- **Thermal Network (*KULI*):**
  The thermal network model includes the cooling circuit of the vehicle.
  Inputs: 9; Outputs: 12; $\delta T = 1s$; $\Delta T = 1s$.

- **Cockpit (*MATLAB/SIMULINK*):**
  The cockpit model describes the driver of the hybrid car. So this system defines the velocity profile and the gear selection.
  Inputs: 1; Outputs: 7; $\delta T = 0.01s$; $\Delta T = 0.05s$.

- **Hybrid Module (*MATLAB/SIMULINK*):**
  The hyrid module includes the parallel hybrid structure of the vehicle. So the

[3]http://www.mathworks.com
[4]http://www.kuli.at
[5]http://www.avl.com/cruise1

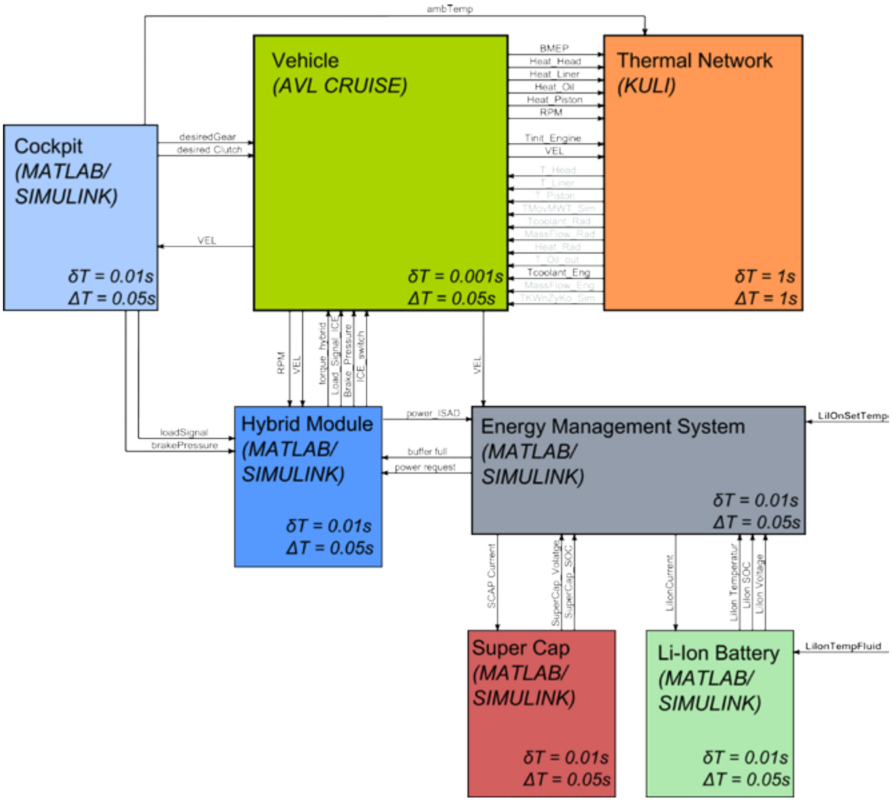


**Figure 5**: Subsystems of the hybrid vehicle

recuperation of the brake energy and the partition of the desired driving torque to the combustion and electric engine are the main tasks of this subsystem.
Inputs: 7; Outputs: 5; $\delta T = 0.01s$; $\Delta T = 0.05s$.

- **Energy Management System (*MATLAB/SIMULINK*):**
  This subsystem includes the energy management of the hybrid vehicle. Especially the energy exchange between the super cap, the electric motor and the Li-Ion battery is represented herein.
  Inputs: 8; Outputs: 6; $\delta T = 0.01s$; $\Delta T = 0.05s$.

- **Super Cap (*MATLAB/SIMULINK*):**
  The super cap subsystem modells the behaviour of the super cap.
  Inputs: 1; Outputs: 4; $\delta T = 0.01s$; $\Delta T = 0.05s$.

- **Li-Ion Battery (*MATLAB/SIMULINK*):**
  The Li-Ion battery subsystem modells the behaviour of the Li-Ion battery.
  Inputs: 2; Outputs: 5; $\delta T = 0.01s$; $\Delta T = 0.05s$.

### 5.2 Real-time co-simulation configuration

For this example the four involved *MATLAB/SIMULINK* modells are compiled and run on an ETAS real-time system (see Fig. 6). The non-real-time part includes the *KULI*, one *MATLAB/SIMULINK* and the *AVL CRUISE* modell. The subsystems of the non-real-time part are coupled via the ICOS Co-Simulation framework to achieve a correct interaction of these subsystems. The essential part of the real-time co-simulation is the synchronisation of the real-time and non-real-time part. This synchronisation mechanism slows down the non-real-time part such that a time correct interaction between the real-time- and non-real-time part is performed. The macro-time-step, where coupling data between the real-time- and non-real-time system gets exchanged, is set to $\Delta T = 200ms$. The overall real-time co-simulation, consisting of the real-time and non-real-time part, use a parallel scheduling while the subsystems in the non-real-time part are sequential scheduled.

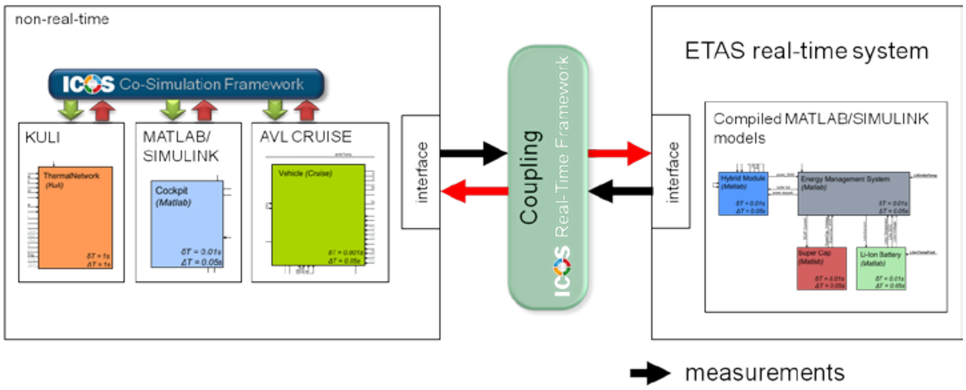


**Figure 6**: Configuration of the real-time co-simulation

### 5.3 Simulation results

Fig. 7 shows the simulation results of the real-time co-simulation compared to a non-real-time realisation of the same system. In the case of the non-real-time co-simulation all subsystems are simulated using the co-simulation platform ICOS. The two simulation results are nearly the same over the simulation time. This means that a subsystem of the non-real-time co-simulation, which runs faster than the *wall-clock* time, can be replaced by a real-time system with the same functionality and the simulation result is still reliable. The small deviations between the simulation results stem from the communication medium between the real-time- and non-real-time part of the real-time co-simualtion. Especially the introduced dead-times influence the overall simulation result. Fig. 8 shows the synchronisation effect of the time correct coupling: All involved subsystems have a simulation speed which is close to the *wall-clock* time due to the slow down mechanism.

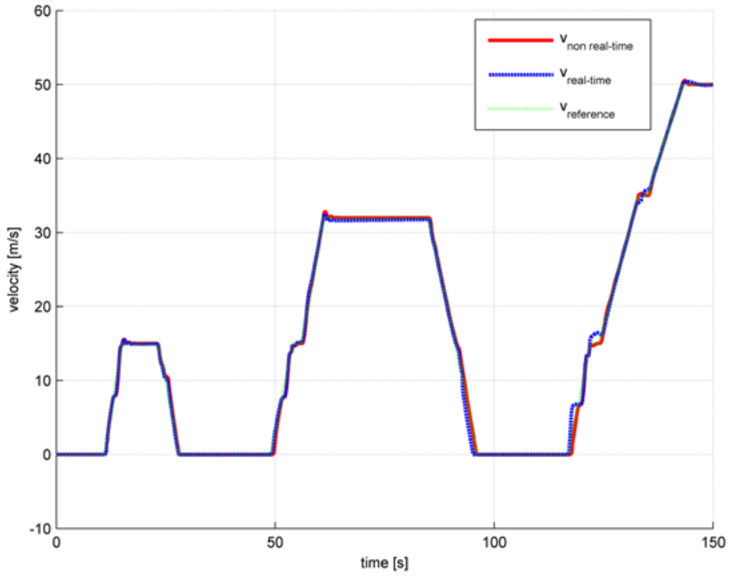


**Figure 7**: Real-time co-simulation result

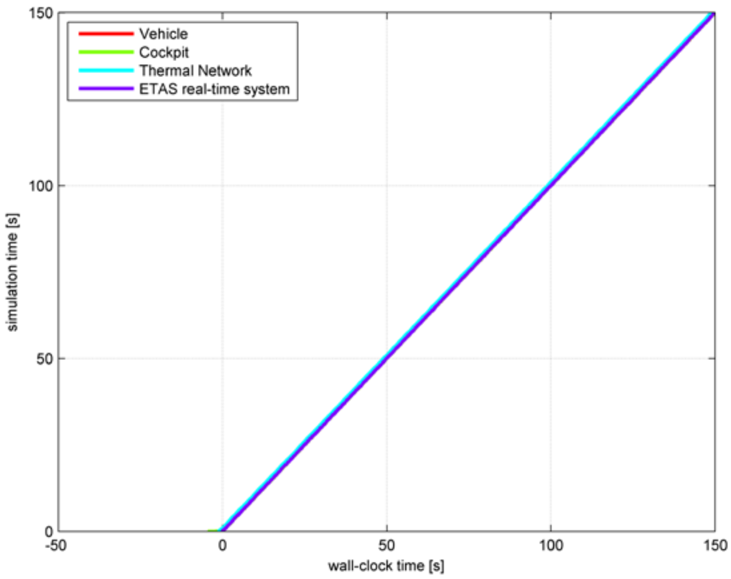


**Figure 8**: Synchronization of the involved online and offline systems

So a correct interaction between the real-time and non-real-time systems is possible. Fig. 9 shows the slow down effect of the real-time co-simulation compared to the non-real-time realisation. In the non-real-time case the whole simulation is done via ICOS where all involved subsystems have a simulation speed which is faster than the *wall-clock* time. With the help of ICOS RT the non-real-time systems slow down and a synchronisation with the real-time system is possible.

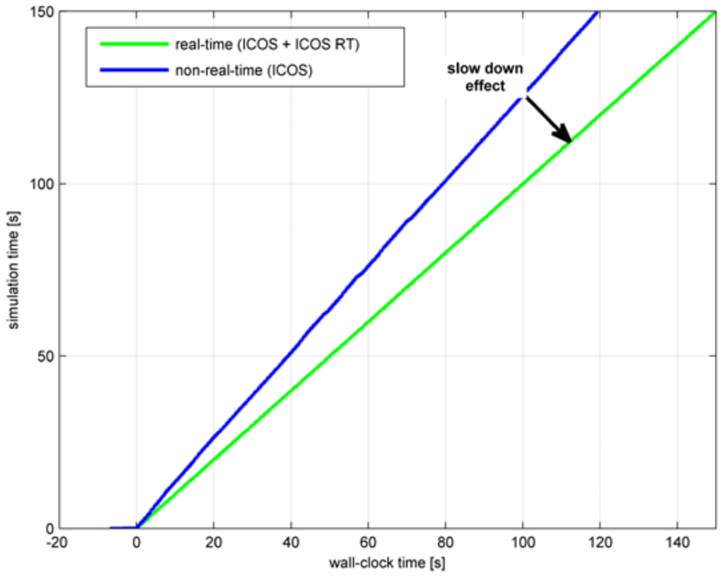


**Figure 9**: Slow down effect of the coupling element

## 6 CONCLUSIONS & OUTLOOK

The integration of hard real-time systems into co-simulation frameworks faces several challenges. The presented approach shows that a real-time co-simulation is handable if the non-real-time systems run faster than the *wall-clock* time. The required synchronisation with respect to the *wall-clock* time is realised via the discussed slow down mechanism. Future activities will concentrate on dead-time handling due to the involved communication medium and on extrapolation techniques which are robust against noisy coulpling signals. These difficulties are focused in the ongoing research project ACoRTA at ViF, together with Porsche, AVL and Uni Klagenfurt. By solving these problems co-simulation under hard-real-time conditions will be possible.

## REFERENCES

[1] Geimler, M.; Krüger, T.; Linsel, P.: "Co-Simulation, gekoppelte Simulation oder Simulatorkopplung? Ein Versuch der Begriffsvereinheitlichung", O+P Zeitschrift für Fluidtechnik (50), 2006, 572-576.

[2] Benedikt, M.; Stippel, H.; Watzenig, D.: "An adaptive Coupling Methodology for Fast Time-Domain Distributed Heterogeneous Co-Simulation", SAE International, 2009, 1-10.

[3] Dronka, S.: "Die Simulation gekoppelter Mehrkörper- und Hydraulik-Modelle mit Erweiterung für Echtzeitsimulation", PhD Thesis, Technische Universität Dresden, 2004.

[4] Arnold, M.: "Multi-Rate Time Integration for Large Scale Multibody System Models", IUTAM Bookseries, 2007, 1-10.

[5] Gonzales, F.; Naya, M.; Luaces, M.; Gonzales, M.: "On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics", Multibody System Dynamics, 2011, 461-483.

[6] Puntigam, W.: "Independent Co-Simulation Platform for transient coupling of individual submodels to establish an efficient energy management for the entire vehicle - taking thermal management as an example", PhD Thesis, Technische Universität Graz, 2007.

[7] Benedikt, M.; Zehetner, J.; Watzenig, D.; Bernasch, J.: "Modern coupling strategies - is co-simulation controllable?", NAFEMS Seminar: The Role of CAE in System Simulation, Wiesbaden, Germany, 2011.

[8] Hermann Kopetz. 1997. Real-Time Systems: Design Principles for Distributed Embedded Applications (1st ed.). Kluwer Academic Publishers, Norwell, MA, USA.

[9] Stettinger, G.; Benedikt, M.; Thek, N.; Zehetner, J.: "Extending Co-Simulation to the Real-Time domain", SAE International, 2013, 1-8.

[10] Benedikt, M.; Watzenig, D.; Zehetner, J.; Hofer, A.: "Macro-Step-Size Selection and Monitoring of the Coupling Error for Weak Coupled Subsystems in the Frequency Domain", V International Conference on Computational Methods for Coupled Problems in Science and Engineering, Ibiza, Spain, 2013.