CHALMERS

Virtual prototyping of vehicular electric steering assistance system using co-simulations

WEITAO CHEN

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2021

Thesis for the degree of doctor of philosophy
in
Machine and Vehicle Systems

Virtual prototyping of vehicular electric steering assistance system using co-simulations

WEITAO CHEN

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2021

Virtual prototyping of vehicular electric steering assistance system using co-simulations

WEITAO CHEN
ISBN 978-91-7905-524-0

Doctoral thesis at Chalmers University of Technology
Series number: 4991
ISSN 0346-718X

Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone: +46 (0)31–772 1000

Chalmers Reproservice
Göteborg, Sweden 2021

"Before I came here I was confused about this subject. Having listened to your lecture I am still confused, but on a higher level."

Enrico Fermi

Same to the author after his PhD

Virtual prototyping of vehicular electric steering assistance system using co-simulations

Weitao Chen
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Virtual prototyping is a practical necessity in vehicle system development. From desktop simulation to track testing, several simulation approaches, such as co-simulation and hardware-in-loop (HIL) simulation, are used. However, due to interfacing problems, the consistency of testing results may not be ensured. Correspondingly, inherent inaccuracies result from numerical coupling error and non-transparent HIL interface, which involves control tracking error, delay error, and attached hardware and noise effects. This work aims to resolve these problems and provide seamless virtual prototypes for vehicle and electric power-assisted steering (EPAS) system development.

The accuracy and stability of explicit parallel co-simulation and HIL simulation are investigated. The imperfect factors propagate in the simulation tools like perturbations, yield inaccuracy, and even instability according to system dynamics. Hence, reducing perturbations (coupling problem) and improving system robustness (architecture problem) are considered.

In the coupling problem, a delay compensation method relying on adaptive filters is developed for real-time simulation. A novel co-simulation coupling method on H_∞ synthesis is developed to improve accuracy for a wide frequency range and achieve low computational cost. In the architecture problem, a force(torque)-velocity coupling approach is employed. The application of a force (torque) variable to a component with considerable impedance, e.g., the steering rack (EPAS motor), yields a small loop gain as well as robust co-simulation and HIL simulation. On a given EPAS HIL system, an interface algorithm is developed for virtually shifting the impedance, thus enhancing system robustness.

The theoretical findings and formulated methods are tested on generic benchmarks and implemented on a vehicle-EPAS engineering case. In addition to the acceleration of simulation speed, accuracy and robustness are also improved. Consequently, consistent testing results and extended validated ranges of virtual prototypes are obtained.

Keywords: Vehicle and mechatronic system, explicit parallel co-simulation, coupling method, HIL simulation, interface algorithm.

Acknowledgments

I would like to express my sincere gratitude to my supervisors, Prof. Bengt Jacobson, associate Prof. Matthijs Klomp, and associate Prof. Fredrik Bruzelius. This project and thesis would not have been completed without your support and encouragement.

Secondly, I wish to express my appreciation to Dr. Shenhai Ran for the technical discussion and feedback; Tushar Chugh, my project partner who I learned from and always gave the best jokes; Mr. Edo Drenth and Prof. Stig Larsson, who assisted on co-simulation problems.

I am also thankful to all my colleagues and fellow PhD students in VEAS for contributing to a great working environment. Special thanks to Simone, Sonja and Britta for their excellent management and to my sisters and brothers in vehicle dynamics group.

I am extremely grateful to my colleagues in Volvo Cars, especially my managers, Åsa and Carl, who recruited me and gave support on administration and needed resources. Thanks to Alejandro and Nrip for the collaboration and sharing the knowledge and pain of solving software problems in the work.

I would like to thank the ITEAM project in the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie (grant agreement no.675999) as well as the SWOPPS project under Electronics, Software and Communication (EMK) (grant agreement no. 2017-05504) as part of the Swedish Strategic Vehicle Research and Innovation (FFI) program.

Last but not the least, I wish to thank my parents and my wife Jingqian. Without your support, I would have definitely given up in completing this work.

Weitao Chen
Göteborg, October 2021

Thesis

This thesis includes the following published papers of the author:

I. Chen, W., Ran, S. and Jacobson, B., *Design of Interface in Co-Simulation for Electric Power Assisted Steering System Development.* In: Proceedings of the 14th International Symposium on Advanced Vehicle Control (AVEC), Beijing, China, 2018.

II. Chen, W., Ran, S. and Jacobson, B., *Integration and Analysis of EPAS and Chassis System in FMI-based Co-Simulation.* In: Proceedings of the 13th International Modelica Conference, Regensburg, Germany, number 157, 2019, `http://dx.doi.org/10.3384/ecp19157717`.

III. Chen, W., Ran, S. and Jacobson, B., *Explicit parallel co-simulation approach: analysis and improved coupling method based on H-infinity synthesis*, journal Multibody System Dynamics, 52, pages 255–279, 2021, `https://doi.org/10.1007/s11044-021-09785-x`.

IV. Chen, W., Ran, S., Klomp, M. and Jacobson, B., *Real-time Co-Simulation Method Study for Vehicle Steering and Chassis System.* In: Proceedings of the 15th IFAC Symposium on Control in Transportation Systems CTS, Savona, Italy, pages 273-278, 2018, `https://doi.org/10.1016/j.ifacol.2018.07.045`.

V. Chen, W., Bruzelius, F., Klomp, M. and Jacobson, B., *A Method to Improve Stability and Transparency for Mechanical Hardware-in-the-Loop Simulation*, journal preprint.

The author was responsible for initiating the ideas, implementation, and paper writing. The co-authors contributed to the technical discussions and revision of manuscripts. The idea behind paper IV was initiated by Dr. Klomp.

The author's other related works (not appended) are as follows.

I. Chen, W., Chugh, T., Klomp, M., Ran, S. and Lidberg, M.,*Design and control of the steering torque feedback in a vehicle driving simulator.* Proceedings of the 25th International Symposium on Dynamics of Vehicles on

Roads and Tracks (IAVSD 2017), Rockhampton, Queensland, Australia, 2017.

II. Chugh, T., Chen, W., Klomp, M., Ran, S. and Lidberg, M.,*Design and control of model based steering feel reference in an electric power assisted steering system.* Proceedings of the 25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2017), Rockhampton, Queensland, Australia, 2017.

III. Chen, W., Klomp, M., Ran, S., Khan, U., Bianchi, A., Ran, S. and Jacobson, B.,*An Architecture of Hardware and Driver in the Loop Simulation for Electric Power Assisted Steering System.* Proceedings of the 26th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2019), Gothenburg, Sweden, 2019.

Acronyms

AD	Autonomous driving
EPAS	Electric power assisted steering system
CAN	Control area network
DAE	Differential-algebraic equation
DCP	Distributed co-simulation protocol
DMSD	Dual mass-spring-damper
DRTS	Digital real-time simulator
DUT	Device under test
ECU	Electric control unit
EKF	Extended Kalman filters
FMI	Functional moch-up interface
FMU	Functional moch-up unit
FOH	First-order hold
HIL	Hardware-in-the-loop
IA	Interface algorithm
LTI	Linear time-invariant
MBD	Multibody dynamics
MHIL	Mechanical hardware-in-the-loop
MSD	Mass-spring-damper
ODE	Ordinary differential equation
PHIL	Power hardware-in-the-loop
RK	Runge-Kutta
RLS	Recursive least square
RMSE	Root mean square error
RT	Real time
SHIL	Signal hardware-in-the-loop
DRTS	Digital Real-time simulator
SOH	Second-order hold
ZOH	Zero-order hold

Table of Contents

Chapter 1

Introduction

This thesis intends to resolve problems associated with the virtual prototyping of vehicle and electric power assisted steering (EPAS) system, i.e., a typical vehicle and mechatronic system interaction. A virtual prototype replaces a physical prototype with digital simulators or hardware-in-the-loop (HIL). However, the integration might not yield reliable and stable results in practise. This work aims to answer why such a system becomes unreliable and how to properly construct virtual prototypes for vehicle development and EPAS system as well as extend their validity ranges. To accomplish this, problems that emerge with the integration must be comprehensively considered, and how different impact factors distort the system dynamics must be answered. Furthermore, methods for reducing undesired factors, and enhancing the robustness of virtual prototypes must be sought. These are the motivations of this research work.

This work does not focus on modelling a vehicle-EPAS system or validating the model to a real system. Many published articles dedicated to the foregoing subjects are found in the literature [16, 86, 52, 75], and the formulated techniques have been successfully applied to the industry. Instead, this thesis focuses more on simulation tools and basic principles, especially on real-time (RT) applications, through co-simulation and HIL simulation.

1.1 Background and motivation

Electrification and automation

In 2019, Volvo Cars launched an ambitious plan aiming to reduce its carbon footprint per car by 40% before 2025 and become a climate-neutral company by 2040. Meanwhile, Ford and Volkswagen announced that they intend to become carbon-neutral by 2050, to align with the Paris Commitment [2, 3]. To achieve these goals, the main strategy is to electrify vehicular subsystems to improve energy efficiency. More specifically, the electrification of gasoline compression ignition

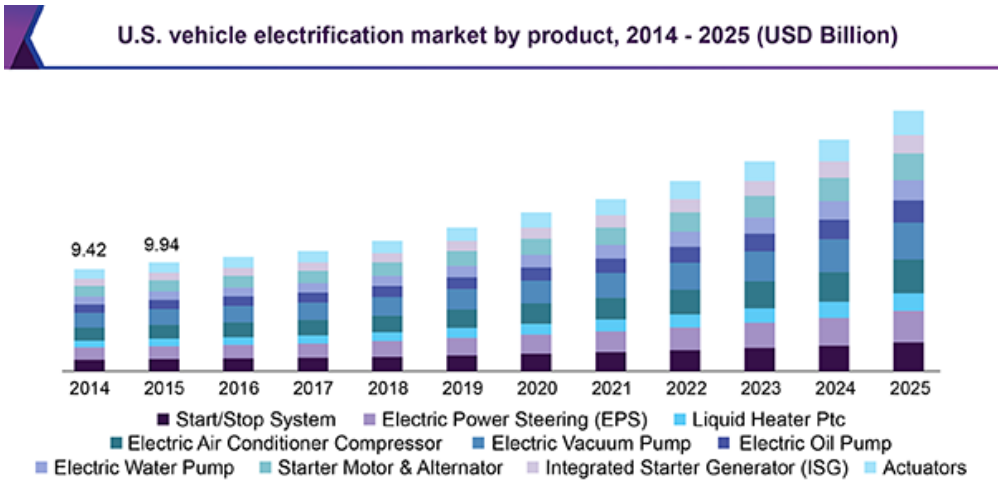


Figure 1.1: Emerging vehicle electrification in US market from the report [1]. Global market size is estimated at 58.92 billion USD in 2016.

engines can reduce fuel consumption by up to 44% [56]. An EPAS system can increase fuel efficiency by up to 6% and reduce CO2 emissions by up to 8 g/km compared with an alternative hydraulic unit [78]. In a conventional braking system, approximately 30% of the total energy is wasted through friction. However, 24% or more of the braking energy can be potentially recovered by braking using an electric machine [95]. Electrification is currently regarded as a major trend in the automotive industry (Fig.1.1) and even in succeeding decades.

The use of more mechatronic subsystems is anticipated to increase functionality to various levels, introducing more electric control units (ECUs) and software codes onboard than ever. A typical example of a subsystem-level function is the anti-lock braking system, which regulates the brake pressure to improve the deceleration capability using pressure sensors, control algorithms and electric valves that are fast reactive [116]. In addition, multiple driving modes (e.g., comfort or sportive) are realised with the active suspension system and EPAS system according to control parameters [76]. Autonomous driving is the highest-level function into which lower-level functions are incorporated; here, the EPAS system is essential. Work on the realisation of full automation is ongoing, involving more codes, improved onboard processing capability, and more sensors when necessary.

Effects on the development

As a vehicle evolves into a machine more akin to a robot or a computer, its development process also changes. To reduce the development time, control codes are expected to be developed concurrently with the hardware. Then, the entire system can be tested at the earliest. This leads to a new form of virtual prototyping. Although it is not novel in vehicle development, it was barely formulated

for software development.

Virtual prototyping refers to the use of simulation tools to replace the physical counterparts for product presentation and testing [119]. In the conceptual phase of a vehicle project, the control codes can be preliminarily tested on the vehicle system only through computer simulation. Then, HIL simulation is performed in later phases for development iterations. This simulation enables more detailed and realistic tests non-destructively (e.g., reliability test and fault-tolerance test) [57]. Furthermore, it can be extended with a human driver-in-the-loop to better understand the virtual prototype. In the virtual development phase, the subjective evaluation of product attributes is possible without a physical prototype.

Requirements

The effective virtual prototyping of a vehicle system requires reliable predictions derived from simulation tools. In this regard, the probable most important requirements on virtual prototypes are *Accuracy* and *consistency*.

> *Accuracy* means the simulation results approach reality to the extent necessary, that is, the tool is validated with respect to reality.

This requires simulation models to have adequate levels of abstraction to capture the characteristics of interest. In general, more accurate models include more system details. The known associated side effect is complexity in terms of parameter identification and computation. These subjects have long been investigated by researchers in each dedicated field. In the industry, the consequent domain-specific models and commercial software tools have been successfully employed. In vehicle development, some examples are Adams/Car and Simpack for chassis multibody simulation [87], MFswift and FTire for simulating the tyre-terrain interaction [44], GT-suite for simulating the powertrain [9], and Bouc–Wen models for the rubber bushing and damper [79, 68]. For the holistic simulation of the vehicle system, the virtual prototype typically consists of multiple software tools.

> *Consistency* means that results derived from different simulation tools are in agreement.

In other words, different simulation tools validate each other (not necessarily with respect to reality). A highly complicated simulation tool must not only introduce new features but also conform with basic predictions. For instance, the torque-speed relation from the HIL simulation of an electric motor must at least, under the steady-state condition, correlates with the prediction from a basic desktop simulation. Discrepancy only occurs due to changes in system model and design parameters. Accordingly, the development of hardware and software can be effectively iterated towards field tests. However, *consistency* is not always ensured as complexity increases; this is known as the interfacing problems of

multi-domain simulation tools [38, 91]. Moreover, virtual prototyping requires development efforts and time, which must be minimised.

1.2 Challenges

On a higher level, the problems in virtual prototyping are associated with knowledge and the application aspects. In terms of knowledge, a multidisciplinary gap exists. A virtual prototype of a mechatronic system involves disciplines, such as mechanical engineering, control engineering and computer science. Each discipline has its own framework and approaches. The disciplinary boundary sometimes constrains the awareness of the solution space for the virtual prototyping of vehicular systems, and a systematic framework on the subject is missing.

In the application aspect, a generic platform for integrating simulation tools is missing in industry, and due to commercial reasons, the prospect of its introduction to the industry is minimal. To integrate different software tools, the commonly used functional-mock-up interface (FMI) standard [25] is initiated by Dassault Systems, developed by the MODELISAR consortium. Thereafter, to integrate simulation tools with RT systems, the distributed co-simulation protocol (DCP) [63] is recently developed by the ACOSAR project led by the Virtual Vehicle Center. Although software obstacles are partially removed, the simulation tools are usually integrated in an ad-hoc manner. Because the applications are highly customised, the integration work is mostly controlled by users instead of tool vendors. From the perspective of vehicle companies, virtual prototyping represents a capability to develop new products.

The forgoing is the background and motivation of this work. The challenges can be divided into following concrete technical problems.

1.2.1 Different simulation techniques

What are the different simulation techniques in virtual prototyping? The most basic is monolithic simulation (Fig.1.2), which uses a single solver to calculate system equations. Alternatively, co-simulation uses multiple solvers to calculate various sets of system equations. In addition to being a common approach for integrating software tools, it is also more suitable for multi-domain problems. Simulations can be categorised as offline and RT, depending on whether the simulation is synchronised with the wall-clock time. Moreover, RT simulation is typically used to integrate with real hardware. It has more computational and feasible requirements on the system model, solver type and computer platform [18]. Fixed-step solvers are generally used due to its predictive computational effort and better synchronization [121, 18]. To achieve RT performance, two options are possible.

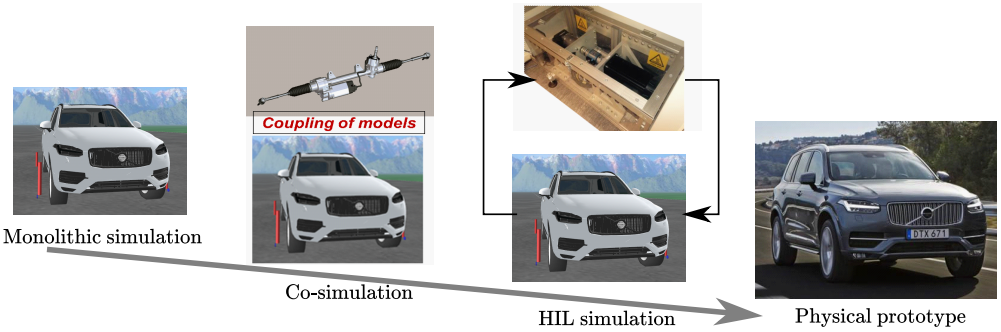


Figure 1.2: Virtual prototyping link desktop simulation to physical prototype.

One is to simplify the model and execute it sufficiently fast with the monolithic simulation. The other is to split the model and run co-simulation on distributed threads or processors [28]. These are the techniques on the simulator side. The accuracy of results are not only determined by system models but also numerical aspects, such as the step-size employed and the implemented co-simulation scheme [30].

In addition to the foregoing simulation techniques, a more advanced and complex application is HIL simulation, which combines the hardware device under test (DUT) and the digital RT simulator (DRTS) [69, 51]. This approach is also widely called X-in-the-loop technique, in which X represents the DUT (e.g., a test-rig, a human-machine interface and a human driver). Furthermore, HIL simulation can be geographically distributed in that different parts are connected by internet communication [36]. Accordingly, this eliminates the cost of building and maintaining test facilities. In addition to the aforementioned numerical aspects, the control of hardware [61] and communication delay also add error to the system and distort the simulation result.

To validate the simulation tools, *consistency* is continuously pursued in related works [36, 91, 61]. Model behaviours must be seamlessly produced with different simulation techniques. The factors that can lead to inconsistency are categorised in two: **architecture** and **coupling problems**. These refer to how the system is built and how coupling variables (signals) are treated, respectively.

1.2.2 Architecture problem

The architecture problem refers to the virtual prototype configuration, i.e., how the system model is distributed and assembled. In co-simulation, different sets of equations are assigned to the solvers. Once the assignment is prepared, the input-output variables (i.e., the variables coupled to sub-models) are fixed. Their flow direction (known as causality) is specified. The architecture setup may yield different performance levels in terms of accuracy and stability, as demonstrated

in a vehicle-track interaction example [80].

In the HIL simulation, the architecture is extended with a virtual-physical interface connecting the DUT and DRTS. Similarly, the coupling variables and causality are determined by corresponding measurements, control variables and control strategy in the interface. Different design choices can yield varying HIL simulation performance levels and influence *consistency*. In addition, the work shows that some auxiliary components can be added to modify system performance. The architecture problem is further discussed in this thesis as well as in **Papers I** and **V**.

1.2.3 Coupling problem

The coupling problem broadly refers to the inaccuracy of coupling variables owing to distribution. Specifically, co-simulation usually has multiple time rates; hence the coupling variables are not constantly updated for the part at a sufficiently fast rate. The coupling variables must be approximated by some means (called coupling methods). The unavoidable coupling error degrades the accuracy of result and may even destabilise the co-simulation [30].

The aforementioned problem also exists in the HIL simulation with its hybrid nature (i.e., a mixture of continuous and discrete time system). Moreover, the coupling variables are transmitted by signals or enforced by control systems. The associated communication delay, measured noises and control errors may further degrade accuracy and stability. To resolve these problems, measures to reduce the coupling error [30], compensate for the communication delay [21], and reduce the control tracking error [29] are implemented. The coupling problem is further discussed in this thesis and covered in **Papers III** and **IV**.

1.3 Research questions

To render the work balanced and sensible, the virtual prototyping of a vehicle-EPAS system (Fig.1.2) is considered as a real engineering case. The theoretical findings and methods are implemented and validated experimentally on such a test bench. The original research question of the project is as follows.

Q. *How can vehicle and mechatronic system development benefit from advanced simulation techniques ?*

As the work progressed, this original question has been refined into the following research questions according to practical necessity and problems:

Q1. *How can the virtual prototype of vehicle-EPAS system be constructed and the interface be designed ?*

This question aims to investigate the architecture problem in a distributed simulation. The effects of interface design must be explained on a fundamental level. This thesis intends to provide guidelines on the vehicle-EPAS system development.

Q2. *How can the performance of vehicle-EPAS system simulation be improved?*

The question aims to investigate the coupling problem and employ coupling methods to improve system stability and accuracy. In addition to a revision on the state-of-the-art methods, new methodologies for overcoming the multi-rate effect are sought. Lastly, this work intends to provide guidelines on how these methods are to be implemented on the test bench.

Q3. *How does HIL simulation differ from computer simulation ?*

The question aims to provide fundamental insights on impact factors in HIL simulation where *consistency* may be degraded in practise. The impact factors on accuracy and stability must be determined.

Q4. *How can the performance of HIL simulation connected to an EPAS system test rig be improved ?*

This question aims to reduce the impact factors described in **Q3**. Methods for resolving the coupling and architecture problems must be developed. Furthermore, experimental results are analysed, and implementation guidelines are formulated.

1.4 Scientific contributions

The scientific contributions of this work are summarised as follows:

1. The different virtual prototypes of vehicle-EPAS system have been created, including an FMI-based co-simulation, a motor-in-the-loop simulation and a driver-in-the-loop simulation.

2. Insights on co-simulation and HIL simulation problems are provided. The simulation techniques are analysed integrally as coupled system dynamics, relying on a linear robust control framework.

3. Proposals for the architecture setup are provided for the design of both co-simulation and HIL simulation.

4. State-of-the-art and new coupling methods are reviewed and developed. An approach using adaptive filters to compensate for the delay effect is developed. To reduce the coupling error due to the multi-rate effect, a frequency-domain optimisation approach (H_∞ method) is also devised.

5. Interface algorithms are developed to improve the HIL simulation performance and implemented on a prototype with a motor-in-the-loop. The validation range of the virtual prototype is considerably extended.

1.5 Limitations

- Only the explicit parallel co-simulation (Gauss-Jacobi scheme) has been discussed. The other types of scheme (e.g., Gauss-Seidel and iterative schemes) are not considered due to the limited support on the use of prototyping tools.
- To facilitate the analysis in a linear robust control framework, the example cases are assumed as linear time-invariant (LTI) systems. Linear approximations are made in the development of coupling methods and interface algorithms.
- Only single-core simulation is performed and full potential of distributed simulation is not investigated. Advanced features on the computer side are not discussed. For example, efficient scheduling techniques are relevant to the speed of distributed simulation [28]; these are not included in the scope of this work.
- The vehicle-EPAS system interaction is mainly considered by the low-frequency lateral dynamics. Consequently, the interfacing problems in this case may be less evident.

Furthermore, considering the broad area related to the subject, the author may be unaware of some interesting concepts and methods.

1.6 Thesis Outline

This thesis has two major parts. The first part focuses on theoretical principles using simple examples to facilitate comprehension. Results can be reproduced with less effort. **Chapter 2** outlines the features of computer simulation. It presents the investigation of co-simulation and the fundamental analysis of stability and accuracy. Next, co-simulation coupling methods are introduced. **Chapter 3** presents a generalisation of HIL simulation. Similarly, the analysis of stability and accuracy is elaborated. Moreover, interface algorithms for enhancing the HIL simulation performance are introduced.

The second part discussed the application of the foregoing to a real engineering case. **Chapter 4** presents the vehicle-EPAS system, mathematical models and dynamic characteristics. The construction of virtual prototypes, implementation,

and test facilities is explained, and experimental results are given. Finally, the summary of this thesis and ideas for future research are presented.

The appended papers cover topics on different aspects (Fig.1.3) that interactively determine the performance of a virtual prototype. **Paper I** presents the co-simulation interface design and causality problem. **Paper II** introduces the virtual prototypes of the vehicle-EPAS system using an FMI-based co-simulation. The acceleration of simulation using different setups is presented. **Paper III** discusses the explicit co-simulation approach in detail. A new coupling method based on H_∞ synthesis to reduce the coupling error is proposed. **Paper IV** presents the delay effect on RT application. Discrete adaptive filters are used for delay compensation. (The latter developed H_∞ method can also be applied to the delay problem.) **Paper V** elaborates on an EPAS motor-in-the-loop prototype. A new interface algorithm is employed to alter the system architecture and improve the HIL simulation performance.

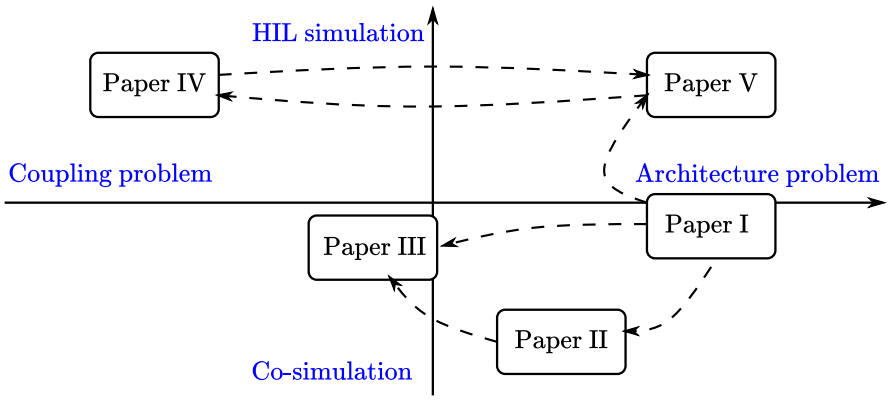


Figure 1.3: A map of appended papers showing the problem dimensions of the work and how the papers are related.

Part I

Theory

Chapter 2

Computer simulation

This chapter introduces the fundamentals of computer simulation. The fundamentals on accuracy and stability of monolithic simulation and co-simulation are elaborated. State-of-the-art coupling methods and a new method formulated by the author are presented. Then, the architectural problem is discussed.

2.1 Modelling of mechanical system

Vehicles and the subsystems can be modelled based on classical mechanical principles. A typical mathematical description of nonlinear ordinary differential equations (ODEs) is as follows:

$$\boldsymbol{M}(q)\ddot{q}(t) = f(q, \dot{q}, t) \tag{2.1}$$

where $q(t) \in \mathbb{R}^{n_q}$ is a minimum set of generalized coordinates, and n_q is the degree of freedom; $\boldsymbol{M}$ is a symmetric positive definite matrix containing the corresponding mass and inertia values; f is the vector of generalized forces. If kinematic constraints are introduced and algebraic equations are added, then differential-algebraic equations (DAEs) are derived [1].

With a moderate order of degrees of freedom, Eq. (2.1) yields basic models to capture the dynamics of a system. For example, in a quarter car model, q may be defined as the vertical motions of the wheel and vehicle body. In a single track model, to study the vehicle's planar motion, q is considered as the yaw and lateral motion [54, 42]. To implement a linear or frequency analysis, the system can be linearised around an equilibrium point $\bar{q}$, where $\ddot{q} = \dot{q} = 0$ and $f = f(\bar{q}, 0, 0)$. Given an external input $u(t) \in \mathbb{R}^{n_u}$, the system can be further expressed into the following state-space form.

[1] DAEs of low index can be reduced into ODEs using projection or derivation. Otherwise, DAE integration can be applied as well, which is not covered in the work.

$$\begin{gathered}\dot{x}(t) = Ax(t) + Bu(t) \\ \text{with} \qquad M = M(\bar{q}) \qquad x(t) \coloneqq \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \\ A \coloneqq \begin{bmatrix} I & \mathbf{0} \\ M^{-1}\partial f/\partial q & M^{-1}\partial f/\partial \dot{q} \end{bmatrix} \qquad B \coloneqq \begin{bmatrix} \mathbf{0} \\ M^{-1}\partial f/\partial u \end{bmatrix}\end{gathered} \tag{2.2}$$

Jacobian matrices $\partial f/\partial q, \partial f/\partial \dot{q}, \partial f/\partial u$ can be approximated using the finite difference method [11]. The Jacobian matrices are updated in certain steps according to their variations. This step requires $2n_q + n_u + 1$ evaluations for each force element f, mainly dominating the computations in a multibody dynamics (MBD) system simulation. In a large-scale MBD system (e.g., a passenger vehicle), n_q may exceed 300.

To facilitate the discussion on stability and accuracy, a single mass-spring-damper (MSD) oscillator is considered (Fig.2.1).

$$m\ddot{q}(t) + d\dot{q}(t) + kq(t) = u(t) \tag{2.3}$$

in the LTI system, A and B are invariants. The state vector is the displacement and velocity, $x = [q, \dot{q}]^T$, then

$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -k/m & -d/m \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ 1/m \end{bmatrix}}_{B} u(t) \tag{2.4}$$

The state matrix eigenvalues, $\lambda(A) \in \mathbb{C}^{2n_q}$ are known to characterise the behaviour of the system [14]. The system is exponentially stable if all eigenvalues are in the complex open left half plane. With complex eigenvalues, the system becomes oscillatory.

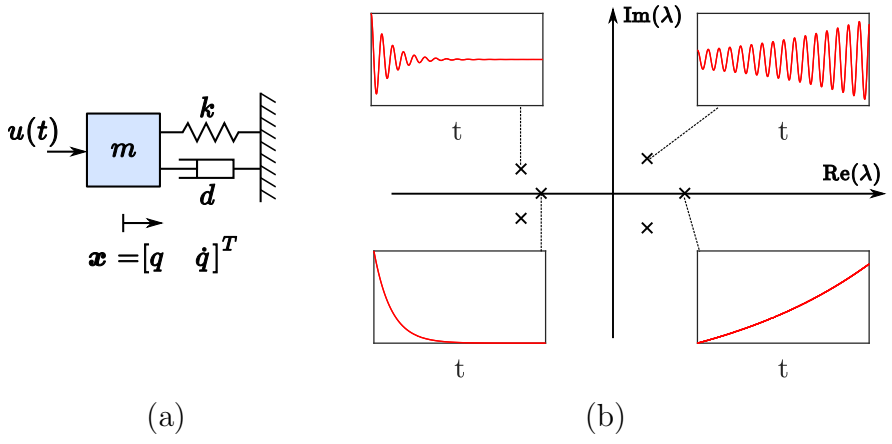


Figure 2.1: (a) An LTI MSD oscillator; (b) Time response of different $\lambda(A)$ values.

2.1.1 Classic ODE integration

In computer simulation, the system model (Eq.(2.4)) can be integrated using standard ODE solvers. Explicit solvers calculate the new states based on the previous values. The simplest solver is the forward Euler method.

$$x_{t_{n+1}} = x_{t_n} + h\dot{x}_{t_n} \qquad \text{with} \qquad h \coloneqq t_{n+1} - t_n \tag{2.5}$$

The well-known explicit Runge–Kutta (RK) method is composed of a linear combination of time derivatives at intermediate points.

$$\begin{aligned} & x_{t_{n+1}} = x_{t_n} + h\sum_{i=1}^{s} b_i k_i \\ \text{with} \quad & k_1 \coloneqq f(x_{t_n}, t_n) = \dot{x}_{t_n} \\ & k_i \coloneqq f(x_{t_n} + h\sum_{j=1}^{i} a_{ij}k_j, t_n + c_i h) \quad (i = 2, 3...., s) \end{aligned} \tag{2.6}$$

For a given stage number s, the coefficients b_i, a_{ij}, c_i are reported in the literature [14, 35]. Due to the evaluation of k_i, the numerical effort is s times the forward Euler method ($s = 1$).

The accuracy of the solver can be checked based on its deviation from the Taylor series. The one-step error of the Euler method (Eq.(2.5)), i.e., the local error, is equal to the residual $\xi_{local} = \boldsymbol{O}(h^2)$. Thus, the error during the simulation time $0 \to t_n$ (i.e., the global error), accumulates as $\xi_{global} = \xi_{local} t_n/h = \boldsymbol{O}(h)$. Similarly, RK methods with $s \leq 4$ yield a global error in the order $\boldsymbol{O}(h^s)$. The order number indicates how fast the method converges to the exact solution when h is reduced. For example, when h reduces by a factor of 10, the simulation accuracy level increases by one digit using the Euler method; the computational effort increases 10 times. With the RK4 method, the accuracy level increases by four more digits, and the computational effort increases 40 times. The system (Eq.(2.4)) is specified with $\lambda(A) = -0.05 \pm 1i$. The initial condition is $\dot{x}(0) = [1, 1]^T$, $u(t) = 0$ and the simulation time is 100 s. The simulation statistics with the Euler and RK4 methods are compared in Table 2.1.

Table 2.1: Example of calculation speed and accuracy level of Euler and RK4 method.

Euler method			
h	Elapsed time	ξ_q	$\xi_{\dot{q}}$
0.01	0.026	1.4e-3	6.1e-3
0.005	0.056	5.9e-4	2.7e-3
0.001	0.275	1.0e-4	4.8e-4

RK4 method			
h	Elapsed time	ξ_q	$\xi_{\dot{q}}$
0.01	0.158	7.6e-11	3.1e-11
0.005	0.307	4.8e-12	2.0e-12
0.001	1.537	7.6e-15	3.3e-15

When implemented on a digital simulator, the stability of the system model (Eq.(2.4)) also changes. Considering a single ODE equation, $\dot{x} = \lambda x$ for one mode in the system (Eq.(2.4)), the explicit solver ($s \leq 4$) transforms the system into a difference equation.

$$\begin{aligned} x_{t_{n+1}} &= x_{t_n} + h\dot{x}_{t_n} + \frac{1}{2}h^2\ddot{x}_{t_n} + \frac{1}{6}h^3\dddot{x}_{t_n} + ... \\ &= (\sum_{i=0}^{s} \frac{(h\lambda)^i}{i!})x_{t_n} \end{aligned} \tag{2.7}$$

The simulated system is asymptotically stable if and only if

$$|\sum_{i=0}^{s} \frac{(h\lambda)^i}{i!}| < 1 \tag{2.8}$$

To solve the model (Eq.(2.4)), all eigenvalues $\lambda(A)$ must remain within the stability region (Fig.2.2). Therefore, numerical stability is determined by the model dynamics and the integration method.

For the model to be solvable using a given method, order-reduction techniques can be applied [37]. There is no explicit criterion on how to select h. However, in practise, the rule of thumb is to use a small h provided that the elapsed time is satisfactory. Nevertheless, dealing with stiff systems in which $|\lambda(A)|_{min} \ll |\lambda(A)|_{max}$ has inconsiderable efficiency [11].

2.2 Co-simulation

Co-simulation is literally a cooperation of simulation in which the solution of the system equation (Eq.(2.1)) is distributed. Co-simulatiom has been demonstrated to be necessary in simulating multi-domain problems. Its well-known advantages include the following: it has better numerical efficiency, enables software tool integration, and can be implemented on multiprocessors and distributed computer systems [48, 28]. The early applications of co-simulation to vehicle simulation emerged in the 90s, such as the integration of software coded in C and mechanical part designed in Matlab [71], and the integration of MBD vehicle model and mechatronic subsystems [62, 117]. In addition to oversimplified models and slow simulation speed, the problem of synchronisation and lack of a unified software format were regarded as major challenges. These problems have been resolved by the latter introduced FMI standard, which relies on a master-slave structure, as shown in Fig.2.3. The master is responsible for exchanging the coupling variables at a specific time step (i.e., the macro-step H). In contrast, the slaves are models whose solution is similar to that of monolithic simulation.

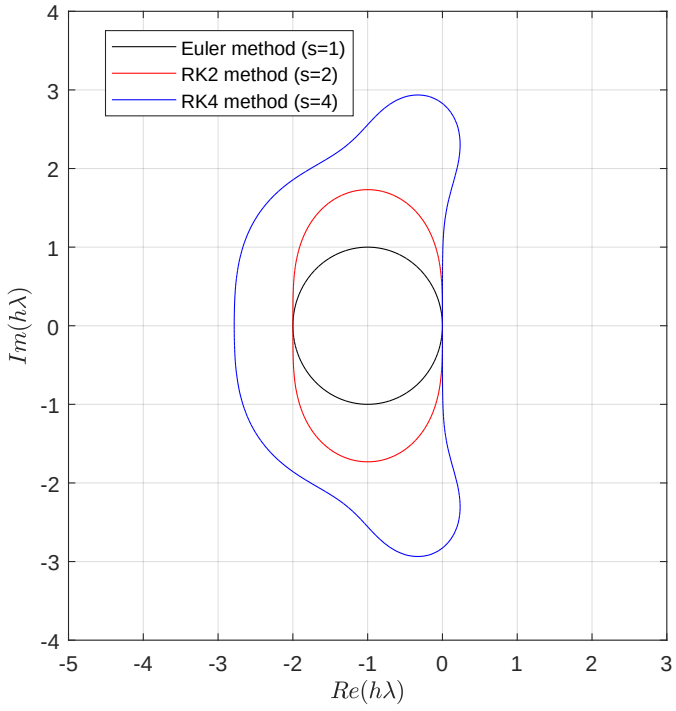


Figure 2.2: Stability regions for explicit solvers. The regions are scaled by the size of h. λ inside the region satisfies Eq.(2.8); thus, the mode is stable during computation.

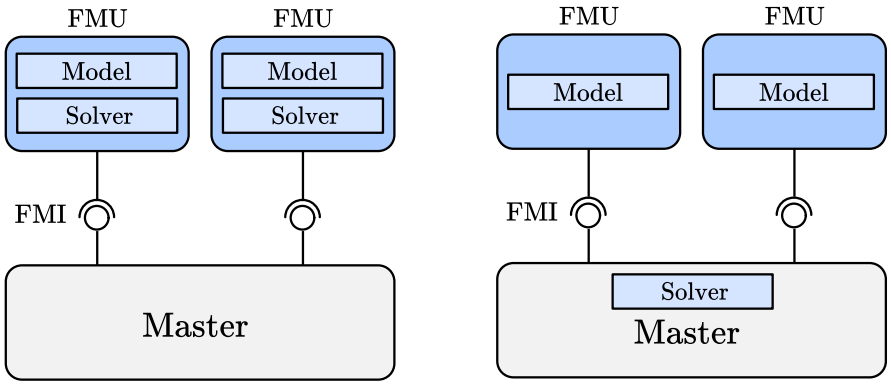


Figure 2.3: FMI standard for co-simulation (left) and model exchange (right). The latter is called tightly-coupled co-simulation in previous references [62, 117, 115]. However, we do not regard this as co-simulation because distribution does not occur in computation.

In the last decade, the fundamental numerical analysis for co-simulation was introduced [65, 10, 30, 12, 13], and the number of contributions to the literature continued to grow. The trend may be motivated by the evolution of computational

power and increasing demand for RT applications. Two well-written reports [46, 74], were found to provide a comprehensive description of the development of the subject including subsequent revisions.

In summary, co-simulation is categorised by master algorithms as non-iterative and iterative schemes [30, 101]. In the iterative scheme, the master proceeds with the steps provided that the error tolerance is satisfied, thus ensuring accuracy. To achieve this, the slave must return to the previous steps and repeat if requested by the master. Therefore, more computational effort and difficulty in implementation are expected with the use of commercial software tools. If the slave conducts iterations in each step, the computational burden can become extremely high. Furthermore, hardware and control software are definitely not applicable to this scheme as slaves. The investigation of other co-simulation variants, such as variable and fixed macro-step approaches, was also reported in the literature [96, 97]. The variable macro-step approach exhibits the aforementioned limitations, especially when RT requirements are involved. This work only focus on the most widely used co-simulation type, i.e., the explicit parallel co-simulation with fixed macro-step (Fig.2.4).

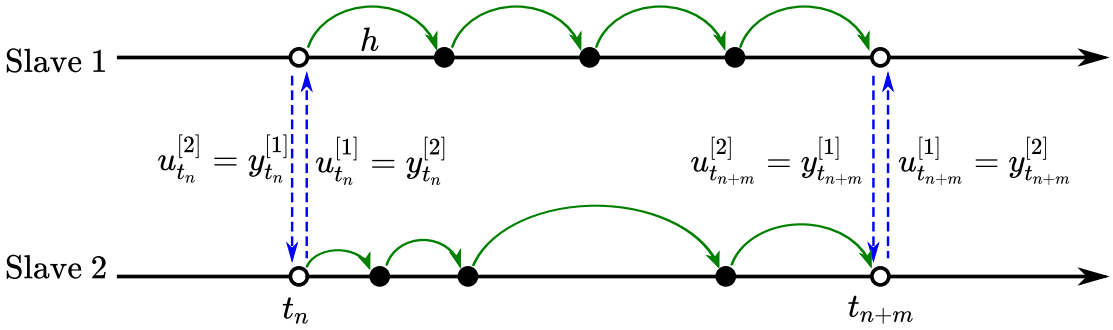


Figure 2.4: Explicit parallel co-simulation consisting of two slaves. Slave 1 is calculated using fixed-step solver at step size h; Slave 2 is solved with variable-step solver. Slaves are coupled at instant t_n and t_{n+m}.

2.2.1 Mathematical description

To elaborate the difference to the aforementioned monolithic simulation, the mathematical description of explicit parallel co-simulation is presented here. An LTI system is considered as a slave model and solved at micro-step h to facilitate analysis. The discrete system is expressed as follows:

$$\begin{aligned} x_{t_{n+1}} &= \mathcal{A}x_{t_n} + \mathcal{B}u_{t_n} \\ y_{t_{n+1}} &= \mathcal{C}x_{t_{n+1}} + \mathcal{D}u_{t_{n+1}} \\ \text{with} \quad \mathcal{A} &:= e^{Ah} \qquad \mathcal{B} := B/A(e^{Ah} - I) \end{aligned} \tag{2.9}$$

$\mathcal{A}, \mathcal{B}$ are calculated by truncating the Taylor series, and the error order is known. $\mathcal{C}, \mathcal{D}$ are output matrices. Given the macro-step $H = mh$, the slave repeats the step (Eq.(2.9)) m times in between communication. Then, the integration steps during the time $t \in [t_n, t_{n+m}]$ can be expressed using state-space realisation.

$$\begin{bmatrix} x_{t_{n+m}} \\ y_{t_n} \\ y_{t_{n+1}} \\ \vdots \\ y_{t_{n+m}} \end{bmatrix} = \left[\begin{array}{c|ccc} \mathcal{A}^m & \mathcal{A}^{m-1}\mathcal{B} & \mathcal{A}^{m-2}\mathcal{B} & \dots & 0 \\ \hline \mathcal{C} & \mathcal{D} & 0 & \dots & 0 \\ \mathcal{C}\mathcal{A} & \mathcal{C}\mathcal{B} & \mathcal{D} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}\mathcal{A}^m & \mathcal{C}\mathcal{A}^{m-1}\mathcal{B} & \mathcal{C}\mathcal{A}^{m-2}\mathcal{B} & \dots & \mathcal{D} \end{array}\right] \begin{bmatrix} x_{t_n} \\ u_{t_n} \\ u_{t_{n+1}} \\ \vdots \\ u_{t_{n+m}} \end{bmatrix} \tag{2.10}$$

The above expression is equal to the monolithic simulation because $u(t)$ is constantly known. In the co-simulation, the slave input during the communication interval $(t_n, t_{n+m}]$ is unknown and approximated using certain coupling methods. In the most common setup, the slave exports the latest output $y_{t_{n+m}}$ and uses a zero-order hold (ZOH) method, i.e., $u(t) = u_{t_n}, \forall t \in [t_n, t_{n+m}]$. Then, the slave system becomes:

$$\begin{aligned} \begin{bmatrix} x_{t_{n+m}} \\ y_{t_{n+m}} \end{bmatrix} &= \left[\begin{array}{c|c} \mathcal{A}^m & \sum_{i=1}^{m} \mathcal{A}^{m-i}\mathcal{B} \\ \hline \mathcal{C}\mathcal{A}^m & \sum_{i=1}^{m} \mathcal{C}\mathcal{A}^{m-i}\mathcal{B} + \mathcal{D} \end{array}\right] \begin{bmatrix} x_{t_n} \\ u_{t_n} \end{bmatrix} \\ &= \left[\begin{array}{c|c} \mathcal{A}_m & \mathcal{B}_m \\ \hline \mathcal{C}_m & \mathcal{D}_m \end{array}\right] \begin{bmatrix} x_{t_n} \\ u_{t_n} \end{bmatrix} \end{aligned} \tag{2.11}$$

The dimensions of the matrices above increases with the use of other higher-order methods. This step only transfers the slave into a system at a rate of H. The stability of the slave remains dependent on $\mathcal{A}$. Using the same form as Eq.(2.11) for each LTI slave, a number of p slaves can be combined into the following:

$$\begin{aligned} \begin{bmatrix} x^{[1]}_{t_{n+m}} \\ \vdots \\ x^{[p]}_{t_{n+m}} \end{bmatrix} &= \bar{\mathcal{A}} \begin{bmatrix} x^{[1]}_{t_n} \\ \vdots \\ x^{[p]}_{t_n} \end{bmatrix} + \bar{\mathcal{B}} \begin{bmatrix} u^{[1]}_{t_n} \\ \vdots \\ u^{[p]}_{t_n} \end{bmatrix} \\ \begin{bmatrix} y^{[1]}_{t_{n+m}} \\ \vdots \\ y^{[p]}_{t_{n+m}} \end{bmatrix} &= \bar{\mathcal{C}} \begin{bmatrix} x^{[1]}_{t_n} \\ \vdots \\ x^{[p]}_{t_n} \end{bmatrix} + \bar{\mathcal{D}} \begin{bmatrix} u^{[1]}_{t_n} \\ \vdots \\ u^{[p]}_{t_n} \end{bmatrix} \end{aligned} \tag{2.12}$$

where $\bar{\mathcal{A}} \coloneqq blkdiag(\mathcal{A}^{[1]}_m, \dots, \mathcal{A}^{[p]}_m)$, $\bar{\mathcal{B}} \coloneqq blkdiag(\mathcal{B}^{[1]}_m, \dots, \mathcal{B}^{[p]}_m)$

$\bar{\mathcal{C}} \coloneqq blkdiag(\mathcal{C}^{[1]}_m, \dots, \mathcal{C}^{[p]}_m)$, $\bar{\mathcal{D}} \coloneqq blkdiag(\mathcal{D}^{[1]}_m, \dots, \mathcal{D}^{[p]}_m)$

the superscript denotes the p th slave system. The notation m is retained for convenience. Unequal micro-steps can be taken by the slaves, which should not

cause confusions here. To form the co-simulation network, the slaves are coupled by a mapping $\mathcal{L}$:

$$\begin{bmatrix} u^{[1]}_{t_{n+m}} \\ \vdots \\ u^{[p]}_{t_{n+m}} \end{bmatrix} = \mathcal{L} \begin{bmatrix} y^{[1]}_{t_{n+m}} \\ \vdots \\ y^{[p]}_{t_{n+m}} \end{bmatrix} \tag{2.13}$$

Finally, all slaves are coupled in a closed-loop manner and yield the following.

$$\begin{bmatrix} x^{[1]}_{t_{n+m}} \\ \vdots \\ x^{[p]}_{t_{n+m}} \end{bmatrix} = \bar{\mathcal{A}} + \bar{\mathcal{B}}\mathcal{L}(I - \bar{\mathcal{D}}\mathcal{L})^{-1}\bar{\mathcal{C}} \begin{bmatrix} x^{[1]}_{t_n} \\ \vdots \\ x^{[p]}_{t_n} \end{bmatrix} \tag{2.14}$$

The states of the co-simulation evolves as described above. If $det(I-\bar{\mathcal{D}}\mathcal{L}) = 0$, then an algebraic loop exists [45]. This means that the slaves have feed-through channels in $\bar{\mathcal{D}}$, and the corresponding channels are connected by $\mathcal{L}$. Then, some of the inputs u_{t_n} are directly dependent on the outputs y_{t_n}, and subsequently become self-dependent. This type of problem can be solved using iteration schemes [66, 102] or artificial bushing elements to break the self-dependency [30, 99]. To avoid the algebraic loop (i.e., $det(I - \bar{\mathcal{D}}\mathcal{L}) \neq 0$), a force-displacement or force-velocity coupling can be selected for the considered mechanical system.

2.2.2 Accuracy and stability

Accuracy is investigated starting from the local error of the state vector in a step of H. The exact and co-simulation results are denoted by $x_{t_{n+1}}$ and $\tilde{x}_{t_{n+1}}$ respectively. Their difference is

$$\begin{aligned} \xi_{x,local} &= x_{t_{n+1}} - \tilde{x}_{t_{n+1}} \\ &= \underbrace{(e^{\mathcal{A}mh} - \mathcal{A}^m)}_{m\boldsymbol{O}(h^{s+1})} x_{t_n} + \underbrace{\sum_{i=1}^{m} \mathcal{A}^{m-i}\mathcal{B}\xi_{u_i,local}}_{m\boldsymbol{O}(H^{k+1})} \\ \text{with} \qquad \xi_{u_i,local} &\coloneqq u_{t_{n+i}} - \tilde{u}_{t_{n+i}} \end{aligned} \tag{2.15}$$

It is known that the approximation error of kth order Lagrange polynomial is $\xi_{u_i,local} = \boldsymbol{O}(H^{k+1})$ [30]. If the slave solver error $m\boldsymbol{O}(h^{s+1})$ is negligible, then, $\xi_{x,local} = m\boldsymbol{O}(H^{k+1}) = \boldsymbol{O}(H^{k+2})$. Following this pattern, the global error $\xi_{x,global}$ propagates from the initial point and then converges with $\boldsymbol{O}(H^{k+1})$. The local error of the output vector can be easily checked using

$$\begin{aligned} \xi_{y,local} &= y_{t_{n+1}} - \tilde{y}_{t_{n+1}} \\ &= \mathcal{C}\xi_{x,local} + D\xi_{u_i,local} \end{aligned} \tag{2.16}$$

which is of the same order as $\xi_{x,local}$. This reveals that the accuracy of co-simulation is determined by the input approximation (i.e,.coupling method). Therefore, to resolve the errors in explicit parallel co-simulation, various coupling methods are developed. Further details are presented in the next section.

The stability of co-simulation is governed by Eq.(2.14). The system is asymptotically stable if and only if $|\lambda(\bar{\mathcal{A}} + \bar{\mathcal{B}}\mathcal{L}(I - \bar{\mathcal{D}}\mathcal{L})^{-1}\bar{\mathcal{C}})| < 1$. The eigenvalues are determined by the coupled behaviour; thus, the stability region cannot be derived as that in the monolithic simulation. To analyse stability, $\bar{\mathcal{B}}\mathcal{L}(I - \bar{\mathcal{D}}\mathcal{L})^{-1}\bar{\mathcal{C}}$ must be specified with the input and output variables of the slaves. This leads to the considerations of the architecture design of co-simulation slaves, such as force-displacement and displacement-displacement coupling [30]. To clarify the analysis, a classic benchmark problem is introduced.

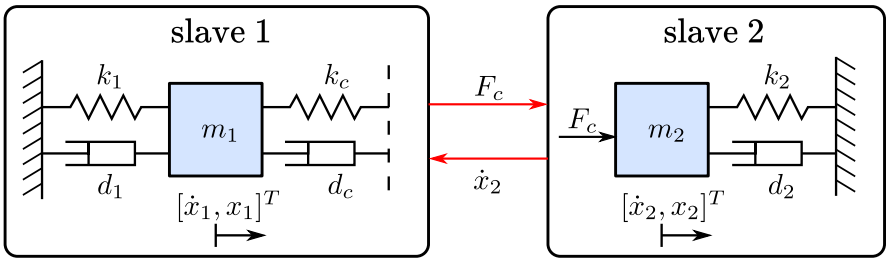


Figure 2.5: Co-simulation setup of the benchmark problem in which the master passes the coupling variables.

***Benchmark**: A typical dual mass-spring-damper (DMSD) system (Fig.2.5) is employed as benchmark. The parameters are $m_1 = m_2 = 0.1$ kg, $k_1 = k_2 = k_c = 10$ N/m, $d_1 = d_2 = d_c = 0.1$ Ns/m. In the explicit co-simulation, a force-velocity coupling approach is used.*

The slave models are as follows

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k_1+k_c}{m_1} & -\frac{d_1+d_c}{m_1} \end{bmatrix}}_{A^{[1]}} \begin{bmatrix} x_1 \\ \dot{x}_1 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{k_c}{m_1} & \frac{d_c}{m_1} \end{bmatrix}}_{B^{[1]}} \begin{bmatrix} x_2 \\ \dot{x}_2 \end{bmatrix} \\
F_c &= \underbrace{\begin{bmatrix} k_c & d_c \end{bmatrix}}_{C^{[1]}} \begin{bmatrix} x_1 \\ \dot{x}_1 \end{bmatrix} + \underbrace{\begin{bmatrix} -k_c & -d_c \end{bmatrix}}_{D^{[1]}} \begin{bmatrix} x_2 \\ \dot{x}_2 \end{bmatrix}
\end{aligned} \tag{2.17}$$

$$\begin{aligned}
\begin{bmatrix} \dot{x}_2 \\ \ddot{x}_2 \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k_2}{m_2} & -\frac{d_2}{m_2} \end{bmatrix}}_{A^{[2]}} \begin{bmatrix} x_2 \\ \dot{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m_2} \end{bmatrix}}_{B^{[2]}} F_c \\
\begin{bmatrix} x_2 \\ \dot{x}_2 \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C^{[2]}} \begin{bmatrix} x_2 \\ \dot{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{D^{[2]}} F_c
\end{aligned} \tag{2.18}$$

First, we can observe the global error $\xi_{x_1,global}$ in Fig.2.6 (with $h = 1e-6$ s). $\xi_{x_1,global}$ converges in $\boldsymbol{O}(H)$ and $\boldsymbol{O}(H^2)$ with the ZOH and FOH methods, respectively. Next, stability can also be checked by the variation of the spectral radius with m. The slaves are discretised with the Euler method at $h = 5e-3$ s. Stability deteriorates when m increases, in this case the FOH method enables a larger macro-step.

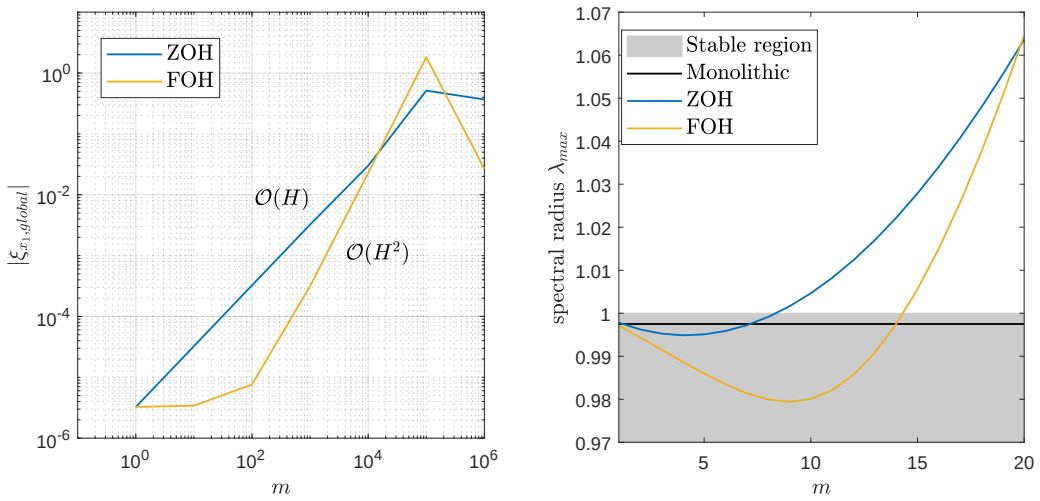


Figure 2.6: Accuracy and stability of the benchmark problem using different coupling methods.

However, stability actually varies with the system parameters, and co-simulation architectures. If different combinations of parameters, coupling methods and architectures are considered, the dimension of the problem significantly increases. This is known as a mutual reaction between the system dynamics and coupling method. Even a simple DMSD system can yield lengthy results from which deriving a conclusion or generalisation for complex engineering applications is difficult [30, 48, 73].

The benchmark system remains an open question in co-simulation research. Other examples such as the non-linear double pendulum [8, 103] and hydraulic system [89] are also reported in the literature. In non-linear cases, numerical tests are first conducted. Performance levels are evaluated a posteriori (i.e., after

the co-simulation results are obtained). Nevertheless, dependency effects on the benchmark system and input excitation persist [88].

Accordingly, the foregoing has motivated the author to investigate the problem in the control framework. By considering the salves by their overall transfer behaviour in the frequency domain, the dimensions of the problem are reduced. The numerical errors of co-simulation are regarded as imperfect perturbations in the system. The equivalent error and stability analysis in the control framework is presented in **Paper III**. Furthermore, the coupling method can be developed with a less case-dependent effect, as discussed in the following.

2.3 Coupling methods

The most widely used methods for approximating the input $u(t)$ are the Lagrange polynomials of degree k ($k = 0, 1, 2$), i.e., zero-order hold (ZOH), first-order hold (FOH) and second-order hold (SOH) methods (Fig.2.7). The k th-order polynomial is determined from $k + 1$ points. Hence, a higher-order polynomial contains more information in the past and yields less information to more recent changes. In another words, its high-frequency precision is compromised. This trend is clearly demonstrated by comparing the approximation error in the frequency domain (Fig.2.8). Note that $\|\xi_u\|$ increases using higher-order methods when $\omega H/2\pi > 0.2$, i,e., $f_u > 0.2/H$ (f_u is the input frequency). Generally, high-frequency input is more critical in co-simulation. Due to this characteristic, higher-order methods more easily become unstable [30, 101, 8]. Consequently, their usage is limited in practise.

To better approximate $u(t)$ in the macro-step, one idea is to disclose most of the system information as possible. For example, the system Jacobian matrices are used [106, 82], the internal states and analytical high-order derivatives are exploited [8], and the analytical derivatives with respect to u and x are added as new features by the standard FMI 2.0 [26]. These approaches must disclose the system information; therefore, software code support is necessary. Another idea is the use of input-output data and considering the slave as a black box. This is applicable to most currently used tools because there is no model disclosure, or only a slight modification is necessary. A loosely defined distinction is used for the revision.

Polynomial-based approach

Polynomial-based approach extrapolates the input like Lagrange polynomials. For example, multiple micro-step values $\left[\ y_{t_n},\ \ldots,\ y_{t_{n-k-1}}\ \right]^T$ can be exported at the communicative instant t_n. Then, the extrapolation can rely on the output at a finer step [21]. With a lower delay, the accuracy improves (Fig.2.9). In another case, a heuristic method is employed. Input data are classified into various

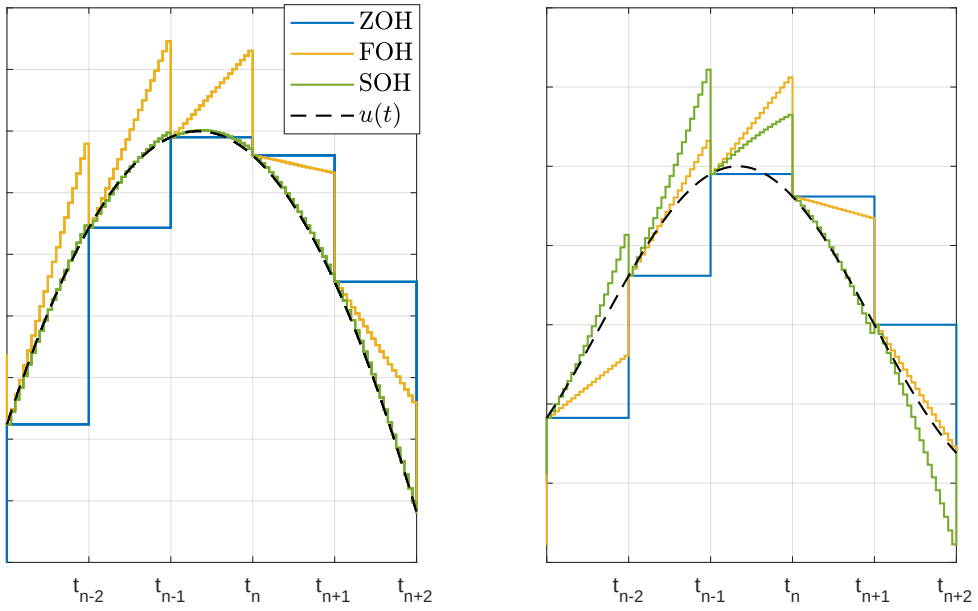


Figure 2.7: Extrapolation of $u(t)$ with Lagrange polynomials based on equidistant macro-step H. Left and right plots show the extrapolation of $u(t)$ in low and high frequencies, respectively.

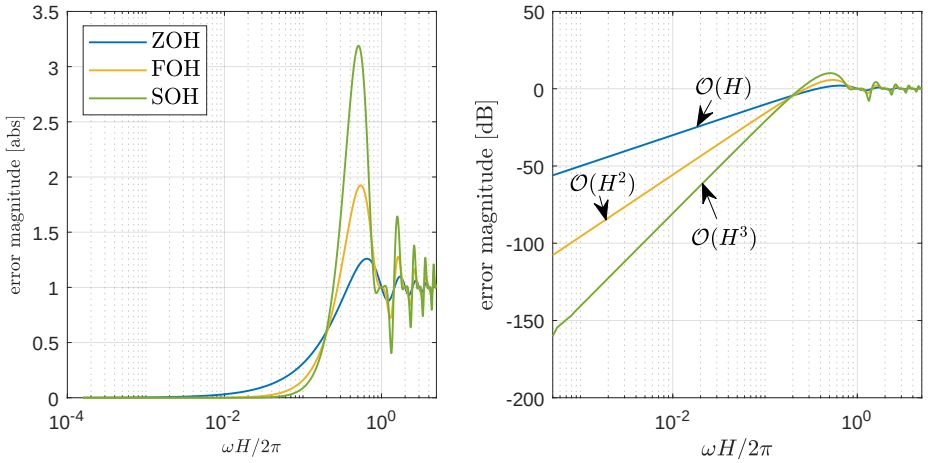


Figure 2.8: Extrapolation error $\|\xi_u\|$ using Lagrange polynomials with respect to normalised frequency $\frac{\omega H}{2\pi}$. Higher-order polynomials yield faster converge rates but lose high-frequency precision.

patterns, then different polynomials are used according to the recognised pattern [59, 19].

In the use of polynomial approaches, a discontinuity problem may occur at the

communicative instant (Fig.2.9). This may degrade the local solver performance, and a correction is added to improve the connection [8]. In terms of mechanical dynamics, this artificial discontinuity is undesired because it may induce a sudden force excitement or even a fast-changing displacement. In essence, the discontinuity is due to the lack of frequency control. However, the input's frequency component may be equally important as the error, as shown by the example in Fig.2.10.

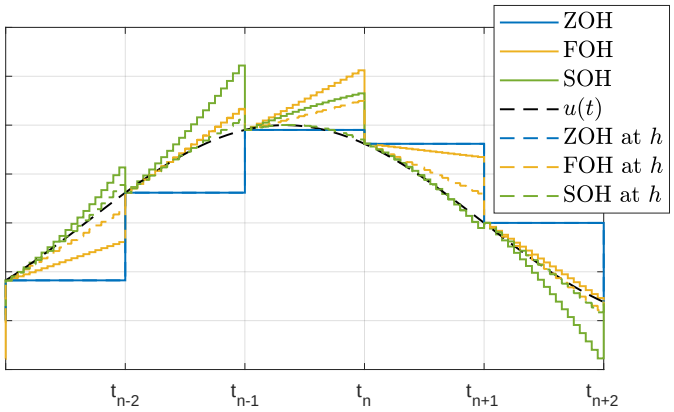


Figure 2.9: Extrapolation of $u(t)$ with Lagrange polynomials based on an equidistant macro step H and the micro step h.

Model-based approach

Model-based approach employs input-output data to identify a reduced model for extrapolation. For example, an MBD model is replaced by an adaptive reduced model to generate the intermediate inputs [84, 85]. The reduced model is a less computationally heavy alternative and can be identified from the input-output data. In these approaches, the extended Kalman filter algorithm and recursive least squares algorithms can also be used [111, 112].

In the author's point of view, the following are the problems unsolved by state-of-the-art explicit coupling methods and some of the expected requirements.

- Can the coupling method reduce the error and preserve the frequency content? As discussed, the coupling approach may exhibit discontinuity, which must be avoided.

- Can the coupling method performance be evaluated a priori? This requisite is twofold. First, if the method can be evaluated alone, it may be applied with a less case-dependent effect. Second, a posteriori evaluation requires complete results and ideal reference, which might be unavailable in practise.

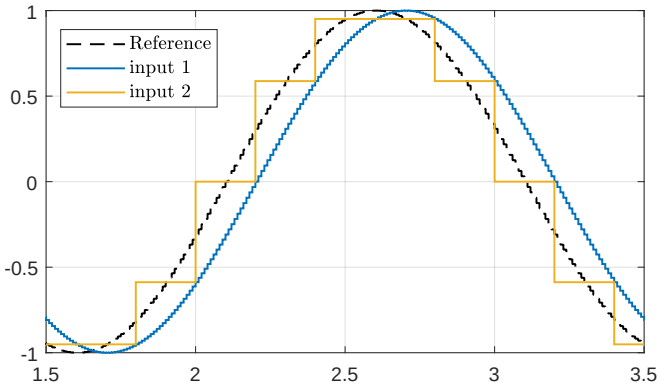


Figure 2.10: Motivating example: input 2 is better than input 1 in terms of maximum error ($0.309 < 0.3129$) and root-mean square error ($0.1286 < 0.2213$). However, delayed input 1 has better performance in terms of frequency content.

- Can the coupling method be transparently tuned, and is it interpretable? As discussed, if the methods can be independently evaluated, then they can be easily tuned for better performance. On the other hand, the parameters involved must be transparent and interpretable for tuning.

The coupling method is selected according to the software capabilities and practical requirements. In this work, a concept that can partially satisfy the identified requirements is proposed.

2.3.1 H_∞ synthesis method

We consider an input-output based approach with a same condition as that in [21], i.e., the finer output values in $t \in [t_{n-m}, t_n]$ are accessible. The objective is to predict $u(t)$ for $t \in [t_n, t_{n+m}]$ for the slave. Moreover, focus is set on a single input-output interface, and the problem is clearly illustrated as a conceptual layout (Fig.2.11). The slave output $y \in \mathbb{R}$ from a slave during a macro-step H is reshaped into a column vector $\bar{y} \in \mathbb{R}^m$. This operation enables to describe the system into a single rate H similar to Eq.(2.11). Here, Φ_1, Φ_2 are the linear operators for generating the output and input respectively. Moreover, the k-th order Lagrange polynomial using finer outputs is obtained by letting $\Phi_1 \in \mathbb{R}^{k\times m}, \Phi_2 \in \mathbb{R}^{m\times k}$ as in the following example.

$$\begin{aligned}
&\text{ZOH method}: \quad \Phi_1 = \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}, \qquad \Phi_2 = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T \\
&\text{FOH method}: \quad \Phi_1 = \begin{bmatrix} 0 & \dots & 1 & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}, \qquad \Phi_2 = \begin{bmatrix} 0 & -1 & \dots & -m \\ 1 & 2 & \dots & m+1 \end{bmatrix}^T
\end{aligned} \tag{2.19}$$

Because only outputs in the previous macro-step are available, $\bar{y}$ is supposed to have a delay of H. If the delay effect is neglected, the operators must satisfy $\Phi_2\Phi_1 = I_m$ for a lossless communication, I_m denotes the $m \times m$ identity matrix. If Φ_2 is given, the optimal Φ_1 is $\Phi_1 = \Phi_2^{-1}$, which is solvable when Φ_1, Φ_2 are non-singular and $m \times m$ matrices. However, this does not provide a solution because it simply requires all outputs in time to generate inputs. The operations may have inputs and outputs of different dimensions as those given in Eq.(2.19). One means for measuring the closeness to the ideal communication is using $||\Phi_2\Phi_1 - I_m||_\infty$, i.e., the maximum singular value of the matrix.

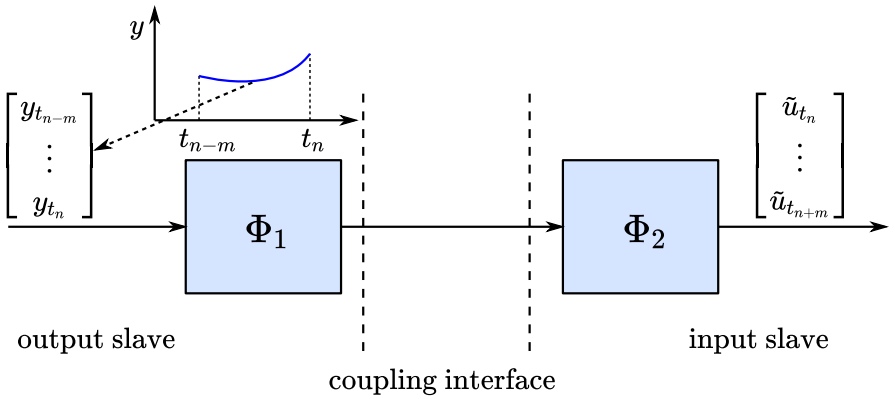


Figure 2.11: Conceptual layout of single input-output interface.

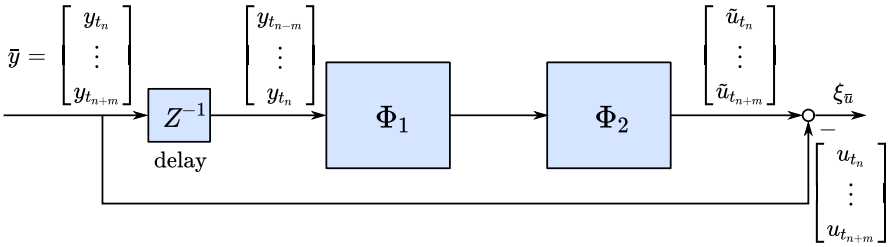


Figure 2.12: Discrete error system for a single output-input interface.

When the macro-step delay is considered, the conceptual layout can be transferred using a discrete error system in the Z-domain (Fig.2.12). An ideal interface path is introduced to derive the input approximation error $\xi_{\bar{\boldsymbol{u}}} \in \mathbb{R}^m$

$$\xi_{\bar{\boldsymbol{u}}} = T_{ye}\bar{\boldsymbol{y}}, \qquad T_{ye} := \Phi_2\Phi_1 Z^{-1} - I_m \tag{2.20}$$

where Z^{-1} denotes the one step delay of H. Note that the ideal path do not necessarily exist in the co-simulation, hence $\xi_{\bar{u}}$ is actually unknown. However, L_2 norm of error $||\xi_{\bar{u}}||_2$ satisfies the following

$$||\xi_{\bar{\boldsymbol{u}}}||_2 = ||T_{ye}\bar{\boldsymbol{y}}||_2 \leq ||T_{ye}||_\infty ||\bar{\boldsymbol{y}}||_2 \tag{2.21}$$

Then, synthesising a pair of Φ_2, Φ_1, which yield a minimum $||T_{ye}||_\infty$, to further bind $||\xi_{\bar{\boldsymbol{u}}}||_2$ is the central idea of the concept. Accordingly, we refer it as the H_∞ coupling method. Here, the concept has been explained using a discrete formulation to be more consistent with the context. In the implementation, an equivalent continuous formulation is actually preferred (Fig.2.13). The reason is that the synthesised continuous operators, i.e.,$K_1(s), K_2(s)$, can be easily implemented with solvers of different rates or with a variable-step solver.

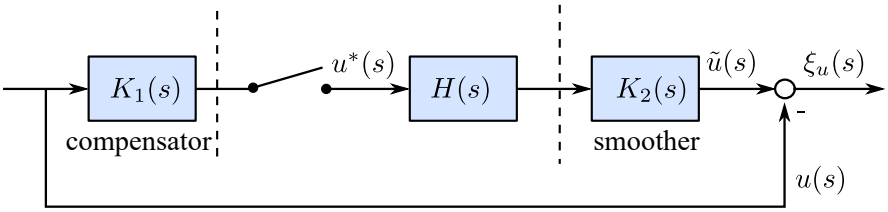


Figure 2.13: Continuous equivalence of error system. $u^*(s)$ and $H(s)$ refer to sampled value and hold process (e.g., ZOH method) respectively.

The resulting T_{ye} is changed using continuous components, however, the principle remains the same (Eq.(2.21)). Detailed information and parameter specification are presented in **Paper III**. In particular, $K_2(s)$ is defined as a low-pass filter to smooth the input, and $K_1(s)$ is the compensator to be optimised. A low-pass penalty function $W_f(s)$ is added to the error bound. The objective is twofold: 1. accuracy can be improved in the specified frequency range; 2. extrapolation error can not be actually bounded in the entire frequency range. Optimisation is implemented offline using the Matlab Robust Control Toolbox. The derived values of $K_1(s)$ and $K_2(s)$ are fixed during the simulation; hence, only a small computational cost is added. The performance of the method in terms of accuracy is discussed next.

Accuracy improvement

Performance of linear methods can be directly revealed as in Fig.2.14. It shows that $\xi_{\bar{\boldsymbol{u}}}$ can be tuned to decrease over a wide frequency range. The convergence rate and balance between low-frequency and high-frequency precision are explicitly dependent on the parameters of the penalty function $W_f(s)$.

The method is tested with a swept input to demonstrate its function (Fig.2.15). The accuracy improvement behaviour is confirmed by the closeness of the smoothed input to the reference, even with a relatively large macro step H. Due to the low-pass behaviour of the smoother, the synthesised compensator distinctly introduces a phase-lead in advance to reduce the error. More numerical test results with basic benchmark systems and the application to the vehicle-EPAS co-simulation are presented in **Paper III**. The implementation of the approach requires the addition of the smoother and compensator to the corresponding port of the model.

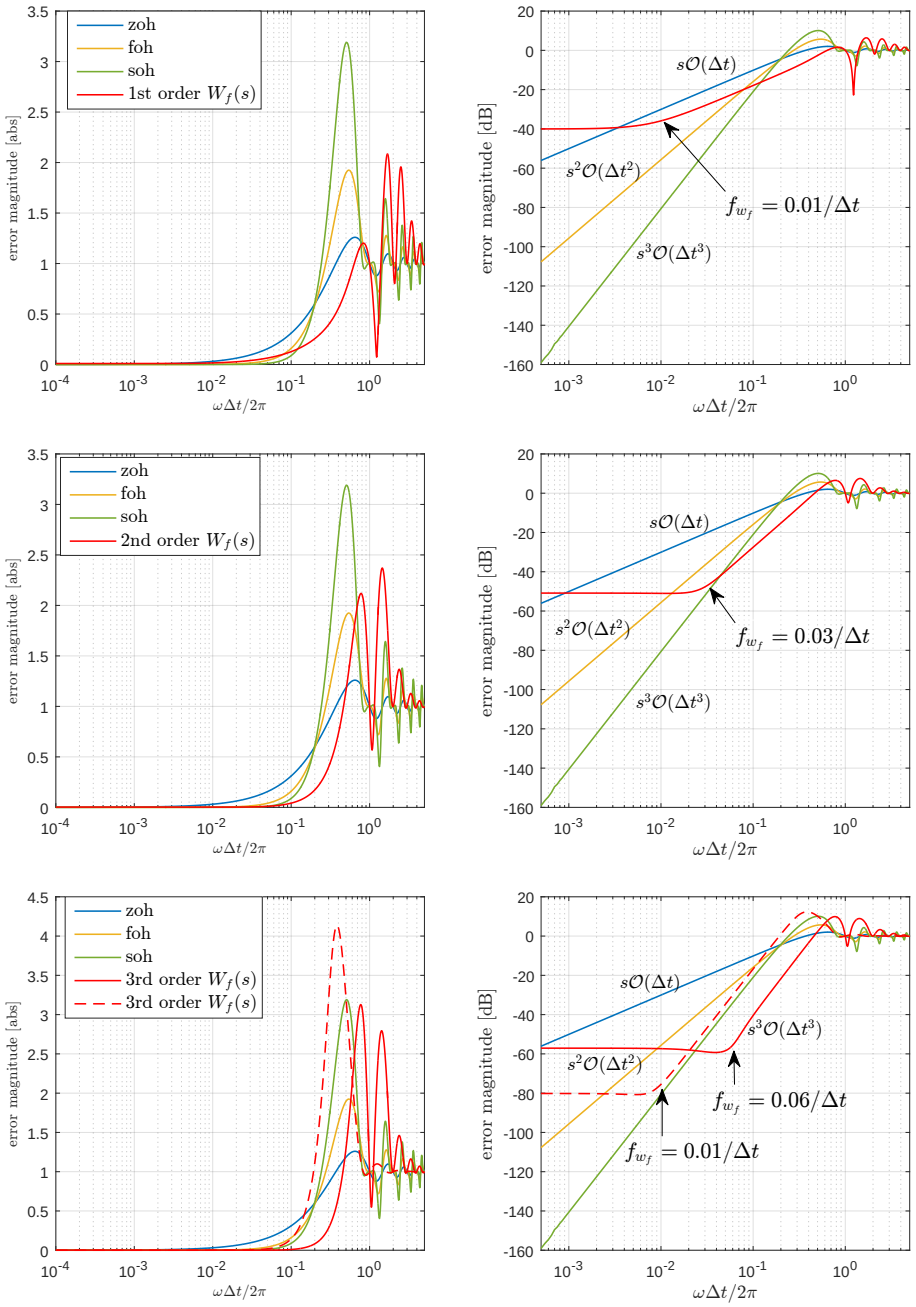


Figure 2.14: The H_∞ method can reduce the error in a wide frequency range. Δt is the macro-step, i,e., H. A higher-order $W_f(s)$ yields an error that converges faster, and the cut-off frequency of the penalty function f_{W_f} determines the balance between low-frequency and high-frequency precision.

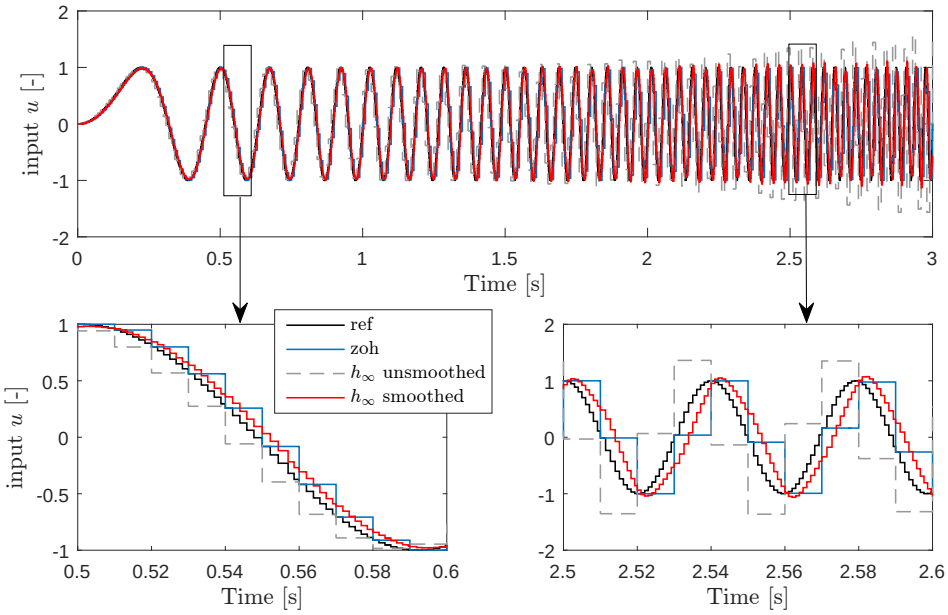


Figure 2.15: Coupling input by H_∞ method: as $K_2(s)$ smooths the input, $K_1(s)$ introduces a phase-lead to reduce the error.

2.3.2 Energy preservation

Thus far, co-simulation stability, accuracy and improving methods are discussed. However, it may be less interpretable and less exposed to the dynamics. In this regard, the energy conservation concept is worth mentioning [22]. As shown in Fig.2.16, two co-simulation slaves are coupled by two variables, e.g., force and velocity, forming power. Ideally, the energy transmitted between these two slaves must be preserved. This is true in monolithic simulation, i.e., $y^{[1]}(t) = u^{[2]}(t), y^{[2]}(t) = u^{[1]}(t)$. In contrast, in co-simulation, the power difference at instant t_{n+m} is given by [2]

$$\begin{aligned} \delta P &= \tilde{u}^{[1]}_{t_{n+m}} y^{[1]}_{t_{n+m}} - \tilde{u}^{[2]}_{t_{n+m}} y^{[2]}_{t_{n+m}} \\ &= u^{[1]}_{t_n} y^{[1]}_{t_{n+m}} - u^{[2]}_{t_n} y^{[2]}_{t_{n+m}} \qquad \text{(with ZOH method)} \end{aligned} \tag{2.22}$$

and it is possible that $\delta P \neq 0$. Then, the energy difference in $[t_n, t_{n+m}]$ is $\delta E = \int_{t_n}^{t_{n+m}} \delta P \, dt \approx \delta P H$ as an approximation [96, 89]. If intermediate results in

[2] Here $\tilde{\bullet}$ indicates that the variable is approximated by coupling methods. y without $\tilde{\bullet}$ means that the output is retrieved directly, instead of the error-free reference.

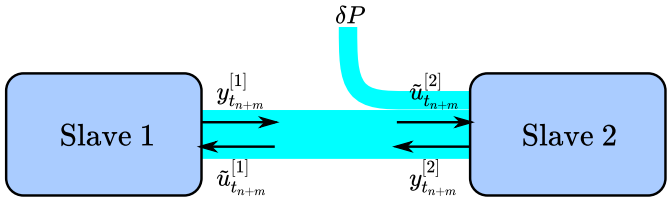


Figure 2.16: Energy transmitted between two coupled slaves in co-simulation. At instant t_{n+m}, inputs are approximated and outputs are retrieved.

$[t_n, t_{n+m}]$ are accessible, a more precise calculation is available.

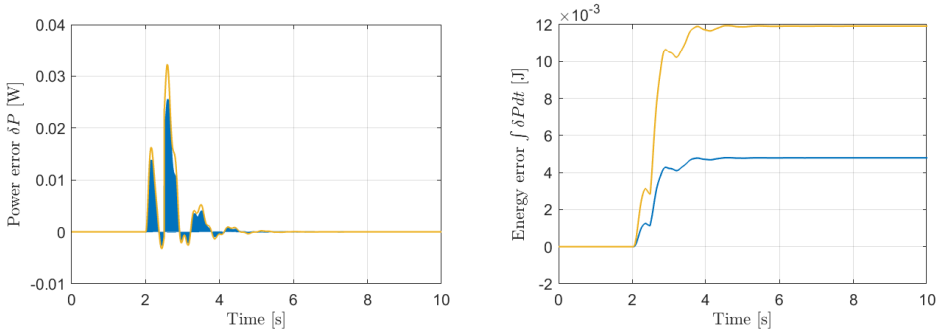
$$\delta E = \left(\begin{bmatrix} \tilde{u}^{[1]}_{t_n} \\ \vdots \\ \tilde{u}^{[1]}_{t_{n+m}} \end{bmatrix}^T \begin{bmatrix} y^{[1]}_{t_n} \\ \vdots \\ y^{[1]}_{t_{n+m}} \end{bmatrix} - \begin{bmatrix} \tilde{u}^{[2]}_{t_n} \\ \vdots \\ \tilde{u}^{[2]}_{t_{n+m}} \end{bmatrix}^T \begin{bmatrix} y^{[2]}_{t_n} \\ \vdots \\ y^{[2]}_{t_{n+m}} \end{bmatrix} \right) h \tag{2.23}$$

This observation is interesting because it introduces a physical interpretation to co-simulation: a portion of the energy may leak from or be added to the system owing to the coupling. Consequently, the dynamics is altered or even becomes unstable.

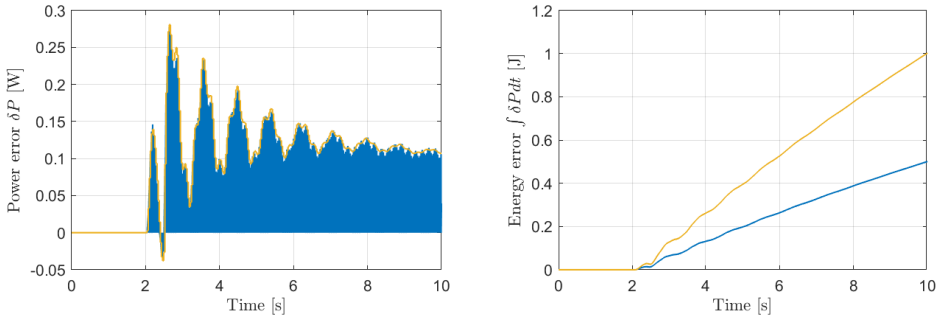
To well explain the idea, the previous co-simulation benchmark (Fig.2.5) is employed and an 1 N force impulse is applied to m_2 to excite the system. As demonstrated in Fig.2.17, the energy transmitted between two slaves is not preserved. When $H = 5$ ms, approximately 0.012 J and 0.0048 J, based on the rough and accurate estimations respectively, are incorrectly added to the system during the simulation. After the energy is dissipated by the damper, the system recovers to the initial state ($\dot{x}_1 = \dot{x}_2 = 0$). In comparison, when $H = 30$ ms, the incorrect energy owing to coupling is consistently added to the system. Without sufficient dissipation, the DMSD system keeps oscillating after the excitation. By monitoring the energy error, co-simulation accuracy can be checked. Furthermore, the incorrectly added energy increases with the macro-step size, which is demonstrated in Fig.2.18.

The energy concept is well-accepted, leading to the formulation of several new techniques e.g., using a normalized δE as an error indicator for a variable macro-step control [96, 97, 47], and using δE for correction [20]. The adjustments are based on δE and δP from the previous macro-step. With explicit approaches, a significant change between two consecutive steps unavoidably results in a drawback. Therefore, the energy can only be nearly conserved.

By continuing with this concept, $y(t)u(t)$ is found as generalised power and not only necessarily as physical power [22]. The product δP is similar to a blended



$H = 5$ ms.



$H = 30$ ms.

Figure 2.17: Residual power δP and accumulated energy error in the coupling of the co-simulation benchmark. Yellow and blue curves are based on the rough (Eq.(2.22)) and accurate (Eq.(2.23)) estimations respectively.

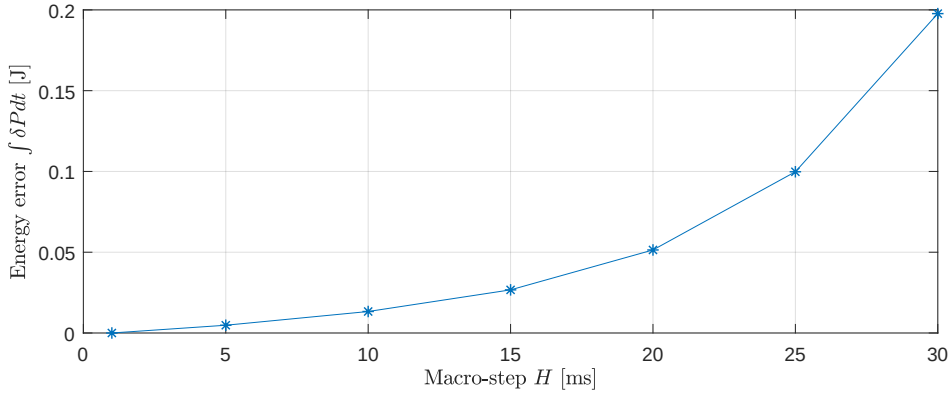


Figure 2.18: Accumulated energy error $\int \delta P dt$ based on the accurate estimation (Eq.(2.23)) in the simulation for 5 s with various macro-step.

error indicator and it yields the following

$$\begin{aligned}\delta P &= \tilde{u}^{[1]}_{t_{n+m}} y^{[1]}_{t_{n+m}} - \tilde{u}^{[2]}_{t_{n+m}} y^{[2]}_{t_{n+m}} \\ &= (u^{[1]}_{t_{n+m}} + \xi_{u^{[1]}_{t_{n+m}}}) y^{[1]}_{t_{n+m}} - (u^{[2]}_{t_{n+m}} + \xi_{u^{[2]}_{t_{n+m}}}) y^{[2]}_{t_{n+m}} \\ &= \xi_{u^{[1]}_{t_{n+m}}} y^{[1]}_{t_{n+m}} - \xi_{u^{[2]}_{t_{n+m}}} y^{[2]}_{t_{n+m}}\end{aligned} \tag{2.24}$$

Similarly, Eq.(2.23) can be written as

$$\delta E = (\xi^T_{\bar{\boldsymbol{u}}^{[1]}} \bar{\boldsymbol{y}}^{[1]} - \xi^T_{\bar{\boldsymbol{u}}^{[2]}} \bar{\boldsymbol{y}}^{[2]})h \tag{2.25}$$

where $\bar{\boldsymbol{y}}$ and $\xi_{\bar{\boldsymbol{u}}}$ are the output and input error vectors in the corresponding macro-step, respectively. It yields

$$\frac{||\delta E||_2}{h} = ||\xi^T_{\bar{\boldsymbol{u}}^{[1]}} \bar{\boldsymbol{y}}^{[1]} - \xi^T_{\bar{\boldsymbol{u}}^{[2]}} \bar{\boldsymbol{y}}^{[2]}||_2 \tag{2.26}$$

Minimising $||\delta E||_2$ may not imply that $||\xi_{\bar{\boldsymbol{u}}^{[1]}}||_2$ and $||\xi_{\bar{\boldsymbol{u}}^{[2]}}||_2$ are also minimised. Coupling variables may still become inconsistent. In specific co-simulation cases, although the energy error is small, incorrect coupling variables (e.g., position) are observed [48, 122]. To correctly apply the concept, each coupling variable needs to be separately fixed [20]. Therefore, to achieve accuracy, energy preservation may be more of a necessary condition rather than a sufficient condition. For sufficiency, additional variables must be monitored [48].

The energy concept is mentioned here because the H_∞ coupling method is relevant and justified in the physical sense. First, $||\xi_{\bar{\boldsymbol{u}}}||_2$ decreases through minimising the upper bound. Second, incorrect energy added to or leaking from each port is minimised as an outcome, because $||\xi^T_{\bar{\boldsymbol{u}}} \bar{\boldsymbol{y}}||_2 \leq ||\xi_{\bar{\boldsymbol{u}}}||_2 ||\bar{\boldsymbol{y}}||_2$ [3].

The energy concept enables co-simulation to be relatively interpretable. However, it may not reflect the weights of coupling variables. In other words, are the coupling variables equally important? Which variable is more critical? The answers to these questions are inferred by the architecture problem elaborated next.

2.4 Architecture problem

The architecture of co-simulation refers to how the slaves or subsystems are partitioned and coupled. This subject is discussed in this thesis, and some of the results are from **Papers II** and **III**.

A mechatronic system typically involves causal parts, such as the EPAS control model. The input-output relationship is evident and dependent on the formulation of the control algorithm. Acausal modelling tools may be preferred for complex mechanic parts, such as the vehicle suspension model in Adams/Car and the gear transmission model in Modelica. Acausal modelling, as a declarative and equation-based approach [41], is more extendable and reusable, because engineers can simply modify the graphic components without deriving low-level equations or presuming a causality. However, casual formats are necessary to run in co-simulation or on distributed processors. For example, directional interfaces must

[3] Cauchy-Schwarz Inequality.

be added to compile the model into the functional mock-up unit (FMU). This process must still be performed manually by engineers and properly implemented with adequate system knowledge. Process automation appears unachievable especially when different software tools are used in combination. Similar to the vehicle-rail-terrain example [80], engineers are uncertain whether the rail must be merged to the vehicle or terrain as well as how to select the causality (e.g., force and displacement direction). The same questions apply to the vehicle-EPAS system interaction in our work [32]. Accordingly, these problems are analysed.

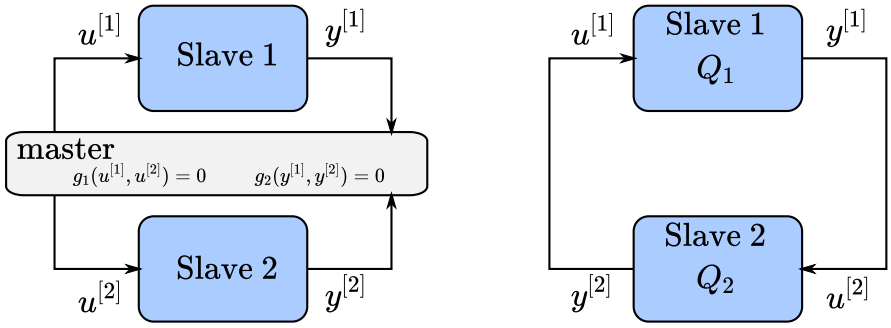


Figure 2.19: Architecture involves coupling of salves by constraint approach (left) and applied force approach (right).

For instance, when splitting a system into two slaves, the slaves may have algebraic dependency (Fig.2.19), e.g., coupling variables are the action and reaction forces or the displacement at the same point [102]. Instead of being regarded as inappropriate, the design is possible when the split is on a rigid body or kinematic constraint. Under this condition, the co-simulation master must implement iterations for every step H until the algebraic equations satisfy a certain tolerance [65, 101]. From the perspective of implementation, the causal conflict must be avoided if possible. Engineers may prefer to retain the constraint problem inside the slave and allow the condition to be enforced by the local solver.

This thesis focuses more on the architecture problem of the applied force approach that is solvable using explicit methods [104]. The causality of one slave is the inverse of the other and the input-output mapping is valid as Eq.(2.13).

2.4.1 Causality of slaves

The coupled salves (LTI) with applied force approach yield an overall system given by Eq.(2.14). Different causalities and splitting can change slave matrices, consequently changing the eigenvalues of the overall system. However, reading system matrices is impractical and complex. For the convenience of analysis, the slaves are assessed based on their transfer behaviours because the system clearly exhibits an input-output behaviour in Eq.(2.14).

By assuming a zero initial condition, the two LTI slaves (Fig.2.19) can be expressed as transfer functions $Q_1(s)$ and $Q_2(s)$ (where s denotes the Laplace domain). Then, the stability of the coupled system is governed by $\frac{1}{1-Q_1(s)Q_2(s)}$. Stability can be checked by the well-known Nyquist stability criterion. A sufficient and conservative stability condition is that $||Q_1(s)Q_2(s)||_\infty < 1$, i.e., the small-gain theorem [60]. The presumption is that slaves $Q_1(s)$ and $Q_2(s)$ are stable, which can be ensured by the local solver. Furthermore, a larger loop gain $||Q_1(s)Q_2(s)||$ generally indicates a less robustness. The following simple example demonstrates the difference resulting from causality.

$$\text{Causality I:} \quad Q_1 = \frac{1}{3}, \quad Q_2 = 2; \qquad \text{Causality II:} \quad Q_1 = 3, \quad Q_2 = \frac{1}{2} \tag{2.27}$$

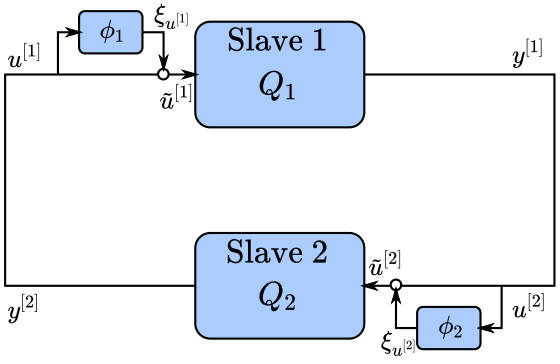


Figure 2.20: Block diagram of coupled salves: the input coupling error is modelled as a multiplicative disturbance.

The slaves present the same equation; however, their causalities are interchanged. A slight delay is allowed to break the algebraic loop. Causality I is asymptotically stable, whereas Causality II is unstable and the error is incrementally propagated inside the system.

A more accurate co-simulation system representation used in **Paper III** is shown in Fig.2.20. The input coupling error can be regarded as perturbations added to the nominal system (i.e., monolithic simulation). This enables the investigation of the property as a control problem. If a linear coupling method is used (e.g., ZOH, FOH and SOH methods), then the perturbations can be modelled in the Laplace domain. Otherwise, a certain error bound λ similar to that employed in the H_∞ coupling method may be used.

$$||\xi_u||_2 \leq ||\phi||_\infty ||u||_2 = \lambda ||u||_2 \tag{2.28}$$

The small-gain theorem provides a more precise stability condition as follows

$$||(1+\phi_1)Q_1(s)(1+\phi_2)Q_2(s)||_\infty < 1 \tag{2.29}$$

the foregoing is satisfied when

$$(1+\lambda)^2||Q_1(s)Q_2(s)||_\infty < 1 \tag{2.30}$$

where ϕ_1, ϕ_2 are assumed to have the same bound λ. This indicates that a larger error bound λ and macro-step are enabled by reducing the system loop gain. However, ξ_u continuously increases at higher frequencies (Fig.2.8). Thus, the error bound λ in Eq.(2.30) can only be estimated at a certain frequency range as discussed.

2.4.2 Sensitivity of coupling variables

How severe is the effect of a coupling error to the co-simulation system? This problem is associated with the sensitivity of error disturbance to the system (Fig.2.20). The propagation paths from the coupling error to slave outputs are as follows.

$$\begin{aligned} \xi_{u^{[1]}} \to y^{[1]}: \quad & \frac{Q_1}{1+L} \qquad \xi_{u^{[1]}} \to y^{[2]}: \quad \frac{(1+\phi_2)Q_1Q_2}{1+L} \\ \xi_{u^{[2]}} \to y^{[2]}: \quad & \frac{Q_2}{1+L} \qquad \xi_{u^{[2]}} \to y^{[1]}: \quad \frac{(1+\phi_1)Q_1Q_2}{1+L} \\ L &\coloneqq -(1+\phi_1)(1+\phi_2)Q_1Q_2 \end{aligned} \tag{2.31}$$

The functions describe the sensitivity of perturbations. Analytically, if $Q_1 > Q_2$ in the frequency range of interest, the corresponding input error $\xi_{u^{[1]}}$ is more critical than $\xi_{u^{[2]}}$. Without a detailed knowledge of the system, the sensitivity of the port may be checked based on the variation in results with two macro-step sizes because such variation increases as Eq.(2.31).

In the physical sense, the input is more sensitive because the perturbation (i.e., coupling error) is exposed to a part that can easily be excited (e.g., Q_1). Hence, reducing the error in the corresponding input may be prioritised. In this case, coupling methods that employ more recent data in finer steps are suitable. By contrast, the effectiveness of improving less sensitive input may not be considerable.

2.5 Discussions

We started from conventional monolithic simulation; then, the formulation of a co-simulation system is mathematically derived. The accuracy and instability problems can potentially degrade the *consistency* of a virtual prototype. Accordingly, the means for preserving *consistency* in two directions is investigated: coupling methods and system architecture.

The main points of the chapter and work results are as follows

1 Input approximation distorts co-simulation results, which can be improved using coupling methods. The usability of methods relies on the capability of the software; however, the effects of case dependency and interpretability must be concerned.

2 Coupling methods with linear operations have satisfactory usability and exhibit a clear frequency behaviour. Within these approaches, the H_∞ coupling method, which can be optimised to achieve certain objectives on the error bound and frequency behaviour, is developed.

3 Coupling variables are dynamics-dependent or frequency-dependent. Therefore, choosing the optimal coupling method in different cases is difficult. Hence, the coupling method has to be robust, and the sensitivity of coupling variable must be identified.

4 In the first place, to avoid a problematic architecture, the distribution within the system must be appropriately arranged by engineers. The architecture can be designed considering the system loop gain. To achieve this, system-level input-output information is necessary.

5 In terms of the work methodology, co-simulation is interpreted as a control system and analysed using a linear robust control framework. Co-simulation stability can be analytically determined based on system eigenvalues. However, to the author´s knowledge, eigenvalues do not imply anything about robust stability. In view of this, the well-established linear robust control theory is applied to the research.

Finally, this chapter ends the discussion on co-simulation at this point; however, the treatment of this subject is incomplete. Although the most general explicit-parallel scheme with equidistant macro-steps is considered, certain problems and research gaps remain unresolved. For example, the relationship given by Eq.(2.31) may not be evident among multiple input variables that are closely coupled. Hence, more work is necessary to improve the interpretability and implementability of co-simulation.

Chapter 3

Hardware-in-the-loop simulation

This chapter introduces the fundamentals of HIL simulation. The technical challenges and factors that may influence the system dynamics are discussed. To improve the performance, state-of-the-art algorithms are discussed and a novel approach is presented in this work.

3.1 Technical challenges

In the HIL simulation, a real testing device is connected to a virtual testing environment. Although this simulation is considered as a more reliable tool than computer simulation, it becomes more complex when more components are included. As an interdisciplinary subject, it is investigated and widely applied in diverse fields, e.g., computer science, embedded system, control engineering and automotive engineering (application area). To better identify the HIL simulation problem and the focus of this work, technical challenges are clustered into three basic ingredients (Fig.3.1).

1 Digital real-time simulator [1] (DRTS): the machine simulating the virtual part of the system.

2 Hardware device under test (DUT): the real testing subject.

3 HIL interface: the part connecting the DRTS and DUT.

DRTS

The model and numerical method must be sufficiently accurate and can be implemented in real time in the DRTS. On the modelling side, an RT vehicle model

[1] A simulator is loosely defined as a machine that imitates a real-word process. It does not necessarily refer to an entire HIL system (e.g, the driving simulator).

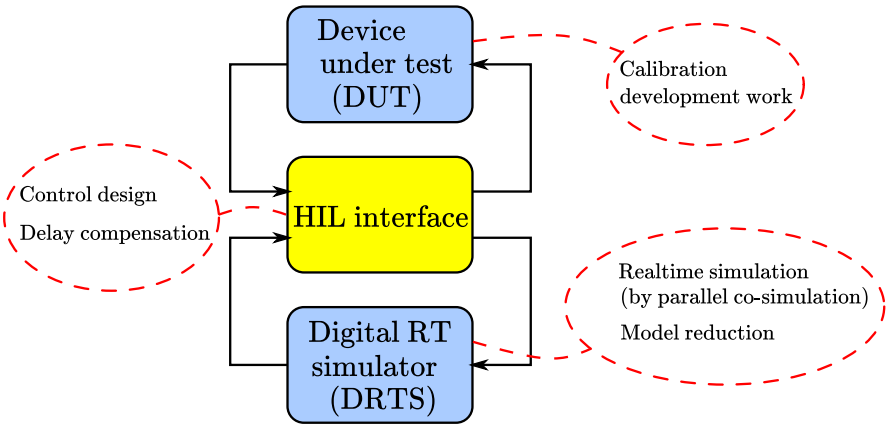


Figure 3.1: Main ingredients of a general HIL simulation and interdisciplinary challenges.

is generally simplified to reduce the computational burden. For example, MBD suspension linkages are replaced with lookup tables, or a more efficient coordinate formulation is used [105]. Moreover, model reduction techniques can be employed for a particular application [37]. In terms of the solver, fixed-step methods are used for a predictable computational time [121]. Typically, the Euler method is employed. This requires that the model be solvable with a reasonable time step, which might be challenging for stiff systems. To avoid this, the stiff system is considered as a real DUT, e.g., HIL simulation of the hydraulic circuit and brake disk [7].

On the computer side, the DRTS features the partitioning of a large system model and parallel computation on distributed processors. The computational power relies on processors and parallelisation. In the first aspect, commercial central processing units (CPUs) are insufficient to simulate fast dynamics, especially the electric system, due to the low update frequency [91]. The field programmable gate array (FPGA) supports much higher update frequency and becomes a standard component for commercial RT computers (e.g., RT-LAB and NI-PXI). In the second aspect, computation power is expandable by taking parallelisation with more processors. In this regard, as discussed, explicit parallel co-simulation is essential. Furthermore, computer scheduling (i.e., the process to assigning parallel tasks to parallel threads) is investigated in distributed simulation [28, 98]. An excellent scheduler can minimise the parallelisation overhead and accelerate the simulation speed.

DUT

The DUT varies with a particular application. According to the physical domain, HIL simulation can be categorised into signal hardware-in-the-loop (SHIL), power hardware-in-the-loop (PHIL) and mechanical hardware-in-the-loop (MHIL) sim-

ulations [27]. In the SHIL simulation, the DUT can be an embedded control unit that connects to the DRTS with control signals. In the PHIL simulation, the DUT can be an actual power devices and connects to the DRTS through voltage and current. In the MHIL simulation, the interface is established by a mechanical system coupled by real forces/torques and motions. For example, the DUT is an engine in the engine-in-the-loop simulation [40]. In the driving simulator case (Fig.3.2), the human driver is the DUT to test the subjective feeling.

The challenges related to the DUT mostly involve the development work on the holistic system, and are not regarded as the general problem of the simulation tool.

HIL interface

The HIL interface is the most distinct part of the HIL simulation. More specifically, it includes actuators, sensors, communication network, and the mechanisms connecting the DUT and DRTS. The objective of the interface is to seamlessly produce the boundary conditions between DUT and DRTS as though they are directly coupled.

In automotive engineering, the technical gap of MHIL simulation is usually associated with the HIL interface; this requires considerable research and implementation efforts. In the engine-in-the-loop simulation, substantial work is devoted to the dynamometer rig control to faithfully produce the load torque and maintain motion [29, 39]. Another advanced application is the driving simulator in which the moving platform establishes the HIL interface (Fig.3.2). The dynamic performance in the old flight simulator is limited to a system intended only for pilot training. Nowadays, the platform is more advanced that it is capable of tracking the simulated vehicle motion. In addition to powerful actuators, advanced motion cueing algorithms (e.g., the model predictive control [17]) are employed to achieve optimality under power and motion space constrains. Consequently, the interface credibility is improved.

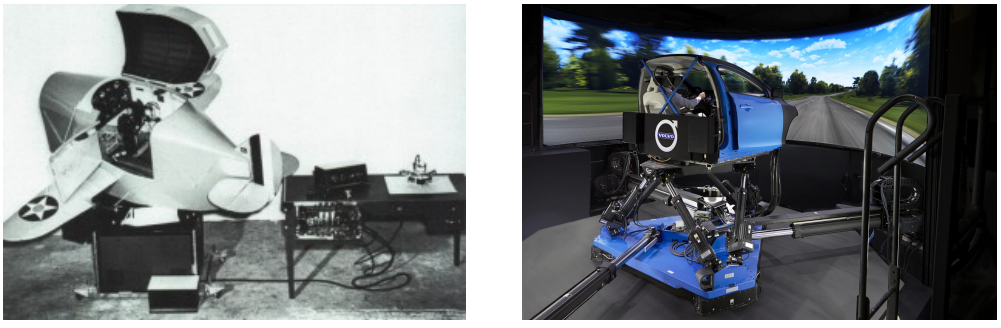


Figure 3.2: Two HIL systems: Link trainer for pilot training during 1940s [34] and driving simulators at Volvo Cars for chassis tuning in 2020s.

In summary, the HIL simulation is not a new approach. Its evolution is primarily driven by improving the HIL interface, which is the main interest of this work.

3.2 Mathematical description

This section mathematically describes the HIL simulation. Accuracy and stability properties are discussed using the benchmark DMSD system as an example (Fig.3.3). The mass-spring-damper (MSD) on the right is simulated by DRTS, and the remaining part is in real mechanics (DUT). The HIL interface consists of a velocity-controlled actuator and force sensor. The actuator attempts to track the reference velocity $\dot{x}_{2,drts}$ and the resulting force $F_{c,drts}$ is sent back to the DRTS.

The causality between the DUT and interface is less clear because they are coupled physically and not by the signal flow. To describe the DUT analytically, a causality has to be assumed, either with a force input or velocity input. Both are mathematically correct. However, when the DUT has the same causality as the DRTS, the interface is an equivalent of the boundary conditions: $F_{c,dut} = F_{c,drts}, \dot{x}_{2,dut} = \dot{x}_{2,drts}$. This is similar to an algebraic loop in co-simulation that must be numerically solved through iterations. To avoid this problem, the DUT is described with an inverse causality as that in [15, 69]. Then, the interface just relates the inputs to the outputs. This allows it to have a familiar block diagram description (Fig.3.4). In essence, it can be analysed as a control system [39].

The block diagram expression of HIL simulation is illustrated in Fig.3.4. In the example case, the DRTS and DUT are as follows

$$\begin{aligned} G_{em} &= \frac{x_2 s}{F_c} = \frac{s}{m_2 s^2 + d_2 s + k_2} \\ G_{dut} &= \frac{F_c}{x_2 s} = -\frac{1}{s\left(\frac{1}{d_c s + k_c} + \frac{1}{m_1 s^2 + d_1 s + k_1}\right)} \end{aligned} \tag{3.1}$$

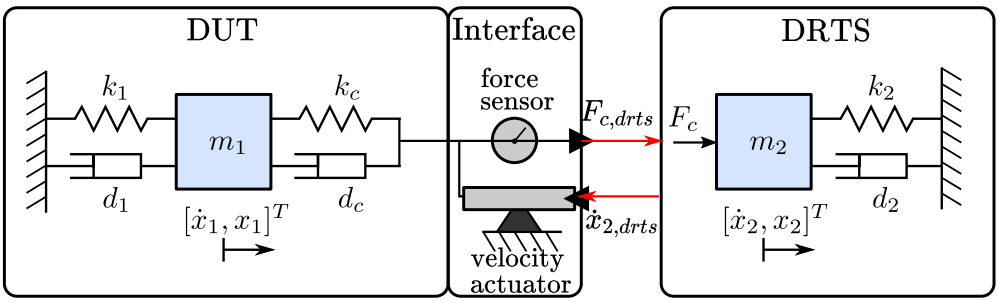


Figure 3.3: Conceptual layout of HIL simulation of DMSD system. HIL interface is a linear actuator mechanism.

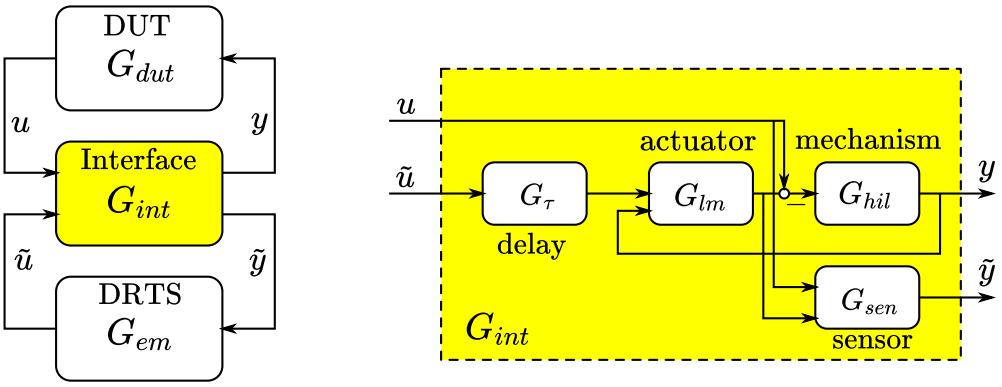


Figure 3.4: General HIL interface with detailed view.

The interface is as follows

$$\begin{bmatrix} y \\ \tilde{y} \end{bmatrix} = G_{int} \begin{bmatrix} u \\ \tilde{u} \end{bmatrix} \implies \begin{bmatrix} \dot{x}_{2,dut} \\ F_{c,drts} \end{bmatrix} = G_{int} \begin{bmatrix} F_{c,dut} \\ \dot{x}_{2,drts} \end{bmatrix} \tag{3.2}$$

Accuracy of HIL simulation is considered next.

3.2.1 Interface transparency

The term *transparency* originates from teleoperation robotics [70]. It is introduced in the HIL simulation to describe the accuracy of the interface [15]. An ideal transparent interface yields the following.

$$G_{int,ideal} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \tag{3.3}$$

The ideal interface does not introduce any distortion and the boundary conditions are equal. In this case, the remaining causes of discrepancy are the DUT and DRTS. With a high-level description, *transparency* is measured as

$$||G_{int} - G_{int,ideal}||_\infty \tag{3.4}$$

which is the infinity norm of the discrepancy transfer matrix proposed in [90, 72]. It is also used when resolving the co-simulation coupling problem in this work. Moreover, G_{int} has an appropriate dimension, which increases with more measurements and actuators. In the driving simulator case, the moving platform tracks the vehicle model motion in six degrees of freedom.

The interface of the example case can be divided into several detailed components (Fig.3.4):

- G_τ refers to the lumped delay effect, which is represented by $G_\tau = e^{-s\tau}$, where τ is the round trip time [36, 109]. A series of delay contributors includes the signal transmission delay, DRTS computation delay and digital-analogue transformation delay [113, 33, 51]. In practice, τ is time-variant instead of a constant.

- G_{lm} is the actuator control (e.g., velocity-control mode). The two inputs are the reference and actual velocity, the control law is

$$F_{actuator} = f_{G_{lm}}(\dot{x}_{2,drts}, \dot{x}_{2,dut}) \tag{3.5}$$

  For a common proportional–integral–derivative (PID) control $G_{lm} = P_{gain} + I_{gain}/s + D_{gain}s$.

- G_{hil} is the overall mechanical load (i.e., mass, stiffness and damping effects of the attached hardware). The linear actuator case is as follows.

$$m_h\ddot{x}_{2,dut} + d_h\dot{x}_{2,dut} = F_{c,dut} - F_{actuator} \implies G_{hil} = \frac{1}{m_h s + d_h} \tag{3.6}$$

- G_{sen} represents the sensor dynamics. First, it accounts for sensor noise and filtering dynamics. Second, it considers the non-collocation of the sensor and the DUT that can degrade the system performance [4]. In practise, the sensor may be mounted on a rigid body between the DUT and actuator. The real measurement is thus a combination of the acting and counteracting forces.

$$F_{measured} = \alpha F_{c,dut} + (1-\alpha)F_{actuator} \tag{3.7}$$

  where α is dependent on the ratio of mass (inertia) on two sides. Under the steady state condition, it is true that $F_{measured} = F_{c,dut} = F_{actuator}$. Under the transient condition ($F_{c,dut} \neq F_{actuator}$), $F_{measured}$ differs from $F_{c,dut}$ with an intermediate hardware effect.

With these detailed components, a component-level description the interface is as follows

$$G_{int} = \frac{1}{1 - G_{lm}G_{hil}} \begin{bmatrix} G_{hil} & -G_\tau G_{lm} G_{hil} \\ \alpha - G_{lm}G_{hil} & G_\tau G_{lm}(1-\alpha) \end{bmatrix} \tag{3.8}$$

To demonstrate their influence on the interface *transparency*, a parameter study is conducted using the example case. The interface components are specified in Table 3.1, and are scaled with factors 0.1 and 10. Individually, $\alpha \in [0.1, 1]$. The interface *transparency* is implied by the singular values (i.e., σ_1 and σ_2) of the transfer matrix (Eq.(3.4)).

Table 3.1: Parameters of the HIL interface components

Specifications	Definitions
$G_{lm} = 10 + 10\frac{1}{s}$	PI control of the actuator
$G_{tau} = e^{-0.02s}$	interface delay $\tau = 0.02$ s
$G_{hil} = \frac{1}{0.1s+0.1}$	mechanical parts $m_h = 0.1$kg, $d_h = 0.1$Ns/m
$\alpha = 0.7$	It varies with the inertia ratio and thus the location.

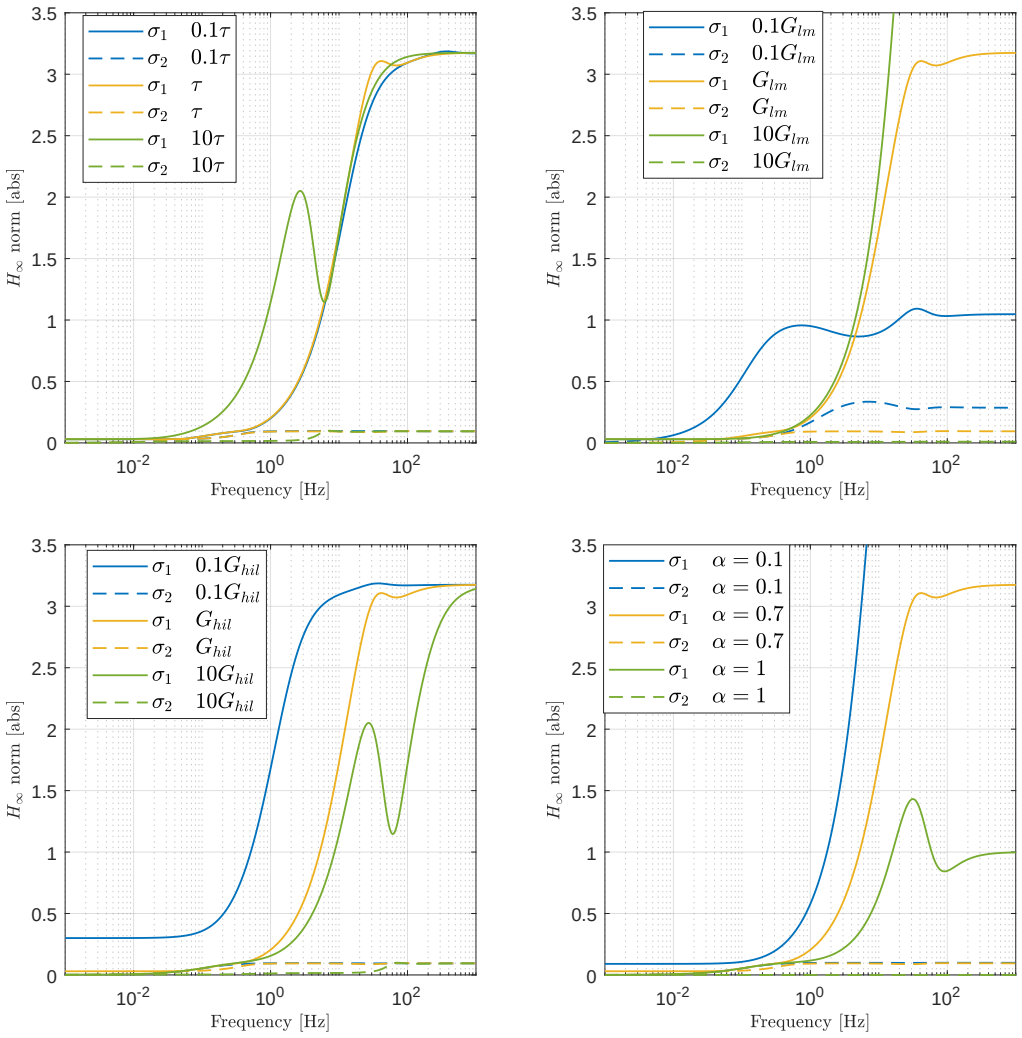


Figure 3.5: Variation of *transparency* expressed by singular values $\sigma\Big(G_{int}(j\omega) - G_{int,ideal}(j\omega)\Big)$ due to different interface components in frequency domain.

As anticipated, the *transparency* deteriorates at a higher frequency and slightly worsens with a larger delay τ (Fig.3.5). The HIL interface consists an inner control loop by $G_{lm}G_{hil}$ in Eq.(3.8), and *transparency* in the low frequency range is improved with a higher inner loop tracking performance (i.e., larger G_{lm} and G_{hil}). Meanwhile, the interface robustness decreases as reflected by the maximum singular values in higher frequency. Then, the inner loop tracking performance is constrained by the level of noises that occur in the real system. Furthermore, *transparency* improves with better sensor collocation ($\alpha = 1$). The measurement is more accurate when the sensor is closer to the DUT. Otherwise, to accomplish realistic results, compensation for the attached hardware must be implemented [61]. This cannot be ignored in the motor-in-the-loop simulation; hence it is considered in **Paper V**.

Similarly, optimal component designs can be obtained by minimising a weighted $||G_{int} - G_{int,ideal}||_\infty$ if the system disturbances (e.g., noises and delay) are explicitly given. Without the disturbance information, the optimisation only confirms the design choices from the parameter study.

To achieve satisfactory *transparency*, compensating for transmission delay τ is implemented using adaptive filters (e.g., extended Kalman filters (EKF) and recursive least square (RLS) algorithms)[110, 109], as discussed in **Paper IV**. However, due to the non-constant τ, system robustness remains necessary to deal with this uncertainty. On the controller side, simply increasing the gains of the actuator controller eventually results in vibrations or instability. An undesirable situation of the HIL simulation is that the vibration or instability problem occurs before achieving sufficient *transparency*. Thus, the HIL simulation stability is analysed next. It is worth noting that *transparency* is an open-loop property of the interface [90]. It does not guarantee the stability of HIL simulation. In fact, it may even counter the system robustness as discussed.

3.2.2 HIL simulation stability

The benchmark problem is employed to analytically demonstrate the HIL system stability. An external input $\begin{bmatrix} y_e & \tilde{y}_e \end{bmatrix}^T$ is added to excite the DUT and DRTS. Combining Eqs.(3.2) and (3.8) yields the following

$$\begin{aligned}
\begin{bmatrix} y \\ \tilde{y} \end{bmatrix} &= G_{int} \begin{bmatrix} G_{dut}(y + y_e) \\ G_{em}(\tilde{y} + \tilde{y}_e) \end{bmatrix} \\
&= \underbrace{\frac{1}{1 - G_{lm}G_{hil}} \begin{bmatrix} G_{hil}G_{dut} & -G_\tau G_{lm}G_{hil}G_{em} \\ (\alpha - G_{lm}G_{hil})G_{dut} & G_\tau G_{lm}(1-\alpha)G_{em} \end{bmatrix}}_{\bar{G}} \begin{bmatrix} y + y_e \\ \tilde{y} + \tilde{y}_e \end{bmatrix}
\end{aligned} \tag{3.9}$$

The external input $\begin{bmatrix} y_e & \tilde{y}_e \end{bmatrix}^T$ produces $\begin{bmatrix} y & \tilde{y} \end{bmatrix}^T$ by the dynamic relationship $-I_2/(I_2 - \bar{G}^{-1})$ [2]. The ideal HIL system with a fully transparent interface yields

$$\begin{bmatrix} y \\ \tilde{y} \end{bmatrix} = \frac{1}{1 - G_{em}G_{dut}} \begin{bmatrix} G_{em}G_{dut} & G_{em} \\ G_{dut} & G_{em}G_{dut} \end{bmatrix} \begin{bmatrix} y_e \\ \tilde{y}_e \end{bmatrix} \tag{3.10}$$

The system's robustness depends on loop gain $G_{em}G_{dut}$. This is similar to the coupled simulation problem in Eq.(2.30). By assuming $\alpha = 1$, the loop gain of the non-ideal HIL system (Eq.(3.9)) becomes

$$H_{loop} = G_{em}G_{dut}G_{\tau}\frac{G_{lm}G_{hil}}{1 - G_{lm}G_{hil}} \tag{3.11}$$

This implies that the inner and outer loops are governed by $G_{lm}G_{hil}$ and $G_{em}G_{dut}G_{\tau}$ respectively. The HIL simulation stability can be checked analytically using the Nyquist criterion [60]. Moreover, the HIL simulation can be stable even with unstable DUT and DRTS if the Nyquist stability criterion is satisfied [3]. A straightforward example is that the DRTS is a virtual inverted pendulum and the DUT is a stabilizing control device and vice versa. According to the small-gain theorem [60, 49], when each part is open-loop stable, the HIL simulation is stable if $||H_{loop}||_{\infty} < 1$. The foregoing is a sufficient and conservative condition and considerably useful in this work.

By comparing the non-ideal case to the reference (Eq.(3.10)), the discrepancy from the ideal interface is found to be given by $G_{em}G_{dut}(1 - G_{\tau}\frac{G_{lm}G_{hil}}{1-G_{lm}G_{hil}})$. As discussed in the section on *transparency*, increasing tracking $G_{lm}G_{hil}$ may marginally reduce the tracking error. This and other perturbations in the outer loop (e.g., transmission delay τ, numerical error in DRTS) are impossible to totally eliminate, and they propagate in the outer loop. If the outer loop is less robust, the perturbations slowly diminish and considerably distort the result. However, in a built HIL system, the DUT G_{dut} is determined by the chosen hardware device, and G_{em} is implemented to simulate the remaining part. This means that the HIL simulation architecture and its stability (Eq.(3.11)) seems unfortunately fixed consequently. However, that is not the case and some methods are elaborated in the next section.

To evaluate the closed-loop accuracy of the HIL system, the zeros and poles including the stability information can be checked [123]. However, this may be tedious when the DUT and DRTS are complex, especially when new poles and zeros are introduced by the interface. *Transparency* is a more practical accuracy indicator when the HIL simulation is known to be stable. In practise, a precise

[2] I_2 refers to the 2×2 identify matrix.

[3] The system is stable, if the number of times $H_{loop}(s)$ encircles the $-1 + j0$ point in an anti-clockwise direction is equal to the number of unstable poles in $H_{loop}(s)$.

stability analysis may be difficult without the full knowledge of DUT. In its implementation, enhancing the stability until the interface achieves high *transparency* is preferable. Therefore, interface algorithms are employed.

3.3 HIL interface algorithms

HIL interface algorithms (IAs) are well-established and mostly implemented in the PHIL simulation for electric engineering applications [92, 83, 107]. However, IAs are less exposed to MHIL simulations and their usability varies in particular cases [114]. Considering the analogy between mechanical and electrical systems, we have generalised their usage and applied to the MHIL simulation. In this section, IAs are interpreted with mechanical systems. The same principle of damping impedance method (DIM) is applied to the MHIL simulation presented in **Paper V**.

For consistency, DUT and DRTS are expressed by impedance Z_{dut}, Z_{em}. This terminology from electric engineering and robotics is widely used in HIL simulations. It indicates the transfer behaviour with a specified causality. For example, the electrical and mechanical impedances are as follows

$$Z_{electric} = \frac{\text{voltage } V}{\text{current } I}, \quad Z_{mechanical} = \frac{\text{force } F}{\text{velocity } v} \tag{3.12}$$

Then, $Z_{em} = G_{em}^{-1}, Z_{dut} = -G_{dut}$ [4] in the example case (Eq.(3.1)).

3.3.1 Ideal transformer model

Given that the interface inner loop is stable, the stability of the HIL system is governed by the outer loop transfer function

$$H_{loop} = G_{em} G_{dut} e^{-s\tau} \tag{3.13}$$

where τ is the lumped time delay. According to the causality, the following is obtained.

$$\begin{aligned} H_{loop,ITM} &= -\frac{Z_{dut}}{Z_{em}} e^{-s\tau} \quad \text{(force to the DRTS)} \\ \text{or} \quad H_{loop,ITM} &= -\frac{Z_{em}}{Z_{dut}} e^{-s\tau} \quad \text{(velocity to the DRTS)} \end{aligned} \tag{3.14}$$

The causality determines whether the actuator is controlled by velocity or force. This is equivalent to the current-type and voltage-type interfaces in the PHIL simulation [67, 83].

[4] The sign change makes the impedance Z_{dut} positive. Z_{dut} reflects the relationship between the flow (velocity) and the effort (acting force).

Algorithm given by Eq.(3.14) refers to the ideal transformer model (ITM) in [83, 53], which preserves the power during the transformation of voltage and current [41]. ITM algorithm is the original setup in HIL simulation, which is simple and straightforward. The ITM algorithm implies that the stability highly depends on the impedance ratio similar to co-simulation. At the component-level, $H_{loop,ITM}$ is governed by the mass ratio between the DUT and DRTS, as discussed in **Paper I**.

3.3.2 Damping impedance method

Damping impedance method (DIM) algorithm is employed to add an impedance Z_{comp} in the HIL system [83]. The benchmark case in which velocity is introduced to the DRTS is considered without loss of generality. The resulting new loop transfer function is

$$H_{loop,DIM} = -\frac{Z_{dut} - Z_{comp}}{Z_{em} + Z_{comp}} e^{-s\tau} \tag{3.15}$$

where Z_{comp} can be regarded as a mass for instance (i.e., indicating that a part of the mass is shifted from the DUT to the DRTS). Comparing with the ITM algorithm, DIM algorithm yields the following if Z_{comp} is specified as a portion of the positive impedance Z_{dut}.

$$H_{loop,DIM} < H_{loop,ITM} \tag{3.16}$$

Therefore, the DIM method can enhance stability. According to the small-gain theorem, if Z_{comp} is specified to cause $\|H_{loop,DIM}\|_\infty < 1$, then stability is ensured. The optimal design in terms of robustness is achieved when $Z_{comp} = Z_{dut}$ because a minimum loop gain is achieved (i.e., $H_{loop,DIM} = 0$). However, this is unusual because the DUT is similar to all those simulated by the DRTS.

How can the DIM algorithm be applied to the MHIL simulation? Change the impedance of DUT by cutting the hardware is certainly unrealistic. Consider the previous example, our realisation is illustrated in Fig.3.6. On the DRTS side, mass m^* is added to the MSD model. To achieve equivalent dynamics, a pseudo force F^* is applied

$$\begin{aligned} F_c + F^* &= (m^* + m_2)\ddot{x}_2 + d_2\dot{x}_2 + k_2 x_2 \\ F^* &= m^*\ddot{x}_2 \end{aligned} \tag{3.17}$$

The pseudo force F^* is derived from the actual $\dot{x}_{2,dut}$ signal and added to the input $F_{c,drts}$ (denoted by the red block). Analytically, compensation for the additional impedance $Z_{comp} = m^*s$ is introduced and overall dynamics is unchanged. However, the resulting block diagram shows that the velocity interface error ($\xi_{x_2} = \dot{x}_{2,dut} - \dot{x}_{2,drts}$) propagates in a modified path. The error first goes

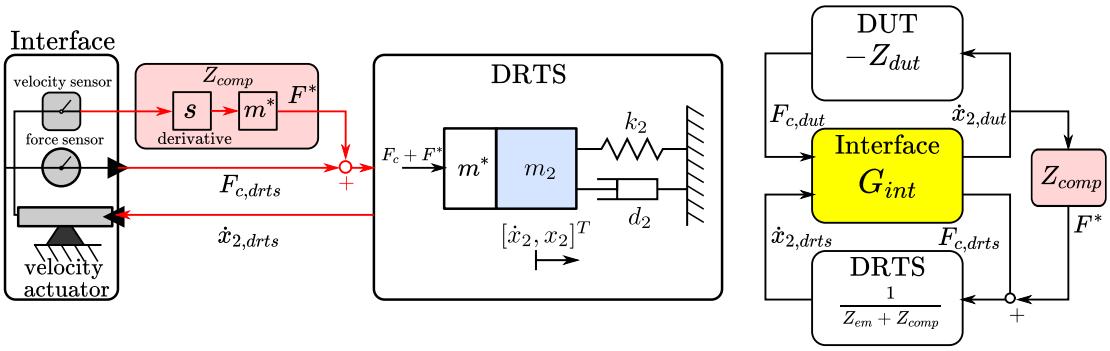


Figure 3.6: Implementation of DIM algorithm on MHIL simulation (left) and resulting block diagram with DIM algorithm (right). The DUT is unchanged and thus not plotted.

through the components $-Z_{dut}$ and Z_{comp} in parallel and then through the DRTS ($\frac{1}{Z_{em}+Z_{comp}}$), yielding a loop gain that is the same as $H_{loop,DIM}$ in Eq.(3.15).

In the PHIL simulation [53], a physical power impedance is inserted. However, F^* is implemented digitally in our implementation. Hence, no hardware and analogue components are added. However, a low-pass filter is necessary in Z_{comp} to avoid sensor noise, particularly when the derivative of the measured velocity $\dot{x}_{2,dut}$ is obtained.

Certain problems are encountered regarding the successful implementation of DIM algorithm. There are multiple choices to obtain F^* from $\dot{x}_2$ in Eq.(3.17). First, F^* can be modelled directly in the DRTS. However, Eq.(3.17) is computed as a whole. Then, the pseudo force is eliminated internally and nothing is changed. Second, Z_{comp} can be implemented in a distributed manner with the signal $\dot{x}_{2,drts}$ from the DRTS or directly $\ddot{x}_{2,drts}$ to avoid derivation. Nevertheless, perturbations from the interface continue to propagate in the same path as the ITM algorithm. Thus, system robustness remains unchanged. Therefore, the implementation of F^* is critical in the realisation. The result of the DIM algorithm is compared in a later section.

3.3.3 Comparisons

To compare the IAs, a simulation study is conducted using the benchmark. Herein, the parameters of the DMSD system and interface are same as that in Table.3.1. In the DIM algorithm, $Z_{comp} = 0.8Z_{dut}$ is specified.

To mimic the HIL simulation, the DRTS is discretised by 1 ms using the Euler method. The remaining parts are simulated with a variable step ODE15 solver with a relative error tolerance $\xi_{tolerance} = 1e-7$. The noise effect is not considered. A first-order Butterworth filter with cut-off frequency $f_c = 20$ Hz is added to Z_{comp}. The result of monolithic simulation is taken as the ground truth.

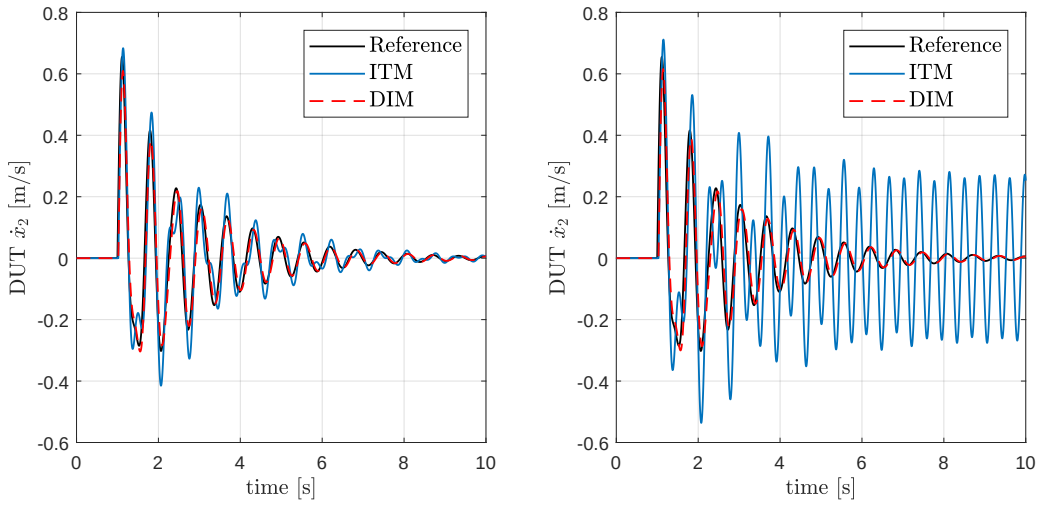


Figure 3.7: Comparison of resulting DUT velocities (Left: $\tau = 20$ ms , right: $\tau = 30$ ms).

A step force input of 1 Nm is applied at 1 s on m_2 to excite the system.

The resulting DUT velocity is plotted in Fig.3.7. Comparing with the ITM algorithm (i.e., the basic HIL simulation), the DIM algorithm is more consistent as reference. Furthermore, the velocity discrepancy is checked by

$$\xi_{\dot{x}_2} = \|\dot{x}_{dut} - \dot{x}_{drts}\|_2 \tag{3.18}$$

and $\xi_{\dot{x}_2,\mathrm{ITM}} = 3.7431, \xi_{\dot{x}_2,\mathrm{DIM}} = 3.0641$. This implies that although the inner loop tracking performance levels are equal using the same controller, the robustness in the outer loop contributes to the agreement of boundary conditions.

The ITM algorithm becomes marginally stable when $\tau \approx 30$ ms. If the delay τ increases to 1 s, the DIM algorithm is consistently stable because the resulting $\|H_{loop,DIM}\|_\infty = 0.67 < 1$ (Fig.3.8). This reveals that the instability and inaccuracy problems are decoupled, rendering the implementation easier. When measures for improving *transparency* (e.g., delay compensation) are tuned, stability can be ensured by the IAs.

The DIM algorithm, with the reference velocity $\dot{x}_{2,drts}$ as input, is also implemented. However, it is not as effective as expected. The interface algorithms are also implemented on a real HIL simulation. The result is presented in **Chapter V** and **Paper V**. In the experimental case, the interface is an electric dynamometer rig, and Z_{dut} and Z_{comp} are mainly the inertia effects. The DIM algorithm is found to be relatively effective for improving the *consistency* of results.

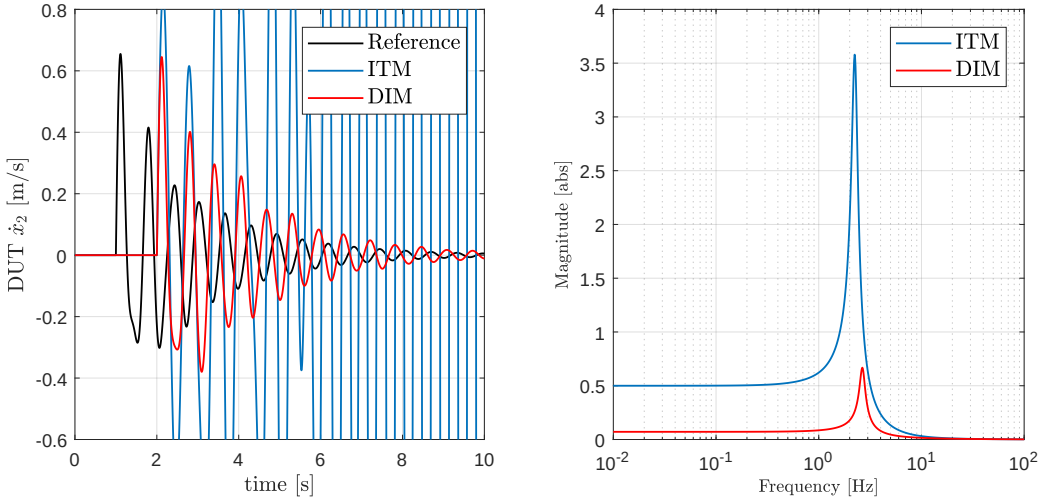


Figure 3.8: Resulting DUT velocities when $\tau = 1$ s. Using DIM algorithm, the result is delayed; however, stability is ensured because $\|H_{loop,DIM}\|_\infty = 0.67 < 1$.

3.4 Discussions

The reliability of the HIL simulation is mostly determined by the HIL interface. The result is distorted due to different DUT-DRTS boundary conditions of the non-ideal interface. Similar to the co-simulation, the HIL simulation can be improved in two directions: reduce non-ideal effects and improve system robustness.

The main points of this chapter and results of this work are as follows.

1 The co-simulation and HIL simulation exhibit many similarities because they are coupled systems, especially analysed in the control framework. The HIL interface has more ingredients that introduce boundary discrepancies, which further propagate inside the HIL system.

2 *Transparency* measures the input-output discrepancy of the interface and is a system-level requirement for accuracy. On the component-level, it is improved by lower delay, more powerful actuator, etc. Investigations on delay compensation and control design have been conducted in many related works.

3 *Transparency* does not ensure the HIL simulation stability, and non-ideal effects remain. Therefore, system robustness is sought using interface algorithms. The ITM algorithm shows that similar to co-simulation, causality is critical in the HIL simulation. In this work, the DIM algorithm is realised in the MHIL simulation and effective for enhancing system robustness.

4 In terms of methodology, the HIL simulation is investigated in the linear robust control framework, which is practical for dealing with uncertainty and time-variant effects. Furthermore, it potentially provides a systematic tool for combining HIL simulation and co-simulation. Although these subjects are frequently investigated together, a unified research framework is missing.

In addition to these findings, certain interesting ideas are not covered in this work. IAs may also be applicable to co-simulations. For example, one of the IAs (e.g., the TLM method) is already effectively used in distributed co-simulations [64, 108]. Furthermore, if the DUT and DRTS are geographically distributed, Z_{comp} must also be delayed with the DIM algorithm. Then, the specification of Z_{comp} and implementation may change.

Part II

Implementation

Chapter 4

Vehicle and EPAS system

The vehicle chassis and EPAS system are the developing subjects in this work. System dynamics, modelling and simulation techniques are presented in this chapter. Computer and HIL simulations are created as development tools. Previous theoretical findings are implemented and verified in the engineering case considered.

4.1 Vehicle chassis system

The vehicle chassis interacting with the steering system mainly consists three subsystems: vehicle body, tyres and suspensions. The modelling details depend on the development phase and test purpose. At the vehicle level, validity is sufficient within a limited frequency range for most vehicle dynamics tasks (e.g., $1-2$ Hz for the vehicle body motion, $4-10$ Hz for the longitudinal oscillations and $10-16$ Hz for the steering oscillations) [100]. However, the inner interaction among subsystems may be incurred at a higher frequency range. At the subsystem level, a stricter validity range is desired .

Vehicle body
The vehicle body is generally regarded as a rigid mass. According to the applied forces, the body has six degrees of freedom in the local coordinate: longitudinal, lateral, vertical, yaw, roll and pitch motions [42, 54]. The important properties include mass m, location of center of gravity and moments of inertia, especially for yaw motion J_{zz}.

Wheel and tyre
Each wheel includes mass, inertia of attached bodies and tyre. It mainly has three degrees of freedom with respect to the body: wheel spinning, steering and jounce. The consequent tyre-road interaction generates the forces: longitudinal force F_x,

lateral force F_y and self-aligning torque M_z in the tyre coordinate (Fig.4.1). In steady state, the tyre forces are functions of the longitudinal slip ratio α_x, lateral slip angle α_y, inclination angle γ and normal load F_z.

$$(F_x, F_y, M_z) = f_{\text{tyre}}(\alpha_x, \alpha_y, \gamma, F_z) \tag{4.1}$$

Here, f_{tyre} is determined by tyre properties, such as compounds, structure, and tyre pressure. The dynamics can be captured by an empirical model, such as the common Magic Formula [81]. Alternatively, first-principle models, such as the brush model [94] or a more detailed finite element model [43], are used. When $\alpha_x = 0$ and α_y is small, Eq.(4.1) is linearised as $F_y = -C_\alpha \alpha_y$, where C_α is the tyre lateral cornering stiffness dependent on F_z.

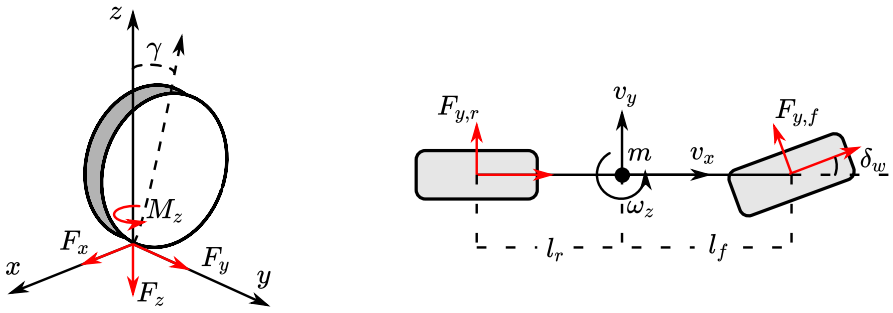


Figure 4.1: Tyre forces in local coordinate and single-track model for lateral dynamics.

Suspension

Suspension includes the linkages connecting the wheel and vehicle body. It suppresses the road disturbance impacted to the body for ride comfort and keeps the wheel on the ground for handling. Kinematically, wheels are constrained to bounce up and down, and can be steered left and right. Dynamically, the suspension linkages transmit the tyre forces to the body. The load on the steering linkage (i.e., tie-rods) is projected to the steering rack according to the kingpin axis geometry (Fig.4.2).

4.1.1 Vehicular lateral dynamics

Vehicle-steering interaction is mainly dominated by vehicular lateral dynamics. The simplest way to describe it is using the well-known linear single-track model.

$$\begin{aligned} ma_y &= m(\dot{v}_y - v_x\omega_z) = F_{y,f} + F_{y,r} \\ J_{zz}\dot{\omega}_z &= F_{y,f}l_f - F_{y,r}l_r \end{aligned} \tag{4.2}$$

The lateral forces are calculated as follows

$$\begin{aligned} F_{y,f} &= -C_{\alpha,f}\alpha_{y,f} \qquad F_{y,r} = -C_{\alpha,r}\alpha_{y,r} \\ \alpha_{y,f} &:= \frac{v_y + \omega_z l_f}{v_x} - \delta_w \quad \text{and} \quad \alpha_{y,r} := \frac{v_y - \omega_z l_r}{v_x} \end{aligned} \tag{4.3}$$

where $C_{\alpha,f}$ and $C_{\alpha,r}$ are the aggregate tyre cornering stiffnesses on the front and rear axles. $\alpha_{y,f}$ and $\alpha_{y,r}$ are the front and rear lateral slip angles respectively.

With the assumption that a_y and δ_f are small, v_x is a large constant, the vehicle lateral dynamics is a second-order LTI system combining Eqs.(4.2) and (4.3). The state vector is $x = \begin{bmatrix} v_y & \omega_z \end{bmatrix}^T$ and the input is the steering angle δ_w on the front wheel. This model is used to analytically describe the relationship $\delta_f \to F_{y,f}$.

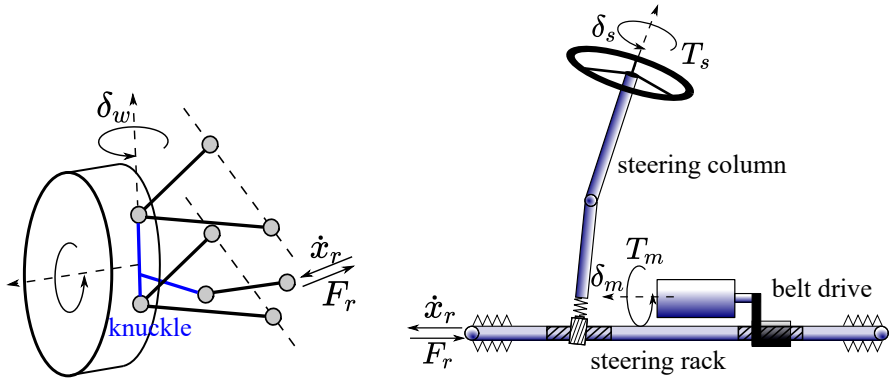


Figure 4.2: The tyre forces are transmitted to the steering rack through the tie-rods. An EPAS motor is attached to the steering rack to apply an assist force.

4.2 EPAS system

The human driver interacts with the vehicle through the steering system. Kinematically, the steering wheel angle δ_s is transmitted to the steering rack displacement x_r. Dynamically, the resultant rack force F_r is transferred to the steering torque T_s perceived by the driver.

EPAS mechanism

The rack-type EPAS system (Fig.4.2) mainly has three degrees of freedom, corresponding to the dynamics on the steering column, motor and steering rack:

$$\begin{aligned} J_{col}\ddot{\delta}_s &= T_s - T_{pinion} - T_{c_{friction}} \\ J_m\ddot{\delta}_m &= T_m - T_{belt} \\ m_r\ddot{x}_r &= F_{pinion} + F_{assist} - F_r - F_{r_{friction}} \end{aligned} \tag{4.4}$$

The definition and values of the parameters are given in **Paper I** and **V**. The forces F_{pinion} and F_{assist} are transferred to torques T_{pinion} and T_{belt} by the rack-pinion and belt transmission respectively. Based on the ratios, these forces are as follows.

$$F_{pinion} = T_{pinion}/i_{pinion}, \quad F_{assist} = T_{belt}/(i_{belt}i_{bs}) \tag{4.5}$$

In an engineering case, the model is more complex than the description. Although detailed transmission dynamics may be considered [120], the details are minimised here.

The gear transmission is preloaded to avoid backslash. However, this results in substantial friction effects. In this work, the friction effects are lumped as an upstream element $T_{c_{friction}}$ on the column and a downstream element $F_{r_{friction}}$ on the rack. The friction elements are modelled using the LuGre model [55]. The friction model is explained in detail and validated using experiment rig data in **Paper I**. Friction negatively affects the steering feedback and 'connecting-to-the-road' feeling. On the other hand, it influences system stability, especially the EPAS control design. If the friction and non-linear effects are omitted, then the EPAS mechanism (Eq.(4.4)) can be further reduced to an EPAS motor rotation (**Paper V**), which may be more intuitive in the HIL simulation design.

EPAS control

The EPAS motor produces a torque T_m to assist the rack movement. The control of T_m has a central role in the steering feedback design and in achieving advanced driving functions. Therefore, it is the primary object tested with a virtual prototype. The basic functions of EPAS include power assist, active damping, inertia compensation and friction compensation [52]. Advanced EPAS functions are dependent on high-level vehicle and traffic information (e.g., Lane keeping aid, automated driving). The EPAS torque is given by the following

$$T_m = f_{EPAS}(T_{pinion}, \delta_s, v_x, \delta_m, \dot{\delta}_m) + f_{ADAS} \tag{4.6}$$

The advanced functions influence vehicle motion; in contrast, the basic functions exert a greater influence at higher-frequency modes on the subsystem level. Therefore, the steering feedback design requires reliable virtual prototypes. Novel concepts, such as HIL simulations [118, 31], are developed for this purpose.

4.3 Vehicle-steering interaction

The EPAS system and vehicle chassis are connected through the tie-rods. Analytically, the two systems (Eqs.(4.2) and (4.4)) are coupled by the rack force F_r and velocity $\dot{x}_r$. The interaction is illustrated in Fig.4.3. Note that directional flows can be changed using different system formulations.

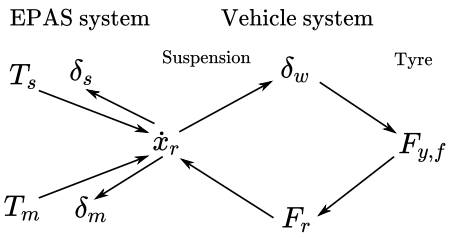


Figure 4.3: Topology of EPAS system and vehicle chassis interaction.

From the driver perspective, the steering torque feedback depends on the relationship $T_s \rightarrow \delta_s$. The perception of road disturbance is determined by $f_{tyre} \rightarrow T_s$ in which f_{tyre} involves all tyre forces. The virtual prototypes must faithfully predict these dynamics. From the EPAS motor perspective, the surrounding dynamics is given by the relationship $T_m \rightarrow \delta_m$. This sets the boundary condition if the EPAS motor is applied as a DUT to the HIL simulation.

4.4 Virtual prototyping

Virtual prototyping is employed to develop the EPAS system. First, it is used to design the EPAS control for the driver´s steering feedback. Second, it is used to verify the control code embedded in the real system. Multiple virtual prototypes, including several co-simulation approaches and an HIL simulation approach, are built. In co-simulation, the architecture problem, effects of relaxed communication (i.e., the macro-step size) and coupling method are investigated. In the HIL simulation, the interface algorithms are implemented, and the *consistency* among the tools is considered.

4.4.1 Co-simulation design

In the computer simulation, the EPAS control is in Matlab s-function and simulated at 1 ms. The remaining mechanism is modelled using Dymola, which is an acausal and monolithic simulation approach. Hence, it is regarded as a relatively error-free reference. The communication with the EPAS control is 1 ms. Otherwise, the simulation result significantly changes owing to an insufficient step size for the fast control dynamics, as presented in **Paper I**. Error may also be introduced by the discrete control functions if they take a different step. Meanwhile, the vehicle-EPAS system can be distributed into co-simulation FMUs in various ways (Fig.4.4).

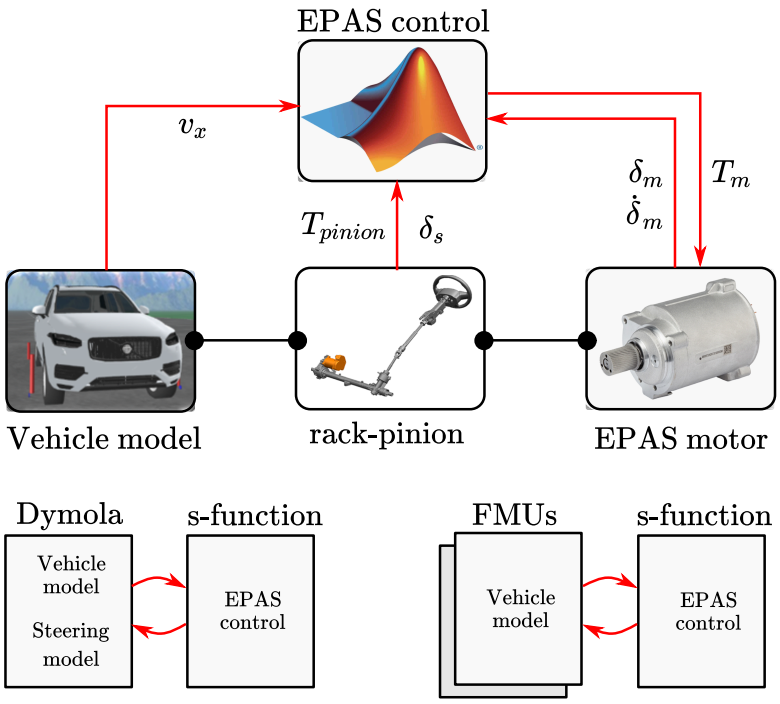


Figure 4.4: Computer simulation tools of the vehicle-EPAS system.

Architecture problem

Distribution can be implemented on the tie-rods or on belt transmission. Then, the rack-pinion mechanism is combined with the vehicle model or EPAS motor. The corresponding coupling variables are F_r and $\dot{x}_r$ or T_{belt} and $\dot{\delta}_m$. Moreover, the causality can be swapped and it offers four design options. Based on **Paper II**, the system is found stable when F_r flows towards the rack or T_{belt} flows towards the EPAS motor. Instability occurs in other options. Due to the gear ratio effect, the EPAS motor is the heavier side (i.e., it has a larger impedance) compared with the vehicular lateral dynamics. This confirms the discussion in **Chapter II** that applying force/torque to the larger impedance, similar to a larger denominator, yields a smaller loop gain and system becomes more robust. This can be roughly verified using the simplified models (Eqs.(4.2) and (4.4)).

In terms of accuracy, the design option with distribution on the tie-rod is more accurate than on the belt transmission (Fig.4.5). The difference is clearer for the motor speed because the motor is directly exposed to the error perturbation on the belt transmission. A minor difference is observed among the resulting steering wheel angles because the bandwidth of $T_s \rightarrow \delta_s$ is typically lower than 2 Hz. This implies that the coupling defect is less important to the slow vehicle-level characteristics.

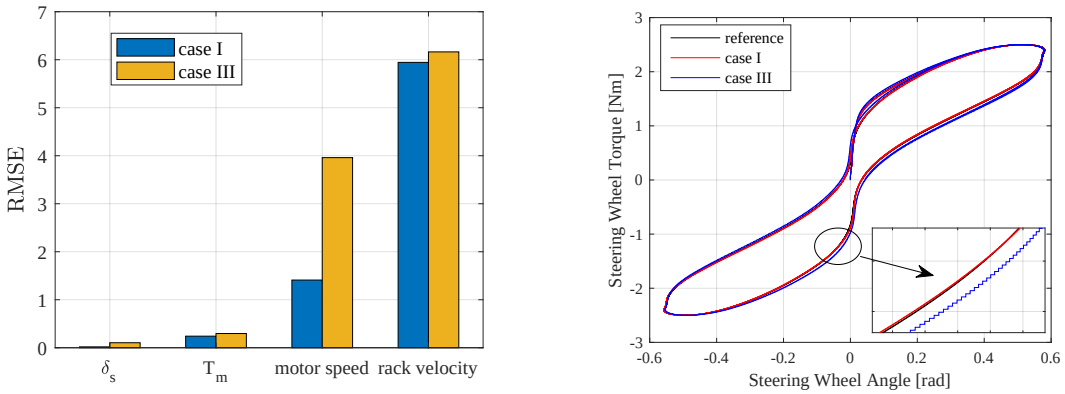


Figure 4.5: Simulation results with sine-wave T_s input and $v_x = 50$ km/h. Root-mean-square error (RMSE) of results and steering hysteresis characteristic of stable co-simulation cases. Case I refers to F_r applied to the rack, and Case III refers to T_{belt} applied to the motor.

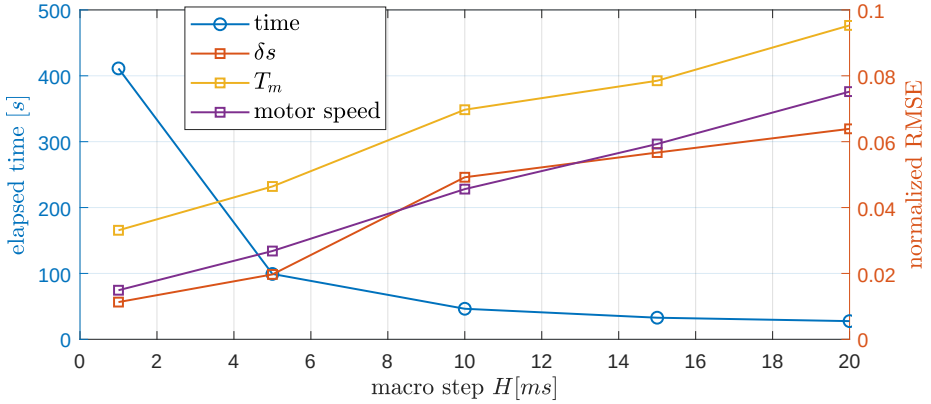


Figure 4.6: Elapsed time and normalised RMSE when a larger macro step is assigned to the vehicle model.

Effect of macro-time step

As reported in **Paper I**, the speed of monolithic simulation is low: 476 s elapses for a 5-s task. Vehicle model stepping is constrained by the fast dynamics of EPAS mechanism and frequent communication. To accelerate the simulation speed, the system on the tie-rods is split according to the previous analysis. The vehicle model is solved by a variable step solver. The EPAS mechanisms model is solved by the RK4 method with a step of 0.25 ms. The communication to the vehicle model is relaxed with a macro-step of H, and the EPAS mechanism communicates every 1 ms.

Only single-core simulations are performed in this work. The comparison

(Fig.4.6) shows that the simulation speed-up effect is evident when $H \leq 10$ ms. Thereafter, it gets saturated. This is because the simulation speed becomes constrained by the local solver integration rather than the communication. The relative error monotonically increases with H. As a result, a macro-step size of approximately 5-10 ms is an acceptable trade-off. According to the normalised RMSE[1], the internal variables (e.g., motor torque and motor speed) show more discrepancies than δ_s. This means that the coupling effect has a greater influence on internal interactions, which may be more of a concern in the subsystem-level development.

Effect of coupling method

Next, coupling methods are employed to further improve the co-simulation prototype accuracy. The ZOH, FOH, SOH and H_∞ methods are implemented to the tie-rods interface. The vehicle and EPAS models communicate with $H = 20$ ms.

H_∞ method is specified in a general way as suggested in **Paper III**. Two simulations with lower and higher-frequency swept steer input are conducted. The consequent normalised RMSE and elapsed time are summarised in Table.4.1. The SOH method is unreliable owing to instability (Fig.4.7), which is also indicated by the large deviations. The FOH and H_∞ methods show error reductions compared with the default ZOH method. In most cases, the new H_∞ method shows more accuracy improvement.

The coupling methods do not add excessive computational load owing to their simplicity. Compared with the increased percentile of the elapsed time, the error reductions are more distinct, indicating that the coupling methods potentially enable a larger macro-step H to reduce the elapsed time. However, in the given case, a further relaxation of H seems unnecessary because of the speed-up saturation.

[1]RMSE of data x is normalised by dividing the value $(x^{\max} - x^{\min})$.

Table 4.1: Comparison of error and elapsed time.

Coupling methods	**Normalised RMSE**				**Elapsed time** [s]
	δ_s	$\dot{x}_r$	T_m	yaw rate	
Low frequency test					
ZOH	0.0071	0.1032	0.0615	0.0489	29.30
FOH	0.0017 (24%)	0.0593 (57%)	0.0558 (91%)	0.0255 (52%)	29.64 (101%)
SOH	4.4502	6.2525	4.1768	1.6053	36.10 (123%)
H_∞	0.0008 (12%)	0.0559 (54%)	0.049 (80%)	0.017 (35%)	30.44 (103%)
High frequency test					
ZOH	0.0028	0.2697	0.0353	0.0329	31.92
FOH	0.0017 (59%)	0.2185 (81%)	0.0342 (97%)	0.0165 (50%)	31.43 (98%)
SOH	8.1357	6.2144	2.2278	1.0361	34.43 (108%)
H_∞	0.004 (143%)	0.1574 (58%)	0.0341 (97%)	0.0121 (37%)	32.89 (103%)
Note: The percentage indicates the normalised RMSE and elapsed time compared with ZOH method.					

Furthermore, the results of stable runs all well agree (Fig.4.7). The *consistency* of results may be sufficient for practical purposes. One may wonder the required extent of accuracy of the virtual prototype. In the present work, a systematic and objective approach has not been found to answer this question. Such requirements may be set empirically and subjectively in engineering work.

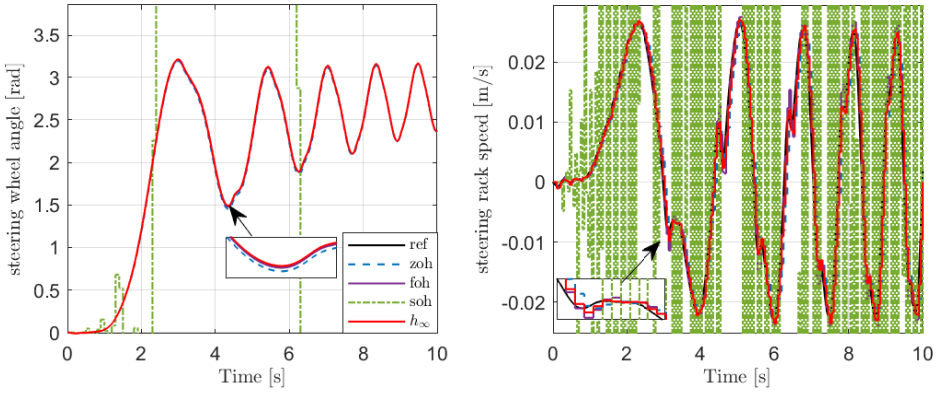


Figure 4.7: Vehicle-EPAS system co-simulation results with different coupling methods. A low-frequency swept steer torque input is applied to the steering system.

4.4.2 HIL-simulation design

For the succeeding development phase, the HIL simulation is designed by replacing the EPAS motor and control with a real DUT. The ingredients of HIL simulation are illustrated in Fig.4.8. The remaining parts (i.e., vehicle and steering mechanism) are emulated by the DRTS. The interface is a motor dynamometer rig consisting of a servo motor, a torque sensor and shaft coupling components.

The system structure is the same as the generalisation presented in **Chapter III**. More specifically, on the DRTS side, the vehicle-steering system is modelled using IPG Carmaker. Unfortunately, the same MBD model in Dymola is not employed here due to the software support on the used RT computer. The system is simulated in real time on an NI PXIe-8880 machine and with a step of 1 ms. The steering mechanism is a built-in model in Carmaker and runs in parallel with a step of 0.1 ms. Moreover, a vehicle control area network (CAN) simulation runs on the same machine with Vector CANoe. The purpose is to map the simulation variables (e.g., T_{pinion}, v_x and δ_s to the EPAS control unit. Other necessary CAN nodes, such as engine ignition, are also emulated. The CAN signals are communicated at 100 Hz, which is same as that on a real vehicle. The RT machine is connected to a host personal computer (PC). The NI Veristand on this PC is used to configure, monitor and host communication for the RT simulations.

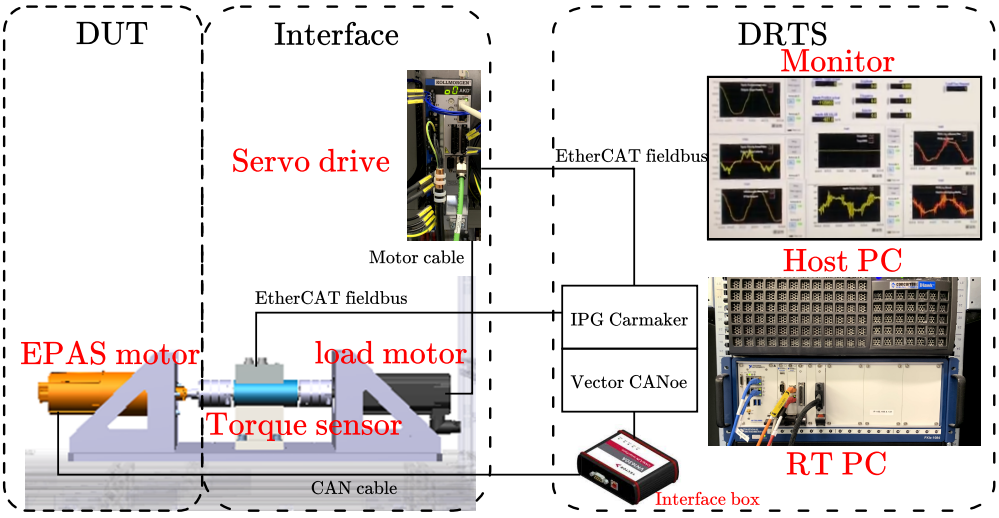


Figure 4.8: HIL simulation setup for vehicle-EPAS system development.

On the DUT side, the EPAS motor and ECU are integrated as a complete power pack unit. The power pack is charged with a 12-V power supply and produces the torque T_m in an open-loop manner according to the received CAN signals. The motor output shaft is clamped to a rotary torque sensor. Thus, the distribution is between the output shaft and belt transmission. The boundary condition (i.e., the relationship between T_{belt} and $\dot{\delta}_m$) is calculated by the DRTS and must be enforced by the HIL interface.

The high-precision torque sensor is Kistler 4503B with a torsional stiffness of 2.7 kNm/rad. It is regarded as a rigid shaft with only an inertia effect. The shaft torque is measured at a rate of 1 kHz. Similar to that in Eq.(3.7), the measured torque results from T_m and the counteracting load torque. The load torque is produced by a Kollmorgen AKM44J servo motor controlled by the servo drive at 8 kHz. The servo motor is in a closed-loop with the DRTS and enables a torque or velocity-controlled mode. Correspondingly, the control reference from the DRTS can be T_{belt} or $\dot{\delta}_m$. The actual motor speed, which is same as the speed of the rig, can be accessed by the resolver inside the servo motor. The DRTS connects to the servo drive via an EtherCAT fieldbus and to the EPAS power pack with a regular CAN cable.

A similar setup is also applied to develop the electric vehicle powertrain [61]. The advantage of this setup is that the motor physical constraints owing to circuit dynamics and thermal behaviour are captured. Moreover, the EPAS control is updated by flashing the motor ECU. Thus, it is verified to be similar to that on a test vehicle. The HIL simulation is also necessary if the power pack unit is supplied without an accurate model. In such a HIL system, the shaft torque measurement may or may not be required [77]. With complete DUT information or an access

to the electric circuit, T_m is directly obtained. This condition does not seem to be a typical MHIL simulation according to the definition in this work, because T_m is not a real mechanical variable. Under this condition, different control methods can be employed, such as compensating the interface distortion with its inverse [6, 5, 93]; however, the problem is different in this work. In our study, the DUT is considered as a black box similar to typical cases, and shaft torque measurement is required.

Preliminary experiment

In the preliminary experiment, the constructed HIL simulation exhibits limited performance and can easily become unstable. For safety reasons, power and velocity thresholds are introduced to the servo drive. When the thresholds are exceeded due to instability, the system switches off to avoid hardware failure. Such a dynamic defect is also reported by other researchers [50], i.e., the system works well only for steering tests in low frequency (< 0.5 Hz) and at high vehicle speed. We found that the limitations are due to

- The inertia attached to the DUT is relatively large comparing with its own. This considerably distorts the simulation results.
- The mechanical impedances of the entire HIL rig, i.e., inertia and damping effects, are extremely small. Moreover, the rig is highly sensitive to the two counteracting motor torques and associated disturbances. In terms of control, the system loop gain is large. Neglecting the noise effect, an overall delay $\tau < 0.04$ s is required according to the estimates in **Paper V**. The delay tolerance increases with the mechanical impedance.
- The DUT inertia is relatively large comparing with that of the DRTS. According to the ITM algorithm (Eq.(3.14)), a large loop gain $-\frac{Z_{dut}}{Z_{em}}e^{-s\tau}$ results in the velocity control mode. Theoretically, the torque control mode is preferred.

The last point, i.e., the causality problem, is discussed in **Paper II** through a simulation study. However, the built-in steering model in Carmaker only accepts an assist torque or force input and outputs $\dot{\delta}_m$. This is practically more suitable for the velocity control mode. To change the mode and accept $\dot{\delta}_m$ as input, an external velocity actuator model must be added, or the in-house steering model in Dymola must be applied if compatible. However, these options are more difficult to implement and do not function smoothly. Accordingly, the velocity control mode was used in the experiment.

Furthermore, the HIL system was extended with a driver-in-the-loop (Fig.4.9). The human driver controls the vehicle model concurrently with the HIL simulation. The EPAS torque and rig dynamics are perceived through a steering force

Figure 4.9: Preliminary experiment using HIL simulation with driver-in-the-loop.

feedback system [58]. In the experiment, the driver conducts a standard on-center steering manoeuvre with a steering input at 0.2 Hz. This tests the responsiveness and steering feel with a low steering input, which is one of the basic system characteristics.

The results at different vehicle speeds are shown in Fig.4.10. The steering hysteresis plot and vehicle yaw rate response correlate well to the real vehicle measurements when $v_x = 120$ and 80 km/h. The driver can clearly identify the steering feedback changes with different settings of the assist torque T_m and rack-pinion ratios i_{pinion}. Even slight vibrations on the steering wheel are perceived in these tests. The vibration is generated from the HIL rig and becomes more distinct when $v_x = 50$ km/h, as reflected in the logged data. The driver must tightly grab the steering wheel to maintain the angle position. In this case, steering feedback cannot be accurately evaluated.

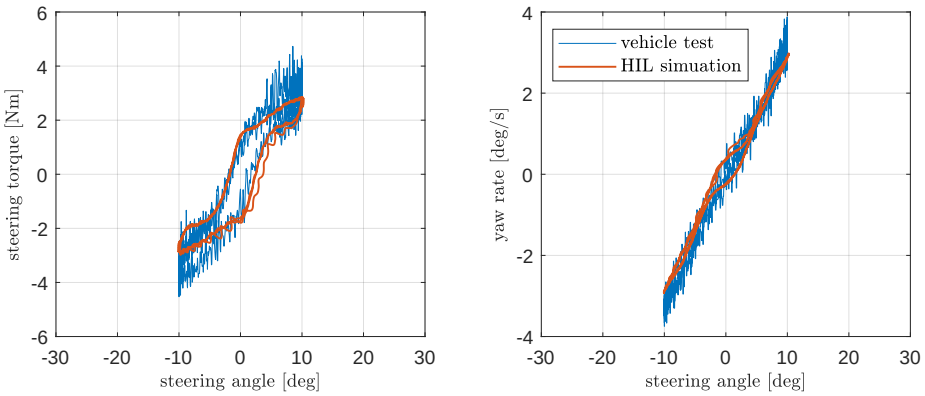
The deterioration may be that the EPAS motor produces a larger assist torque in low speed manoeuvres, consequently amplifying the associated disturbance to the rig. In the energy perspective, more of the energy is incorrectly transferred to the rig as kinetic energy. Then, high-frequency vibrations or fast rotations in one direction occur. In more aggressive manoeuvres, when the driver applies a higher-frequency steering input, increased vibrations are also felt.

Preliminary results reveal that the use of the virtual prototype is limited. In tests at low speed and high frequency, the HIL simulation is found unreliable to predict the dynamics due to the lack of robustness.

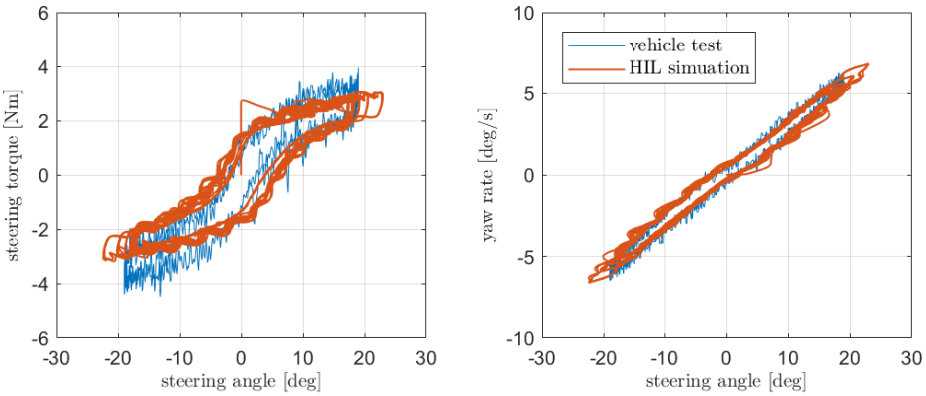
Effect of interface algorithm

To resolve the aforementioned problem and extend the validation range of HIL simulation, the introduced DIM algorithm is applied. The DIM algorithm is similarly implemented as the benchmark MHIL simulation (Fig.3.6). In this case, Z_{dut} primarily consists of the motor inertia $J_m = 1.614e-4\text{kgm}^2$ and coupling shaft inertia $J_{h_1} = 1.527e-4\text{kgm}^2$. As introduced in **Paper V**, the following term is specified.

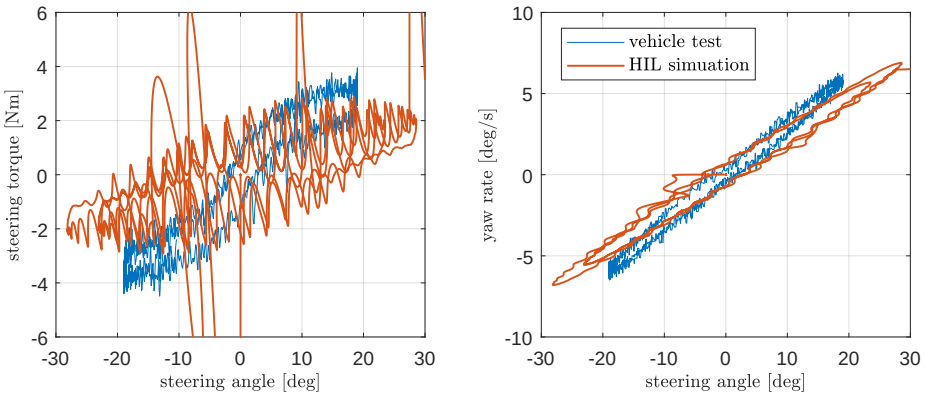
$$Z_{comp} = \lambda Z_{dut} \tag{4.7}$$



Vehicle speed $v_x = 120$ km/h.



Vehicle speed $v_x = 80$ km/h.



Vehicle speed $v_x = 50$ km/h.

Figure 4.10: On-center steering test results of δ_s versus T_s and ω_z at various vehicle speeds. The HIL simulation results are unreliable when $v_x = 50$ km/h due to instability.

It is modelled in Matlab, complied to an RT executable and added to the shaft torque measurement. To preserve the dynamics, the steering rack mass in the DRTS is modified using $m_r^* = m_r + Z_{comp} i_{belt}^{-2} i_{bs}^{-2}$ similar to Eq.(3.17). Analytically, a portion of DUT impedance is compensated and shifted to the DRTS.

The constant λ determines the ratio of the shifted part. In the original setup, i.e., $\lambda = 0$, the system is highly unstable. The simulation frequently terminates when the safety thresholds are exceeded. A partial compensation ($Z_{comp} = J_{h_1}$) and an optimal compensation ($Z_{comp} = J_{h_1} + J_m$) are applied. In the former, the attached hardware inertia is approximately compensated, and the HIL simulation becomes stable in all tests. However, vibrations remain, as demonstrated in Fig.4.11. The steering input is at 0.5 Hz and the vehicle speed is $v_x = 40$ km/h in the shown case. This case actually fails in the preliminary test. Under the optimal condition, an increased impedance is shifted to the DRTS, and undesired vibrations are removed as demonstrated in Fig.4.11 and Fig.4.12.

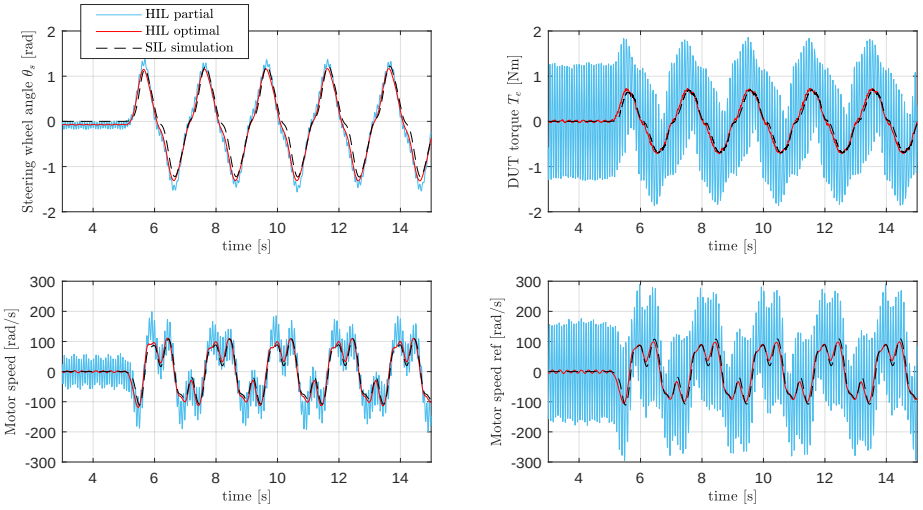


Figure 4.11: SIL and MHIL simulation results of sine steer tests with optimal and partial compensations of DIM algorithm. The steering input is at 0.5 Hz and $v_x = 40$ km/h.

A SIL simulation, i.e., a pure computer simulation approach, with the same EPAS control software, is applied for comparison. The HIL simulation results well agree with the SIL reference, implying improved *consistency* by the DIM algorithm.

Furthermore, a correlation study in various testing scenarios is conducted to validate the tool. The correlation is measured statistically using Pearson's correlation coefficient $\rho \in [-1, 1]$ [23]. When $\rho = 1$, HIL and SIL simulations are perfectly correlated; $\rho = -1$ implies a reverse correlation; $\rho = 0$ implies unrelated results (i.e., the HIL simulation is unreliable). The deviation value $(1 - \rho)$ listed

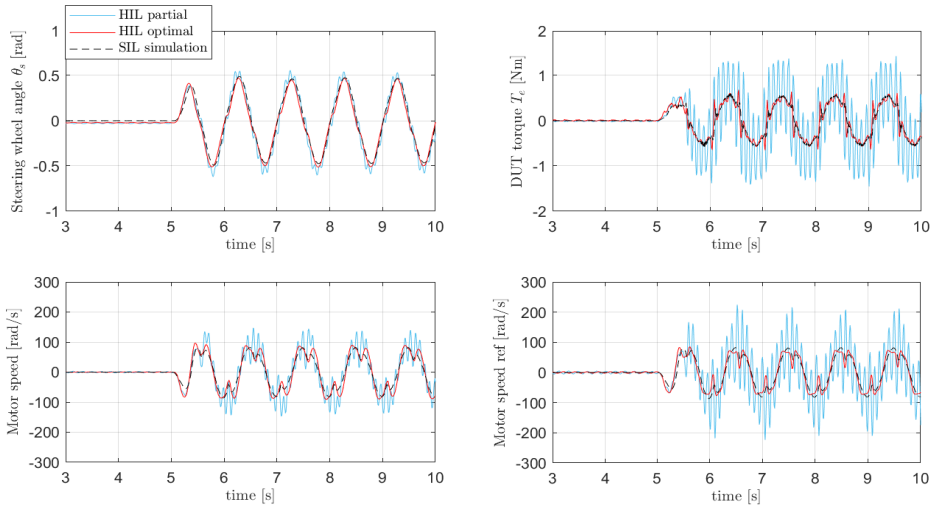


Figure 4.12: SIL and MHIL simulation results of sine steer tests with optimal and partial compensations of DIM algorithm. The steering input is at 1 Hz and $v_x = 90$ km/h.

Table 4.2: Deviation from perfect correlation $(1 - \rho)$ in various scenarios

Measurement	40 km/h [%]						90 km/h [%]						140 km/h [%]					
	0.5 Hz		1 Hz		2 Hz		0.5 Hz		1 Hz		2 Hz		0.5 Hz		1 Hz		2 Hz	
T_s	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	1
θ_s	1	2	2	1	4	2	3	13	1	5	9	7	7	84	2	82	11	64
T_e	1	49	4	41	13	75	1	62	6	59	58	91	2	89	11	83	44	127
motor speed $\dot{\theta}$	1	10	2	3	2	1	10	9	4	1	3	3	9	83	14	94	5	36
vehicle yaw rate	1	1	2	1	4	2	3	9	1	1	9	3	7	88	1	58	10	57
vehicle speed	1	1	1	1	1	1	1	5	1	2	0	1	1	192	0	113	0	183
tracking performance	1	65	1	35	1	13	3	122	4	92	3	64	5	127	13	128	6	137
Note: The first and second values are the optimal and partial compensations respectively.																		

in Table.4.2 of **Paper V** shows the following.

- The DIM algorithm effectively improves the HIL simulation accuracy. The optimal case is more accurate than the partial case.
- As anticipated, the deviation increases with frequency. The deviation is more distinct at the internal interaction $T_m \rightarrow \dot{\delta}_m$.
- The interface *transparency*, observing from the correlation between the reference and actual $\dot{\delta}_m$, also improves. Because the less oscillatory control reference decreases the tracking error with a same servo controller, as shown in Fig.4.11 and Fig.4.12.

- The high-speed test result is less accurate than that of the low speed test.

The last point may be contradictory with more vibrations observed in low-speed. However, it is reasonable because in the high-speed test, the true values of T_m and δ_s are small, and the imprecision factors, e.g., noises and delay, become relatively large. In terms of stability, the low-speed test remains to be the problematic scenario.

Eventually, the validation range of the HIL system is effectively extended in the work, showing considerable potential as an EPAS development tool.

4.5 Discussions

This chapter introduced several virtual prototypes constructed in the work. The purpose is to enable a consistent and validated vehicle-EPAS system development process in different phases. The goal is achieved by solving the instability and inaccuracy problems in co-simulation and HIL simulation. In addition to solving real engineering problems, the theoretical guidelines are followed, and the findings are validated.

The main points of the chapter and results of this work are as follows.

1 Numerical efficiency can be considerably improved by separating the EPAS system from the vehicle model due to the fast modes, e.g.,mechatronic subsystem and friction. The simulation is accelerated to a certain extent using a relaxed macro-step. Thus, a variable macro-step control can be advantageous to improve performance.

2 Regarding co-simulation, architecture and causality demonstrate a dominant effect on performance. The EPAS motor must receive the torque due to the larger impedance. This may be the same for other mechatronic systems on vehicles with a reduction gear.

3 The developed coupling method improves accuracy with a minor computational load. However, in the considered case, the improvement is marginal because the problem is already partially solved by the architecture design. Thus, the coupling method is regarded as an add-on for further improvement. In engineering work, knowing the level of accuracy is desirable and musts achieved.

4 In the HIL simulation, the architectures of DUT and DRTS are also crucial. The imprecision factors from the HIL interface outweigh the numerical factors inside the DRTS. An acausal tool with a flexible interface is more suitable for different HIL simulations.

5 The DIM algorithm greatly improves the HIL simulation performance. Although the servo control, which is a PI contoller, is tuned for better tracking performance, the system behaviour remains unsatisfactory. Thus, the effect of an advanced servo motor control may be marginal in the investigated case. However, the interface algorithm solved most problems.

Chapter 5

Conclusions

The work arises from the necessity of virtual prototypes for the vehicle-EPAS system development. Reliable virtual prototypes are necessary to indicate design changes. Simulations should reveal these changes rather than errors originating from the simulation environment. Given the same design, *consistency* in results is necessary. However, such may not be the case due to tool integration.

Associated technical problems are thoroughly identified. Virtual prototype architecture, numerical error in co-simulation, delay error in transmission, control error and noise disturbances are considered. To resolve the corresponding defects, methods are developed. Eventually, the scope of the work is satisfied by achieving improved *consistency* among the prototypes (e.g., reduced co-simulation and HIL simulation inaccuracy), and validated over an extended range (e.g., improved HIL simulation results in more testing scenarios).

5.1 Summary

The outcomes are summarised in this section. First, the research questions are answered. In addition to technical solutions, the ideas are integrated to formulate and propose a working procedure.

5.1.1 Answers to research questions

Q1. *How can the virtual prototype be constructed and the interface of vehicle–EPAS system be designed ?*

Because the interface unavoidably introduces errors, low-frequency coupling variables are less distorted. As implemented in practice, having the interface close to some compliant parts is more advantageous. Next, the causality of force-velocity coupling considerably impacts stability. To reduce the system loop gain, the force

or torque must flow towards the larger impedance and velocity to the other. In vehicle-EPAS system, interfaces can be implemented on the rack and tie-rods or belt transmission. In the former, the rack force is applied to the rack, and in the latter, the transmitted torque is applied to the motor. Other designs easily become unstable.

Q2. *How can the simulation performance of the vehicle–EPAS system be improved ?*

In terms of numerical efficiency, the inclusion of a detailed EPAS system to an MBD vehicle model is unsuitable. Multi-rate co-simulation is preferred especially for RT performance. The stability and accuracy of the virtual prototype are observed to degrade. However, these effects are manageable and have many solutions. In addition to the robust interface design discussed in **Q1**, coupling error can be reduced by implementing smaller macro-steps and applying explicit coupling methods, e.g., the H_∞ method. Furthermore, more accurate co-simulation schemes and coupling methods can be employed if these are enabled by the software.

Q3. *How does HIL simulation differ from computer simulation ?*

The HIL interface distinguishes the HIL simulation from computer simulation. In addition to numerical errors, more imprecision disturbances (e.g., control error, transmission delay, and sensor and actuator noises) are introduced to the system. The equivalence of boundary conditions (i.e., interface *transparency*) indicates the HIL simulation accuracy. It can be determined online by monitoring the tracking errors and filtered signals. Their influence depends on the overall closed-loop behaviour.

Q4. *How can the performance of HIL simulation connected to an EPAS system test rig be improved ?*

The HIL interface is determined by the selected DUT, which is an EPAS motor in this study. First, closed-loop stability must be ensured. Interface algorithms can be applied to shift the impedance balance between the vehicle model (DRTS) and EPAS motor (DUT). Thus, system stability and robustness can be improved. The next step is to further improve the interface *transparency* by amplifying the tracking performance and delay compensation.

5.1.2 A proposed working procedure

A simple working procedure for building a co-simulation or HIL simulation prototype is proposed by the author (Fig.5.1). The empirical guides are as follows.

1. Start by partitioning and integrating the model for co-simulation and HIL simulation. Ideally, knowledge on system dynamics is possessed. Desktop simulation can be useful to identify the dynamics. Interface and causality can be designed based on the approximate information, e.g., impedance ratio.

2. Under an unstable condition, coupling variables in co-simulation can be scaled down and tracking performance in HIL simulation can be reduced until the system becomes stable. This step reduces the interaction of coupled systems and the loop gain. The problematic connection may be found. Otherwise, the design in the previous step should be reviewed.

3. Under a stable condition, performance can be improved. In terms of co-simulation, performance refers to simulation speed and accuracy. Thus, communication can be relaxed to accelerate the speed and monitor the result variations. In the HIL simulation, interface *transparency* can be improved by higher tracking performance and delay compensation.

4. Before further improving the performance, robustness can be enhanced by applying IAs, which can also be used in Step 2.

5. Coupling methods can be applied to co-simulation. In the HIL simulation, interface *transparency* can be improved as that in Step 3.

5.2 Outlook

Virtual prototyping has long been recognised as an industrial necessity. Compared with the delivered results of the product, motivation and focus on how the results are generated seem inadequate. Consequently, this has led to marginal knowledge as to alternatives when the development tools become more complexed, advanced and expensive. Simple and correct tools are ideally preferred. However, it is not generally the case in the industry. Accordingly, the author believes that comprehending this subject must be endeavoured through theoretical and systematic means. Further, the author is optimistic that this work can contribute to such an objective.

In the course of this work, further efforts have been devoted to improve the usage of virtual prototyping. Application-wise, commercial software tools must

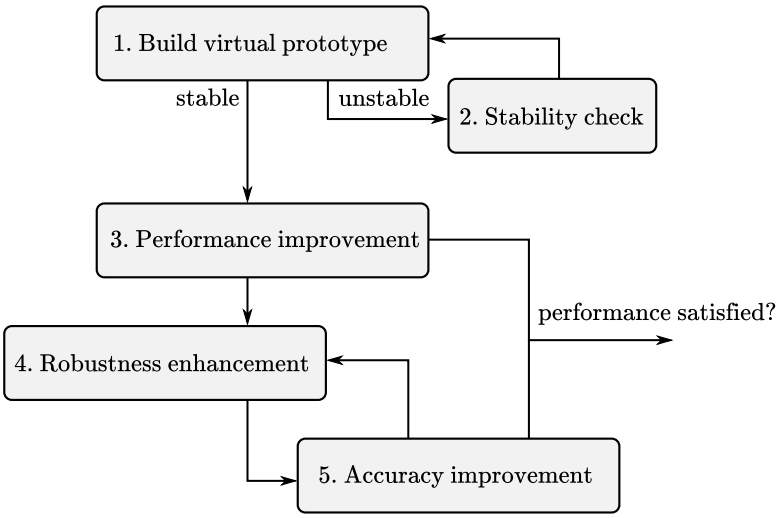


Figure 5.1: Proposed working procedure for virtual prototyping.

be more open and flexible for tool integration. Although non-proprietary standards, such as FMI and DCP, removed the integration obstacles to some extent, commercial tools only support limited features. Some tools enable the import of FMU and run as a co-simulation master. However, they may not be ready to be exported as slaves. This definitely constraints the usability in different applications. Moreover, the support of FMU on an RT system is even more limited and may require an additional license. Therefore, more software capability and not increased associated licensing cost are necessary to further exploit the techniques.

In terms of the problem investigated in the thesis, the potential of interface algorithms to co-simulation deserves further exploration. An analogy of co-simulation to HIL simulation is demonstrated in this work. Furthermore, the TLM algorithm has been well used to resolve numerical problems in distributed co-simulations. The DIM algorithm may also be effective and the modification on the model is required anyway. Another problem that must be resolved is determining the practical level of accuracy for a virtual prototype. Apparently, the error effects should not cause instability and large vibrations. However, the extent to which accuracy is necessary with the coupling method must be established. This question is not covered in this work, because it may be another research topic. Moreover, defining such a requirement objectively and quantitatively is difficult. For example, for the subjective evaluation with a driver-in-the-loop, the accuracy level may be less than that demanded by the subsystem-level objective evaluation.

Finally, because the problems and findings are general, extensions to other advanced applications are expected. To develop advanced driver assistance system functions, a camera-in-the-loop simulation at Volvo Cars is developed and oscillatory effects are reported [24]. The developed methodologies may also im-

prove the test results. A geographically distributed simulation seems an ultimate tool to share development resources in the future, and research outcomes can be advantageous to this application.

Bibliography

[1] *Vehicle electrification market reportvehicle electrification market analysis by product, by hybridization and segment forecasts, 2018 - 2025*, tech. rep., Grand View Research, 2018.

[2] *Ford expands climate change goals, sets target to become carbon neutral by 2050: Annual Sustainability Report*, 2020. URL: https://media.ford.com/content/fordmedia/fna/us/en/news/2020/06/24/ford-expands-climate-change-goals.html.

[3] *How we become CO2-neutral*, 2021. URL: https://www.volkswagenag.com/en/news/stories/2020/10/29-climate-measures-of-the-volkswagen-group.html#.

[4] G. Aguirre-Ollinger, J. E. Colgate, M. A. Peshkin, and A. Goswami, *Design of an active one-degree-of-freedom lower-limb exoskeleton with inertia compensation*, The International Journal of Robotics Research, 30 (2011), pp. 486–499.

[5] H. Akpolat, G. M. Asher, and J. C. Clare, *A practical approach to the design of robust speed controllers for machine drives*, IEEE Transactions on Industrial Electronics, 47 (2000), pp. 315–324.

[6] Z. H. Akpolat, G. M. Asher, and J. C. Clare, *Dynamic emulation of mechanical loads using a vector-controlled induction motor-generator set*, IEEE Transactions on Industrial Electronics, 46 (1999), pp. 370–379.

[7] A. Aksjonov, V. Ricciardi, K. Augsburg, V. Vodovozov, and E. Petlenkov, *Hardware-in-the-loop test of an open-loop fuzzy control method for decoupled electrohydraulic antilock braking system*, IEEE Transactions on Fuzzy Systems, 29 (2020), pp. 965–975.

[8] C. Andersson, *Methods and tools for co-simulation of dynamic systems with the functional mock-up interface*, PhD thesis, Lund University, 2016. Available at https://lup.lub.lu.se/search/publication/7dcb80cb-af81-4eed-b501-2733bd93e6eb.

[9] J. Andert, F. Xia, S. Klein, D. Guse, R. Savelsberg, R. Tharmakulasingam, M. Thewes, and J. Scharf, *Road-to-rig-to-desktop: Virtual development using real-time engine modelling and powertrain co-simulation*, International Journal of Engine Research, 20 (2019), pp. 686–695.

[10] M. Arnold, *Stability of Sequential Modular Time Integration Methods for Coupled Multibody System Models*, Journal of Computational and Nonlinear Dynamics, 5 (2010).

[11] M. Arnold, B. Burgermeister, C. Führer, G. Hippmann, and G. Rill, *Numerical methods in vehicle system dynamics: State of the art and current developments*, Vehicle System Dynamics, 49 (2011), pp. 1159–1207.

[12] M. Arnold, C. Clauss, and T. Schierz, *Error analysis and error estimates for co-simulation in fmi for model exchange and co-simulation v2. 0*, in Progress in Differential-Algebraic Equations, Springer, 2014, pp. 107–125.

[13] M. Arnold, S. Hante, and M. A. Köbis, *Error analysis for co-simulation with force-displacement coupling*, in Proceedings in Applied Mathematics and Mechanics (PAMM), vol. 14, Wiley Online Library, 2014, pp. 43–44.

[14] U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*, vol. 61, Society for Industrial and Applied Mathematics (SIAM), 1998. ISBN: 978-0-898714-12-8.

[15] M. Bacic, *On hardware-in-the-loop simulation*, in Proceedings of the 44th IEEE Conference on Decision and Control, IEEE, 2005, pp. 3194–3198.

[16] A. Badawy, J. Zuraski, F. Bolourchi, and A. Chandy, *Modeling and analysis of an electric power steering system*, tech. rep., SAE Technical Paper, 1999.

[17] M. Baseggio, A. Beghi, M. Bruschetta, F. Maran, and D. Minen, *An mpc approach to the design of motion cueing algorithms for driving simulators*, in 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, 2011, pp. 692–697.

[18] J. Bélanger, P. Venne, and J.-N. Paquin, *The what, where and why of real-time simulation*, Planet Rt, 1 (2010), pp. 37–48.

[19] A. Ben Khaled-El Feki, L. Duval, C. Faure, D. Simon, and M. Ben Gaid, *Choptrey: contextual online polynomial extrapolation for enhanced multi-core co-simulation of complex systems*, SIMULATION: Transactions of The Society for Modeling and Simulation International, 93 (2017), pp. 185–200.

[20] M. Benedikt and A. Hofer, *Guidelines for the application of a coupling method for non-iterative co-simulation*, in 2013 8th EUROSIM Congress on Modelling and Simulation, IEEE, 2013, pp. 244–249.

[21] M. Benedikt, D. Watzenig, and A. Hofer, *Modelling and analysis of the non-iterative coupling process for co-simulation*, Mathematical and Computer Modelling of Dynamical Systems, 19 (2013), pp. 451–470.

[22] M. Benedikt, D. Watzenig, J. Zehetner, and A. Hofer, *Nepce-a nearly energy-preserving coupling element for weak-coupled problems and co-simulations*, in COUPLED V: proceedings of the V International Conference on Computational Methods for Coupled Problems in Science and Engineering:, CIMNE, 2013, pp. 1021–1032.

[23] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Pearson correlation coefficient*, in Noise reduction in speech processing, Springer, 2009, pp. 1–4.

[24] F. Björklund and E. Karlström, *Enabling testing of lateral active safety functions in a multi-rate hardware in the loop environment*, Master's thesis, Linköping University, 2017. Available at `https://www.diva-portal.org/smash/get/diva2:1118174/FULLTEXT01.pdf`.

[25] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauss, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, et al., *The functional mockup interface for tool independent exchange of simulation models*, in Proceedings of the 8th International Modelica Conference, Linköping University Press, 2011, pp. 105–114.

[26] T. Blockwitz, M. Otter, J. Akesson, M. Arnold, C. Clauss, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, et al., *Functional mockup interface 2.0: The standard for tool independent exchange of simulation models*, in Proceedings, 2012.

[27] A. Bouscayrol, *Different types of hardware-in-the-loop simulation for electric drives*, in 2008 IEEE International Symposium on Industrial Electronics, IEEE, 2008, pp. 2146–2151.

[28] R. Braun, *Distributed System Simulation Methods: For Model-Based Product Development*, PhD thesis, Linköping University Electronic Press, 2015. Available at `http://liu.diva-portal.org/smash/record.jsf?pid=diva2%3A872653&dswid=-4660`.

[29] B. J. Bunker, M. A. Franchek, and B. E. Thomason, *Robust multivariable control of an engine-dynamometer system*, IEEE Transactions on Control Systems Technology, 5 (1997), pp. 189–199.

[30] M. Busch, *Zur effizienten Kopplung von Simulationsprogrammen*, PhD thesis, Kassel University, 2012. Available at `https://www.uni-kassel.de/upress/online/frei/978-3-86219-296-0.volltext.frei.pdf`.

[31] W. Chen, M. Klomp, U. Khan, A. Bianchi, S. Ran, and B. Jacobson, *An architecture of hardware and driver in the loop simulation for electric power assisted steering system*, in The IAVSD International Symposium on Dynamics of Vehicles on Roads and Tracks, Springer, 2019, pp. 1449–1456.

[32] W. Chen, S. Ran, and B. Jacobson, *Design of interface in co-simulation for electric power assisted steering system development*, in 14th International Symposium on Advanced Vehicle Control, 2018.

[33] M. Dargahi, A. Ghosh, and G. Ledwich, *Stability synthesis of power hardware-in-the-loop (phil) simulation*, in 2014 IEEE PES General Meeting— Conference & Exposition, IEEE, 2014, pp. 1–5.

[34] J. De Angelo, *The link flight trainer: A historic mechanical engineering landmark, roberson museum and science center, binghamton, new york, june 10, 2000.*, ASME, (2000).

[35] J. R. Dormand and P. J. Prince, *A reconsideration of some embedded Runge-Kutta formulae*, Journal of Computational and Applied Mathematics, 15 (1986), pp. 203–211.

[36] T. Ersal, M. Brudnak, A. Salvi, J. L. Stein, Z. Filipi, and H. K. Fathy, *Development and model-based transparency analysis of an Internet-distributed hardware-in-the-loop simulation platform*, Mechatronics, 21 (2011), pp. 22–29.

[37] T. Ersal, H. K. Fathy, D. G. Rideout, L. S. Louca, and J. L. Stein, *A review of proper modeling techniques*, Journal of Dynamic Systems, Measurement and Control (ASME), 130 (2008), pp. 0610081–06100813.

[38] M. Faruque, V. Dinavahi, M. Steurer, A. Monti, K. Strunz, J. Martinez, G. Chang, J. Jatskevich, R. Iravani, and A. Davoudi, *Interfacing issues in multi-domain simulation tools*, IEEE Transactions on Power Delivery, 27 (2011), pp. 439–448.

[39] H. K. Fathy, Z. S. Filipi, J. Hagena, and J. L. Stein, *Review of hardware-in-the-loop simulation and its prospects in the automotive area*, Modeling and Simulation for Military Applications, 6228 (2006), p. 62280E.

[40] Z. Filipi, H. Fathy, J. Hagena, A. Knafl, R. Ahlawat, J. Liu, D. Jung, D. Assanis, H. Peng, and J. Stein, *Engine-in-the-loop testing for evaluating hybrid propulsion concepts and transient emissions–hmmwv case study*, SAE Transactions, (2006), pp. 23–41.

[41] P. Fritzson, *Principles of object-oriented modeling and simulation with Modelica 3.3: a cyber-physical approach*, John Wiley & Sons, 2014. ISBN: 978-1118859124.

[42] G. Genta and L. Morello, *The Automotive Chassis*, vol. 2, Springer, 2001. ISBN: 978-3-030-35708-5.

[43] M. Gipser, *Ftire: a physically based application-oriented tyre model for use with detailed mbs and finite-element suspension models*, Vehicle System Dynamics, 43 (2005), pp. 76–91.

[44] M. Gipser, *FTire - The tire simulation model for all applications related to vehicle dynamics*, Vehicle System Dynamics, 45 (2007), pp. 139–151.

[45] S. Glumac and Z. Kovacic, *Relative consistency and robust stability measures for sequential co-simulation*, in Proceedings of the 13th International Modelica Conference, Regensburg, Germany, March 4–6, 2019, no. 157, Linköping University Electronic Press, 2019.

[46] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, *Co-simulation: State of the art*, arXiv preprint, (2017).

[47] F. Gonzalez, S. Arbatani, A. Mohtat, and J. Kövecses, *Energy-leak monitoring and correction to enhance stability in the co-simulation of mechanical systems*, Mechanism and Machine Theory, 131 (2019), pp. 172–188.

[48] F. González, M. Á. Naya, A. Luaces, and M. González, *On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics*, Multibody System Dynamics, 25 (2011), pp. 461–483.

[49] M. Green and D. J. Limebeer, *Linear robust control*, Courier Corporation, 2012. ISBN: 978-0-486-48836-3.

[50] M. C. Grunewald, *Functional testing of an electric power steering using hil simulations*, in 5th International Munich Chassis Symposium 2014, Springer, 2014, pp. 455–469.

[51] E. Guillo-Sansano, M. H. Syed, A. J. Roscoe, G. M. Burt, and F. Coffele, *Characterization of time delay in power hardware in the loop setups*, IEEE Transactions on Industrial Electronics, 68 (2020), pp. 2703–2713.

[52] M. Harrer and P. Pfeffer, *Steering handbook*, Springer, 2017. ISBN: 978-3-319-05448-3.

[53] T. Hatakeyama, A. Riccobono, and A. Monti, *Stability and accuracy analysis of power hardware in the loop system with different interface algorithms*, in 2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL), IEEE, 2016, pp. 1–8.

[54] B. Jacobson, *Vehicle dynamics compendium for course mmf062; edition 2016*, tech. rep., Chalmers University of Technology, 2016.

[55] K. Johanastrom and C. Canudas-De-Wit, *Revisiting the lugre friction model*, IEEE Control Systems Magazine, 28 (2008), pp. 101–114.

[56] A. Joshi, *Review of Vehicle Engine Efficiency and Emissions*, SAE Technical Papers, 2020-April (2020), pp. 2479–2507.

[57] M. Karpenko and N. Sepehri, *Hardware-in-the-loop simulator for research on fault tolerant control of electrohydraulic actuators in a flight control application*, Mechatronics, 19 (2009), pp. 1067–1077.

[58] D. Katzourakis, M. Gerard, E. Holweg, and R. Happee, *Design issues for haptic steering force feedback on an automotive simulator*, in 2009 IEEE International Workshop on Haptic Audio visual Environments and Games, IEEE, 2009, pp. 1–6.

[59] A. B. Khaled, L. Duval, M. E. M. B. Gaïd, and D. Simon, *Context-based polynomial extrapolation and slackened synchronization for fast multi-core simulation using fmi*, in MODELICA, Linköping University Electronic Press, 2014, pp. 225–234.

[60] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*, vol. 3, Prentice hall Upper Saddle River, NJ, 2002. ISBN: 978-0130673893.

[61] S. Klein, F. Xia, K. Etzold, J. Andert, N. Amringer, S. Walter, T. Blochwitz, and C. Bellanger, *Electric-motor-in-the-loop: Efficient testing and calibration of hybrid power trains*, IFAC-PapersOnLine, 51 (2018), pp. 240–245.

[62] W. Kortüm and M. Valasek, *4th IAVSD-Herbertov Workshop: 'Modelling and simulation of mechatronic vehicles: Tools, standards and industry demands' - objectives, issues and summary of results*, Vehicle System Dynamics, 33 (2000), pp. 191–201.

[63] M. Krammer, M. Benedikt, T. Blochwitz, K. Alekeish, N. Amringer, C. Kater, S. Materne, R. Ruvalcaba, K. Schuch, J. Zehetner, et al., *The distributed co-simulation protocol for the integration of real-time systems and simulation environments*, in Proceedings of the 50th Computer Simulation Conference, 2018, pp. 1–14.

[64] P. Krus, *Modeling of mechanical systems using rigid bodies and transmission line joints*, Journal of Dynamic Systems, Measurement, and Control (ASME), 121 (1999), pp. 606–611.

[65] R. Kübler and W. Schiehlen, *Modular Simulation in Multibody System Dynamics*, Multibody System Dynamics, 4 (2000), pp. 107–127.

[66] R. Kübler and W. Schiehlen, *Two methods of simulator coupling*, Mathematical and Computer Modelling of Dynamical Systems, 6 (2000), pp. 93–113.

[67] R. Kuffel, R. Wierckx, H. Duchen, M. Lagerkvist, X. Wang, P. Forsyth, and P. Holmberg, *Expanding an analogue hvdc simulator's modelling capability using a real-time digital simulator (rtds)*, in ICDS'95. First International Conference on Digital Power System Simulators, IEEE, 1995, pp. 199–204.

[68] N. Kwok, Q. Ha, M. Nguyen, J. Li, and B. Samali, *Bouc–wen model parameter identification for a mr fluid damper using computationally efficient ga*, ISA Transactions, 46 (2007), pp. 167–179.

[69] G. F. Lauss, M. O. Faruque, K. Schoder, C. Dufour, A. Viehweider, and J. Langston, *Characteristics and design of power hardware-in-the-loop simulations for electrical power systems*, IEEE Transactions on Industrial Electronics, 63 (2015), pp. 406–417.

[70] D. A. Lawrence, *Stability and transparency in bilateral teleoperation*, IEEE Transactions on Robotics and Automation, 9 (1993), pp. 624–637.

[71] P. Le Marrec, C. A. Valderrama, F. Hessel, A. A. Jerraya, M. Attia, and O. Cayrol, *Hardware, software and mechanical cosimulation for automotive applications*, Proceedings of the International Workshop on Rapid System Prototyping, (1998), pp. 202–206.

[72] S. Lentijo, S. D'Arco, and A. Monti, *Comparing the dynamic performances of power hardware-in-the-loop interfaces*, IEEE Transactions on Industrial Electronics, 57 (2010), pp. 1195–1207.

[73] P. Li, *On the numerical stability of co-simulation methods*, PhD thesis, Technical University of Darmstadt, July 2017. Available at `https://tuprints.ulb.tu-darmstadt.de/6606/`.

[74] L. Lichtenstein, F. Ries, M. Völker, J. Höll, C. König, J. Zehetner, O. Kotte, I. Coral, L. Mikelsons, N. Amringer, et al., *Literature review in the fields of standards, projects, industry and science*, tech. rep., 2016.

[75] J. Loof, I. Besselink, and H. Nijmeijer, *Implementation and validation of a three degrees of freedom steering-system model in a full vehicle model*, Vehicle System Dynamics, 57 (2019), pp. 86–107.

[76] T. Melman, J. de Winter, X. Mouton, A. Tapus, and D. Abbink, *How do driving modes affect the vehicle's dynamic behaviour? comparing renault's multi-sense sport and comfort modes during on-road naturalistic driving*, Vehicle System Dynamics, 59 (2021), pp. 485–503.

[77] R. W. Newton, R. E. Betz, and H. B. Penfold, *Emulating dynamic load characteristics using a dynamic dynamometer*, in Proceedings of 1995 International Conference on Power Electronics and Drive Systems. PEDS 95, IEEE, 1995, pp. 465–470.

[78] Nexteer, *Nexteer Produces 60 Millionth Electric Power Steering System*, tech. rep., Feb. 2019.

[79] J. Ok, W. Yoo, and J. Sohn, *New nonlinear bushing model for general excitations using bouc-wen hysteretic model*, International Journal of Automotive Technology, 9 (2008), pp. 183–190.

[80] B. Olivier, O. Verlinden, and G. Kouroussis, *Effect of applied force cosimulation schemes on recoupled vehicle/track problems*, Multibody System Dynamics, 50 (2020), pp. 337–353.

[81] H. B. Pacejka and E. Bakker, *The magic formula tyre model*, Vehicle System Dynamics, 21 (1992), pp. 1–18.

[82] P. Pannu, C. Andersson, C. Führer, and J. Åkesson, *Coupling model exchange fmus for aggregated simulation by open source tools*, in Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015, no. 118, Linköping University Electronic Press, 2015, pp. 903–909.

[83] S. Paran, *Utilization of impedance matching to improve damping impedance method-based phil interface*, PhD thesis, Florida State University, 2013. Available at `https://fsu.digital.flvc.org/islandora/object/fsu%3A183854`.

[84] A. Peiret, F. González, J. Kövecses, and M. Teichmann, *Multibody system dynamics interface modelling for stable multirate co-simulation of multiphysics systems*, Mechanism and Machine Theory, 127 (2018), pp. 52–72.

[85] ——, *Co-simulation of multibody systems with contact using reduced interface models*, Journal of Computational and Nonlinear Dynamics, 15 (2020).

[86] P. E. Pfeffer, M. Harrer, and D. Johnston, *Interaction of vehicle and steering system regarding on-centre handling*, Vehicle System Dynamics, 46 (2008), pp. 413–428.

[87] P. Porteš, P. Kučera, V. Píštěk, J. Fojtášek, and L. Zháňal, *Modern tools for vehicle development*, Engineering Mechanics, 2017 (2017), pp. 54–57.

[88] J. Rahikainen, F. González, and M. Á. Naya, *An automated methodology to select functional co-simulation configurations*, Multibody System Dynamics, 48 (2020), pp. 79–103.

[89] J. Rahikainen, F. González, M. Á. Naya, J. Sopanen, and A. Mikkola, *On the cosimulation of multibody systems and hydraulic dynamics*, Multibody System Dynamics, (2020), pp. 1–25.

[90] W. Ren, *Accuracy Evalaution of Power Hardware-in-the-Loop (PHIL) Simulation*, PhD thesis, Florida State University, 2007. Available at `https://www.proquest.com/openview/effd1ea0813a34dde56f0030feff5221/1?pq-origsite=gscholar&cbl=18750`.

[91] W. Ren, M. Sloderbeck, M. Steurer, V. Dinavahi, T. Noda, S. Filizadeh, A. Chevrefils, M. Matar, R. Iravani, C. Dufour, et al., *Interfacing issues in real-time digital simulators*, IEEE Transactions on Power Delivery, 26 (2010), pp. 1221–1230.

[92] W. Ren, M. Steurer, and T. L. Baldwin, *Improve the stability and the accuracy of power hardware-in-the-loop simulation by selecting appropriate interface algorithms*, IEEE Transactions on Industry Applications, 44 (2008), pp. 1286–1294.

[93] M. Rodic, K. Jezernik, and M. Trlep, *Dynamic emulation of mechanical loads: an advanced approach*, IEE Proceedings-Electric Power Applications, 153 (2006), pp. 159–166.

[94] L. Romano, F. Bruzelius, and B. Jacobson, *Brush tyre models for large camber angles and steering speeds*, Vehicle System Dynamics, (2020), pp. 1–52.

[95] J. Ruan, P. D. Walker, P. A. Watterson, and N. Zhang, *The dynamic performance and economic benefit of a blended braking system in a multi-speed battery electric vehicle*, Applied Energy, 183 (2016), pp. 1240–1258.

[96] S. Sadjina, L. T. Kyllingstad, S. Skjong, and E. Pedersen, *Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation*, Engineering with Computers, 33 (2017), pp. 607–620.

[97] S. Sadjina and E. Pedersen, *Energy conservation and coupling error reduction in non-iterative co-simulations*, Engineering with Computers, 36 (2020), pp. 1579–1587.

[98] S. E. Saidi, N. Pernet, Y. Sorel, and A. B. Khaled, *Acceleration of fmu co-simulation on multi-core architectures*, in Japanese Modelica Conference, vol. 124, 2016, pp. 106–112.

[99] F. Schneider, M. Burger, M. Arnold, and B. Simeon, *A new approach for force-displacement co-simulation using kinematic coupling constraints*, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 97 (2017), pp. 1147–1166.

[100] D. Schramm, M. Hiller, and R. Bardini, *Vehicle dynamics Modeling and Simulation*, vol. 151, Springer, 2014. ISBN: 978-3-540-36045-2.

[101] B. Schweizer, P. Li, and D. Lu, *Explicit and implicit cosimulation methods: stability and convergence analysis for different solver coupling approaches*, Journal of Computational and Nonlinear Dynamics, 10 (2015).

[102] B. Schweizer and D. Lu, *Predictor/corrector co-simulation approaches for solver coupling with algebraic constraints*, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 95 (2015), pp. 911–938.

[103] ——, *Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints*, Multibody System Dynamics, 34 (2015), pp. 129–161.

[104] B. Schweizer, D. Lu, and P. Li, *Co-simulation method for solver coupling with algebraic constraints incorporating relaxation techniques*, Multibody System Dynamics, 36 (2016), pp. 1–36.

[105] R. Serban and E. J. Haug, *Globally independent coordinates for real-time vehicle simulation*, Journal of Mechanical Design (ASME), 122 (2000), pp. 575–582.

[106] S. Sicklinger, V. Belsky, B. Engelmann, H. Elmqvist, H. Olsson, R. Wüchner, and K.-U. Bletzinger, *Interface jacobian-based co-simulation*, International Journal for Numerical Methods in Engineering, 98 (2014), pp. 418–444.

[107] J. Siegers and E. Santi, *Improved power hardware-in-the-loop interface algorithm using wideband system identification*, in 2014 IEEE Applied Power Electronics Conference and Exposition-APEC 2014, IEEE, 2014, pp. 1198–1204.

[108] M. Sjölund, P. Fritzson, P. Krus, and R. Braun, *Towards efficient distributed simulation in modelica using transmission line modeling*, in 3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools; Oslo; Norway; October 3, no. 047, Citeseer, 2010, pp. 71–80.

[109] G. Stettinger, M. Benedikt, M. Horn, J. Zehetner, and C. Giebenhain, *Control of a magnetic levitation system with communication imperfections: A model-based coupling approach*, Control Engineering Practice, 58 (2017), pp. 161–170.

[110] G. Stettinger, M. Benedikt, M. Tranninger, M. Horn, and J. Zehetner, *Recursive fir-filter design for fault-tolerant real-time co-simulation*, in 2017 25th Mediterranean Conference on Control and Automation (MED), IEEE, 2017, pp. 461–466.

[111] G. Stettinger, M. Horn, M. Benedikt, and J. Zehetner, *A model-based approach for prediction-based interconnection of dynamic systems*, in 53rd IEEE Conference on Decision and Control, IEEE, 2014, pp. 3286–3291.

[112] ——, *Model-based coupling approach for non-iterative real-time co-simulation*, in 2014 European Control Conference (ECC), IEEE, 2014, pp. 2084–2089.

[113] M. Törngren, *Fundamentals of implementing real-time control applications in distributed computer systems*, Real-time systems, 14 (1998), pp. 219–250.

[114] O. Tremblay, H. Fortin-Blanchette, R. Gagnon, and Y. Brissette, *Contribution to stability analysis of power hardware-in-the-loop simulators*, IET Generation, Transmission & Distribution, 11 (2017), pp. 3073–3079.

[115] O. Vaculín, W. R. Krüger, and M. Valášek, *Overview of coupling of multibody and control engineering tools*, Vehicle System Dynamics, 41 (2004), pp. 415–429.

[116] A. T. Van Zanten et al., *Evolution of electronic control systems for improving the vehicle dynamic behavior*, in Proceedings of the 6th International Symposium on Advanced Vehicle Control, vol. 2, Citeseer, 2002, pp. 9–17.

[117] A. Veitl, T. Gordon, A. Van De Sand, M. Howell, M. Valasek, O. Vaculin, and P. Steinbauer, *Methodologies for coupling simulation models and codes in mechatronic system analysis and design*, Vehicle System Dynamics, 33 (2000), pp. 231–243.

[118] F. Vinattieri, T. Wright, R. Capitani, C. Annicchiarico, and G. Danisi, *Target setting and structural design of an eps-in-the-loop test bench for steering feeling simulation*, tech. rep., SAE Technical Paper, 2016.

[119] G. G. Wang, *Definition and review of virtual prototyping*, Journal of Computing and Information Science in Engineering (ASME), 2 (2002), pp. 232–236.

[120] F. Wilhelm, T. Tamura, R. Fuchs, and P. Müllhaupt, *Friction compensation control for power steering*, IEEE Transactions on Control Systems Technology, 24 (2015), pp. 1354–1367.

[121] B. Wittenmark, J. Nilsson, and M. Torngren, *Timing problems in real-time control systems*, in Proceedings of 1995 American Control Conference-ACC'95, vol. 3, IEEE, 1995, pp. 2000–2004.

[122] C. Wu, *Co-simulation methods for epas and chassis systems development*, Master's thesis, Chalmers University of Technology, 2018. Available at `https://hdl.handle.net/20.500.12380/255683`.

[123] I. D. Yoo and A. M. Gole, *Compensating for interface equipment limitations to improve simulation accuracy of real-time power hardware in loop simulation*, IEEE Transactions on Power Delivery, 27 (2012), pp. 1284–1291.