Jarkko Rahikainen

ON THE DYNAMIC SIMULATION OF COUPLED MULTIBODY AND HYDRAULIC SYSTEMS FOR REAL-TIME APPLICATIONS

ACTA UNIVERSITATIS LAPPEENRANTAENSIS 881

LUT University

Jarkko Rahikainen

ON THE DYNAMIC SIMULATION OF COUPLED MULTIBODY AND HYDRAULIC SYSTEMS FOR REAL-TIME APPLICATIONS

Dissertation for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 1325 at Lappeenranta-Lahti University of Technology LUT, Lappeenranta, Finland on the 29th of November, 2019, at noon.

Acta Universitatis
Lappeenrantaensis 881

Supervisors Professor Aki Mikkola
LUT School of Energy Systems
Lappeenranta-Lahti University of Technology LUT
Finland

Professor Jussi Sopanen
LUT School of Energy Systems
Lappeenranta-Lahti University of Technology LUT
Finland

Reviewers Professor Bernhard Schweizer
Department of Mechanical Engineering
Technical University Darmstadt
Germany

Dr. Radu Serban
Department of Mechanical Engineering
University of Wisconsin-Madison
USA

Opponents Professor József Kövecses
Department of Mechanical Engineering
McGill University
Canada

Dr. Radu Serban
Department of Mechanical Engineering
University of Wisconsin-Madison
USA

ISBN 978-952-335-446-3
ISBN 978-952-335-447-0 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Lappeenranta-Lahti University of Technology LUT
LUT Yliopistopaino 2019

Abstract

Jarkko Rahikainen
On the dynamic simulation of coupled multibody and hydraulic systems for real-time applications
Lappeenranta, 2019
57 pages
Acta Universitatis Lappeenrantaensis 881
Diss. Lappeenranta-Lahti University of Technology LUT
ISBN 978-952-335-446-3, ISBN 978-952-335-447-0 (PDF), ISSN-L 1456-4491, ISSN 1456-4491

The computer simulation of mechanical systems enables engineers to perform fast prototyping and in-depth system analysis even before a single physical model has been built. However, since mechanical systems are often actuated by other physical systems which have their own internal dynamics, the multibody equations that describe the movement of the mechanism are in these cases not sufficient to model the entire system alone. This has become increasingly important, as the available software solutions have matured such that their use has become an industry standard. For these reasons, coupled simulation has a significant real-life impact.

This dissertation contributes to the coupled simulation of multibody and hydraulic dynamics, the objective being to develop methods and gain insight into the problem. The scope of the work covers real-time applicable methods. Both monolithic and co-simulation approaches are considered in this work and in the included publications. A monolithic formulation based on a semi-recursive multibody method has been proposed and further improved with the introduction of an efficient description for hydraulics. Regarding the co-simulation approaches, the effect and selection of the co-simulation configuration have been studied. The results may prove useful for both analysts and researchers in the field.

Keywords: multibody dynamics, hydraulic dynamics, monolithic simulation, co-simulation, real-time methods

Acknowledgements

This research has been carried out in the Laboratory of Machine Design during the years 2015-2019 at LUT University and from February 2018 to August 2018 at the Mechanical Engineering Laboratory at the University of La Coruña, Campus Ferrol, Spain.

I would like to thank my supervisors Aki Mikkola and Jussi Sopanen for providing the opportunity for the research and for their guidance during these years, and the preliminary examiners, Dr. Radu Serban and professor Bernhard Schweizer, for their efforts in reviewing this work. The participation of the opponents, professor József Kövecses and Dr. Radu Serban, is also highly appreciated.

As it stands, this work would have not been possible without the research visit hosted by Professor Javier Cuadrado at the University of La Coruña. I would like to express my gratitude to Javier and to the whole team at Laboratorio de Ingeniería Mecánica for all of their support. Special thanks go to Francisco "Fran" González and Miguel Naya, whose work is greatly appreciated. I guess I still owe a couple of lunches to some of you there.

I would also like to thank all of the current and the former members of the Laboratories of Machine Design and Machine Dynamics with whom I had a chance to work. I have come a long way to get here, sometimes stressful and full of highs and lows, but it has all been worth the effort.

Finally, thanks to all others who have supported me along the way.

Jarkko Rahikainen
November 2019
Lappeenranta, Finland

This dissertation includes a total of four refereed internationally-published journal articles. The articles are presented below in publication order.

Publication I

Rahikainen, J., Mikkola, A., Sopanen, J., Gerstmayr, J. "Combined semi-recursive formulation and lumped fluid method for monolithic simulation of multibody and hydraulic dynamics." *Multibody System Dynamics*, 44(3), pp. 293-311 November 2018.

Publication II

Rahikainen, J., Kiani, M., Sopanen, J., Jalali, P., Mikkola, A. "Computationally efficient approach for simulation of multibody and hydraulic dynamics." *Mechanism and Machine Theory*, 130, pp. 435-446 December 2018.

Publication III

Rahikainen, J., González, F., Naya, M.A. "An automated methodology to select functional co-simulation configurations." *Multibody System Dynamics*, Accepted August 2019.

Publication IV (Submitted)

Rahikainen, J., González, F., Naya, M.A., Sopanen, J., Mikkola, A. "On the co-simulation of multibody systems and hydraulic dynamics." Submitted to *Multibody System Dynamics*

This section details the contribution of the author in the writing of the included articles. The articles are written under the supervision of Professors Aki Mikkola and Jussi Sopanen from LUT University, Professor Johannes Gerstmayr from the University of Innsbruck, and Doctors Francisco González and Miguel Naya from the University of A Coruña. This dissertation was written under the supervision of Professors Aki Mikkola and Jussi Sopanen at LUT University.

Publication I

The topic of the work was selected as a joint effort by the author and Professors Aki Mikkola and Jussi Sopanen, while Johannes Gerstmayr provided assistance in the development of the methods. The author implemented the chosen methods and was responsible for most of the writing. The other co-authors provided guidance and assistance, and the article was finalized together.

Publication II

The topic of the paper was selected as a joint effort by the authors. The author implemented the method with assistance from the second co-author, Mehran Kiani, and was responsible for most of the writing. Mehran Kiani and Payman Jalali were responsible for the aspects regarding the method based on the singular perturbation theory. Together the authors finalized the paper.

Publication III

The topic of the third publication was decided as a joint effort between the authors. The author implemented the configuration search method with the help of the other co-authors. The original co-simulation implementation was provided by the second co-author. The paper was written and finalized together.

Publication IV

The topic of the paper was decided by the first three authors. The co-simulation implementation, and the main body of the writing process, was a joint effort between the author and the second co-author. Professors Aki Mikkola and Jussi Sopanen provided guidance and assistance, and the authors finalized the paper together.

Alphabetical symbols

A_1, A_2	Area
$\boldsymbol{b}$	Joint-dependent element of velocity transformation matrix
B_c	Bulk modulus of a container
B_e, B_{e1}, B_{e2}, B_{e3}	Effective bulk modulus
B_h	Bulk modulus of hose
B_o	Bulk modulus of oil
c	Viscous coefficient
C, D, B, Q , R	Point
C_t	Throttle valve discharge coefficient
C_v	Semi-empirical coefficient
$\boldsymbol{C}$	Damping matrix
$\boldsymbol{d}$	Joint-dependent element of acceleration transformation
δE	Residual energy
f	Function
$\boldsymbol{f}$	Residual vector
F	Cylinder force
F_d	End-damper force
F_μ	Friction force
g	Gravity
h_1, h_2, h_m, h_f	Internal step size of a subsystem
$\boldsymbol{h}$	Function of pressures
H	Communication step size
i	Iteration number
k	Power bond index
$\boldsymbol{K}$	Stiffness matrix
l	Length of a cylinder
l_1	Piston-side length
l_2	Rod-side length
L, L_h	Length of a rod
$\boldsymbol{M}$, $\bar{\boldsymbol{M}}$ $\bar{\bar{\boldsymbol{M}}}$	Mass matrix
n	Time step index
m	Extrapolation order
m_f	Number of degrees of freedom
m_n	Number of dependent variables
m_m	Number of constraint equations
p, p_1, p_2, p_3	Pressure

p_P, p_T	
Δp	Pressure difference
δP	Residual power
$\boldsymbol{q}$	Vector of dependent coordinates
$\dot{\boldsymbol{q}}$	Vector of dependent velocities
$\ddot{\boldsymbol{q}}$	Vector of dependent accelerations
$\delta\dot{\boldsymbol{q}}$	Vector of virtual velocities
Q_{in}, Q_{out}, Q_{31}, Q_{2V}, Q_{V3}	Volume flow
$\boldsymbol{Q}$, $\bar{\boldsymbol{Q}}$ $\bar{\bar{\boldsymbol{Q}}}$	Force vector
$\dot{\boldsymbol{r}}_b$	Body b velocity
$\boldsymbol{R}$	Velocity transformation matrix
$\boldsymbol{R}_d$	Block-diagonal velocity transformation matrix
Δt	Step size in monolithic formulations
s, s_1, s_2	Actuator length
$\dot{s}$	Rate of actuator stroke
$\ddot{s}$	Acceleration of actuator stroke
t	Time
t_n	Time step n
$\boldsymbol{T}$	Path matrix
u_{k_1}, u_{k_2}	Input of a subsystem via power bond k
$\boldsymbol{u}$, $\boldsymbol{u}_1$, $\boldsymbol{u}_2$	Vector of inputs
U	Control signal
V, V_1, V_2, V_3	Volume
$\boldsymbol{w}$, $\boldsymbol{w}_2$	External input
x_0	Point of derivative approximation
X, Y, Z	Coordinate axis
y_{k_1}, y_{k_2}	Output of a subsystem via power bond k
$\boldsymbol{y}$, $\boldsymbol{y}_1$, $\boldsymbol{y}_2$	Vector of outputs
$\boldsymbol{z}$	Vector of relative coordinates
$\dot{\boldsymbol{z}}$	Vector of relative velocities
$\ddot{\boldsymbol{z}}$	Vector of relative accelerations
$\boldsymbol{Z}$	Velocity-level body coordinates
$\boldsymbol{Z}_b$	Velocity-level body coordinates of body b

Greek symbols

α_P, α_T	Pressure coefficient
$\boldsymbol{\alpha}, \boldsymbol{\Omega}, \boldsymbol{\mu}$	Diagonal matrix of penalty coefficients
δ	Differentiation increment
$\boldsymbol{\Phi}$	Vector of constraint equations

$\dot{\boldsymbol{\Phi}}$	Time derivative of the constraint vector
$\ddot{\boldsymbol{\Phi}}$	Second time derivative of the constraint vector
$\boldsymbol{\Phi_q}$	Constraint Jacobian
$\boldsymbol{\lambda}$	Lagrange multipliers
$\boldsymbol{\lambda}^*$	Iterated Lagrange multipliers
$\boldsymbol{\omega}$	Angular velocity vector
γ_+, γ_-	Flow correction factor
θ_1, θ_2	Angle of body

Abbreviations

CPU	Central processing unit
FMI	Functional Mock-up Interface
FWE	Forward Euler
HiL	Human in the loop
IM	Interface model
I1AL	Index-1 augmented Lagrangian
I3AL-P	Index-3 augmented Lagrangian with projections
LF	Lumped fluid
SitL	System in the loop
RK4	Runge-Kutta 4th order
SP	Singular perturbation
SR	Semi-recursive
TR	Trapezoidal rule

Chapter 1

Introduction

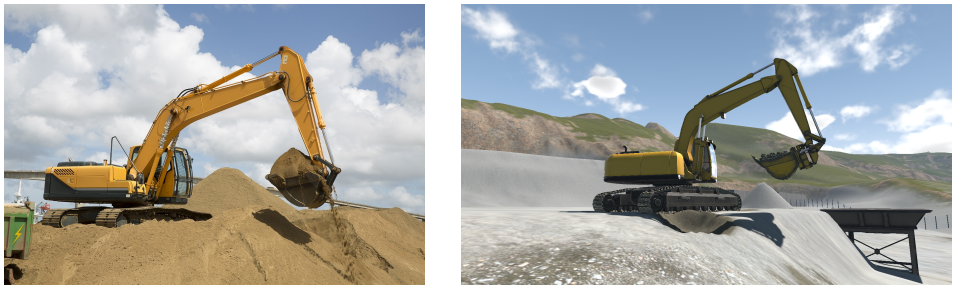
A computer simulation can be defined as the act of imitation of a real-life system or process in a computational environment. The starting point of any simulation is the construction of a model, which is an approximation of the original system or process itself. In a computational environment, this model is a mathematical description of the original system or process, whereas the simulation itself describes the behaviour of the model over time. The use of the simulation tools allow engineers, and in some cases also end-users, better to understand the original system or process, thus providing a tool for improvement in design and performance.

1.1 Motivation for the study

In practical engineering, the possibility for fast prototyping and in-depth system analysis provided make simulation tools attractive, as product design tends to be highly iterative and products tend to become more complex over time. In the field of mechanical engineering, simulation tools can be used provide insight into the behaviour of the studied mechanical system, for instance, the effect of force dispersion between the wheels of a rover [18], and to avoid costly experiments with actual hardware. In addition, in some cases, such as in landing gears of airplanes, simulation tools may provide the only practical and safe way to evaluate the design before it is used in practice.

Simulation tools can also be coupled with physical systems, such as controllers, to verify their behaviour before applying them in the final application, which leads to a setup called *system-in-the-loop* (SitL) simulation. In later phases of a product life cycle, in turn, simulation tools can be used to provide operator training in a realistic environment even before a single physical product has been completed. The latter setup is also known as *human-in-the-loop* (HiL) simulation. In these

cases, a special subcategory of simulation, real-time simulation, illustrated in Fig. 1.1, is required. In contrast, a more common approach is non-real-time, often referred as offline simulation, as in real-time simulation the solution for the system response needs to be obtained within the real-time limit. This requirement arises when a system contains a component, such as a human being, which may not operate in a correct way on other time scales. For these reasons, real-time simulation poses efficiency requirements for both the model and the solution process of the system.



(a) **(b)**

Figure 1.1. A physical system and a corresponding real-time model. Figure **(b)** courtesy of Mevea Ltd (https://mevea.com/).

Whether online or offline simulation is used, the construction of model and the solution process follow the same overall arc. For practical reasons, that is to be able to make generalizable computer implementations, a systematic approach to construct the model is required in both cases. The field of *multibody dynamics* addresses this aspect by providing a methodological approach in modelling mechanisms, i.e. system of bodies linked to each other by kinematic constraints, also referred as joints. Numerical time integration of the force equilibrium equations obtained via the multibody approach, hereafter referred as equations of motion, corresponds to the solution process.

However, in practice, the multibody equations are not sufficient to describe an entire system comprehensively since the mechanical components are often driven by actuator systems. These systems may be, for instance, hydraulics based, and therefore, require different modelling approaches to describe their dynamic behaviour. It should be noted that both the mechanical components and the actuators form a system where dynamics are inherently coupled. However, from the modelling point of view, the different modelling approaches required for each subsystem may favour a decoupled approach. Additionally, there may be issues with different time scales and other aspects with each subsystem, as will be explained later. In any case, for an appropriate and robust approach to system-

level modelling, accurate and effective descriptions must be formed for all dynamic subsystems as well as an accurate and efficient coupling between them.

1.2 Multibody dynamics

A *multibody system* can be defined as a set of two or more bodies connected via joints that constrain their relative motion, and multibody dynamics refers to the movement of such systems. The roots of the field are in classical mechanics, and it became its own field in the 1960s with the introduction of computers, whereas commercial solutions become available in the 1980s [36]. Currently, the approach is used in a wide variety of engineering problems, such as in vehicle system dynamics [1] and biomechanics [41], which are both of great academic interest to multibody system dynamics researchers. A common denominator for the multibody approaches are algorithms aimed for computer implementation. As the applications for multibody dynamics were for a long period of time – and still are to an extent – limited by available and affordable computational power, efficiency has been a key aspect in many of the developed formalisms.

The multibody approaches can be, in general, categorized into two main groups based on the set of coordinates for building the equations of motion. Firstly, the global approaches use a set of coordinates that independently define the position and orientation of each body, whereas, secondly, the semi and fully recursive methods use the relative coordinates between the bodies in a chain [14, 40]. The latter family of methods is also sometimes referred as topological methods, as the system topology plays a role in the formation of the equations. As the fully recursive formulations have been in less active use for reasons such as complex expressions with closed-loop kinematics, hereafter only the semi-recursive formulations are discussed.

Regarding the selection of a multibody method for a problem, firstly, efficiency-wise the optimal method tends to be case dependent [13, 11]. The factors that affect the efficiency of a certain method include, among others, system size, which is the total number of coordinates required to model the system, and system topology. In the solution phase, in turn, the used time integration scheme may also have a significant effect on the efficiency of the simulation [13, 16]. In addition, the computer implementation itself and the tools used, such as automated differentiation, sparse matrix and parallelization methods, may also have a significant effect on efficiency [9, 24]. Thus, as can be seen, no single optimal solution exists for the multibody problem. However, since the multibody approach leads to a system of nonlinear equations, which may be computationally expensive to solve, it can often be assumed, as a rule of thumb, that the approach that leads to the minimum set of variables to be solved tends to become more efficient when the system size increases.

1.3 Hydraulic dynamics

As hydraulic systems have a relatively high power-to-size ratio and relatively simple technology, they have gained ground on applications such as mobile working machinery. They can be modelled based, for instance, on the lumped fluid theory [43], which leans on the assumption that the hydraulic circuit can be split into volumes with internally equally distributed pressure. From the modelling point of view, hydraulic circuits can be challenging, as they often contain components such as valves, which create small volumes into the system, thus introducing high numerical stiffness to the problem [30]. For this reason, when compared to multibody dynamics, the used step sizes in the numerical solution tend to be, in general, an order of magnitude or more smaller, depending on the used integrators.

1.4 Simulation of coupled dynamical systems

To put it in practical terms, the coupling between two dynamical models often reduces to one or more equations in each subsystem, which are dependent on the states of the other subsystem. In the case of multibody and hydraulic dynamics, these state variables can be, for instance, a pressure in a hydraulic cylinder needed to compute the resultant force at the multibody side, and at the hydraulics side, the length and rate of the cylinder, which are needed to compute the pressures.

For the simulation of such systems, in general, two main categories can be established. Firstly, in *monolithic* approaches, the model consists of a single set of equations that are integrated in forward of time within the same solver. Secondly, in *co-simulation* approaches, the sets of equations that describe the model are formed and integrated separately, with information being exchanged between the two simulations at either fixed or adaptive time intervals, often referred to as communication time steps or instances. The latter is also sometimes referred to as *co-integration*, should the same software tool be used to model and integrate each subsystem separately with potentially different integration parameters. An early example of the decoupling of a dynamic problem is the work of Gear and Wells [17], which discusses the use of different step sizes for different parts of a system. More recently, as simulation softwares have matured, efforts have been made to standardize the interfaces between the software programmes, e.g the Functional Mockup Interface (FMI) [7, 6], to ease co-simulation in industrial purposes. It should be noted that co-simulation can be further divided into continuous time, discrete time, and hybrid co-simulation [19, 37], multibody and hydraulic dynamics belonging to the continuous time systems.

Whether the monolithic approach or the co-simulation should be used depends on the needs and practical aspects of the simulation. From the modelling perspective,

simple and accurate coupling of monolithic approaches favours them over the co-simulation methods. However, the use of the monolithic schemes requires all of the equations to be available within the same software, that is, the same software needs to encompass all of the engineering areas required for the simulation. This is rarely practical since each engineering field often has tailored modellers and solvers for their particular needs, or may even be impossible, and therefore, co-simulation schemes are required. Other reasons for the use of the co-simulation schemes include, for instance, performance issues, i.e. the computational load needs to be parallelized, as demonstrated in [33, 29], or the system may be a SitL setup, e.g. as in [25], in which case monolithic approaches are simply not possible. Additionally, co-simulation allows hiding intellectual property by delivering the model and the solver as a "black box" for the analyst, should that be necessary. Subsystems of a different nature also often work on different time scales, as pointed out in the previous subsection, and therefore, the efficiency of the monolithic approaches might suffer.

Regarding the technical challenges of monolithic schemes, as the single set of equations used to model the system eliminates coupling errors, they need relatively little extra effort for an accurate solution to be obtained. The errors arise simply from the modelling simplifications and from the accuracy of the numerical integration. The other challenges may relate to the different magnitudes of variables and different time scales in the coupled systems, which may require the tuning of integration parameters for an efficient solution to be found. Some research efforts have focused on monolithic schemes in the context of multibody and hydraulic dynamics, such as [15]. In [28], in turn, a monolithic scheme was proposed with a combination of penalty-based global formulation and the lumped fluid method. Although relatively high penalty factors were required, the method did obtain good convergence properties. This is likely due to an analytically formed tangent matrix for the numerical integration, which, on the other hand, may not be the most generalizable solution, and to a relatively good efficiency when compared to the multirate method used in the study. Overall, however, research on this area has been relatively scarce, which most likely is due to the practical aspects above that favour co-simulation, and to the issues with the co-simulation that require more research effort.

The decoupling of the original problem in co-simulation schemes, in turn, inherently introduces a number of technical challenges and different options to overcome it. In general, the co-simulation schemes work, such that each subsystem proceeds its states independently, and after a certain number of steps or after a time limit has been reached, the integration is stopped and information is exchanged between other subsystems. Then, the procedure starts again. This discrete-time information exchange poses challenges to the accuracy and stability of the solution, as the independent integration of states inherently leads to *coupling errors*. This

problem is partially solved by the *Gauss-Seidel scheme*, where one subsystem is integrated first, and interpolated values are used in the other subsystem. The alternative is the *Jacobi scheme*, where both subsystems are integrated in parallel. A detailed introduction to these concepts can be found in [31]. In the latter case, the decoupling is more comprehensive, which leads to greater coupling errors, but on the other hand, the parallel structure allows for more efficient computer implementations than in the Gauss-Seidel scheme. Both approaches can, in theory, be made implicit, which alleviates the coupling error issue but comes at the cost of efficiency. Such schemes are presented, for instance, in [38, 27, 39]. The Jacobi scheme in its non-iterative form is well suited for performance critical applications, such as real-time simulation, and is therefore in the focus of this work.

The non-iterative Jacobi scheme, however, often requires additional effort to guarantee the accuracy and stability of the results. In practice, this often means numerical treatment, such as polynomial extrapolation, of the coupling variables. A number of examples can be found from the literature. In [21], the effect of multirate co-simulation techniques was studied, and the results demonstrate that the correct extrapolation method can change depending on the model parameters. An online predictive extrapolation scheme was, in turn, proposed in [4] to optimize the co-simulation performance. Other candidates include a system-energy based method [22], where it was assumed that the co-simulated systems provided their internal energy at the co-simulation interface. In [5], in turn, the energy error at the co-simulation interface was used as a basis of a nearly energy-preserving coupling element. In a similar manner, in [35] the concept of power bonds [32] was utilized to monitor the energy error at the co-simulation interface to select the communication step size adaptively. In the case of co-simulation of mechanical systems, a reduced interface based approach was proposed in [33] to approximate the coupling variables between the communication instances.

While research on co-simulation has been active especially within the past decade in multiple fields of science, it cannot be considered as a solved problem, and further research is needed [19, 34, 37]. The configuration of the co-simulation still seems to be largely based on trial and error, in lack of plug-and-play type approaches. Additionally, the methods for the assessment of co-simulation stability seem still to be progressing [42], as is the effect of the configuration on the accuracy and the stability of the solution. In the context of multibody and hydraulic dynamics, co-simulations have been performed in the industry, but research on solely this topic seems to be limited. The particularities of this coupling include, among others, discontinuities and high stiffness of the hydraulic system, as well as nonlinearity in both of the coupled subsystems.

1.5 Objective and scope of the dissertation

The main objective of this dissertation is to develop methods and to gain insight into the coupled simulation of mechanical and hydraulic systems. This aim is divided into two subcategories. Firstly, monolithic schemes are considered since monolithic solutions can be used as a reference for co-simulation, and secondly, co-simulation approaches are investigated. The scope of this work and the included publications are limited to the field of real-time applications. Thus, efficiency plays a key role in both the monolithic and co-simulation solutions. Regarding the latter, the work presented here is limited to the non-iterative Jacobi scheme co-simulation, and in the case of *Publication III*, to SitL and HiL simulators especially. Regarding the implementation aspects, the scope of the work is limited to the efficiency of algorithms. Therefore, only small academic examples are considered, and efficient programming solutions are not discussed in this work.

The rest of this dissertation is organized as follows: The theoretical background of the study is presented in chapter 2. This includes the used multibody methods, the lumped fluid method and the application of the singular perturbation theory to it, and the used co-simulation schemes. Additionally, a method to compute the coupling error is presented. An overview of the results is given in chapter 3, and conclusions and future research topics are presented in chapter 4.

1.6 Scientific contributions

This dissertation considers the coupled dynamic simulation of multibody and hydraulic systems in the context of real-time capable applications. Both monolithic and co-simulation approaches are utilized, which yields two main categories of scientific contributions.

Firstly, monolithic coupling is considered. A method based on the use of an efficient semi-recursive formulation, coupled with the lumped fluid method, is proposed. The computation of the tangent matrix for the time integration is performed numerically to provide a general approach, and to improve convergence, the differentiation parameter is selected adaptively. The results indicate higher computational efficiency for the proposed method compared to the method presented in the literature, and good convergence properties. As a further improvement to the proposed method, the lumped fluid approach is later exchanged for a method based on the singular perturbation theory. This modification is introduced to prevent the numerical stiffness that may occur with the lumped fluid method to excessively limit the achievable integration step size in a monolithic simulation. The results show further improvement in computational efficiency. These aspects are dealt with in *Publication I* and *Publication II* in more detail.

Secondly, the weak coupling of the explicit Jacobi scheme co-simulation is considered. A key aspect in this scheme is the configuration of the coupling itself, e.g. the extrapolation order and internal solvers, which often need to be selected by trial and error. Based on the energy residuals at the co-simulation interface, a method has been put forward to select a functional configuration in an automated manner in a case where one of the subsystems has an external input or control signal that affects its behaviour. This aspect has been presented in *Publication III*. To gain better understanding of the effect of each configuration parameter on the stability and accuracy of co-simulation, a benchmark study is also presented. Several factors are evaluated, and as a result, their effects on the co-simulation are determined. For instance, it seems that the use of the state variables in the information exchange should be recommended since they may remove direct feed-through from the system. These results are covered in detail in *Publication IV*.

Chapter 2

Coupled simulation of multibody and hydraulic dynamics

As explained in the previous chapter, two main categories, namely monolithic and co-simulation, can be established for the coupled simulation of dynamic systems. This chapter presents the theoretical background of the modelling and simulation approaches applied in mechanical and hydraulic systems as well as the details of the coupling methods. Since the scope of the work covers real-time capable methods, only the non-iterative Jacobi scheme is considered in the co-simulation methods.

2.1 Equations of motion for multibody systems

The field of multibody dynamics provides a methodological approach to analyse the motion of mechanical systems. While multiple approaches exist within the field, two main divisions can be established based on the used coordinates [14]. In the global approaches, the set of coordinates uniquely defines each body, whereas the topological approaches use relative coordinates between the bodies. Regarding the naming conventions of the sets of coordinates employed, in this work, the independent coordinates refer to the minimum set of coordinates, and dependent coordinates refer to the expanded set of coordinates, which are interrelated via constraint equations. This section explains the methods used in the publications.

A brief discussion on the selection of the methods may, however, also be useful. In general, the equation of motion to which the multibody approach leads is simply a force equilibrium equation that consists of external, internal, and constraint forces. While the external and internal forces appear in the equations in a similar manner regardless of the formulation, the introduction of constraints often follows three distinct approaches. These three options are the Lagrange multiplier method,

penalty method, and velocity transformation method, and each of them can be applied also to address the closed loop kinematics in the topological methods [14]. As this work deals also with hydraulics, which introduces numerical stiffness, and the scope is within real-time methods, the multibody approaches are selected accordingly.

Regarding the selection between global and topological methods, the latter leads to smaller problems, i.e it can be expected to lead to a more efficient solution when the system size grows [13] and is, therefore, selected for the proposed monolithic approaches. For co-simulation studies, in turn, the focus is on coupling, and subsequently, the global approaches are utilized for their easier implementation. Regarding the constraint enforcement, penalty methods seem to be a suitable solution for numerically stiff problems [12, 28] and may also suit the simulation of systems with discontinuities, e.g. hydraulics, as a result of the relaxed constraint enforcement. Thus, the penalty approach is used for the constraint enforcement for both global and topological formulations. Additionally, although less importantly from the perspective of this work, penalty methods also have the advantage of being capable of handling singular positions since they produce a positive-definite leading matrix [10, 12, 20], and in the case of the method of relative coordinates, it ensures the selected coordinates to remain valid throughout the simulation [12].

2.1.1 Methods in global coordinates

Consider a multibody system defined with m_n dependent coordinates $\boldsymbol{q}$ and constrained by m_m constraint equations

$$\boldsymbol{\Phi}(\boldsymbol{q}, t) = \mathbf{o}, \tag{2.1}$$

where t is time. The equation of motion for such a system can be derived via the principle of virtual power [14]. To this end, let $\delta\dot{\boldsymbol{q}}$ be the virtual velocities that satisfy the constraints at a stationary time, i.e.

$$\dot{\boldsymbol{\Phi}} = \boldsymbol{\Phi}_{\boldsymbol{q}}^T \delta\dot{\boldsymbol{q}} = \mathbf{o}. \tag{2.2}$$

The principle of virtual power states that the sum of acting forces in a system must equal zero. Since the constraint forces cancel each other out – that is, they produce no virtual work – the virtual power equation can be written as

$$\delta\dot{\boldsymbol{q}}^T \left(\boldsymbol{M}\ddot{\boldsymbol{q}} - \boldsymbol{Q}\right) = \mathbf{o}, \tag{2.3}$$

where $\boldsymbol{M}$ is the mass matrix of the system, $\ddot{\boldsymbol{q}}$ is the accelerations of the dependent coordinates, and $\boldsymbol{Q}$ consists of the external, centrifugal and Coriolis forces. Since the constraint forces do not appear in the above equations, the term in parenthesis,

$\boldsymbol{M\ddot{q}}-\boldsymbol{Q}$, cannot, in general, be set to zero. Consequently, the method of Lagrange multipliers can be applied to add forces in the directions of the constraint violations. The directions are given by the constraint Jacobian $\boldsymbol{\Phi_q}$, whereas their magnitudes $\boldsymbol{\lambda}$ are unknown. Since $\delta\dot{\boldsymbol{q}}^T\boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\lambda}=\mathbf{o}$, Eq. (2.3) can be rewritten as

$$\delta\dot{\boldsymbol{q}}^T\left(\boldsymbol{M\ddot{q}}-\boldsymbol{Q}+\boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\lambda}\right)=\mathbf{o}. \tag{2.4}$$

Since it is always possible to find the set of m_m Lagrange multipliers $\boldsymbol{\lambda}$, the virtual velocities $\delta\dot{\boldsymbol{q}}$ can now be eliminated, that is Eq. (2.4) can be rewritten as

$$\boldsymbol{M\ddot{q}}+\boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\lambda}=\boldsymbol{Q}. \tag{2.5}$$

Index-1 augmented Lagrangian

Eq. (2.5), however, contains m_m+m_n unknowns and only m_m equations. A possible solution to this is a penalty based approach, namely the augmented Lagrangian [14], which approximates the true Lagrange multipliers with the expression

$$\boldsymbol{\lambda}\approx\boldsymbol{\lambda}^*+\boldsymbol{\alpha}\left(\ddot{\boldsymbol{\Phi}}+2\boldsymbol{\Omega\mu}\dot{\boldsymbol{\Phi}}+\boldsymbol{\Omega}^2\boldsymbol{\Phi}\right), \tag{2.6}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\Omega}$, and $\boldsymbol{\mu}$ are diagonal matrices that contain the penalty coefficients, $\dot{\boldsymbol{\Phi}}$ and $\ddot{\boldsymbol{\Phi}}$ are the first and second time derivatives of the constraint equations, respectively, and $\boldsymbol{\lambda}^*$ are the iterated Lagrange multipliers. The substitution of Eq. (2.6) to Eq. (2.5) yields

$$\boldsymbol{M\ddot{q}}+\boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\alpha}\left(\ddot{\boldsymbol{\Phi}}+2\boldsymbol{\Omega\mu}\dot{\boldsymbol{\Phi}}+\boldsymbol{\Omega}^2\boldsymbol{\Phi}\right)+\boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\lambda}^*=\boldsymbol{Q}. \tag{2.7}$$

Notice that if the weighted constraint term is fully satisfied, i.e. $\boldsymbol{\alpha}\Big(\ddot{\boldsymbol{\Phi}}+2\boldsymbol{\Omega\mu}\dot{\boldsymbol{\Phi}}+\boldsymbol{\Omega}^2\boldsymbol{\Phi}\Big)=\mathbf{o}$, then $\boldsymbol{\lambda}^*=\boldsymbol{\lambda}$ and Eq. (2.7) becomes Eq. (2.5). To solve Eq. (2.7) without the addition of algebraic constraint equations, an iterative solution can be established based on Eq. (2.6):

$$\boldsymbol{\lambda}_{i+1}=\boldsymbol{\lambda}_i+\boldsymbol{\alpha}\left(\ddot{\boldsymbol{\Phi}}+2\boldsymbol{\Omega\mu}\dot{\boldsymbol{\Phi}}+\boldsymbol{\Omega}^2\boldsymbol{\Phi}\right)_{i+1}, \tag{2.8}$$

where i is the iteration counter and $\boldsymbol{\lambda}_0=\mathbf{o}$ is used for the initial iteration. This procedure can be embedded into Eq. (2.7) to obtain an iterative solution to the accelerations:

$$\left(\boldsymbol{M}+\boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\alpha\Phi}_{\boldsymbol{q}}\right)\ddot{\boldsymbol{q}}_{i+1}=\boldsymbol{M\ddot{q}}_i-\boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\alpha}\left(\dot{\boldsymbol{\Phi}}_{\boldsymbol{q}}\dot{\boldsymbol{q}}+\dot{\boldsymbol{\Phi}}_t2\boldsymbol{\Omega\mu}\dot{\boldsymbol{\Phi}}+\boldsymbol{\Omega}^2\boldsymbol{\Phi}\right), \tag{2.9}$$

where $\boldsymbol{M\ddot{q}}_0=\boldsymbol{Q}$ is used to initialize the iterative process.

Index-3 augmented Lagrangian with projections

In Eq. (2.9), accelerations are employed as the primary variables in the numerical integration, and the constraints are enforced such that the weighted term $\boldsymbol{\alpha}\left(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\Omega}\boldsymbol{\mu}\dot{\boldsymbol{\Phi}} + \boldsymbol{\Omega}^2\boldsymbol{\Phi}\right)$ equals zero within the given tolerances. However, as Bayo and Ledesma [3] have shown, it is also possible to enforce each constraint $\ddot{\boldsymbol{\Phi}}$, $\dot{\boldsymbol{\Phi}}$, and $\boldsymbol{\Phi}$ individually to zero within numerical accuracy during the integration process. This is achieved with mass-orthogonal projections, which are applied after Newton's iteration has converged. While [3] presents two variants of this method, one for accelerations and another for positions as the primary variables in the integration process, this work utilizes the latter case. This results from the modification proposed in [10], where the mass-orthogonal projections are replaced with mass-damping-stiffness orthogonal projections, which can be used to reduce the computational effort in the projection step.

When positions are used as the primary variables, Eq. (2.7) can be rewritten as

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\alpha}\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\lambda}^* = \boldsymbol{Q}. \tag{2.10}$$

In this case, the iteration of the Lagrange multipliers takes the form

$$\boldsymbol{\lambda}_{i+1} = \boldsymbol{\lambda}_i + \boldsymbol{\alpha}\boldsymbol{\Phi}_{i+1}, \tag{2.11}$$

where, according to [10], $\boldsymbol{\lambda}_0$ is taken to be the $\boldsymbol{\lambda}^*$ obtained from the previous time step.

The index-3 formulation with projections is tied to the used integration scheme, which is, in this case, the implicit single-step trapezoidal rule. The difference equation for velocities at time step $n+1$ takes the form

$$\dot{\boldsymbol{q}}_{n+1} = \frac{2}{\Delta t}\boldsymbol{q}_{n+1} + \hat{\dot{\boldsymbol{q}}}_n = \frac{2}{\Delta t}\boldsymbol{q}_{n+1} - \left(\frac{2}{\Delta t}\boldsymbol{q}_n + \dot{\boldsymbol{q}}_n\right), \tag{2.12}$$

and for accelerations,

$$\ddot{\boldsymbol{q}}_{n+1} = \frac{4}{\Delta t^2}\boldsymbol{q}_{n+1} + \hat{\ddot{\boldsymbol{q}}}_n = \frac{4}{\Delta t^2}\boldsymbol{q}_{n+1} - \left(\frac{4}{\Delta t^2}\boldsymbol{q}_n + \frac{4}{\Delta t}\dot{\boldsymbol{q}}_n + \ddot{\boldsymbol{q}}_n\right). \tag{2.13}$$

These equations can be introduced into Eq. (2.10) to establish a dynamic equilibrium where $\boldsymbol{q}_{n+1}$ are the unknowns, i.e. $\boldsymbol{f}(\boldsymbol{q}_{n+1}) = \boldsymbol{0}$. This system of nonlinear equations can be solved with the Newton-Rhapson iteration, the residual vector being

$$\boldsymbol{f}(\boldsymbol{q}) = \frac{\Delta t^2}{4}\left(\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\alpha}\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\boldsymbol{q}}\boldsymbol{\lambda}^* - \boldsymbol{Q}\right). \tag{2.14}$$

The tangent matrix, in turn, can be approximated numerically, or analytically with

$$\frac{\partial \boldsymbol{f}(\boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\left(\boldsymbol{\Phi}_{\boldsymbol{q}}\boldsymbol{\alpha}\boldsymbol{\Phi}_{\boldsymbol{q}} + \boldsymbol{K}\right), \tag{2.15}$$

where $\boldsymbol{C} = -\frac{\partial \boldsymbol{Q}}{\partial \dot{\boldsymbol{q}}}$ represents the damping and $\boldsymbol{K} = -\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{q}}$ the stiffness in the system.

After the solution to $\boldsymbol{q}_{n+1}$ has converged, the obtained positions satisfy Eq. (2.10), and constraint conditions $\boldsymbol{\Phi} = 0$ are achieved. To also satisfy $\dot{\boldsymbol{\Phi}} = 0$ and $\ddot{\boldsymbol{\Phi}} = 0$, which are not taken into account during the iteration, mass-damping-stiffness orthogonal projections are used. Denoting the obtained velocities and accelerations as $\dot{\boldsymbol{q}}^*$ and $\ddot{\boldsymbol{q}}^*$, respectively, the final solutions $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ can be computed from

$$\frac{\partial \boldsymbol{f}(\boldsymbol{q})}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = \left[\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K}\right]\dot{\boldsymbol{q}}^* - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\alpha}\boldsymbol{\Phi}_t \tag{2.16}$$

for velocities, and for accelerations, from

$$\frac{\partial \boldsymbol{f}(\boldsymbol{q})}{\partial \boldsymbol{q}}\ddot{\boldsymbol{q}} = \left[\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K}\right]\ddot{\boldsymbol{q}}^* - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\alpha}\left(\dot{\boldsymbol{\Phi}}_{\boldsymbol{q}}\dot{\boldsymbol{q}} + \dot{\boldsymbol{\Phi}}_t\right). \tag{2.17}$$

2.1.2 A method in relative coordinates

As the global formulations use a set of coordinates that is often larger than the minimum set of variables required to model the system, their efficiency may be hindered. For this reason, the family of semi-recursive formulations is often utilised in performance critical applications. This work follows the method proposed in [12].

The method defines a mapping between a set of velocity-level body coordinates, which are used to define the dynamic terms, and relative coordinates, which are used in the final equation of motion. The first set of coordinates can be written for each body b as

$$\boldsymbol{Z}_b = \begin{bmatrix}\dot{\boldsymbol{r}}_b \\ \boldsymbol{\omega}_b\end{bmatrix}, \tag{2.18}$$

where $\dot{\boldsymbol{r}}_b$ is the velocity of the body b particle coincident with the origin of the global coordinate system at the current instant in time, and $\boldsymbol{\omega}_b$ is the angular velocity vector. Accelerations can be expressed as $\dot{\boldsymbol{Z}}_b = \begin{bmatrix}\ddot{\boldsymbol{r}}_b^T & \dot{\boldsymbol{\omega}}_b^T\end{bmatrix}^T$. The recursive expressions for the body velocities take form

$$\boldsymbol{Z}_b = \boldsymbol{Z}_{b-1} + \boldsymbol{b}_b\dot{\boldsymbol{z}}_b, \tag{2.19}$$

and for accelerations

$$\dot{\boldsymbol{Z}}_b = \dot{\boldsymbol{Z}}_{b-1} + \boldsymbol{b}_b\ddot{\boldsymbol{z}}_b + \boldsymbol{d}_b. \tag{2.20}$$

The terms $\boldsymbol{b}_b$ and $\boldsymbol{d}_b$ are matrices or vectors, the forms of which depend on the joint type between bodies b and $b-1$, and $\dot{\boldsymbol{z}}$ and $\ddot{\boldsymbol{z}}$ are the first and second derivatives of the relative positions, respectively, which form the second set of coordinates used in the method. A matrix of the body velocities and accelerations can be written as $\boldsymbol{Z} = \begin{bmatrix}\boldsymbol{Z}_1^T & \boldsymbol{Z}_2^T & \dots\end{bmatrix}^T$ and $\dot{\boldsymbol{Z}} = \begin{bmatrix}\dot{\boldsymbol{Z}}_1^T & \dot{\boldsymbol{Z}}_2^T & \dots\end{bmatrix}^T$. A velocity transformation matrix $\boldsymbol{R}$ relates the body and joint coordinates as

$$\boldsymbol{Z} = \boldsymbol{R}\dot{\boldsymbol{z}} = \boldsymbol{T}\boldsymbol{R}_d\dot{\boldsymbol{z}}. \tag{2.21}$$

This relation can be expressed for acceleration as

$$\dot{\boldsymbol{Z}} = \boldsymbol{R}\ddot{\boldsymbol{z}} + \dot{\boldsymbol{R}}\dot{\boldsymbol{z}} = \boldsymbol{T}\boldsymbol{R}_d\ddot{\boldsymbol{z}} + \boldsymbol{T}\dot{\boldsymbol{R}}_d\dot{\boldsymbol{z}}, \tag{2.22}$$

where $\boldsymbol{T}$ is a path matrix that describes the system topology, and $\boldsymbol{R}_d$ is a block diagonal matrix that contains the $\boldsymbol{b}_b$ of each body.

The equations of motion for an open-loop system can be derived in a similar manner as with the global formulation. The virtual power for the open loop takes the form $\delta\boldsymbol{Z}\left(\bar{\boldsymbol{M}}\dot{\boldsymbol{Z}} - \bar{\boldsymbol{Q}}\right)$, where $\bar{\boldsymbol{M}}$ and $\bar{\boldsymbol{Q}}$ are the mass matrix and sum of external and quadratic forces, respectively, expressed in terms of $\boldsymbol{Z}$. With the introduction of Eqs. (2.21) and (2.20), the equation of motion for open loop dynamics can be obtained:

$$\boldsymbol{R}_d^T\boldsymbol{T}^T\bar{\boldsymbol{M}}\boldsymbol{R}_d\boldsymbol{T}\ddot{\boldsymbol{z}} = \boldsymbol{R}_d^T\left(\boldsymbol{T}^T\left(\bar{\boldsymbol{Q}} - \bar{\boldsymbol{M}}\boldsymbol{T}\dot{\boldsymbol{R}}_d\dot{\boldsymbol{z}}\right)\right). \tag{2.23}$$

A penalty formulation similar to the index-3 augmented Lagrangian is used to address the closed loops. Denoting $\bar{\bar{\boldsymbol{M}}} = \boldsymbol{R}_d^T\boldsymbol{T}^T\bar{\boldsymbol{M}}\boldsymbol{R}_d\boldsymbol{T}$ and $\bar{\bar{\boldsymbol{Q}}} = \boldsymbol{R}_d^T\left(\boldsymbol{T}^T\left(\bar{\boldsymbol{Q}} - \bar{\boldsymbol{M}}\boldsymbol{T}\dot{\boldsymbol{R}}_d\dot{\boldsymbol{z}}\right)\right)$, the equation of motion for closed loops takes the form

$$\bar{\bar{\boldsymbol{M}}}\ddot{\boldsymbol{z}} + \boldsymbol{\Phi}_{\boldsymbol{z}}^T\boldsymbol{\alpha}\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\boldsymbol{z}}^T\boldsymbol{\lambda}^* = \bar{\bar{\boldsymbol{Q}}}, \tag{2.24}$$

which can now be solved with the previously given process for the global method, but with relative coordinates. The constraint Jacobian $\boldsymbol{\Phi}_{\boldsymbol{z}}$ can be computed via the chain rule of differentiation, i.e $\boldsymbol{\Phi}_{\boldsymbol{z}} = \boldsymbol{\Phi}_{\boldsymbol{q}}\boldsymbol{q}_{\boldsymbol{z}}$.

2.2 Modelling of hydraulic systems

In contrast to multibody dynamics, the hydraulic systems have more case-dependencies in the modelling procedure, as different approaches may be required to model each component. While the lumped fluid theory used in this study provides a general framework for the structures of the equations, the details may differ, as will be explained next. This section also briefly introduces an alternative method, which can alleviate the numerical stiffness often present in the hydraulic models.

2.2.1 Lumped fluid method

Figure 2.1 illustrates the key concept of the lumped fluid method [43]. In the figure, a volume flow Q_{in} flows into volume V, which has internally equally distributed pressure p, an effective bulk modulus B_e, and an outgoing volume flow Q_{out}. When this method is applied, the hydraulic circuit under study is divided into such volumes for which the pressure is solved.

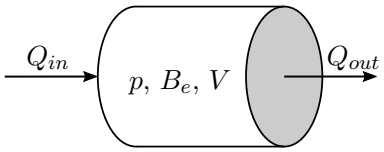


Figure 2.1. An idealised volume according to the lumped fluid theory.

The difference in the ingoing and outgoing volume flows and the stiffness of the volume causes a change in the pressure, the rate of which can be described with a first order differential equation of the form

$$\dot{p} = \frac{B_e}{V}\left(Q_{in} - Q_{out}\right). \tag{2.25}$$

To illustrate the method in practice, consider the hydraulic circuit depicted in Fig. 2.2. The circuit contains a pump and a tank with constant pressures p_P and p_T, respectively, a 4/3 directional valve controlled by a voltage U, a throttle valve, and a hydraulic cylinder of the total length l and lengths l_1 and l_2 at the each side of the piston. The figure depicts the positive directions of volume flows Q_{2V}, Q_{V3}, and Q_{31}. The circuit is split into three volumes, V_1, V_2, and V_3, which, respectively, have the pressures p_1, p_2, and p_3, and effective bulk moduli B_{e1}, B_{e2}, and B_{e3}. The application of Eq. (2.25) to the circuit of Fig. 2.2 yields differential equations for each pressure:

$$\dot{p}_1 = \frac{B_{e1}}{V_1}(Q_{31} - A_1\dot{s}), \tag{2.26}$$

$$\dot{p}_2 = \frac{B_{e2}}{V_2}(A_2\dot{s} - Q_{2V}), \tag{2.27}$$

$$\dot{p}_3 = \frac{B_{e3}}{V_3}(Q_{V3} - Q_{31}), \tag{2.28}$$

where A_1 and A_2 are areas at the each side of the piston, and $\dot{s}$ is the actuator rate. Bulk moduli describe the compressibility of the oil and containers, and take the form

$$B_{e1} = \left(B_o^{-1} + \left(\frac{V_1 B_c}{A_1 l_1}\right)^{-1} + \left(\frac{V_1 B_h}{V_{h_1}}\right)^{-1}\right)^{-1}, \tag{2.29}$$

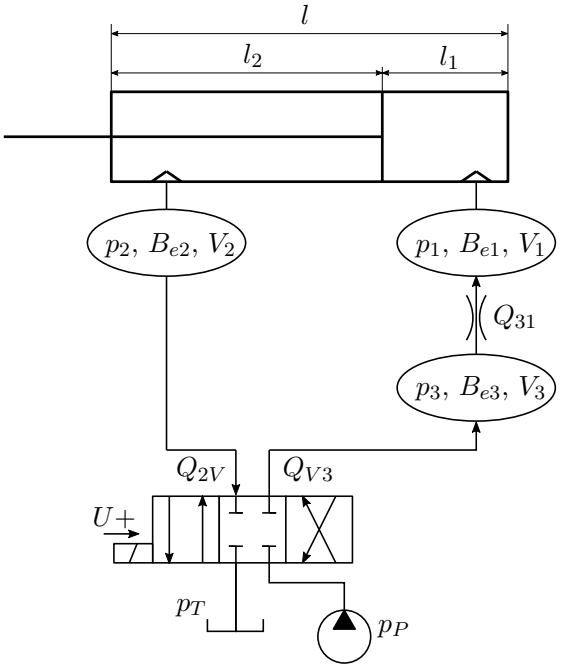


Figure 2.2. A hydraulic circuit.

$$B_{e2} = \left(B_o^{-1} + \left(\frac{V_2 B_c}{A_2 l_1} \right)^{-1} + \left(\frac{V_2 B_h}{V_{h_2}} \right)^{-1} \right)^{-1}, \tag{2.30}$$

$$B_{e3} = \left(B_o^{-1} + B_h^{-1} \right)^{-1}, \tag{2.31}$$

where B_o, B_c, and B_h are the bulk moduli of the oil, the cylinder, and the hoses, respectively, and V_{h_1} and V_{h_2} are the hose volumes.

A semi-empirical model [23], which allows a simple yet accurate model to be built, can be used to model the valves. For a throttle valve, the expression can be written as

$$Q = \begin{cases} C_t \sqrt{|\Delta p|}, & \Delta p > 0, \\ 0, & \Delta p = 0, \\ -C_t \sqrt{|\Delta p|}, & \Delta p < 0, \end{cases} \tag{2.32}$$

where C_t is the semi-empirical coefficient, computed analytically in the case of the simple throttle valve as $C_t = C_d A_t \sqrt{\frac{2}{\rho}}$, C_d being the discharge coefficient, A_t is the throttle area, ρ is the oil density, and Δp is the pressure difference over the

valve. For the directional valve, in turn, the semi-empirical model takes the form

$$Q = \begin{cases} C_v U \sqrt{|\Delta p|}, & \Delta p > 0, \\ 0, & \Delta p = 0, \\ -C_v U \sqrt{|\Delta p|}, & \Delta p < 0, \end{cases} \tag{2.33}$$

where C_v is the semi-empirical coefficient of the directional valve and U is a feedback signal that corresponds to the spool position.

The cylinder produces a force which can be computed based on the pressure difference at each side of the piston and on the friction and end damper contact forces:

$$F = p_1 A_1 - p_2 A_2 - F_\mu + F_d, \tag{2.34}$$

where $F_\mu = F_\mu(\dot{s})$ is a friction force caused by the sealing, and $F_d = F_d(s, \dot{s})$ is the end damper force.

2.2.2 A method based on the singular perturbation theory

The use of Equation (2.25) tends to lead to numerically stiff equations, should one of the idealized volumes be relatively small and high in stiffness. This is often the case when valves are to be modelled. The numerical stiffness introduced by the small volume reduces the achievable integration step size, and therefore, the computational efficiency. The application based on the singular perturbation theory presented in [30] and utilized in *Publication II* allows the set of equations that describe a hydraulic system, formed according to Equation (2.25), to be reduced such that the small volume is described with an algebraic equation instead of a differential equation. This allows the small volume to be ignored by the integrator, which thus increases the achievable step size.

As shown in [30] and discussed in more detail in *Publication II*, in the case of the hydraulic circuit depicted in Fig. 2.2, the computation of pressure p_3 can be reduced to

$$p_3 = \begin{cases} \frac{p_1 + \alpha_P p_P}{1 + \alpha_P}, & U > 0, \\ \frac{p_1 + \alpha_T p_T}{1 + \alpha_T}, & U < 0, \end{cases} \tag{2.35}$$

where, in this case,

$$\alpha_P = \alpha_T = \left(\frac{U C_v}{C_t} \right)^2. \tag{2.36}$$

The pressure p_3 is now evaluated algebraically within the integration process instead of numerical integration. This modification, however, introduces errors to the volume flow Q_{31} since it is dependent on the value of p_3, which can be, in

turn, corrected by the formula

$$Q_{31} = \begin{cases} \gamma_+ \bar{Q}_{31}, & U > 0, \\ \gamma_- \bar{Q}_{31}, & U < 0, \end{cases} \tag{2.37}$$

where $\bar{Q}_{31}$ denotes the uncorrected volume flow, and γ_+ and γ_- are the correction factors [30].

2.3 Monolithic coupling

A straightforward approach to couple the multibody and hydraulic dynamics during time integration is the use of a monolithic approach, in which the same step size is used for the entire system. *Publication I* and *Publication II* study this approach in the case where the semi-recursive formulation is coupled with the hydraulic dynamics.

In the monolithic approach, the model for the system takes the form

$$\begin{aligned} \bar{\bar{\boldsymbol{M}}}\ddot{\boldsymbol{z}} + \boldsymbol{\Phi}_{\boldsymbol{z}}^T \boldsymbol{\alpha} \boldsymbol{\Phi} + \boldsymbol{\Phi}_{\boldsymbol{z}}^T \boldsymbol{\lambda}^* &= \bar{\bar{\boldsymbol{Q}}}\left(\boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{p}\right), \\ \dot{\boldsymbol{p}} &= \boldsymbol{h}\left(\boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{p}\right), \end{aligned} \tag{2.38}$$

where $\boldsymbol{p}$ is the pressure vector, $\dot{\boldsymbol{p}}$ contains their derivatives, and $\boldsymbol{h}\left(\boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{p}\right)$ are the pressure variation equations. This set of equations is integrated according to the index-3 formulations described earlier. However, to avoid the analytical computation of the directional matrix, which can become complex and prone to errors, and to provide a general method, numerical computation is proposed for the monolithic approach. Forward difference is used for this purpose, the general form of which is

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0 + \delta) - f(x_0)}{\delta}, \tag{2.39}$$

where x_0 is the point at which the derivative is approximated. The differentiation increment δ is computed at each time step with a formula motivated by Brenan et al. [8] to address the difference in magnitude at position and pressure variables:

$$\delta = 1 \times 10^{-8} \max\left(1 \times 10^{-2}, |x_0|\right), \tag{2.40}$$

where 1×10^{-2} limits the minimum value of δ to 1×10^{-10}.

For the projection of the constraints, according to Eqs. (2.16) and (2.17), the weight matrix needs to be computed, should the numerical approach for the tangent matrix be used. This can be achieved with

$$\left[\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K}\right] = \frac{\partial \boldsymbol{f}(\boldsymbol{z})}{\partial \boldsymbol{z}} - \frac{\Delta t^2}{4}\left(\boldsymbol{\Phi}_{\boldsymbol{q}}^T \boldsymbol{\alpha} \boldsymbol{\Phi}_{\boldsymbol{z}}\right), \tag{2.41}$$

where $\frac{\partial \boldsymbol{f}(\boldsymbol{z})}{\partial \boldsymbol{z}}$ corresponds to the rows and columns of the tangent matrix that correspond to the multibody problem.

2.4 Co-simulation

In contrast to the monolithic approaches, in co-simulation, the multibody and hydraulic models are integrated separately and information is exchanged between the systems. This is illustrated in Fig. 2.3, where, for the sake of simplicity, two subsystems are integrated in forward of time and an orchestrator algorithm manages the overall simulation process. In the figure, $\int_1$ and $\int_2$ are the internal integrators and h_1 and h_2 are the internal step sizes of each subsystem, and in the manager, H is the communication step size, i.e. the time difference between each information exchange. As the notation suggests, all three step sizes can, in general, differ from each other. The figure also illustrates the flow of information during the simulation process, vectors $\boldsymbol{u}_1$ and $\boldsymbol{y}_1$ being the inputs and outputs as reported by Subsystem 1, respectively, and $\boldsymbol{u}_2$ and $\boldsymbol{y}_2$ are the inputs and outputs of Subsystem 2, respectively. Finally $\boldsymbol{w}_2$ illustrates the possible external inputs, such as those from a human operator in HiL setups, that may be present in the system.

Figure 2.3. A generic co-simulation scheme.

In general, the co-simulation schemes can be divided into two categories based on the integration order, and further, into iterative and non-iterative schemes. Regarding the integration order, these categories are, as discussed earlier, the Gauss-Seidel scheme, where one subsystem is integrated first with extrapolated inputs, followed by the second subsystem with interpolated inputs, and the Jacobi scheme, where both subsystems are integrated in parallel using extrapolated inputs. Naturally, with an otherwise identical setup, the Gauss-Seidel scheme leads to lesser extrapolation errors, but this comes with an efficiency cost as the integration needs to be sequential. Thus, the Jacobi scheme seems to be more suited to real-time applications and is utilised in this work. Similarly, iterative schemes can alleviate the problem with the extrapolation error, but the introduced efficiency cost makes them less suited for performance critical applications, and thus, the non-iterative form is selected for this work.

To illustrate the information exchange in the used Jacobi scheme co-simulation, Fig. 2.4 presents the process with two subsystems between communication instances t_n and t_{n+1} with scalar inputs and outputs. At t_n, each subsystem sends

and receives inputs $u_k(t_n)$ and outputs $y_k(t_n)$ and proceeds to integrate its states until t_{n+1} is reached, at which point outputs $y_k(t_{n+1})$ are reported and inputs $u_k(t_{n+1})$ are received, and the process starts again. As each subsystem integrates its states independently, this scheme allows parallel execution.

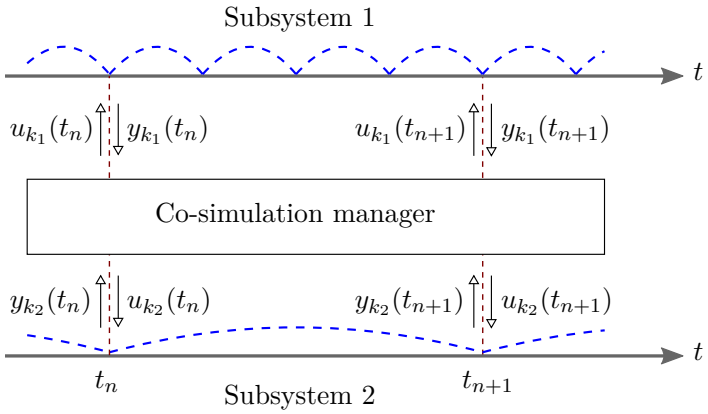


Figure 2.4. Jacobi scheme co-simulation.

2.4.1 Accuracy evaluation

In the Jacobi scheme co-simulation, states of the other subsystems are only known at the communication instances, and therefore, in general, the information needs to be extrapolated until the next communication instance is reached. This is often performed via simple polynomial extrapolation based on, for instance, Lagrange polynomials or least squares fitting, the former of which is utilized in this work. While approaches based on data extrapolation can improve the stability and accuracy of the co-simulation [4, 21], the extrapolated data cannot be expected, in general, to match the actual solution, and therefore, extrapolation errors are introduced to the solution. This eventually poses challenges to the accuracy and, consequently, to the stability of the entire solution. A detailed discussion on the topic is presented in [35], proposing an approach based on the concept of power bonds [32] to compute the introduced error and to use it for adaptive step size control.

The method presented in [35] is utilized in *Publication III* and *Publication IV* of this study and is, for the sake of clarity, presented here. To illustrate the method, consider again the information exchange in Fig. 2.4. In the figure, subindex k refers to power bond k between the subsystems, k_1 and k_2 being the power ports. Subsystem 1 reports the total transmitted power P_{k_1} at the communication time t as

$$P_{k_1}(t) = \tilde{u}_{k_1}(t) y_{k_1}(t), \tag{2.42}$$

where $\tilde{u}_{k_1}(t)$ is the extrapolated value of the input u_{k_1} at the end of the communication step, and y_{k_1} is the computed output. Similarly, Subsystem 2 reports the total transmitted power via k_2 as

$$P_{k_2}(t) = \tilde{u}_{k_2}(t) y_{k_2}(t), \tag{2.43}$$

where $\tilde{u}_{k_2}(t)$ and $y_{k_2}(t)$ are the input and output, respectively.

In an ideal case, no power is lost in the coupling, and the following balance holds:

$$-(P_{k_1} + P_{k_2}) = 0. \tag{2.44}$$

However, in practice, coupling errors are an inherent issue in non-iterative co-simulation, which results in the violation of Eq. (2.44), and therefore,

$$-(P_{k_1} + P_{k_2}) \neq 0. \tag{2.45}$$

As energy is power integrated over time, this spurious power introduced in the co-simulation interface leads to the accumulation of energy errors during the simulation.

However, the violation described above can be exploited to monitor the coupling accuracy. Eq. (2.45) can be rewritten as

$$\delta P_k = -(P_{k_1} + P_{k_2}), \tag{2.46}$$

where δP_k is the residual power, i.e. the incorrectly added power, of power bond k. For multiple power bonds, the equation reads

$$\delta P = -\tilde{\boldsymbol{u}}^T \boldsymbol{y}, \tag{2.47}$$

where $\tilde{\boldsymbol{u}}$ and $\boldsymbol{y}$ are the vectors of inputs and outputs, respectively. For time step t_{n+1}, following the notation in Fig. 2.4, the power residual reads

$$\delta P_k(t_{n+1}) = -\tilde{\boldsymbol{u}}_k(t_{n+1})^T \boldsymbol{y}_k(t_{n+1}), \tag{2.48}$$

which can be implemented in the orchestrator software in a straight, assuming the extrapolation method is known or the extrapolated values $\tilde{\boldsymbol{u}}_k(t_{n+1})$ are otherwise available.

The corresponding residual energy can now be computed with a piece-wise defined integral

$$\delta E_k(t_{n+1}) = \int_{t_n}^{t_{n+1}} \delta P_k(t)\, dt, \tag{2.49}$$

which gives the exact amount of incorrectly added or removed energy in the interface. Should $m = 0$, where m is the extrapolation order to be used as was the case in [35], the residual energy can be approximated as

$$\delta E_k(t_{n+1}) \approx \delta P_k(t_{n+1}) H(t_{n+1}), \tag{2.50}$$

where $H(t_{n+1})$ is the communication step size. For higher order extrapolation, a higher order quadrature formula should be used. The trapezoidal rule reads, in this case, as

$$\delta E_k(t_{n+1}) \approx -\frac{H}{2} \left(\delta P_k(t_n) + \delta P_k(t_{n+1}) \right), \tag{2.51}$$

which can be assumed to be appropriate for $m = 1$. While higher order quadrature rules would provide a more accurate estimation in the case of $m > 1$, the outputs $\boldsymbol{y}_k$ are only known at the communication instances, and therefore, their use is limited. For this reason, the trapezoidal rule is used in this work for $m \geq 1$.

In theory, the size of the above described energy residuals is $\mathcal{O}(H^{m+1})$ when the simulation is stable [35]. In *Publication III* this property is monitored with different configurations to automatically determine functional settings for the co-simulation. A more detailed discussion on the method can be found in section 3.2.

Chapter 3

Summary of the findings

The objective of this work is to develop methods and gain insight into the coupled simulation of mechanical and hydraulic systems. To this end, numerical examples were implemented and then used to evaluate the proposed methods. Monolithic approaches were considered first, followed by studies on the non-iterative Jacobi scheme co-simulation. This chapter summarizes the results obtained in *Publication I, II, III*, and *IV*, and has been divided into two main sections according to the approaches taken.

3.1 Monolithic approaches

Monolithic methods for the coupled simulation of multibody and hydraulic systems were developed in the first two publications. For this purpose, a numerical example illustrated in Fig. 3.1 was constructed to evaluate the proposed methods. In both publications, the mechanism was the same, whereas for the latter publication, the size of small volume between the valves in the hydraulic circuit, illustrated in Fig. 2.2, was decreased. In Fig. 3.1, p_1 and p_2 are the pressures at the each side of the piston, $\boldsymbol{s}$ is the cylinder length expressed in global coordinates, C is the cut-joint location, D is the cylinder attachment point, and the revolute joint between the first link and ground is located at the origin of the global coordinate system.

Publication I

A monolithic formulation that couples the semi-recursive formulation with the lumped fluid method, both of which have been described in the previous chapter, was studied in the first included publication. The work can be seen as a

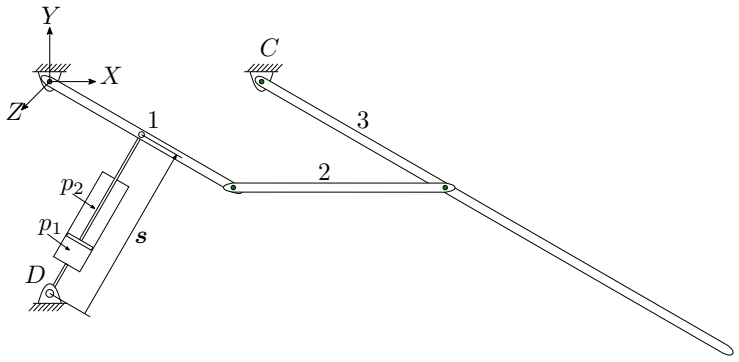


Figure 3.1. A four-bar mechanism actuated by a hydraulic cylinder used in *Publication I* and *Publication II*.

continuation to the work of Naya et al. [28], who used a global formulation. In addition to the use of the semi-recursive method, which tends to become faster when the system grows [12], a numerically obtained tangent matrix was used for its generality and ease of implementation. To improve the convergence properties, which may suffer as the magnitude of position and pressure variables can be rather large, the differentiation increment was computed adaptively based on the magnitude of the variable on which the differentiation was performed. The method shown in [28] was used with Cartesian coordinates as a point of comparison.

The proposed method was evaluated with the numerical case example described above. As the hydraulic system introduces stiffness to the system, for good convergence properties to be obtained the penalty factor α of Eq. (2.24) needed to be increased to 1×10^{11} from the typical range of 1×10^7 to 1×10^9. A comparison to [28] suggests that the penalty factor may need to be selected based on the hydraulic circuit. This penalty factor does not affect to the solution itself, but it has an effect on the rate of convergence in the Newton's iteration. Table 3.1 shows the average and maximum number of iterations with different step sizes and relative CPU times. In the table, Δt is the integration step size, SR (num.) is the semi-recursive formulation with a fully numerical tangent matrix, SR (sym.) is a partially analytic, and AL is the augmented Lagrangian of Eq. (2.10). As the table shows, the proposed method may lead to a low number of iterations in the process, and therefore, can be considered real-time capable. Regarding efficiency, the method was implemented in the Matlab environment, and therefore, no absolute computing times are shown. However, the programme was constructed to enable comparative relative performance. As Table 3.1 shows, the proposed method exhibits better computational efficiency compared to its global counterpart.

To verify the correctness of the method, Fig. 3.2 displays the energy balance E_b during the simulation. The figure shows acceptable convergence properties even

Table 3.1. Average and maximum number of iterations, and relative CPU times.

Δt	Iterations - SR (num.)		Iterations - SR (sym.)		Iterations - AL		CPU times
	Avg.	Max.	Avg.	Max.	Avg.	Max.	AL/SR(num.)
0.5 ms	1.63	4	1.63	4	2.10	5	4.18
1 ms	1.78	3	1.78	3	2.19	5	3.99
2 ms	1.85	4	1.85	4	2.11	4	3.53
3 ms	1.90	4	1.90	4	2.15	4	3.78
4 ms	2.03	6	2.03	6	2.27	7	3.69
5 ms	2.37	17	2.37	17	2.60	17	3.56

for the largest converged step of 8 ms, the maximum energy drift being about 9.6 J and otherwise less than $\pm 5\,J$. To put this in context, the actuator work and change in the potential energy were about 1400 J during the simulation, i.e. the energy drift was mostly less than $\pm 0.4\,\%$ of the variations of the potential energy. These results also indicate that the robustness of the multibody method presented in [12] is preserved when coupled with hydraulic dynamics.

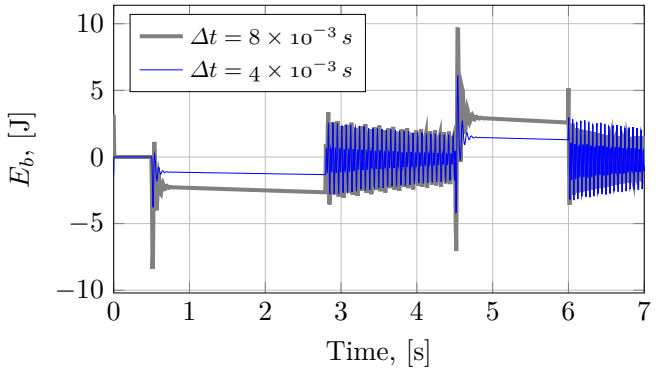


Figure 3.2. Energy balance during the simulated work cycle.

In conclusion, *Publication I* presents a monolithic method that couples the semi-recursive method with the lumped fluid theory. The resultant approach can achieve convergence properties good enough for the number of iterations to be fixed to a low constant, thus yielding a real-time capable solution. The computational efficiency of the method seems to be an improvement on its global counterpart.

Publication II

To prevent the numerical stiffness of the hydraulic system to excessively reduce the integration step size, the method based on the singular perturbation theory [30]

was introduced to the monolithic scheme to improve computational efficiency. As the method based on the singular perturbation theory is relatively new, few studies have dealt with its coupling to mechanical systems. The proposed method is compared to that of *Publication I*.

The theoretical background of the method is presented in [30] and implementation details in section 2.2.2. As explained, the differential equation of Eq. (2.28) is replaced with an algebraic equation, i.e. Eq. (2.35), without a significant change in the accuracy of the solution [30]. Fig. 3.3 depicts both the practical effect of this modification and the volume flow over the throttle valve. As the figure shows, the solution obtained with the lumped fluid method oscillates slightly, whereas the method based on the singular perturbation theory removes this high frequency component from the solution. While minor differences can be seen, the overall solutions with both of the methods agree.

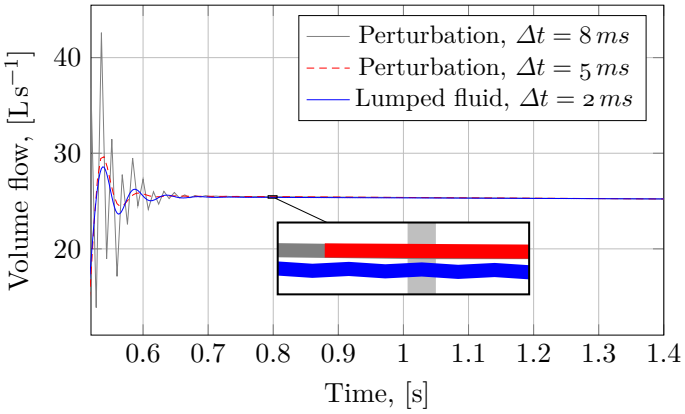


Figure 3.3. The volume flow Q_{31} with the lumped fluid and the perturbation method after opening of the valve.

As the proposed method leads to reduced overall numerical stiffness in the system, larger step sizes can be achieved, and the computational efficiency is therefore improved. Table 3.2 presents the number of iterations and relative CPU times for the compared methods, and as can be seen, the results support the hypothesis. The average number of iterations remains lower than with the lumped fluid theory, and also efficiency obviously improves. Similarly to *Publication I*, a penalty factor of 1×10^{11} was required for the achievement of good convergence properties. However, these results may be partially contributed by the lower problem size that the removal of the differential equation yields. Nevertheless, the method based on the singular perturbation theory enables the monolithic simulation to converge with significantly larger step sizes.

To conclude, the method proposed in *Publication II* shows improved computational

Table 3.2. Number of iterations and relative CPU times.

	Iterations - lumped fluid (LF)		Iterations - perturbation (P)		CPU times	
Δt	Avg.	Max.	Avg.	Max.	P/LF	P/LF(2 ms)
0.5 ms	1.76	4	1.62	4	0.83	2.34
1 ms	2.26	4	1.76	4	0.71	1.24
2 ms	2.52	4	1.81	3	0.66	0.66
3 ms	fail	fail	1.83	4	-	0.42
5 ms	fail	fail	1.85	3	-	0.27
8 ms	fail	fail	1.90	4	-	0.18

efficiency over the method proposed in *Publication I*, most significantly by enabling larger steps to be taken during the integration process. Solutions for both of the studied methods agree, although the method proposed in *Publication II* requires the correction factor of Eq. (2.37) to be tuned. However, to address the trial and error based search for the correction factors, more research is needed to automate this part of the modelling process.

3.2 Co-simulation

The viewpoint on the coupling problem was switched from monolithic to co-simulation for *Publication III* and *Publication IV*. While multiple co-simulation approaches exist, the discussion here is limited to the non-iterative Jacobi scheme co-simulation. Similarly to the monolithic approaches, the numerical examples were recycled also in the co-simulation experiments. Two cases are utilized in these two publications. The hydraulic circuit again corresponds to that presented in section 2.2, whereas the mechanical system is illustrated in Fig. 3.4. The system is a planar mechanism that is modelled in natural coordinates, the set of coordinates being in this case $\boldsymbol{q} = \begin{bmatrix} x_P & y_P & \theta_1 & x_R & y_R \end{bmatrix}^T$. The mechanism is attached to the ground via a revolute joint at point O, and the cylinder is attached to the ground at point B and to body 1 at point P. A revolute joint connects the bodies at point Q, L and L_h being the lengths of the links. At point R, a point mass is attached to link 2. The cylinder length is denoted as s_1, g is gravity, and θ_2 is the angle of the second link.

Fig. 3.5 presents the second case example. The mechanical properties of the system are the same as in the single actuator model. However, from the co-simulation point of view, this model is significantly more challenging, as the cylinders are added as independent circuits and are therefore only indirectly coupled via the mechanism.

The co-simulation is arranged such that for *Publication III*, Lagrange polynomials of the order m and the integration of exchanged variables were implemented,

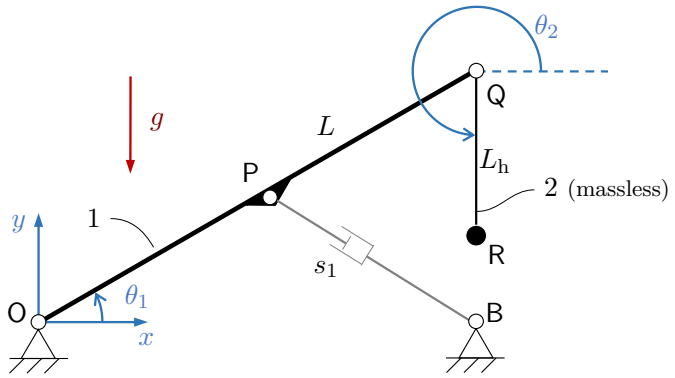


Figure 3.4. A two-dimensional single actuator model used in *Publication III* and *Publication IV*.

and for *Publication IV*, an interface model (IM) [33] was also added. For the mechanical subsystem, two solution methods have been implemented: I1AL and I3AL-P. For the hydraulic subsystem, in turn, forward Euler (FWE), and trapezoidal rule (TR) integrators have been implemented. The extrapolation methods were implemented in the co-simulation manager. Additionally, the coupling was arranged such that the mechanical subsystem sends length, rate, and acceleration of the actuator stroke, that is s, $\dot{s}$, and $\ddot{s}$, respectively, to the hydraulic subsystem, which, in turn, can send either cylinder force F_s or pressures p_1 and p_2 to the mechanical system. It has been assumed that $H = h_m$, where h_m is the step size in the mechanical subsystem, and that h_h, which is the step size in the hydraulic subsystem, forms a matching grid with H. Table 3.3 summarizes the available configuration parameters.

Table 3.3. Possible configurations for the co-simulation in *Publication III*.

Option	Mech.	Hyd.
Extrapolation	$m = 0$	Orders 0-4
Integration of exchanged variables	-	s and $\dot{s}$
Integrator	I3AL-P/I1AL	TR/FWE
Step size	$h_\mathrm{m} = H$	$h_\mathrm{h} = 0.1\,ms < H$

Publication III

In a monolithic simulation, the coupling of the problems is straightforward and the configuration of the simulation, i.e. integration step sizes, etc., is similar to

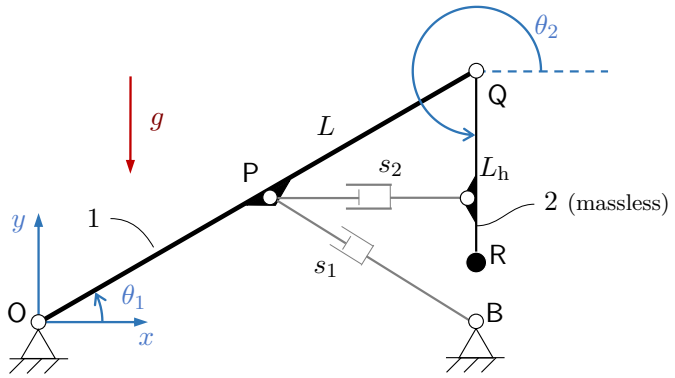


Figure 3.5. A double actuator model used in *Publication III* and *Publication IV*.

that of a single discipline simulation. In co-simulation, in turn, there are, firstly, two or more solvers to be configured, and secondly, the information exchange, i.e. the selection of the extrapolation method and possibly its order, to be configured before a stable and efficient system-level simulation can be obtained. As can be concluded from the result presented in [21], this process may be laborious, as the optimal configuration may depend on the model parameters even if the general structure stays the same. *Publication III* puts forward a method to address this issue, helping to identify a functional configuration in an automated manner.

The foundations of the proposed method are in the residual energy ([35]) described in section 2.4.1. Fig. 3.6 illustrates the behaviour of the accumulated energy residual with a simulation with different extrapolation methods, m being the order of Lagrange polynomials and $I2$ referring to the integration of the position and velocity variables within the hydraulic subsystem. In theory, these curves should be the size of $\mathcal{O}(H^{m+1})$ [35]. When the simulation becomes unstable, this property is lost, i.e. $\sum_i |\delta E_k(t_{n+1})| >> \mathcal{O}(H^{m+1})$. Since the discontinuities in the hydraulic system may deteriorate the theoretical result, the proposed method for selecting functional configurations assumes that in the stable region, the accumulated energy residual behaves approximately linearly. The method assumes that the internal integrators of the simulators are accurate enough for the purposes of the simulation, and the scope of the method is in the cases where stable simulation can be considered accurate enough, such as often in HiL and SitL simulators.

The algorithm itself is presented in Fig. 3.7. In the figure, H_{ini} and H_{inc} are, respectively, the initial communication step size and the increment that is added to it until simulation stability is lost for the current configuration. In the next

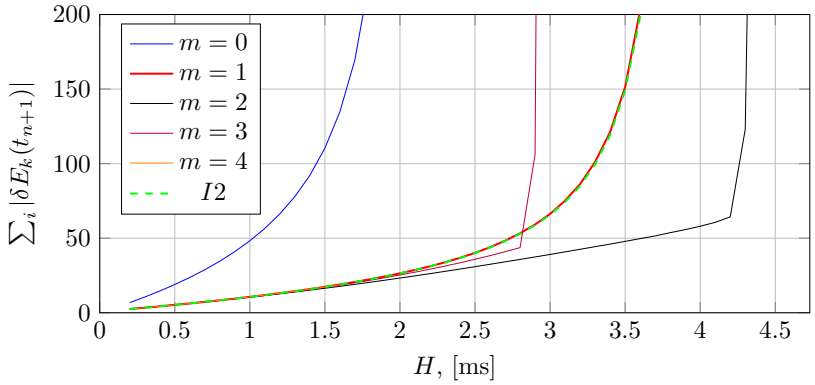


Figure 3.6. Energy residuals for the single actuator model.

step, $\boldsymbol{w}$ are the external inputs, which, as they may not be known in advance, are replaced with test inputs described in *Publication III*. The rest of the process is relatively straightforward: the simulation runs with experimental inputs, and H is increased until stability is lost. This process is repeated until each possible configuration is experimented, at which point the results are parsed for the user.

Table 3.4 shows the parsed results for each test produced by the algorithm for the single-actuator model. As can be seen, the method provides, if not an optimal configuration, a functional one for the studied co-simulation model, should the configuration with the smallest H be selected as the final solution. In the table, $\int_m$ and $\int_h$ refer to the integrators in the mechanical and hydraulic subsystems, respectively, and $H_{0.2\,ms}$ is the final value of H with stable results, incremented with $H_{\text{inc}} = 0.2\,ms$.

Table 3.4. Results of the pre-defined test for the single-actuator model.

Test function	Extrapolation	$\int_m$	$\int_h$	$H_{0.2\,ms}$
Step	$m = 2$	I1AL	Euler	4.0 ms
Ramp	$m = 2$	I1AL	Euler	5.0 ms
Impulse	$m = 2$	I1AL	Euler	3.6 ms
Sinusoidal 17.5 Hz	$m = 2$	I1AL	Euler	2.6 ms
Sinusoidal 35 Hz	$m = 3$	I1AL	Euler	2.8 ms
Sinusoidal 70 Hz	$m = 2$	I1AL	Euler	4.4 ms

Publication IV

As discussed above, the selection of the configuration may have a significant effect on the achievable communication step size and on the introduced coupling

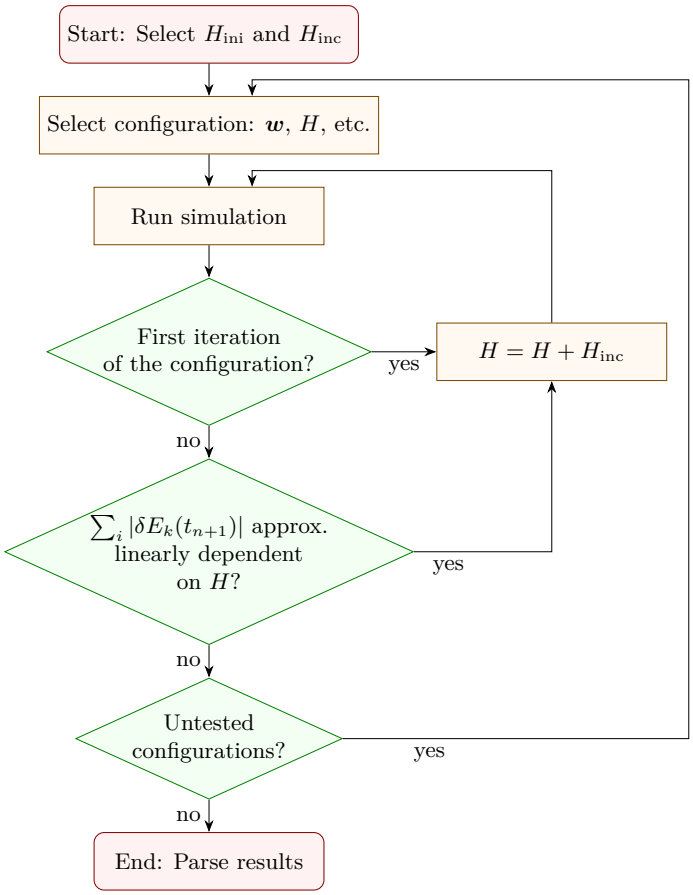
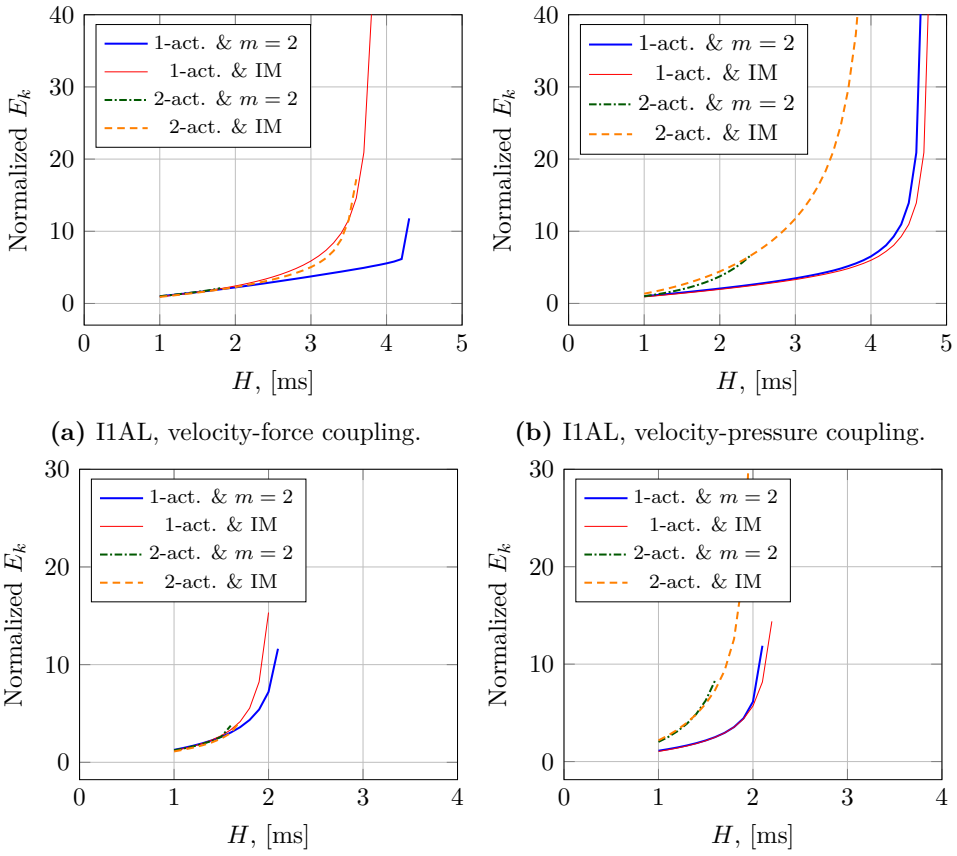


Figure 3.7. Configuration search algorithm.

error. To evaluate the effect of each parameter, a benchmark study regarding the achievable communication step sizes and solution accuracy was performed. In practice, the benchmark took into account the effect of coupling variables, multibody algorithms, integrators in the hydraulic subsystem, and the coupling method. Table 3.3 presents the available parameter space, with the exception of the extrapolation order m being kept as a constant for each model, based on the results obtained in *Publication III* and the Runge-Kutta 4th order (RK4) being implemented for the hydraulics.

Fig. 3.8 displays the energy residuals for the single and double actuator models with the forward Euler integrator with different coupling options and multibody algorithms. In the figure, the energy residual has been normalized such that it equals 1 with I1AL and $m = 2$, the integrator of the hydraulic system is

forward Euler, and $H = h_m = 1\,ms$. As can be seen, the coupling variables can significantly affect the coupling error. This is especially evident with the double actuator model, where the indirect coupling of the hydraulic circuits limits the achievable step size in general. In the single actuator model, where no indirect coupling exists, the effect is visible but less pronounced. This difference originates from the cylinder force equation, i.e. Eq. (2.34), which is dependent on the states of both of the subsystems. Should this equation be evaluated within the hydraulic subsystem, the cylinder rate $\dot{s}$ is extrapolated from the previous communication instance. In other words, in this case, the system contains a direct feed-through, i.e. the inputs and the outputs are directly interdependent, which may be detrimental to the solution stability in co-simulation [26, 2].



(a) I1AL, velocity-force coupling.

(b) I1AL, velocity-pressure coupling.

(c) I3AL-P, velocity-force coupling.

(d) I3AL-P, velocity-pressure coupling.

Figure 3.8. Energy residuals with the forward Euler integrator for the hydraulic subsystem.

Fig. 3.8 also suggests that the solution method within the mechanical subsystem may have a significant effect on the achievable communication step size and on the introduced error in the co-simulation interface. On the other hand, the system level error, as Fig. 3.9 shows, may be rather large even if the coupling accuracy is not significantly degraded. This likely results from the explicit integrator used with the I1AL formulation, which requires no extrapolation, i.e. is not sensitive to extrapolation errors, but on the other hand, may be less accurate than its implicit counterparts and thus lose energy over time.

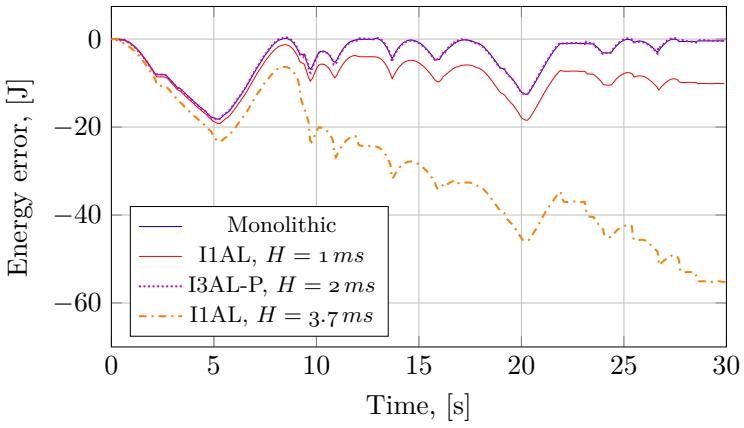


Figure 3.9. Energy balance during the simulation of the single-actuator model.

Regarding the other studied aspects, the use of the interface model (IM) instead of polynomial extrapolation allowed larger steps to be taken. This, however, came at the cost of a degraded velocity level solution. The effect of the internal integrator of the hydraulics, in turn, had little effect in the achievable communication step size when h_h was kept constant and small enough for each integrator to produce stable results. On the other hand, h_h itself had a significant effect on the achieved communication step size when it was increased from the small value.

Chapter 4

Conclusions

This dissertation has discussed the problem of the coupled dynamic simulation of multibody and hydraulic systems. Multibody dynamics is a field of science that focuses on the simulation of mechanical systems, and, in practical applications, they are often actuated by other dynamical systems, such as hydraulics. Two main coupling approaches can be established, monolithic and co-simulation, both of which have been applied in this study.

Monolithic simulation allows easy and accurate coupling between dynamic systems of different nature, as the equations that describe each subsystem are directly coupled and solved within the same numerical solver. The main benefit of this approach is its simple and accurate coupling, but it comes at the cost of the modularity of the system model. Additionally, the efficiency of the monolithic simulation may suffer if one subsystem requires relatively small integration steps compared to other subsystems. This aspect has been addressed in *Publication I* and *Publication II*.

Publication I has proposed a monolithic scheme based on an efficient semi-recursive formulation and on the lumped fluid method. The trapezoidal rule integrator was implemented as a numerical solver. The results indicate that the method can achieve relatively good computational efficiency and good convergence even with a numerically computed tangent matrix. This, however, required the use of high penalty factors and an adaptively selected differentiation increment. The latter publication, in turn, improved the performance of the method presented in *Publication I* by the introduction of a method based on the singular perturbation theory. The results show improved computational efficiency, which is contributed by both lower numerical stiffness and fewer integrated variables.

Co-simulation, in turn, provides a powerful tool for building multidisciplinary

simulations of large systems, should the monolithic approach not be practical or even possible. While multiple approaches exist within this field [19, 37], this work focuses on the non-iterative Jacobi-scheme co-simulation in continuous time for its applicability in real-time solutions. With this method a parallel execution of subsystems is possible, which, in practical implementations, enables efficiency gains over the other possibilities.

A number of challenges, however, are related to the selected co-simulation approach, one of them being the selection of the coupling method and the internal integrators, which is still often based on a trial and error basis. This aspect has been addressed in *Publication III*, which proposes an automated methodology to select functional co-simulation schemes. The results indicate potential for the method; that is, the method is able to provide a functional configuration for the nonlinear case example used. To obtain a better understanding of the effect of the different configurations, a benchmark study has been carried out in *Publication IV*. The results of this study indicate that the use of state variables in the information exchange may be advisable, as it can eliminate direct feed-through from the system and thus improve the stability of the solution. Additionally, explicit integrators perform well in the co-simulation from the coupling accuracy point of view since they can, in some cases, eliminate the need for data extrapolation.

4.1 Suggestions for future work

The work presented in this dissertation and the included publications can be continued in several directions. Below are some suggestions for future research topics.

A comparison of monolithic approaches to equivalent co-simulation approaches could provide useful insight on the efficiency differences. This, however, should be performed both on an appropriate scale, i.e. with a model representative of real-life problems, and in an appropriate computational environment – that is, the implementation is carried out with C++ or another efficient programming language.

As mentioned earlier, a number of co-simulation approaches suited also for real-time purposes have been introduced in the literature. However, very few comparative studies seem to exist. The work presented here contributes to the field but could be extended to consider other coupling options, such as the method presented in [4]. This could prove valuable for both the scientists in the field and the engineers who design the actual applications, as the lack of comprehensive benchmarks limits the available information when the methods are being selected.

[1] Arnold, M., Burgermeister, B., Führer, C., Hippmann, G., and Rill, G. Numerical methods in vehicle system dynamics: State of the art and current developments. *Vehicle System Dynamics 49*, 7 (2011), 1159–1207.

[2] Arnold, M., Clauss, C., and Schierz, T. Error analysis and error estimates for co-simulation in FMI for model exchange and co-simulation v2.0. In *Progress in Differential-Algebraic Equations* (Berlin, Heidelberg, 2014), S. Schöps, A. Bartel, M. Günther, E. J. W. ter Maten, and P. C. Müller, Eds., Springer Berlin Heidelberg, pp. 107–125.

[3] Bayo, E., and Ledesma, R. Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. *Nonlinear Dynamics 9*, 1-2 (1996), 113–130.

[4] Ben Khaled-El Feki, A., Duval, L., Faure, C., Simon, D., and Gaid, M. B. CHOPtrey: Contextual online polynomial extrapolation for enhanced multi-core co-simulation of complex systems. *Simulation 93*, 3 (2017), 185–200.

[5] Benedikt, M., Watzenig, D., Zehetner, J., and Hofer, A. A nearly energy-preserving coupling element for holistic weak-coupled system co-simulations. In *NAFEMS World Congress 2013* (Salzburg, Austria, 2013).

[6] Blochwitz, T., Otter, M., Akesson, J., Arnold, M., Clauss, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., Neumerkel, D., et al. Functional mockup interface 2.0: The standard for tool independent exchange of simulation models. In *Proceedings of the 9th International MODELICA Conference; September 3-5, 2012, Munich, Germany* (2012), no. 076, Linköping University Electronic Press, pp. 173–184.

[7] Blochwitz, T., Otter, M., Arnold, M., Bausch, C., Elmqvist, H., Junghanns, A., Mauss, J., Monteiro, M., Neidhold, T., Neumerkel, D., et al. The functional mockup interface for tool independent exchange of simulation models. In *Proceedings of the 8th International Modelica*

Conference; March 20th-22nd; Technical Univeristy; Dresden; Germany (2011), no. 063, Linköping University Electronic Press, pp. 105–114.

[8] Brenan, K. E., Campbell, S. L., and Petzold, L. R. *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*. Elsevier, 1989.

[9] Callejo, A., Narayanan, S., de Jalón, J., and Norris, B. Performance of automatic differentiation tools in the dynamic simulation of multibody systems. *Advances in Engineering Software 73* (2014), 35–44.

[10] Cuadrado, J., Cardenal, J., Morer, P., and Bayo, E. Intelligent simulation of multibody dynamics: Space-state and descriptor methods in sequential and parallel computing environments. *Multibody System Dynamics 4*, 1 (2000), 55–73.

[11] Cuadrado, J., Dopico, D., Barreiro, A., and Delgado, E. Real-time state observers based on multibody models and the extended kalman filter. *Journal of Mechanical Science and Technology 23*, 4 (2009), 894–900.

[12] Cuadrado, J., Dopico, D., Gonzalez, M., and Naya, M. A combined penalty and recursive real-time formulation for multibody dynamics. *Journal of Mechanical Design, Transactions of the ASME 126*, 4 (2004), 602–608.

[13] Cuadrado, J., Dopico, D., Naya, M. A., and Gonzalez, M. Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators. *Multibody System Dynamics 12*, 2 (2004), 117–132.

[14] de Jalón, J. G., and Bayo, E. *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*. Springer-Verlag, New-York, 1994.

[15] Docquier, N., Poncelet, A., Delannoy, M., and Fisette, P. Multiphysics modelling of multibody systems: application to car semi-active suspensions. *Vehicle System Dynamics 48*, 12 (2010), 1439–1460.

[16] Dopico, D., Lugris, U., Gonzalez, M., and Cuadrado, J. IRK vs structural integrators for real-time applications in MBS. *Journal of Mechanical Science and Technology 19*, 1 (2005), 388–394.

[17] Gear, C. W., and Wells, D. R. Multirate linear multistep methods. *BIT Numerical Mathematics 24*, 4 (1984), 484–502.

[18] Ghotbi, B., González, F., Kövecses, J., and Angeles, J. Mobility evaluation of wheeled robots on soft terrain: Effect of internal force distribution. *Mechanism and Machine Theory 100* (2016), 259–282.

[19] Gomes, C., Thule, C., Broman, D., Larsen, P. G., and Vangheluwe, H. Co-simulation: A survey. *ACM Computing Surveys 51*, 3 (2018), 49:1–49:33.

[20] González, F., Dopico, D., Pastorino, R., and Cuadrado, J. Behaviour of augmented Lagrangian and Hamiltonian methods for multibody dynamics in the proximity of singular configurations. *Nonlinear Dynamics 85*, 3 (2016), 1491 – 1508.

[21] González, F., Naya, M. A., Luaces, A., and González, M. On the effect of multi-rate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. *Multibody System Dynamics 25*, 4 (2011), 461–483.

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

[23] Handroos, H. M., and Vilenius, M. J. Flexible semi-empirical models for hydraulic flow control valves. *Journal of Mechanisms, Transmissions, and Automation in Design 113*, 3 (1991), 232–238.

[24] Hidalgo, A., and De Jalón, J. Real-time dynamic simulations of large road vehicles using dense, sparse, and parallelization techniques. *Journal of Computational and Nonlinear Dynamics 10*, 3 (2015).

[25] Kirchner, M., and Eberhard, P. Simulation model of a gear synchronisation unit for application in a real-time hil environment. *Vehicle System Dynamics 55*, 5 (2017), 668–680.

[26] Kübler, R., and Schiehlen, W. Modular simulation in multibody system dynamics. *Multibody System Dynamics 4*, 2 (2000), 107–127.

[27] Meyer, T., Li, P., Lu, D., and Schweizer, B. Implicit co-simulation method for constraint coupling with improved stability behavior. *Multibody System Dynamics 44*, 2 (2018), 135–161.

[28] Naya, M. A., Cuadrado, J., Dopico, D., and Lugris, U. An efficient unified method for the combined simulation of multibody and hydraulic dynamics: Comparison with simplified and co-integration approaches. *Archive of Mechanical Engineering 58*, 2 (2011), 223–243.

[29] Negrut, D., Serban, R., Mazhar, H., and Heyn, T. Parallel computing in multibody system dynamics: why, when, and how. *Journal of Computational and Nonlinear Dynamics 9*, 4 (2014), 041007.

[30] Oshtorjani, M. K., Mikkola, A., and Jalali, P. Numerical treatment of singularity in hydraulic circuits using singular perturbation theory. *IEEE/ASME Transactions on Mechatronics 24*, 1 (2019), 144–153.

[31] Palensky, P., Van Der Meer, A. A., Lopez, C. D., Joseph, A., and Pan, K. Cosimulation of intelligent power systems: Fundamentals, software architecture, numerics, and coupling. *IEEE Industrial Electronics Magazine 11*, 1 (2017), 34–50.

[32] Paynter, H. Analysis and design of engineering systems; class notes for M.I.T. course 2.751, 1961.

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

[34] Sadjina, S., Kyllingstad, L. T., Rindarøy, M., Skjong, S., Æsøy, V., and Pedersen, E. Distributed co-simulation of maritime systems and operations. *Journal of Offshore Mechanics and Arctic Engineering 141*, 1 (2019), 011302.

[35] Sadjina, S., Kyllingstad, L. T., Skjong, S., and Pedersen, E. Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation. *Engineering with Computers 33*, 3 (2017), 607–620.

[36] Schiehlen, W. Multibody system dynamics: Roots and perspectives. *Multibody System Dynamics 1*, 2 (1997), 149–188.

[37] Schweiger, G., Gomes, C., Engel, G., Hafner, I., Schoeggl, J., Posch, A., and Nouidui, T. An empirical survey on co-simulation: Promising standards, challenges and research needs. *Simulation Modelling Practice and Theory* (2019).

[38] Schweizer, B., Li, P., and Lu, D. Implicit co-simulation methods: Stability and convergence analysis for solver coupling approaches with algebraic constraints. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 96*, 8 (2016), 986–1012.

[39] Schweizer, B., Li, P., Lu, D., and Meyer, T. Stabilized implicit cosimulation method: Solver coupling with algebraic constraints for multibody systems. *Journal of Computational and Nonlinear Dynamics 11*, 2 (2016), 021002.

[40] Shabana, A. A. *Computational Dynamics*. John Wiley & Sons, 2009.

[41] Sherman, M. A., Seth, A., and Delp, S. L. Simbody: multibody dynamics for biomedical research. *Procedia IUTAM 2* (2011), 241–261. IUTAM Symposium on Human Body Dynamics.

[42] Skjong, S., and Pedersen, E. On the numerical stability in dynamical distributed simulations. *Mathematics and Computers in Simulation 163* (2019), 183–203.

[43] Watton, J. *Fluid Power Systems: Modeling, Simulation, Analog and Microcomputer Control*. Prentice-Hall, Inc., 1989.

Publication I

Rahikainen, J., Mikkola, A., Sopanen, J., Gerstmayr, J.

Combined semi-recursive formulation and lumped fluid method for monolithic simulation of multibody and hydraulic dynamics

Reprinted with permission from
Multibody System Dynamics
Vol. 44(3), pp. 293–311, November 2018

Multibody Syst Dyn (2018) 44:293–311
https://doi.org/10.1007/s11044-018-9631-x

Combined semi-recursive formulation and lumped fluid method for monolithic simulation of multibody and hydraulic dynamics

Jarkko Rahikainen[1] · **Aki Mikkola**[1] · **Jussi Sopanen**[1] · **Johannes Gerstmayr**[2]

Received: 24 November 2017 / Accepted: 26 April 2018 / Published online: 15 May 2018

Abstract The use of multibody simulation tools allows complex machinery to be described in detail while still providing a solution for the system in real time. As mechanical components are often accompanied by other dynamical systems, such as hydraulics, description of each subsystem is required to fully describe the dynamics of complex machinery. A potential candidate for solving the multiphysics problem at hand is known as the unified or monolithic approach. This strongly coupled approach yields a single set of equations to be integrated and, compared to co-simulation and co-integration approaches, a relatively simple integration procedure. In this paper, a monolithic formulation for a combined simulation of multibody and hydraulic dynamics using an efficient semi-recursive formulation and the lumped fluid method is introduced. The results indicate that the proposed method shows potential for efficient simulation of combined multibody and hydraulic problems. The robustness of the multibody method is maintained when combined with the hydraulic dynamics description and higher efficiency is observed than with an equivalent global approach.

Keywords Multibody system dynamics · Hydraulic modelling · Real-time simulation · Coupled simulation

J. Rahikainen
jarkko.rahikainen@lut.fi

A. Mikkola
aki.mikkola@lut.fi

J. Sopanen
jussi.sopanen@lut.fi

J. Gerstmayr
johannes.gerstmayr@uibk.ac.at

1 Department of Mechanical Engineering, Lappeenranta University of Technology, Skinnarilankatu 34, Lappeenranta, Finland

2 Institute of Mechatronics, Leopold-Franzens-Universität Innsbruck, Technikerstrasse 13, Innsbruck, Austria

1 Introduction

The use of multibody techniques enables the description of complex products, such as mobile machinery, with a high level of detail while still solving the equations of motion in real time. Multibody based real-time simulation has been utilized in user training and, more recently, in product development [21]. For product development, real-time simulation makes it possible to address machine user needs early on in the concept development phase.

A dynamic system rarely consists of mechanical components alone, since the mechanical components are often driven by actuator systems, which may be, for instance, hydraulics based. It is important to note that the actuator system has its own dynamic behavior, which significantly affects the performance of the system under investigation. It is for this reason that multibody equations alone are not sufficient to describe the dynamics of the system. For a more appropriate and robust approach, accurate and effective descriptions must be formed for all dynamic subsystems.

The problem of combined multibody and hydraulic dynamics is often addressed using co-simulation or co-integration approaches. In the co-simulation approach, specialized software may be used to model each subsystem, as presented in [14]. Co-integration approaches, in turn, employ multi-rate schemes such as [13] to integrate the different subsystems. These approaches allow different time steps to be used for each subsystem, thus optimizing simulation performance for the full system. An alternative strategy to solve a coupled system is usage of a monolithic method such as presented in [23] and [8], in which a single set of equations is constructed and integrated forward in time.

In real-time applications, computational efficiency is a key aspect and, therefore, both the solution method strategy and the descriptions of the individual subsystems should be selected accordingly. To obtain a computationally efficient description for hydraulics, simplified models such as the lumped fluid method [26] can be used. In this approach, details affecting the subsystem dynamics, such as valve dynamics, are simplified and rare phenomena, such as pressure waves within a pipeline, are ignored. The method can be used, as demonstrated in [1], to model complex hydraulic circuits in real-time.

Approaches based on multibody dynamics can, in general, be divided into two main categories. In global methods [17, 24], bodies in the system are defined by a set of coordinates that independently describe the position and orientation of each body, whereas in topological approaches, also known as recursive methods [17, 24], bodies are described with respect to the previous body in a chain. In addition to the efficiency requirement in a real-time environment, the multibody method and the integrator it uses must also be able to deal with numerically stiff equations due to the presence of the equations that describe the hydraulics. Ease of implementation and generalization should also be considered.

As the topological methods exploit chain structures in the description, they often lead to more effective algorithms than global approaches. This is particularly the case when system size increases [6]. These methods can be further divided into fully and semi-recursive methods, depending on the need to solve a system of equations. Fully-recursive formulations, which do not need solution of a system of equations, have been found effective but fail at singular configurations [5]. Also, due to the complex treatment of closed kinematic chains, they can be more difficult to generalize than global approaches [5, 6].

To overcome the issues with fully-recursive formulations, semi-recursive formulations that use efficient recursive formulae to describe the open-loop system have been developed. In the case of closed kinematic loops, three main approaches to address the loop-closure conditions can be used in semi-recursive formulations [12, 18]. Firstly, it is possible to introduce constraint equations with the Lagrange multiplier method [17] or, secondly, to use

a penalization scheme [7] similar to an augmented Lagrangian [17]. Thirdly, it is possible to perform a second velocity transformation and thus obtain an independent subset of the relative coordinates [12].

It has been shown that the penalization scheme (proposed by Cuadrado et al. in [7]), similar to the augmented Lagrangian [17], can lead to robust analysis as it can handle singular configurations and numerical stiffness. The method uses a trapezoidal rule for integration and mass-damping-stiffness-orthogonal projections [2, 6] to enforce constraints in velocity and acceleration levels. Further research by the authors included study of the integrator effect [9, 10] and extension to the use of flexible bodies [19]. More recently, in a study of automated differentiation tools in multibody context [4], numerical differentiation of certain terms was proposed to improve generality of the method. Similarly, in [16] parallelization and sparse matrix tools were used to improve efficiency and generality.

In light of this previous research, the semi-recursive method [7] seems to be a potential candidate for real-time simulation. When it comes to the case of multiphysics problems, however, selection of the approach may be less clear. On the other hand, assuming both the equations of motion for the multibody problem and hydraulic description are available within a single environment, such as seen in certain simulator solutions [1], use of a monolithic approach would yield a simple solution for the problem. For instance, work by Naya et al. [20] presents a monolithic approach where the augmented Lagrangian based on position-level constraints and mass-damping-stiffness-orthogonal projections was used for the multibody dynamics.

The objective of this paper is to introduce a monolithic formulation that uses the semi-recursive formulation and the lumped fluid method for combined simulation of multibody and hydraulic dynamics. This is a novel combination that has been overlooked in the literature. In addition, and in contrast to [20] and most research regarding the semi-recursive formulation, the tangent matrix used in Newton's iteration is formed numerically to improve the generality of the method, and due to the complicated form of the partial derivatives of the pressure descriptions. A method based on work presented in [20] is utilized, with the proposed modification to the tangent matrix and with often used reference point coordinates, as an efficiency benchmark for the proposed method.

This paper is structured as follows: The semi-recursive formulation used in this study is described in Sect. 2. For sake of clarity, the integration scheme for the multibody system is also described in Sect. 2. Section 3 explains the hydraulic circuit modelling based on the lumped fluid method. Coupling of the multibody and hydraulic dynamics with modifications to the tangent matrix and projections is presented in Sect. 4. This section also discusses reasons for using the numerical tangent matrix. In Sect. 5, a numerical example of a four-bar linkage actuated by a hydraulic cylinder is introduced. Discussion related to the numerical example and coupling method are presented in Sect. 6. Section 7 presents key conclusions of this work.

2 Multibody formulation

The semi-recursive formulation as proposed by Cuadrado [7] uses a topological method to describe the open-loop kinematic chains and penalty formulation at cut-joints to close kinematic loops. A fixed time step integrator is used, as the approach is intended for real-time use. The topological components are derived from Featherstone's [11] articulated inertia method. Further developments of the approach are presented in [6] and Rodríquez [12, 22].

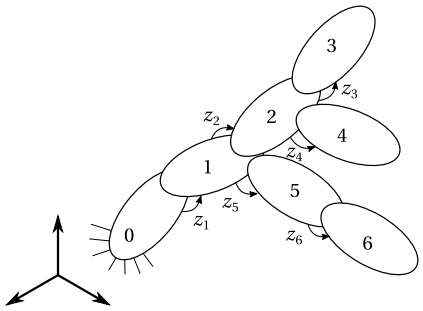


Fig. 1 An example of an open-loop system

In this method, a double set of coordinates are defined: six Cartesian velocities for each body and relative coordinates for the full system. Cartesian velocities for a body k can be expressed as

$$\mathbf{Z}_k = \begin{bmatrix} \dot{\mathbf{r}}_k \\ \boldsymbol{\omega}_k \end{bmatrix}, \tag{1}$$

where $\dot{\mathbf{r}}_k$ represents the velocity of body k's particle located at the current instance of time at the origin of the global coordinate system. Vector $\boldsymbol{\omega}_k$ contains angular velocities of the body. Time derivatives can be written correspondingly as

$$\dot{\mathbf{Z}}_k = \begin{bmatrix} \ddot{\mathbf{r}}_k \\ \dot{\boldsymbol{\omega}}_k \end{bmatrix}, \tag{2}$$

where $\ddot{\mathbf{r}}_k$ represents acceleration of the particle, and $\dot{\boldsymbol{\omega}}_k$ are the angular accelerations.

Consider an open-loop system of m bodies such as the example presented in Fig. 1. The bodies are connected via joints, allowing the positions of the system to be described using relative coordinates $\mathbf{z} = [\mathbf{z}_1^{\mathrm{T}} \ \mathbf{z}_2^{\mathrm{T}} \ \ldots \ \mathbf{z}_m^{\mathrm{T}}]^{\mathrm{T}}$. Cartesian velocities and accelerations for body $k \in [1, m]$ in the chain can now be expressed in recursive form as follows:

$$\mathbf{Z}_k = \mathbf{Z}_{k-1} + \mathbf{b}_k \dot{\mathbf{z}}_k \tag{3}$$

and

$$\dot{\mathbf{Z}}_k = \dot{\mathbf{Z}}_{k-1} + \mathbf{b}_k \ddot{\mathbf{z}}_k + \mathbf{d}_k, \tag{4}$$

where $\dot{\mathbf{z}}_k$ and $\ddot{\mathbf{z}}_k$ are the relative velocities and accelerations of the joint k. Note that indexes k and $k-1$ may not be successive as the system may branch. In Eqs. (3) and (4), $\mathbf{b}_k$ is a joint-dependent representation that contains values of $\mathbf{Z}_k$ in a special case where unit value is given to $\dot{\mathbf{z}}_k$, while other relative velocities $\dot{\mathbf{z}}$ are set to zero. In other words, joint k moves with a unit velocity, while other joints do not move. Similarly, vector $\mathbf{d}_k$ describes differences in accelerations between bodies k and $k-1$ when all accelerations in $\ddot{\mathbf{z}}$ are set to zero. For the system of m bodies, these Cartesian velocities and accelerations can be expressed in matrix forms as $\mathbf{Z} = [\mathbf{Z}_1^{\mathrm{T}} \ \mathbf{Z}_2^{\mathrm{T}} \ \ldots \ \mathbf{Z}_m^{\mathrm{T}}]^{\mathrm{T}}$ and $\dot{\mathbf{Z}} = [\dot{\mathbf{Z}}_1^{\mathrm{T}} \ \dot{\mathbf{Z}}_2^{\mathrm{T}} \ \ldots \ \dot{\mathbf{Z}}_m^{\mathrm{T}}]^{\mathrm{T}}$, respectively.

As Eqs. (3) and (4) suggest, a mapping between the Cartesian and relative coordinates can be formed. Introducing a velocity transformation matrix $\mathbf{R}$ yields

$$\mathbf{Z} = \mathbf{R}\dot{\mathbf{z}} = \mathbf{T}\mathbf{R}_d\dot{\mathbf{z}} \tag{5}$$

and

$$\dot{\mathbf{Z}} = \mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} = \mathbf{T}\mathbf{R}_d\ddot{\mathbf{z}} + \mathbf{T}\dot{\mathbf{R}}_d\dot{\mathbf{z}}, \tag{6}$$

where $\mathbf{R}_d$ is a block-diagonal matrix that contains $\mathbf{b}_k$ in ascending order, while $\dot{\mathbf{R}}\dot{\mathbf{z}}$ is simply a vector containing $\mathbf{d}_k$. The topology of the system is described with path matrix $\mathbf{T}$, which is a lower block triangular matrix with entries from diagonal to left being 6×6 unit matrices, $\mathbf{I}_6$, if the corresponding body lies between the considered body and root of the system. The example of an open-loop system presented in Fig. 1 can be used to further illustrate the form of the path matrix. In this case, the matrix $\mathbf{T}$ takes the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_6 & \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_6 & \mathbf{I}_6 & \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_6 & \mathbf{I}_6 & \mathbf{0} & \mathbf{I}_6 & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_6 & \mathbf{0} \\ \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_6 & \mathbf{I}_6 \end{bmatrix}. \tag{7}$$

When employing coordinates $\mathbf{Z}_k$, the mass matrix of a body k with a mass of m_k takes the form

$$\overline{\mathbf{M}}_k = \begin{bmatrix} m_k \mathbf{I}_3 & -m_k \tilde{\mathbf{g}}_k \\ m_k \tilde{\mathbf{g}}_k & \mathbf{J}_k - m_k \tilde{\mathbf{g}}_k \tilde{\mathbf{g}}_k \end{bmatrix}, \tag{8}$$

where $\mathbf{I}_3$ is a 3×3 unit matrix and $\tilde{\mathbf{g}}_k$ is a skew-symmetric matrix associated with the global position vector $\mathbf{g}_k$. Inertia tensor $\mathbf{J}_k$ can be obtained from the inertial tensor representation with respect to the local frame of reference as

$$\mathbf{J}_k = \mathbf{A}_k^{\mathrm{T}} \bar{\bar{\mathbf{J}}}_k \mathbf{A}_k, \tag{9}$$

where $\mathbf{A}_k$ is a rotation matrix of the body and $\bar{\bar{\mathbf{J}}}_k$ is an inertia tensor defined at the center of mass in the frame of reference of the body.

The force vector, in turn, takes the form:

$$\overline{\mathbf{Q}}_k = \begin{bmatrix} \mathbf{f}_k - \tilde{\boldsymbol{\omega}}_k (\tilde{\boldsymbol{\omega}}_k m_k \mathbf{g}_k) \\ \mathbf{n}_k - \tilde{\boldsymbol{\omega}}_k \mathbf{J}_k \boldsymbol{\omega}_k + \tilde{\mathbf{g}}_k (\mathbf{f}_k - \tilde{\boldsymbol{\omega}}_k (\tilde{\boldsymbol{\omega}}_k m_k \mathbf{g}_k)) \end{bmatrix}, \tag{10}$$

where $\mathbf{f}_k$ is a vector of external forces and $\mathbf{n}_k$ are the external moments with respect to the center of mass of the body k.

The virtual power of an open-loop system can now be written as

$$\delta \mathbf{Z} (\overline{\mathbf{M}} \dot{\mathbf{Z}} - \overline{\mathbf{Q}}) = \mathbf{0}, \tag{11}$$

where $\overline{\mathbf{M}}$ is a block-diagonal matrix consisting of mass matrices $\overline{\mathbf{M}}_k$ and the force vector takes the form $\overline{\mathbf{Q}} = [\overline{\mathbf{Q}}_1^{\mathrm{T}} \ \ \overline{\mathbf{Q}}_2^{\mathrm{T}} \ \ \ldots \ \ \overline{\mathbf{Q}}_m^{\mathrm{T}}]^{\mathrm{T}}$. Substituting Eqs. (5) and (6) in Eq. (11) yields an expression of virtual power in terms of the relative coordinates. As virtual velocities are independent, they can be eliminated, yielding a set of differential equations describing the open-loop motion:

$$\mathbf{R}_d^{\mathrm{T}} \mathbf{T}^{\mathrm{T}} \overline{\mathbf{M}} \mathbf{T} \mathbf{R}_d \ddot{\mathbf{z}} = \mathbf{R}_d^{\mathrm{T}} \left(\mathbf{T}^{\mathrm{T}} (\overline{\mathbf{Q}} - \overline{\mathbf{M}} \mathbf{T} \dot{\mathbf{R}}_d \dot{\mathbf{z}}) \right). \tag{12}$$

This expression can be simplified using the following assignments:

$$\mathbf{M} = \mathbf{R}_d^{\mathrm{T}} \mathbf{T}^{\mathrm{T}} \overline{\mathbf{M}} \mathbf{T} \mathbf{R}_d \tag{13}$$

and

$$\mathbf{Q} = \mathbf{R}_d^{\mathrm{T}}\big(\mathbf{T}^{\mathrm{T}}(\overline{\mathbf{Q}} - \overline{\mathbf{M}}\mathbf{T}\dot{\mathbf{R}}_d\dot{\mathbf{z}})\big), \tag{14}$$

thus yielding a simple expression of $\mathbf{M}\ddot{\mathbf{z}} = \mathbf{Q}$.

Closed loops can be accounted for by cutting joints and introducing constraint equations $\boldsymbol{\Phi}$ for these locations. A penalty technique can be used to enforce $\boldsymbol{\Phi} = \mathbf{0}$. The full equation of motion can now be written according to the augmented Lagrangian formulation with position level constraints:

$$\mathbf{M}\ddot{\mathbf{z}} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\boldsymbol{\lambda}^* = \mathbf{Q}, \tag{15}$$

where $\boldsymbol{\Phi}_{\mathbf{z}}$ is the Jacobian matrix of the constraint equations with respect to the relative coordinates and $\boldsymbol{\lambda}^*$ is the vector of the iterated Lagrangian multipliers. The penalty factor α can be set the same for all constraints.

The system is integrated using an implicit trapezoidal scheme [25]. Applying the rule $\mathbf{z}_{n+1} = \mathbf{z}_n + \Delta t\frac{1}{2}(\dot{\mathbf{z}}_n + \dot{\mathbf{z}}_{n+1})$ to positions $\mathbf{z}_{n+1}$ and velocities $\dot{\mathbf{z}}_{n+1}$ and solving for velocities and accelerations at time step $n+1$ yields:

$$\begin{aligned}
\dot{\mathbf{z}}_{n+1} &= \frac{2}{\Delta t}\mathbf{z}_{n+1} + \hat{\dot{\mathbf{z}}}_n = \frac{2}{\Delta t}\mathbf{z}_{n+1} - \left(\frac{2}{\Delta t}\mathbf{z}_n + \dot{\mathbf{z}}_n\right),\\
\ddot{\mathbf{z}}_{n+1} &= \frac{4}{\Delta t^2}\mathbf{z}_{n+1} + \hat{\ddot{\mathbf{z}}}_n = \frac{4}{\Delta t^2}\mathbf{z}_{n+1} - \left(\frac{4}{\Delta t^2}\mathbf{z}_n + \frac{4}{\Delta t}\dot{\mathbf{z}}_n + \ddot{\mathbf{z}}_n\right).
\end{aligned} \tag{16}$$

Introducing these equations to the equation of motion (15) leads to a dynamic equilibrium at time step $n+1$,

$$\frac{4}{\Delta t^2}\mathbf{M}\mathbf{z}_{n+1} + \boldsymbol{\Phi}_{\mathbf{z}_{n+1}}^{\mathrm{T}}\big(\alpha\boldsymbol{\Phi}_{n+1} + \boldsymbol{\lambda}_{n+1}^*\big) - \mathbf{Q}_{n+1} + \mathbf{M}\hat{\ddot{\mathbf{z}}}_n = \mathbf{0}, \tag{17}$$

where $\mathbf{z}_{n+1}$ is the set of unknowns. Equation (17) may be scaled by a factor of $\frac{\Delta t^2}{4}$, as a result of which

$$\mathbf{M}\mathbf{z}_{n+1} + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\mathbf{z}_{n+1}}^{\mathrm{T}}\big(\alpha\boldsymbol{\Phi}_{n+1} + \boldsymbol{\lambda}_{n+1}^*\big) - \frac{\Delta t^2}{4}\mathbf{Q}_{n+1} + \frac{\Delta t^2}{4}\mathbf{M}\hat{\ddot{\mathbf{z}}}_n = \mathbf{0}. \tag{18}$$

Equation (18) is a nonlinear system of equations, denoted $\mathbf{f}(\mathbf{z}_{n+1}) = \mathbf{0}$, which can be solved iteratively by employing the Newton–Raphson method as follows:

$$\left[\frac{\partial\mathbf{f}(\mathbf{z})}{\partial\mathbf{z}}\right]_{n+1}^{(i)}\Delta\mathbf{z}_{n+1}^{(i)} = -\big[\mathbf{f}(\mathbf{z})\big]_{n+1}^{(i)}, \tag{19}$$

where n is the time step and (i) is the iteration step. The next iteration step can be calculated as follows:

$$\mathbf{z}_{n+1}^{(i+1)} = \mathbf{z}_{n+1}^{(i)} + \Delta\mathbf{z}_{n+1}^{(i)}. \tag{20}$$

The vector of Lagrange multipliers, $\boldsymbol{\lambda}^*$, is iterated as follows:

$$\boldsymbol{\lambda}_n^{*(i+1)} = \boldsymbol{\lambda}_n^{*(i)} + \alpha\boldsymbol{\Phi}_n^{(i+1)}, \tag{21}$$

where (i) is the iteration step. The value for $\boldsymbol{\lambda}_n^{*(0)}$ is taken to be the final value of $\boldsymbol{\lambda}^*$ calculated in the previous time-step [7].

The residual vector of Eq. (19) can be obtained from the equation of motion (15) as follows:

$$\left[\mathbf{f}(\mathbf{z})\right]_{n+1}^{(i)} = \left[\frac{\Delta t^2}{4}\left(\mathbf{M}\ddot{\mathbf{z}} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\boldsymbol{\lambda}^{*} - \mathbf{Q}\right)\right]_{n+1}^{(i)}. \tag{22}$$

The approximated tangent matrix takes the form

$$\left[\frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}}\right]_{n+1}^{(i)} = \left[\mathbf{P} + \frac{\Delta t^2}{4}\left(\boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{z}}\right)\right]_{n+1}^{(i)}, \tag{23}$$

where

$$\mathbf{P} = \mathbf{M} + \frac{\Delta t}{2}\mathbf{C} + \frac{\Delta t^2}{4}\mathbf{K}. \tag{24}$$

In Eq. (24) $\mathbf{C}$ and $\mathbf{K}$ represent the damping and stiffness contributions in the system. These terms can be written as partial derivatives of the force vector with respect to $\mathbf{z}$ and $\dot{\mathbf{z}}$, respectively, namely

$$\mathbf{K} = -\frac{\partial \mathbf{Q}}{\partial \mathbf{z}} \tag{25}$$

and

$$\mathbf{C} = -\frac{\partial \mathbf{Q}}{\partial \dot{\mathbf{z}}}. \tag{26}$$

After convergence of Newton's iteration, a set of positions $\mathbf{z}_{n+1}$ that satisfy the equation of motion (15) and constraint conditions $\boldsymbol{\Phi} = \mathbf{0}$ is obtained. However, velocities and accelerations have not been accounted for in the solution, and therefore constraints $\dot{\boldsymbol{\Phi}} = \mathbf{0}$ and $\ddot{\boldsymbol{\Phi}} = \mathbf{0}$ may not be satisfied. Using mass-damping-stiffness-orthogonal projections [2, 6] in velocities and accelerations enables these constraint conditions to be fulfilled. Denoting velocities and accelerations obtained from the Newton–Raphson iteration as $\dot{\mathbf{z}}^{*}$ and $\ddot{\mathbf{z}}^{*}$, use of the mass-damping-stiffness-orthogonal projections yields final values for $\dot{\mathbf{z}}$ and $\ddot{\mathbf{z}}$ as follows:

$$\frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}}\dot{\mathbf{z}} = \mathbf{P}\dot{\mathbf{z}}^{*} - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{t} \tag{27}$$

and

$$\frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}}\ddot{\mathbf{z}} = \mathbf{P}\ddot{\mathbf{z}}^{*} - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\left(\dot{\boldsymbol{\Phi}}_{\mathbf{z}}^{\mathrm{T}}\dot{\mathbf{z}} + \dot{\boldsymbol{\Phi}}_{t}\right), \tag{28}$$

where matrix $\dot{\boldsymbol{\Phi}}_{\mathbf{z}}^{\mathrm{T}}\dot{\mathbf{z}}$ can be calculated using the chain rule of differentiation. Vector $\boldsymbol{\Phi}_{t}$ is the partial derivative of the constraints with respect to time. As can be seen, matrix $\mathbf{P}$ is now the same both in the projections and in the tangent matrix.

3 Hydraulic circuit modeling

The lumped fluid theory [26] can be used to describe hydraulic pressures within a hydraulic circuit. In this approach, the hydraulic circuit is partitioned into volumes V_l in which pressure is assumed to be equally distributed. Pressure waves occurring within volumes where pressure is calculated are thus assumed to be negligible.

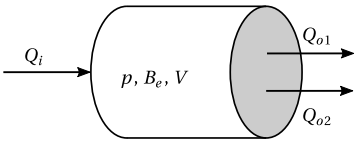


Fig. 2 Idealized volume with an internally constant pressure and bulk modulus

Figure 2 illustrates the hydraulic modeling concept. In the figure, a volume V is presented with one ingoing and two outgoing volume flows, which are denoted as Q_i and Q_{o1} and Q_{o2}, respectively. The time derivative of pressure p is computed within this volume as a function of effective bulk modulus B_e, volume V and ingoing and outgoing volume flows.

The following differential equation can be used to describe the pressure within each volume l:

$$\dot{p}_l = \frac{B_{el}}{V_l} \sum_{j=1}^{n_f} Q_{lj}, \tag{29}$$

where B_{el} is the effective bulk modulus of the volume, and n_f is the number of volume flows Q_{lj} going in or out of the volume. The effective bulk modulus of V_l can be calculated as

$$B_{el} = \left(B_o^{-1} + \sum_{j=1}^{n_c} \frac{V_j}{V_l B_j} \right)^{-1}, \tag{30}$$

where B_o is the bulk modulus of oil, n_c is the number of volumes V_j, such as pipes and hoses that form the volume V_l, and B_j is the bulk modulus of the corresponding volume.

3.1 Valves

A semi-empirical modelling approach [15] can be used to describe the valves in the system. This approach allows a single valve to be modelled as a system of independent throttles. For a simple throttle valve use of the semi-empirical approach yields the following volume flow Q over the valve:

$$Q = \begin{cases} C_v \sqrt{|\Delta p|}, & \Delta p > 0, \\ 0, & \Delta p = 0, \\ -C_v \sqrt{|\Delta p|}, & \Delta p < 0, \end{cases} \tag{31}$$

where C_v is the semi-empirical flow rate coefficient. As throttle valve has a rather simple construction, an analytical form can be used to compute the semi-empirical flow rate coefficient

$$C_v = C_d A_t \sqrt{\frac{2}{\rho}}, \tag{32}$$

where C_d is the flow discharge coefficient, A_t is the area of the throttle valve, and ρ is the oil density.

For a directional valve, the volume flow can be modeled for each valve position as follows:

$$Q = \begin{cases} C_v U \sqrt{|\Delta p|}, & \Delta p > 0, \\ 0, & \Delta p = 0, \\ -C_v U \sqrt{|\Delta p|}, & \Delta p < 0, \end{cases} \tag{33}$$

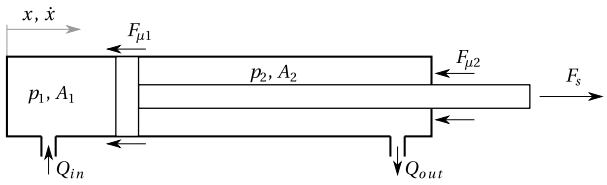


Fig. 3 A hydraulic cylinder

where C_v is the semi-empirical flow rate constant of the valve obtained from manufacturer data, U is relative spool or poppet position, and Δp is pressure difference over the valve. If the pressure difference is small, that is, less than 2 bar, volume flow can be assumed to be laminar, and Eqs. (31) and (33) are modified such that volume flow and pressure difference have a linear relation. The variable U takes valve behavior into account and can be written as a first order differential equation as

$$\dot{U} = \frac{U_{\text{ref}} - U}{\tau}, \tag{34}$$

where U_{ref} is the reference signal (i.e., control signal) and τ is the time constant, which can be read from the Bode-diagram of the valve, describing the valve behavior.

3.2 Cylinders

Figure 3 presents a hydraulic cylinder. Volume flow due to cylinder motion can be expressed as:

$$\begin{aligned} Q_{in} &= \dot{x} A_1, \\ Q_{out} &= \dot{x} A_2, \end{aligned} \tag{35}$$

where $\dot{x}$ is the velocity of the piston, and A_1 and A_2 are areas at each side of the piston.

Force produced by the cylinder can be calculated as follows:

$$F_s = p_1 A_1 - p_2 A_2 - F_\mu, \tag{36}$$

where F_μ is the total friction force caused by sealing. Pressures p_1 and p_2 acting in the chambers can be calculated using Eq. (29).

4 Coupling of multibody and hydraulic dynamics

The description of hydraulic systems given in the previous section can be added to the multibody system in a straightforward manner with use of the monolithic approach. The first stages of the method presented here follow the method presented in [20], where coupling between hydraulics and an augmented Lagrangian with position level constraints is described, but, in contrast to [20], here a numerically obtained tangent matrix is used. For this reason, as will be explained, the weight matrix used in the mass-damping-stiffness-orthogonal projections is obtained with a different method. Moreover, in contrast to [20], cylinder length is not considered as a variable in the multibody system, because cylinder length can be computed as a function of the relative coordinates and therefore yields a smaller system to be solved. Details of this approach are illustrated in Sect. 5.

In the semi-recursive formulation, force vector $\mathbf{Q}$ can be written as a function of the joint coordinates, their first time derivatives and, when hydraulics are present, pressures. Pressure variation equations, presented in Eq. (29), can also be written in terms of the same variables. The following system of equations can thus be obtained:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{z}} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\boldsymbol{\lambda}^{*} &= \mathbf{Q}(\mathbf{z}, \dot{\mathbf{z}}, \mathbf{p}), \\ \dot{\mathbf{p}} &= \mathbf{h}(\mathbf{p}, \mathbf{z}, \dot{\mathbf{z}}), \end{aligned} \tag{37}$$

where $\mathbf{p}$ is a vector of the pressures in the chambers and $\mathbf{h}(\mathbf{p}, \mathbf{z}, \dot{\mathbf{z}})$ are the pressure variation equations.

The implicit single-step trapezoidal rule is, again, adopted as the integration scheme. Difference equations for velocities and accelerations, which are the same as presented in Eq. (16), are now complemented with pressure derivatives of the following form:

$$\dot{\mathbf{p}}_{n+1} = \frac{2}{\Delta t}\mathbf{p}_{n+1} + \hat{\dot{\mathbf{p}}}_{n} = \frac{2}{\Delta t}\mathbf{p}_{n+1} - \left(\frac{2}{\Delta t}\mathbf{p}_{n} + \dot{\mathbf{p}}_{n}\right). \tag{38}$$

Similarly to the multibody dynamics integration scheme presented in Sect. 2, Eq. (38) is introduced into Eq. (37), thus establishing a dynamic equilibrium at time step $n+1$. Again, these equations are scaled by $\frac{\Delta t^2}{4}$, yielding

$$\begin{aligned} \mathbf{M}\mathbf{z}_{n+1} + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\mathbf{z}_{n+1}}^{\mathrm{T}}(\alpha\boldsymbol{\Phi}_{n+1} + \boldsymbol{\lambda}_{n+1}) - \frac{\Delta t^2}{4}\mathbf{Q}_{n+1} + \frac{\Delta t^2}{4}\mathbf{M}\hat{\ddot{\mathbf{z}}}_{n} &= \mathbf{0}, \\ \frac{\Delta t}{2}\mathbf{p}_{n+1} - \frac{\Delta t^2}{4}\mathbf{h}_{n+1} + \frac{\Delta t^2}{4}\hat{\dot{\mathbf{p}}}_{n} &= \mathbf{0}, \end{aligned} \tag{39}$$

which can be written symbolically as $\mathbf{f}(\mathbf{x}_{n+1}) = \mathbf{0}$, where $\mathbf{x} = [\mathbf{z}^{\mathrm{T}} \;\; \mathbf{p}^{\mathrm{T}}]^{\mathrm{T}}$.

Newton–Raphson iteration is used to solve the nonlinear system of Eq. (39), namely

$$\left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right]_{n+1}^{(i)} \Delta\mathbf{x}_{n+1}^{(i)} = -\left[\mathbf{f}(\mathbf{x})\right]_{n+1}^{(i)}, \tag{40}$$

where the residual vector can now be written as

$$\left[\mathbf{f}(\mathbf{x})\right]_{n+1}^{(i)} = \frac{\Delta t^2}{4}\begin{bmatrix} \mathbf{M}\ddot{\mathbf{z}} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\boldsymbol{\lambda}^{*} - \mathbf{Q} \\ \dot{\mathbf{p}} - \mathbf{h} \end{bmatrix}_{n+1}^{(i)}. \tag{41}$$

In [20], a symbolic tangent matrix was formed in terms of dependent fully-Cartesian coordinates. The same tangent matrix can also be written in terms of relative coordinates as follows:

$$\left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right]_{n+1}^{(i)} = \begin{bmatrix} \mathbf{P} + \frac{\Delta t^2}{4}\left(\boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{z}}\right) & -\frac{\Delta t^2}{4}\frac{\partial \mathbf{Q}}{\partial \mathbf{p}} \\ -\frac{\Delta t}{2}\left(\frac{\Delta t}{2}\frac{\partial \mathbf{h}}{\partial \mathbf{z}} + \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{z}}}\right) & \frac{\Delta t}{2}\left(\mathbf{I}_{np} - \frac{\Delta t}{2}\frac{\partial \mathbf{h}}{\partial \mathbf{p}}\right) \end{bmatrix}_{n+1}^{(i)}, \tag{42}$$

where $\mathbf{I}_{np}$ is an identity matrix size of $\mathbf{p}$. The obtained velocities $\dot{\mathbf{z}}^{*}$ and accelerations $\ddot{\mathbf{z}}^{*}$ are projected back to constraint manifolds using Eqs. (27) and (28), respectively.

As discussed in [4] and [16], for a general-purpose method, however, the damping and stiffness matrices present in the weight matrix $\mathbf{P}$ should be obtained numerically, because in certain cases analytical methods can become too cumbersome to use in practice. For in-

stance, a spring might be described with spline or tabulated data and deriving a symbolic derivative may not be appropriate. Additionally, the partial derivatives of the force vector and pressure variation equations seen in Eq. (42) may become complicated in practical applications. Consequently, suitable analytical expressions may exists only for the mass matrix $\mathbf{M}$ and the constraint Jacobian $\mathbf{\Phi_z}$.

For the reasons stated above, a fully numerically obtained tangent matrix is used in this study. Components in the matrix correspond to those presented in Eq. (42), but no analytical expression is used. Use of the mass-damping-stiffness-orthogonal projections, however, requires the weight matrix $\mathbf{P}$ to be solved. Using Eq. (23) it can be seen that

$$\mathbf{P} = \frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}} - \frac{\Delta t^2}{4}\left(\mathbf{\Phi}_\mathbf{z}^{\mathrm{T}} \alpha \mathbf{\Phi}_\mathbf{z}\right), \tag{43}$$

where the tangent matrix can now be obtained numerically using the forward differentiation:

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0+\epsilon) - f(x_0)}{\epsilon}, \tag{44}$$

where ϵ is the differentiation increment. To avoid problems with very large and small state variables, the differentiation increment ϵ is computed, motivated by the differentiation formula given in [3], as follows:

$$\epsilon = 1 \times 10^{-8} \max\left(1 \times 10^{-2}, |x_0|\right), \tag{45}$$

where 1×10^{-2} limits the minimum value for the differentiation increment. In the coupled system, Eq. (43) is applied such that the corresponding components are selected from the tangent matrix to form the weight matrix.

5 Numerical example

In this study, a case example is used to verify the introduced approach. The example consists of a double-acting cylinder and a four bar linkage, a mechanism common in hydraulically actuated machinery, of bodies 1–3 as depicted in Fig. 4. Details of the hydraulic circuit are presented in Fig. 5.

To minimize the size of the system, total cylinder length $|\mathbf{s}|$ is expressed as a function of joint coordinates. Bodies k are assumed to be rectangular beams with a cross-section of 0.05 m × 0.05 m that have length of $L_1 = L_2 = 1$ m and $L_3 = 2.5$ m, and masses $m_1 = m_2 = 50$ kg and $m_3 = 250$ kg. Revolute joints are used to connect the bodies, as depicted in Fig. 4. Similarly, two revolute joints are used to attach the cylinder to the system. The lower

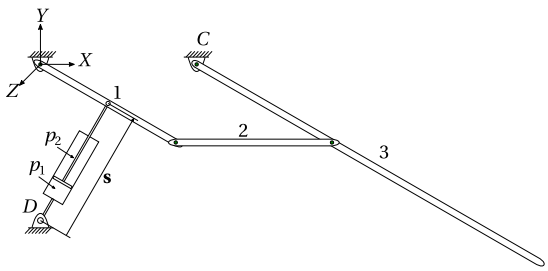


Fig. 4 A four bar linkage actuated by a hydraulic cylinder

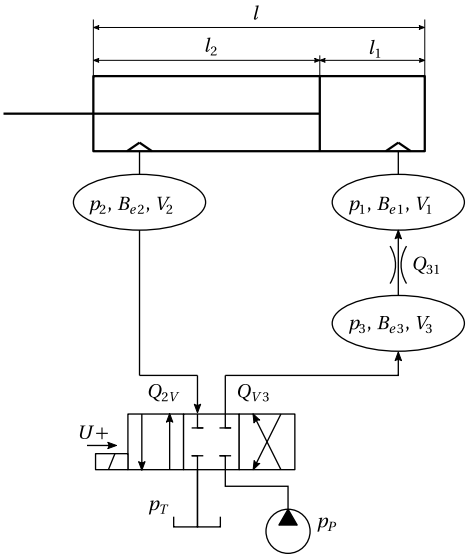


Fig. 5 Hydraulic circuit used in the example

end is attached to the global reference coordinate system at point D at $\mathbf{r}_D = [0 \quad -1 \quad 0]^{\mathrm{T}}$ and the upper end to the midpoint of the body 1. Gravity is assumed to act in the negative Y-direction, and the gravitational constant is $g = 9.81\ \mathrm{m/s^2}$.

Figure 5 presents a detailed view of the hydraulic circuit. Pressure variables p_1 and p_2 correspond to those in Fig. 4. The system consists of a pump and a tank, the pressures of which are assumed to be constant values of p_P and p_T, respectively. The hydraulic circuit also has a directional valve with control signal U, a throttle valve, a cylinder and connecting hoses. Positive volume flow and control signal directions are depicted in Fig. 5. A full list of the parameters of the hydraulic circuit is presented in Table 1. In the table, l_{1_0} and l_{2_0} define the initial position of the piston, and V_{h1} and V_{h2} are the hose 1 and 2 volumes, respectively.

The set of variables defining this problem is

$$\mathbf{x} = \left[\mathbf{z}^{\mathrm{T}} \quad \mathbf{p}^{\mathrm{T}}\right]^{\mathrm{T}} = [z_1 \quad z_2 \quad z_3 \quad p_1 \quad p_2 \quad p_3]^{\mathrm{T}}, \tag{46}$$

where z_1, z_2 and z_3 are the relative coordinates of the joints, and p_1, p_2 and p_3 are pressures in the chambers depicted in Figs. 4 and 5. The vector form of the actuator length can be written as a function of the joint coordinates as follows:

$$\mathbf{s} = \mathbf{r}_1 - \mathbf{r}_D, \tag{47}$$

where $\mathbf{r}_1$ is the vector from the origin of the global reference coordinate system to the cylinder attachment point in body 1. Kinematic relations can be applied to obtain $\mathbf{r}_1 = \mathbf{A}_1[\frac{L_1}{2} \quad 0 \quad 0]^{\mathrm{T}}$, where $\mathbf{A}_1$ is the rotation matrix and L_1 is the length of the first body.

The elongation velocity needed to compute pressure derivatives and external forces can be written as

$$\dot{s} = \frac{d|\mathbf{s}|}{dt} = \dot{\mathbf{s}} \cdot \frac{\mathbf{s}}{|\mathbf{s}|} = \dot{\mathbf{r}}_1 \cdot \frac{\mathbf{s}}{|\mathbf{s}|}, \tag{48}$$

where $\dot{\mathbf{r}}_1$ is the velocity of the attachment point, expressed as $\dot{\mathbf{r}}_1 = \tilde{\boldsymbol{\omega}}_1 \mathbf{r}_1$.

Table 1 Parameters of the example circuit

Parameter	Symbol	Value
Length of the piston	l	0.7 m
	l_{1_0}	0.2 m
	l_{2_0}	0.5 m
Initial actuator length	s_0	0.866 m
Fluid density	ρ	850 kg/m^3
Oil bulk modulus	B_o	1500 MPa
Cylinder bulk modulus	B_c	31500 MPa
Hose bulk modulus	B_h	550 MPa
Volume of hose 1	V_{h1}	3.14×10^{-5} m^3
	V_{h2}	7.85×10^{-5} m^3
	V_{h3}	4.71×10^{-5} m^3
Diameter of piston	d_1	80 mm
Diameter of piston rod	d_2	35 mm
Tank pressure	p_T	0.1 MPa
Pump pressure	p_P	7.6 MPa
Cylinder efficiency	η	0.88
Throttle discharge coefficient	C_d	0.8
Semi-empirical coefficient	C_v	2.138×10^{-8} m$^3/\text{s}\sqrt{\text{Pa}}$

The force produced by the hydraulic cylinder needs to be written in a vector form, as Eq. (10) indicates. Using the elongation length, the hydraulic force vector can be expressed as follows:

$$\mathbf{F}_s = \left[\frac{s_X}{|\mathbf{s}|} F_s \quad \frac{s_Y}{|\mathbf{s}|} F_s \quad \frac{s_Z}{|\mathbf{s}|} F_s \right]^{\mathrm{T}}, \tag{49}$$

where F_s is the cylinder force calculated according to Eq. (36), and s_X, s_Y and s_Z are the respective X, Y and Z components of the vector $\mathbf{s}$.

The cylinder has a length of l and the chamber lengths are referred to as l_1 and l_2. These variable lengths are expressed as:

$$\begin{aligned} l_1 &= l_{1_0} + s_0 - |\mathbf{s}|, \\ l_2 &= l_{2_0} + |\mathbf{s}| - s_0, \end{aligned} \tag{50}$$

where l_{1_0} and l_{2_0} are initial chamber lengths, and s_0 is the actuator length at $t = 0$. Volumes can now be written as:

$$\begin{aligned} V_1 &= V_{h1} + A_1 l_1, \\ V_2 &= V_{h2} + A_2 l_2, \\ V_3 &= V_{h3}, \end{aligned} \tag{51}$$

where V_{h1}, V_{h2} and V_{h3} are the hose volumes, and A_1 and A_2 are the piston areas. Effective bulk modules can be written according to Eq. (30) and volume flows over the valves according to Eq. (33). Differential equations for pressures can now be written according to

Table 2 Initial values used in all simulations

$\mathbf{z}_0$ [°]		$\dot{\mathbf{z}}_0$ [°/s]		$\mathbf{p}_0$ [Pa]		$\dot{\mathbf{p}}_0$ [Pa/s]	
z_1	-30	$\dot{z_1}$	0	p_1	4.14×10^6	$\dot{p_1}$	0
z_2	30	$\dot{z_2}$	0	p_2	3.50×10^6	$\dot{p_2}$	0
z_3	150	$\dot{z_3}$	0	p_3	4.14×10^6	$\dot{p_3}$	0

Eq. (29) as:

$$\begin{aligned}
\dot{p}_1 &= h_1 = \frac{B_{e1}}{V_1}(Q_{31} - A_1\dot{s}), \\
\dot{p}_2 &= h_2 = \frac{B_{e2}}{V_2}(A_2\dot{s} - Q_{2V}), \\
\dot{p}_3 &= h_3 = \frac{B_{e3}}{V_3}(Q_{V3} - Q_{31}),
\end{aligned} \tag{52}$$

where Q_{31}, Q_{2V} and Q_{V3} are the volume flows, B_{e1}, B_{e2} and B_{e3} are the effective bulk modules, and V_1, V_2 and V_3 are the volumes, as depicted in Fig. 5. The control parameter U, computed with Eq. (34), is used to compute the volume flows over the directional valve. The system ascends with positive values and descends with negative ones and stands on the hydraulic cylinder at $U = 0$ V.

To avoid instabilities in the integration process, simulation is started at static equilibrium with zero velocities. The first body is initially at $-30°$ angle, i.e., perpendicular to the actuator. Therefore initial hydraulic force F_{s_0} is

$$F_{s_0} = \frac{\sqrt{3}}{2} g \left(m_1 + 2m_2 + \frac{L_3}{L_1} m_3 \right). \tag{53}$$

In this static configuration, pressures p_1 and p_3 are equal. It is also assumed that no friction forces exist in the static configuration. The relation between the initial pressures can in such a case be solved from Eq. (36):

$$p_1 = \frac{F_{s_0} + A_2 p_2}{A_1}, \tag{54}$$

where the initial value for p_2 is given by the user. The full set of initial values used in the simulations is presented in Table 2. Note that while these values are presented in degrees the actual computations are performed in radians.

A work cycle of 7 seconds is used to actuate the system. The reference signal takes the following form:

$$U_{\text{ref}} = \begin{cases} 0 & t < 0.5, \\ 10 & 0.5 \leq t < 2.8, \\ 0 & 2.8 \leq t < 4.5, \\ -10 & 4.5 \leq t < 6, \\ 0 & t \leq 6, \end{cases} \tag{55}$$

where U_{ref} is the reference signal passed to Eq. (34).

6 Numerical results and discussion

As explained in the previous section, the example was simulated for 7 seconds using the combined semi-recursive and lumped fluid method. An overview of the working cycle is presented in Figs. 6 and 7, where angle of the first body and pressure p_1 are depicted. As can be seen, the most demanding moments for the proposed method occur at opening of the valve, and between closing and opening of the valve, when hydraulics act as a stiff spring supporting the system. Use of the fully numerical tangent matrix is also verified. Efficiency is evaluated with a comparison to a method with an equivalent global method for the mechanical components of the system, based on work by Naya et al. [20]. Different result sets are referred to by the name of the multibody method.

While the penalty factor α of Eq. (15) typically takes values from 1×10^7 to 1×10^9, stiffness introduced by the hydraulics required that it be increased to 1×10^{11} for good convergence properties in the Newton–Raphson scheme to be obtained. Similarly, a high penalty factor was also required for the case study presented by Naya et al. [20], where $\alpha = 1 \times 10^{10}$ was used. The difference can be explained by the hydraulic circuit modeled, as the model presented in [20] did not include the volume between the valves, which has relatively small volume and high bulk modulus, and which thus increases the numerical stiffness.

The energy balance depicted in Fig. 8 shows acceptable conservation properties for the proposed method even at the largest converged time step of 8×10^{-3} s, and, as can be seen, the energy conservation properties improve as the time step decreases. The peak energy drift at the valve closing is about 9.6 J during the simulation with $\Delta t = 8 \times 10^{-3}$ s and 6.1 J with $\Delta t = 4 \times 10^{-3}$ s, whereas otherwise an energy drift of less than ± 5 J, i.e., less than $\pm 0.4\%$ of the variations of about 1400 J in potential energy and actuator work, can be observed. The magnitude of constraint violation, and its first and second derivative depicted in Fig. 9, show, in turn, that the constraints are fulfilled with good accuracy in all cases. Therefore, the robustness of the multibody method demonstrated in the literature [7] is conserved when applied to a monolithic simulation of multibody and hydraulic dynamics.

The maximum and average number of iterations, and relative CPU-times are, in turn, presented in Table 3. In the table, SR (num.) is the semi-recursive method with fully numerical tangent matrix, SR (sym.) is the semi-recursive method with partially symbolic tangent

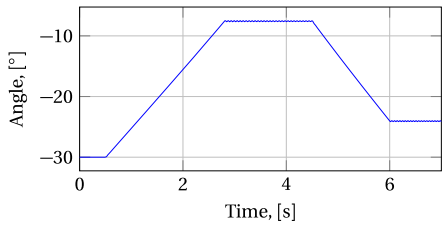


Fig. 6 Angle of the body 1 during the work cycle

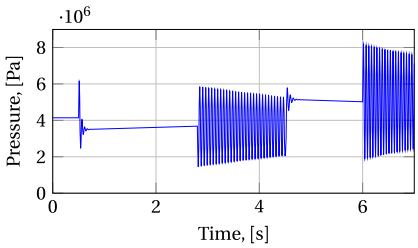
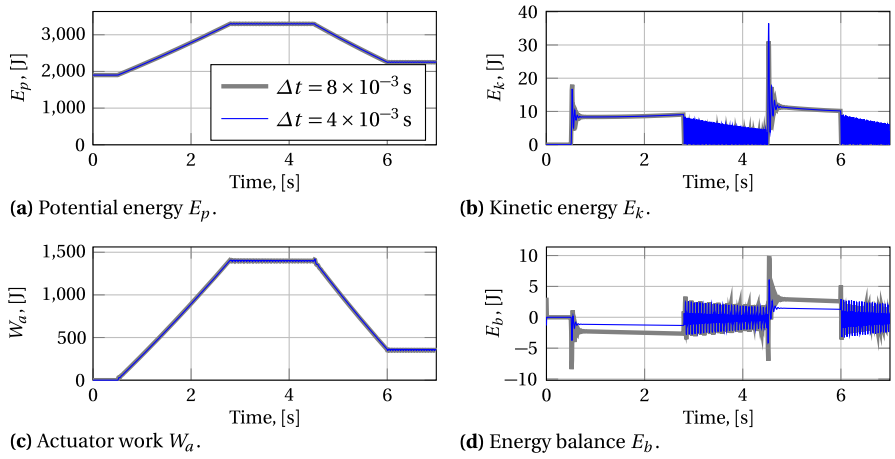


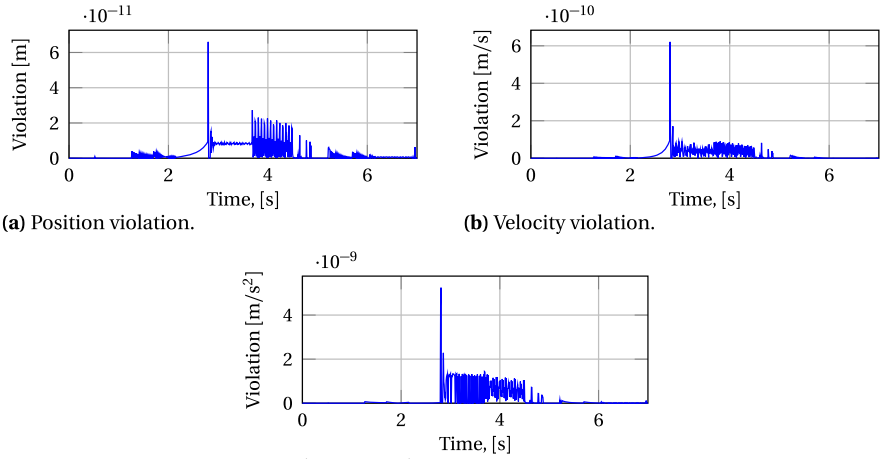
Fig. 7 Pressure p_1, i.e., the pressure of the lower part of the cylinder during the work cycle



(a) Potential energy E_p. **(b)** Kinetic energy E_k.

(c) Actuator work W_a. **(d)** Energy balance E_b.

Fig. 8 Comparison of potential and kinetic energy, actuator work, and energy balance during the simulations with time steps of 4×10^{-3} s and 8×10^{-3} s



(a) Position violation. **(b)** Velocity violation.

(c) Acceleration violation.

Fig. 9 Magnitude of the constraint violation at the cut-joint with the time step of 8×10^{-3} s

matrix, i.e., upper left-corner terms of Eq. (42) are obtained analytically, and Gl is the global method based on the work presented in [20]. The table shows, as can be expected, identical results between the fully and partially numerical tangent matrices. Error tolerance was kept in all cases as a constant of 1×10^{-7} m for positions and 1×10^{2} Pa for pressures. Convergence was obtained with the proposed method up to 8 ms, but above $\Delta t = 5$ ms no smooth results for hydraulics were obtained in the studied case.

Table 3 and Fig. 10, the latter of which depicts the number of iterations at 1 ms time step, also give an indication of the efficiency and real-time capability of the method. The average and maximum number of iterations are with the semi-recursive formulation consistently the same or lower than with the equivalent global method. The maximum number of iterations occurred during opening and closing of the valve, and, as Fig. 7 suggests and Fig. 10 confirms, when the system was standing on hydraulics. Otherwise one iteration was

Table 3 Average and maximum number of iterations, and relative CPU times

Δt	Iterations—SR (num.)		Iterations—SR (sym.)		Iterations—Gl		CPU time
	Avg.	Max	Avg.	Max	Avg.	Max	Gl/SR (num.)
0.5 ms	1.63	4	1.63	4	2.10	5	4.18
1 ms	1.78	3	1.78	3	2.19	5	3.99
2 ms	1.85	4	1.85	4	2.11	4	3.53
3 ms	1.90	4	1.90	4	2.15	4	3.78
4 ms	2.03	6	2.03	6	2.27	7	3.69
5 ms	2.37	17	2.37	17	2.60	17	3.56

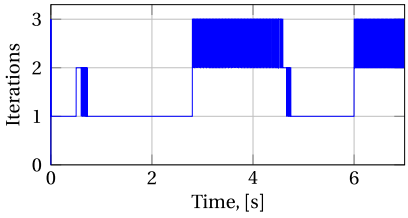


Fig. 10 Number of iterations in the Newton–Raphson scheme with 1 ms simulation

enough. These results indicate that the number of iterations can be fixed to a low value, thus fulfilling the fixed-step requirement of real-time applications.

The relative CPU times, that is, CPU time of the global method divided by CPU time of the semi-recursive method, given in Table 3, indicate, in turn, that the efficiency benefit of the semi-recursive method mostly arises from the smaller problem size. While in the literature, e.g., in [7], the semi-recursive approach becomes faster only when the problem size increases, here, likely due to implementation details, a difference can be seen already with a small example. Both approaches are implemented in Matlab environment, and the similar structure of the equations of motion is exploited such that the same number of function calls per iteration was achieved with both methods. In practice, this was achieved with symbolical precomputation and function generation of the matrix products, such as seen in Eq. (13), which seems to reduce the overhead associated with the semi-recursive method. The efficiency effect of the numerical tangent matrix was also evaluated, the partially analytical formulation being on average 13% slower in the case of the semi-recursive and 7% faster in the case of the global method when compared to the fully numerical tangent matrix with the same formulation. This, in turn, suggests that as the problem size increases, the analytical formulation of the tangent matrix becomes more efficient.

Sequential computation was used in this paper, and thus only limited conclusions of the effect of tangent matrix computation can be made. For a more definitive conclusions, a large-scale example, implemented in a lower level language such as C++ or Fortran, should be considered to generalize the performance evaluation summarized in Table 3 and the following paragraphs. In such study, consideration should also be given to possible tools to increase computational efficiency, such as the sparse matrix and parallelization techniques demonstrated in [16], which is out of scope of this paper.

7 Conclusions

In this paper, a monolithic method that uses a semi-recursive formulation and the lumped fluid method for combined simulation of multibody and hydraulic dynamics is presented. In

contrast to most existing publications utilizing the semi-recursive formulation, a fully numerical Jacobian is used in the integration process to improve the generality of the proposed method. The efficiency of the method is evaluated against an equivalent monolithic method that utilizes global coordinates.

The proposed method exhibits superior computational efficiency when compared to the equivalent global approach and has convergence properties that allow it to be applied to real-time applications. Use of the fully numerical tangent matrix in the integration process seems advisable for general purpose applications, as a number of components in the tangent matrix are often computed numerically in any case and practically no difference can be seen in the results. The proposed monolithic formulation preserves the previously demonstrated robustness of the semi-recursive method.

Acknowledgements The authors acknowledge support of this project by SIM Platform (www.lut.fi/sim) at Lappeenranta University of Technology.

References

1. Baharudin, M.E., Rouvinen, A., Korkealaakso, P., Mikkola, A.: Real-time multibody application for tree harvester truck simulator. Proc. Inst. Mech. Eng., Proc., Part K, J. Multi-Body Dyn. **228**(2), 182–198 (2014)
2. Bayo, E., Ledesma, R.: Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. Nonlinear Dyn. **9**(1–2), 113–130 (1996)
3. Brenan, K.E., Campbell, S.L., Petzold, L.R.: The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations. Elsevier, Amsterdam (1989)
4. Callejo, A., Narayanan, S., Javier García de Jalón, J., Norris, B.: Performance of automatic differentiation tools in the dynamic simulation of multibody systems. Adv. Eng. Softw. **73**, 35–44 (2014). https://doi.org/10.1016/j.advengsoft.2014.03.002
5. Cuadrado, J., Cardenal, J., Bayo, E.: Modeling and solution methods for efficient real-time simulation of multibody dynamics. Multibody Syst. Dyn. **1**(3), 259–280 (1997)
6. Cuadrado, J., Cardenal, J., Morer, P., Bayo, E.: Intelligent simulation of multibody dynamics: space-state and descriptor methods in sequential and parallel computing environments. Multibody Syst. Dyn. **4**(1), 55–73 (2000)
7. Cuadrado, J., Dopico, D., Gonzalez, M., Naya, M.: A combined penalty and recursive real-time formulation for multibody dynamics. J. Mech. Des. **126**(4), 602–608 (2004). https://doi.org/10.1115/1.1758257
8. Docquier, N., Poncelet, A., Delannoy, M., Fisette, P.: Multiphysics modelling of multibody systems: application to car semi-active suspensions. Veh. Syst. Dyn. **48**(12), 1439–1460 (2010). https://doi.org/10.1080/00423110903509335
9. Dopico, D., Lugris, U., Gonzalez, M., Cuadrado, J.: IRK vs structural integrators for real-time applications in MBS. J. Mech. Sci. Technol. **19**(1), 388–394 (2005). https://doi.org/10.1007/BF02916159
10. Dopico, D., Lugris, U., Gonzalez, M., Cuadrado, J.: Two implementations of irk integrators for real-time multibody dynamics. Int. J. Numer. Methods Eng. **65**(12), 2091–2111 (2006). https://doi.org/10.1002/nme.1544
11. Featherstone, R.: Robot Dynamics Algorithms. Kluwer Academic, Boston (1987)
12. García De Jalón, J., Álvarez, E., De Ribera, F., Rodríguez, I., Funes, F.: A fast and simple semi-recursive formulation for multi-rigid-body systems. Comput. Methods Appl. Sci. **2**, 1–23 (2005)
13. Gear, C.W., Wells, D.R.: Multirate linear multistep methods. BIT Numer. Math. **24**(4), 484–502 (1984). https://doi.org/10.1007/BF01934907
14. González, F., González, M., Mikkola, A.: Efficient coupling of multibody software with numerical computing environments and block diagram simulators. Multibody Syst. Dyn. **24**(3), 237–253 (2010). https://doi.org/10.1007/s11044-010-9199-6
15. Handroos, H., Vilenius, M.: Flexible semi-empirical models for hydraulic flow control valves. J. Mech. Transm. Autom. Des. **113**(3), 232–238 (1991)

16. Hidalgo, A., García De Jalón, J.: Real-time dynamic simulations of large road vehicles using dense, sparse, and parallelization techniques. J. Comput. Nonlinear Dyn. **10**(3), 031005 (2015). https://doi.org/10.1115/1.4028794
17. de Jalón, J.G., Bayo, E.: Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge. Springer, New York (1994)
18. de Jalón, J.G., Hidalgo, A., Callejo, A.: Improved semi-recursive formulation for the dynamic simulation of multibody systems (2011)
19. Lugrís, U., Naya, M., González, F., Cuadrado, J.: Performance and application criteria of two fast formulations for flexible multibody dynamics. Mech. Based Des. Struct. Mach. **35**(4), 381–404 (2007). https://doi.org/10.1080/15397730701617947
20. Naya, M., Cuadrado, J., Dopico, D., Lugris, U.: An efficient unified method for the combined simulation of multibody and hydraulic dynamics: comparison with simplified and co-integration approaches. Arch. Mech. Eng. **58**(2), 223–243 (2011). https://doi.org/10.2478/v10180-011-0016-4
21. Nokka, J., Montonen, J.H., Bin Baharudin, E., Immonen, P., Rouvinen, A., Laurila, L., Lindh, T., Mikkola, A., Sopanen, J., Pyrhönen, J.: Multi-body simulation based development environment for hybrid working machines. Int. Rev. Model. Simul. **8**(4), 466–476 (2015)
22. Rodríguez, J.I., Jiménez, J.M., Funes, F.J., García de Jalón, J.: Recursive and residual algorithms for the efficient numerical integration of multi-body systems. Multibody Syst. Dyn. **11**(4), 295–320 (2004). https://doi.org/10.1023/B:MUBO.0000040798.77064.bc
23. Samin, J.C., Brüls, O., Collard, J.F., Sass, L., Fisette, P.: Multiphysics modeling and optimization of mechatronic multibody systems. Multibody Syst. Dyn. **18**(3), 345–373 (2007). https://doi.org/10.1007/s11044-007-9076-0
24. Shabana, A.A.: Computational Dynamics. Wiley, New York (2009)
25. Wanner, G., Hairer, E.: Solving Ordinary Differential Equations II. Springer, Berlin (1991)
26. Watton, J.: Fluid Power Systems: Modeling, Simulation, Analog and Microcomputer Control. Prentice Hall, New York (1989)

Publication II

Rahikainen, J., Kiani, M., Sopanen, J., Jalali, P., Mikkola, A.

Computationally efficient approach for simulation of multibody and hydraulic dynamics

Reprinted with permission from
Mechanism and Machine Theory
Vol. 130, pp. 435–446, December 2018

Mechanism and Machine Theory 130 (2018) 435–446

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmachtheory

Research paper

Computationally efficient approach for simulation of multibody and hydraulic dynamics

Jarkko Rahikainen[a,*], Mehran Kiani[b], Jussi Sopanen[a], Payman Jalali[b], Aki Mikkola[a]

[a] Department of Mechanical Engineering, Lappeenranta University of Technology, Skinnarilankatu 34, 53850, Finland
[b] Department of Energy Technology, Lappeenranta University of Technology, Skinnarilankatu 34, 53850, Finland

ARTICLE INFO

Article history:
Received 20 April 2018
Revised 21 August 2018
Accepted 22 August 2018
Available online 15 September 2018

Keywords:
Multibody dynamics
Hydraulic dynamics
Monolithic integration
Real-time simulation
Singular perturbation theory

ABSTRACT

A realistic real-time simulation of a complex system, such as an excavator, requires detailed description of the machinery and its components. To take into account the dynamics of entire systems, the model must encompass descriptions of non-mechanical systems, such as hydraulics. For the multibody systems, use of the semi-recursive methods has often been found to be the most efficient solution when the system size increases. For the hydraulic dynamics, in turn, the recently introduced application of the singular perturbation method is a potential candidate for the real-time applications. The main benefit of the application of the singular perturbation method over the conventionally used lumped fluid method is that it overcomes the challenges that the lumped fluid method encounters when numerical stiffness caused by small hydraulic volumes is present in the circuit. Objective of this paper is to improve a recently proposed monolithic formulation for the combined simulation of multibody and hydraulic dynamics via the introduction of the singular perturbation method. Results indicate that the proposed method improves efficiency and robustness when compared to the formulation proposed earlier.

1. Introduction

The use of the multibody-based real-time simulation tools has become more widespread as computation power has grown more affordable. Applications of this real-time simulation are found, for instance, in user training and, recently, in product development [1]. Since real-world mechanical systems are often accompanied by other dynamic systems, such as hydraulics, descriptions of the both dynamic systems are required if an accurate model is to be built.

Several approaches have been proposed to couple the descriptions of the dynamic subsystems, such as co-simulation [2,3] and co-integration [4,5], both of which allow different time steps to be used for each subsystem. These methods are often used to address the significantly different time scale that certain systems, such as multibody and hydraulic dynamics, possess, or, more commonly, to allow the use of the domain-specific tools and solvers developed for each field. This increases modularity in the design process and adds a possibility to hide the internal details of subsystems, for intellectual property protection, but it increases the complexity of the coupling interface.

* Corresponding author.
E-mail addresses: jarkko.rahikainen@lut.fi (J. Rahikainen), mehran.kiani@lut.fi (M. Kiani), jussi.sopanen@lut.fi (J. Sopanen), payman.jalali@lut.fi (P. Jalali), aki.mikkola@lut.fi (A. Mikkola).

https://doi.org/10.1016/j.mechmachtheory.2018.08.023
0094-114X/

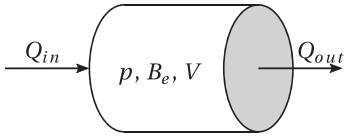


Fig. 1. Idealisation of a volume in the lumped fluid method.

However, if equations for each subsystem are available in the same simulation environment, as seen in [6], use of a monolithic approach, which yields a single set of nonlinear equations to be solved, can provide a straightforward approach for the coupling. This approach has been presented multiple times in the literature, for instance, in [7,8]. In the latter work, the augmented Lagrangian method based on position level constraints and projections [9,10] for constraint stability is coupled with the lumped fluid [11] method and integrated with the implicit single-step trapezoidal rule. As the penalty scheme with iterated Lagrange multipliers and implicit integrator can provide a robust approach for multibody dynamics [10], the work presented in [8] seems interesting from the coupled simulation point of view.

The efficiency of the method proposed in [8] is, however, hindered by the global approach used for the multibody dynamics. To address this, Rahikainen et al. [12] proposed to use the semi-recursive formulation proposed by Cuadrado et al. in [13] for the monolithic simulation of multibody and hydraulic dynamics, based on work presented in [8]. The multibody method uses the same penalty formulation as the augmented Lagrangian method for closed loop constraints and uses the mass-damping-stiffness orthogonal projections [10] to stabilise them, and it has indicated potential for real-time simulation in multiple applications, as demonstrated in studies of automated differentiation tools [14] and dense, sparse, and parallelisation techniques [15].

However, while the multibody method was selected real-time simulation in mind in [12], the lumped fluid method [11] may in certain cases encounter problems, that hinder its computational efficiency. As demonstrated in the literature [16,17], small volumes in a hydraulic circuit can increase numerical stiffness of the system, and thereby require small time steps for a solution to be sought. This can become an issue especially in the context of the monolithic approach wherein the same time step is also used for the multibody solution.

Recently, Kiani et al. [18] proposed application of the singular perturbation theory [19] to address the issue of numerical stiffness introduced by the small volumes. The results of the study indicated potential for the real-time simulation of a hydraulic system, as the behaviour of the examined circuit examined was captured with high accuracy while the computational effort was lower than in the lumped fluid approach. Use of the perturbation method allows to increase the integration time step, which, in turn, decreases the time scale difference between the multibody and hydraulic dynamics.

The objective of this paper is to improve the real-time applicability of the method presented in [12] by employing the approach based on the singular perturbation theory presented in [18]. Accordingly, this is the first study to couple the application of the singular perturbation theory [18] with the multibody dynamics. A case example seen in [12] is modified such that the issue of small volumes with the lumped fluid method is demonstrated, and real-time applicability is evaluated relative to the method presented in [12].

The rest of this paper is structured as follows: Section 2 includes the hydraulic modelling with use of the lumped fluid method and provides a brief introduction to the singular perturbation theory, multibody modelling, and the integration scheme. To better illustrate the application of the singular perturbation theory, Section 3 details the differences in the hydraulic modelling with respect to the lumped fluid method. That section also details the hydraulic circuit and the mechanical model used in the example. Results are presented in Section 4, and Section 5 presents the key conclusions.

2. Methods

In this section the lumped fluid method, which is traditionally used to model the hydraulic circuits, along with the singular perturbation theory, the method applied for the hydraulic circuit here, are briefly introduced. Models of hydraulic components related to the case example, and the implementation details of the singular perturbation theory can be found further on, in Section 3. The semi-recursive multibody method and the monolithic integration also are briefly discussed in this section.

2.1. The lumped fluid method

In the lumped fluid method [11] the hydraulic circuit is divided into volumes V in which pressure is assumed to be equally distributed. This is illustrated in Fig. 1, where volume flows Q_{in} and Q_{out}, both of which may consist of multiple separate flows, flow into volume V, which has internally constant pressure p and effective bulk modulus B_e.

In a general form, the pressure variation for each volume l can be expressed as a first-order differential equation as follows:

$$\dot{p}_l = \frac{B_{el}}{V_l}\left(\sum_{j=1}^{n_f} Q_{lj} - \dot{V}_l\right) \tag{1}$$

where $\dot{p}_l$ is the time derivative of the pressure, B_{el} is the effective bulk modulus, V_l is the volume, $\dot{V}_l$ is the rate of change of volume V_l, n_f is the number of incoming and outgoing volume flows, and, finally, Q_{lj} are the incoming and outgoing volume flows. Effective bulk modulus B_{el} represents the compressibility of the volume, and it takes the fluid compressibility due to dissolved air in the fluid and flexibility of containers into account. The general form for the effective bulk modulus can be written as

$$B_{el} = \left(B_o^{-1} + \sum_{j=1}^{n_c} \frac{V_j}{V_l} B_j^{-1}\right)^{-1} \tag{2}$$

where B_o is the bulk modulus of the oil, n_c is the number of subvolumes V_j that form the volume V_l, and B_j is the bulk modulus of each volume j. Subvolumes V_j are components of the circuit, such as pipes and hoses, in which the container has a consistent bulk modulus B_j, and their volume can be computed from the container geometry. Each volume V_l, in turn, is limited by restrictors, such as valves and pistons, that allow volume flows into and from of the volume. The computation of these volume flows is dependent on the component, and in some cases, such as valves, dependent also on the modelling method selected.

2.2. Singular perturbation theory

Consider the pressure derivative given in Eq. (1). Rearranging the equation such that the ratio between the effective bulk modulus and volume is moved to the left side yields an equation with a small-valued parameter. In the perturbation problems, in general, the effect of a parameter that has a small value, as in the case illustrated above, has a key role. For the regular perturbation problems, the solution can be obtained by equating the small valued parameter to zero, whereas in the singular perturbation problems this parameter cannot be neglected without a significant loss of accuracy in the results. Under certain conditions, the pressure variation equations of the lumped fluid theory fall into the latter category.

A well-known approach for perturbation problems of the latter type is the singular perturbation theory. As mentioned in [18], with this theory the ordinary differential equations, that contain the infinitesimal parameter ϵ, are transferred into a quasi-steady state model, in which the accuracy is maintained in the order of ϵ, based on Tikhonov's theorem [19] in singular perturbation theory and the strategy employed by Wang et al. [20]. To illustrate the method, consider the following system of singular equations:

$$\dot{\phi} = f(\phi, \psi, t, \epsilon) \quad \phi(t_0) = \zeta(\epsilon) \tag{3}$$

$$\dot{\psi} = g(\phi, \psi, t, \epsilon) \quad \psi(t_0) = \xi(\epsilon) \tag{4}$$

where ϵ is the infinitesimal value, t is time, and ϕ and ψ are the state variables. According to [21], the above system can be reduced to

$$\dot{\phi} = f(\phi, h(t, \phi), t, \epsilon) \tag{5}$$

$$\bar{\psi} = h(t, \bar{\phi}) \tag{6}$$

where the overbar denotes the perturbed variables and h is an algebraic equation that is determined during the order reduction. For the reduction to be possible, functions f and g are assumed to be smooth enough and their boundary layers to be exponentially stable. In addition, the reduced-order system of Eq. (6) is assumed to be exponentially stable and to have unique solutions on a convex set.

To elaborate it in a simple manner, the set of singular equations of (4) can be transformed into a set of ordinary equations in which the small parameter ϵ can be simply set to zero without a loss of accuracy. The order of Eq. (4) is reduced and Eq. (6) is an algebraic instead of a first order differential equation. Thus, this method reduces the numerical stiffness and, therefore, enables us to use a larger time step in integration. It is important to note, that $\bar{\phi}$ in Eq. (6) is not unknown but an algebraic equation with an explicit solution, and, therefore, Eqs. (5) and (6) do not constitute a set of differential algebraic equations. The application details in context of the lumped fluid method will be described later with the case example in Section 3.2.

2.3. Multibody modeling

In this paper, the semi-recursive method described by Cuadrado et al. [13] is used to describe the multibody dynamics. In the modelling phase fully Cartesian or natural coordinates are used to define constraint equations and the geometry of each body whereas positions and velocities of the open loops are computed recursively. Hence, the method uses a double set of coordinates.

The first set of coordinates, used to define the dynamic terms, is defined for a body k at the velocity level as

$$\mathbf{Z}_k = \begin{bmatrix} \dot{\mathbf{r}}_k^0 \\ \boldsymbol{\omega}_k \end{bmatrix} \tag{7}$$

where $\dot{\mathbf{r}}_k^0$ is the velocity 3×1 vector of the body k particle located at the current instant in time at the origin of the global coordinate system and $\boldsymbol{\omega}_k$ is the angular velocity vector of the same size, both expressed in the global frame. The use of these coordinates allows the recursive expressions to be written without a velocity transformation matrix, as follows:

$$\mathbf{Z}_k = \mathbf{Z}_{k-1} + \mathbf{b}_k \dot{\mathbf{z}}_k \tag{8}$$

where $\dot{\mathbf{z}}_k$ is the relative velocity of the joint k. The term $\mathbf{b}_k$ is a matrix whose number of columns corresponds to the degrees of freedom of the joint between bodies k and $k-1$, and its form depends on the joint type. Matrix forms of these Cartesian velocities can be expressed, for a system of m bodies, as $\mathbf{Z} = [\mathbf{Z}_1^{\mathrm{T}} \quad \mathbf{Z}_2^{\mathrm{T}} \quad \dots \quad \mathbf{Z}_m^{\mathrm{T}}]^{\mathrm{T}}$.

A velocity transformation matrix $\mathbf{R}$ can be defined between Cartesian velocities $\mathbf{Z}$ and relative velocities $\dot{\mathbf{z}}$ as follows:

$$\mathbf{Z} = \mathbf{R}\dot{\mathbf{z}} = \mathbf{T}\mathbf{R}_d\dot{\mathbf{z}} \tag{9}$$

where $\mathbf{T}$ is a path matrix describing the system topology and matrix $\mathbf{R}_d$ is a block-diagonal matrix containing $\mathbf{b}_k$.

The derivation of the equations of motion with the selected set of coordinates has been presented in literature on numerous occasions [13,22]. The virtual power of an open-loop system can be expressed in terms of $\mathbf{Z}$, $\bar{\mathbf{M}}_k$ being the mass matrix and $\bar{\mathbf{Q}}_k$ the force vector. The velocity transformations can be introduced to the virtual power equation, which yields independent virtual velocities that can be eliminated from the equation. Accordingly, the following set of differential equations describing the open-loop motion can be obtained:

$$\mathbf{R}_d^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\mathbf{T}\mathbf{R}_d\ddot{\mathbf{z}} = \mathbf{R}_d^{\mathrm{T}}\left(\mathbf{T}^{\mathrm{T}}\left(\bar{\mathbf{Q}} - \bar{\mathbf{M}}\mathbf{T}\dot{\mathbf{R}}_d\dot{\mathbf{z}}\right)\right) \tag{10}$$

Eq. (10) can be simplified by denoting the left-hand-side system matrix as $\mathbf{M}$ and the right-hand-side force vector as $\mathbf{Q}$. This yields the simple expression of $\mathbf{M}\ddot{\mathbf{z}} = \mathbf{Q}$.

The method presented in [13] uses a penalty technique to deal with closed kinematic chains, resulting in an equation of motion similar to the augmented Lagrangian method with position-level constraints [9]:

$$\mathbf{M}\ddot{\mathbf{z}} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\boldsymbol{\lambda}^* = \mathbf{Q} \tag{11}$$

where $\boldsymbol{\Phi}$ are the constraint equations, $\boldsymbol{\Phi}_{\mathbf{z}}$ is the constraint Jacobian, and α is the penalty factor. While this expression does not take nonholonomic constraints into account, the multibody method can work also with them [23], as they are imposed in the projection phase presented in the next section. Vector $\boldsymbol{\lambda}^*$ contains the iterated Lagrangian multipliers, which are iterated at each time step n as follows:

$$\boldsymbol{\lambda}_n^{*(i+1)} = \boldsymbol{\lambda}_n^{*(i)} + \alpha\boldsymbol{\Phi}_n^{(i+1)} \tag{12}$$

where i is the iteration number in Newton's method. The final value of $\boldsymbol{\lambda}^*$ calculated in the time step $n-1$ is reused for $\boldsymbol{\lambda}_n^{*(0)}$.

2.4. Monolithic integration

An integration scheme for the monolithic simulation is presented in [12]. That method, followed here, has its origins in the work presented by Naya et al. [8]. After the multibody and the hydraulic system equations are assembled, the following set of equations can be obtained:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{z}} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{z}}^{\mathrm{T}}\boldsymbol{\lambda}^* &= \mathbf{Q}(\mathbf{z}, \dot{\mathbf{z}}, \mathbf{p}) \\ \dot{\mathbf{p}} &= \mathbf{h}(\mathbf{p}, \mathbf{z}, \dot{\mathbf{z}}) \end{aligned} \tag{13}$$

where $\mathbf{p}$ is the vector of pressures and $\mathbf{h}$ denote the pressure variation equations.

The implicit single-step trapezoidal rule is applied for Eq. (13). This yields the following dynamic equilibrium at the time step $n+1$:

$$\begin{aligned} &\mathbf{M}\mathbf{z}_{n+1} + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\mathbf{z}_{n+1}}^{\mathrm{T}}\left(\alpha\boldsymbol{\Phi}_{n+1} + \boldsymbol{\lambda}_{n+1}^*\right) - \frac{\Delta t^2}{4}\mathbf{Q}_{n+1} + \frac{\Delta t^2}{4}\mathbf{M}\hat{\mathbf{z}}_n = \mathbf{0} \\ &\frac{\Delta t}{2}\mathbf{p}_{n+1} - \frac{\Delta t^2}{4}\mathbf{h}_{n+1} + \frac{\Delta t^2}{4}\hat{\mathbf{p}}_n = \mathbf{0} \end{aligned} \tag{14}$$

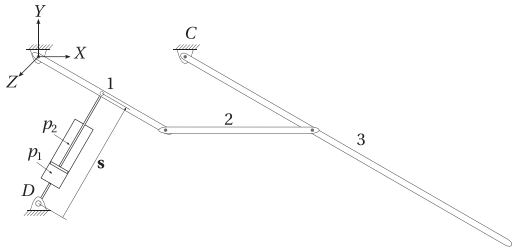


Fig. 2. A four-bar linkage actuated by a hydraulic cylinder [12].

where $\mathbf{z}_{n+1}$ and $\mathbf{p}_{n+1}$ form the vector of unknowns, and Δt is the time step, and

$$\begin{aligned}\hat{\ddot{\mathbf{z}}}_n &= -\left(\frac{4}{\Delta t^2}\mathbf{z}_n + \frac{4}{\Delta t}\dot{\mathbf{z}}_n + \ddot{\mathbf{z}}_n\right)\\ \hat{\dot{\mathbf{p}}}_n &= -\left(\frac{2}{\Delta t}\mathbf{p}_n + \dot{\mathbf{p}}_n\right)\end{aligned} \tag{15}$$

Eq. (14) can be written in a simpler form as $\mathbf{f}(\mathbf{x}_{n+1}) = \mathbf{0}$, where $\mathbf{x} = [\mathbf{z}^{\mathrm{T}} \quad \mathbf{p}^{\mathrm{T}}]^{\mathrm{T}}$. This nonlinear system of equations can be solved by means of the well-known Newton–Rhapson iteration. To address magnitude differences in $\mathbf{z}$ and $\mathbf{p}$ during the iteration, integration tolerances are specified individually for these vectors.

While a symbolic formulation of the tangent matrix required for the Newton–Rhapson scheme is possible [8], numerical computation yields a more general-purpose approach [14,15] and is less prone to implementation errors. In this study, the forward differentiation is used:

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0+\delta) - f(x_0)}{\delta} \tag{16}$$

where δ is computed as motivated by the differentiation formula given by Brenan et al. [24], as follows:

$$\delta = 1 \times 10^{-8}\max(1 \times 10^{-2}, |x_0|) \tag{17}$$

where 1×10^{-2} limits the minimum value for the differentiation increment to 1×10^{-10}.

In the Newton's iteration, a solution for positions $\mathbf{z}_{n+1}$ is sought such that the equation of motion, i.e. Eq. (11), and constraint conditions $\mathbf{\Phi} = \mathbf{0}$ are satisfied. Velocities and accelerations, however, are not accounted for during the iteration, and, therefore, $\dot{\mathbf{\Phi}} = \mathbf{0}$ and $\ddot{\mathbf{\Phi}} = \mathbf{0}$ are not expected to be satisfied. This issue can be addressed via mass–damping–stiffness orthogonal projections [9] to project the velocities and accelerations to the corresponding constraint manifolds. Let $\dot{\mathbf{z}}^*$ and $\ddot{\mathbf{z}}^*$, correspondingly, be the velocities and accelerations obtained from the Newton's iteration. When the rows and columns of the tangent matrix that correspond to the multibody problem are denoted as $\frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}}$, the final values for $\dot{\mathbf{z}}$ and $\ddot{\mathbf{z}}$ can be solved for via

$$\frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}}\dot{\mathbf{z}} = \mathbf{P}\dot{\mathbf{z}}^* - \frac{\Delta t^2}{4}\mathbf{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\mathbf{\Phi}_t \tag{18}$$

$$\frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}}\ddot{\mathbf{z}} = \mathbf{P}\ddot{\mathbf{z}}^* - \frac{\Delta t^2}{4}\mathbf{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\left(\dot{\mathbf{\Phi}}_{\mathbf{z}}^{\mathrm{T}}\dot{\mathbf{z}} + \dot{\mathbf{\Phi}}_t\right) \tag{19}$$

where weight matrix $\mathbf{P}$ can be extracted from the tangent matrix as follows:

$$\mathbf{P} = \frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}} - \frac{\Delta t^2}{4}\left(\mathbf{\Phi}_{\mathbf{z}}^{\mathrm{T}}\alpha\mathbf{\Phi}_{\mathbf{z}}\right) \tag{20}$$

3. Case study

A case study seen in [12] is reused with a minor modification in this paper to compare the perturbation and lumped fluid methods. The example, a hydraulically actuated four-bar linkage, is depicted in Fig. 2, and the hydraulic circuit is presented in Fig. 3. Details of the hydraulic circuit and the above mentioned differences are presented next, followed by the implementation of the singular perturbation approach, in Section 3.2, and the mechanical model and the coupling, in Section 3.3.

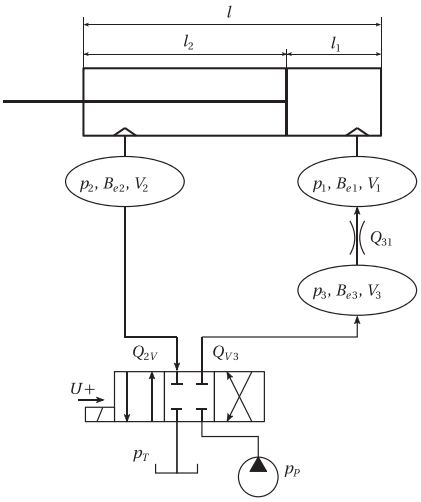


Fig. 3. The case example modified from the work of Rahikainen et al. [12].

Table 1
Parameters for the example circuit modified from Rahikainen et al. [12].

Parameter	Symbol	Value
Length of the piston	l	0.7 m
	l_{1_0}	0.2 m
	l_{2_0}	0.5 m
Initial actuator length	s_0	0.866 m
Fluid density	ρ	850 kg/m^3
Oil bulk modulus	B_o	1500 MPa
Cylinder bulk modulus	B_c	31500 MPa
Hose bulk modulus	B_h	550 MPa
Volume of hose 1	V_{h1}	3.14×10^{-5} m^3
	V_{h2}	7.85×10^{-5} m^3
	V_{h3}	7.85×10^{-7} m^3
Diameter of piston	d_1	80 mm
Diameter of piston rod	d_2	35 mm
Tank pressure	p_T	0.1 MPa
Pump pressure	p_P	7.6 MPa
Cylinder efficiency	η	0.88
Throttle discharge coefficient	C_d	0.8
Semi-empirical coefficient	C_v	2.138×10^{-8}m^3/s$\sqrt{Pa}$
Correction factor	γ_+	1.16
Correction factor	γ_-	1.05

3.1. Hydraulic circuit

For a better demonstration of the application of the perturbation theory, the case-example depicted in Fig. 3 is considered. In the figure, p_P is the pump pressure, p_T is the tank pressure, U is the control signal and l, l_1, and l_2 are the total, piston-side and rod-side lengths, respectively. The positive direction of the volume flows Q_{2V}, Q_{V3}, and Q_{31} is also shown in the figure.

Compared to the example presented in [12], the volume V_3, shown in Fig. 3, is further reduced from the original, to introduce a small volume to the system. This increases the numerical stiffness of the system and, thus, makes finding the numerical solution more challenging. Otherwise the example has been left unaltered. A full list of the parameters regarding the hydraulic circuit can be found in Table 1.

3.1.1. Modelling of the valves

A semi-empirical modelling approach [25], which allows determination of the flow rate coefficients on the basis of empirical data, is used to model the valves in the system. Volume flow Q_{31} over the throttle valve can be expressed as:

$$Q_{31} = \begin{cases} C_t\sqrt{|\Delta p|}, & \Delta p > 0 \\ 0, & \Delta p = 0 \\ -C_t\sqrt{|\Delta p|}, & \Delta p < 0 \end{cases} \tag{21}$$

where C_t is the semi-empirical flow rate coefficient and Δp is the pressure difference over the valve. As the construction of a throttle valve is rather simple, the semi-empirical coefficient is computed analytically here:

$$C_t = C_d A_t \sqrt{\frac{2}{\rho}} \tag{22}$$

where C_d is the flow discharge coefficient, A_t is the area of the throttle valve, and ρ is the oil density.

For the directional valve, the use of the semi-empirical approach yields the following general form for the volume flow:

$$Q = \begin{cases} C_v U\sqrt{|\Delta p|}, & \Delta p > 0 \\ 0, & \Delta p = 0 \\ -C_v U\sqrt{|\Delta p|}, & \Delta p < 0 \end{cases} \tag{23}$$

where the semi-empirical coefficient C_v can be determined from $Q/(U\sqrt{\Delta p})$ when the volume flow is known at one operational point, and U is relative spool or poppet position. Eq. (23) can be used to compute Q_{2V} and Q_{V3}, the positive directions of which are depicted in Fig. 3.

It must be noted that the numerical solution of Eqs. (21) and (23) can become problematic when the pressure difference is small. Hence, for small pressure differences, that is less than 2 bar, the flow is assumed to be laminar, and the corresponding equations are modified such that volume flow rate and pressure difference are related linearly. In practice, firstly, the volume flow is computed at $\Delta p = 2$ bar, with the current spool position in the case of Eq. (23), and, secondly, the actual volume flow is interpolated between zero and the computed value. The sign of the volume flow follows that presented in Eqs. (21) and (23).

Spool position U, not to be mistaken for the control signal U_{ref}, takes valve behaviour into account and can be expressed via a first-order differential equation as:

$$\dot{U} = \frac{U_{ref} - U}{\tau} \tag{24}$$

where τ is the time constant of the valve, which can be obtained from the Bode-plot provided by the manufacturer.

3.1.2. Cylinder model

A hydraulic cylinder and the related quantities have been presented in Figs. 2 and 3. The rate of change in volumes V_1 and V_2 can be written as thus:

$$\begin{aligned} \dot{V}_1 &= \dot{s}A_1 \\ \dot{V}_2 &= \dot{s}A_2 \end{aligned} \tag{25}$$

where $\dot{s}$ is the piston velocity and A_1 and A_2 are the areas on either side of the piston.

Cylinder force F_s can be expressed for the depicted case as follows:

$$F_s = p_1 A_1 - p_2 A_2 - F_\mu \tag{26}$$

where F_μ is the friction force caused by the sealing, computed in this case as $F_\mu = (p_1 A_1 - p_2 A_2)(1 - \eta)\dot{s}$.

3.2. Application of the singular perturbation theory

For the example shown in Fig. 3, the use of the lumped fluid theory yields the following expressions for the pressure variations in volumes V_1, V_2, and V_3:

$$\dot{p}_1 = \frac{B_{e1}}{V_1}\left(Q_{31} - \dot{V}_1\right) \tag{27}$$

$$\dot{p}_2 = \frac{B_{e2}}{V_2}\left(\dot{V}_2 - Q_{2V}\right) \tag{28}$$

$$\dot{p}_3 = \frac{B_{e3}}{V_3}\left(Q_{V3} - Q_{31}\right) \tag{29}$$

where B_{e1}, B_{e2}, and B_{e3} are the effective bulk moduli of the volumes and $\dot{s}$ is the cylinder elongation velocity.

Application of the singular perturbation theory [18], in turn, allows the use of algebraic equations instead of differential equations for the small volumes in the system. However, as discussed in Section 2.2, for the singular perturbation theory to be applied, a certain set of conditions must be fulfilled. As the singular perturbation theory concerns singular systems, two distinct solutions for the system are obtained if the small parameter is set to zero, or it is left unchanged. Also, the boundary layers of the following equations must be exponentially stable:

$$\epsilon \dot{p}_1 = \left(Q_{31} - \dot{V}_1\right) \tag{30}$$

$$\epsilon \dot{p}_2 = \left(\dot{V}_2 - Q_{2V}\right) \tag{31}$$

$$\epsilon \dot{p}_3 = \left(Q_{V3} - Q_{31}\right) \tag{32}$$

where ϵ represents the small volume divided by the bulk modulus.

As Kiani et al. have shown [18], in the case examined Eq. (32) alone satisfies all the conditions required by the singular perturbation theory. Therefore, it is applied only to computation of pressure p_3 only, as follows:

$$p_3 = \begin{cases} \dfrac{p_1 + \alpha_P p_P}{1 + \alpha_P} & U > 0 \\ \dfrac{p_1 + \alpha_T p_T}{1 + \alpha_T} & U < 0 \end{cases} \tag{33}$$

where

$$\alpha_P = \left(\frac{U C_v}{C_t}\right)^2 \tag{34}$$

and, in this case, $\alpha_T = \alpha_P$. In Eq. (34), C_v is the semi-empirical coefficient for the directional valve, given in Table 1, and C_t is the semi-empirical coefficient for the throttle valve, computed on the basis of Eq. (21).

While Eq. (33) allows computation of p_3 with an algebraic equation that has an explicit solution, it should be mentioned that the application of the singular perturbation theory to Eq. (29) introduces error to the value computed for volume flow, Q_{31}, since its value depends on p_3. Therefore, a correction factor is introduced to obtain the correct value of volume flow, and to remove the perturbation effects on the system. The modified flow rate is introduced by simply multiplying the uncorrected value by factor γ:

$$Q_{31} = \begin{cases} \gamma_+ \bar{Q}_{31} & U > 0 \\ \gamma_- \bar{Q}_{31} & U < 0 \end{cases} \tag{35}$$

where, bar symbol denotes the uncorrected variables computed with the perturbation theory, and γ_+ and γ_- are the correction factors, for either side of the spool. Details of the method are presented in [18], but it is still worth noting here that a manual tuning of the correction factors was suggested. The values presented in Table 1 are the final values used in the simulations.

3.3. Combination with a mechanical system

The mechanism, a four-bar linkage actuated by a hydraulic cylinder, as depicted in Fig. 2, consists of rectangular beams with a cross-section of 0.05 m × 0.05 m that have the lengths of $L_1 = L_2 = 1$ m and $L_3 = 2.5$ m, and masses $m_1 = m_2 = 50$ kg and $m_3 = 250$ kg. The bodies are connected via revolute joints and the cut-joint is located at point C. Similarly, the upper end of the cylinder end is attached to the mechanism at the midpoint of body 1, and the lower end, in the global reference coordinate system, at point D at $\mathbf{r}_D = [0 \quad -1 \quad 0]^{\mathrm{T}}$. Gravity is assumed to act in the negative Y direction, and the gravitational constant is $g = 9.81$ m/s^2 .

Total cylinder length $|\mathbf{s}|$ can be expressed as a function of joint coordinates to minimise the problem size. Thereby, the set of integrated variables defining the problem is as follows:

$$\mathbf{x} = \left[\mathbf{z}^{\mathrm{T}} \quad \mathbf{p}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{36}$$

where $\mathbf{z}$ contains the relative coordinates of the multibody system and $\mathbf{p}$ contains the pressures. As was demonstrated above, the length of the latter vector is dependent on the method used for hydraulic formulation. In the case of the lumped fluid method, all three pressures are integrated, yielding the three Eqs. (27)–(29). With the singular perturbation theory, in turn, p_3 is computed algebraically via Eq. (33), so the pressure vector has, in this example, a length of two.

The length of the cylinder can be expressed in terms of the relative coordinates as follows:

$$\mathbf{s} = \mathbf{r}_1 - \mathbf{r}_D \tag{37}$$

where $\mathbf{r}_1$ is the location of the cylinder attachment point in body 1 and $\mathbf{r}_D$ is the location of the cylinder attachment point at ground. The cylinder elongation velocity is required also, and it takes the following form:

$$\dot{s} = \dot{\mathbf{r}}_1 \cdot \frac{\mathbf{s}}{|\mathbf{s}|} \tag{38}$$

Table 2
Initial values.

z_0 [°]		$\dot{z}_0$ [°/s]		p_0 [Pa]		$\dot{p}_0$ [Pa/s]	
z_1	−30	$\dot{z}_1$	0	p_1	4.14×10^6	$\dot{p}_1$	0
z_2	30	$\dot{z}_2$	0	p_2	3.50×10^6	$\dot{p}_2$	0
z_3	150	$\dot{z}_3$	0	p_3	4.14×10^6	$\dot{p}_3$	0

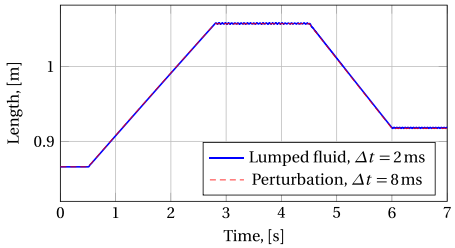


Fig. 4. Cylinder length during the work cycle.

where $\dot{\mathbf{r}}_1$ is the velocity of the attachment point.

Cylinder force F_s can be computed from Eq. (26), and its vector form $\mathbf{F}_s$, required by the force vector $\bar{\mathbf{Q}}_k$, can be obtained as:

$$\mathbf{F}_s = \left[\frac{s_X}{|\mathbf{s}|} F_s \quad \frac{s_Y}{|\mathbf{s}|} F_s \quad \frac{s_Z}{|\mathbf{s}|} F_s \right]^{\mathrm{T}} \tag{39}$$

where s_X, s_Y, and s_Z are the respective X, Y, and Z components of the cylinder length $\mathbf{s}$.

Control signal U_{ref} for the directional valve is defined as follows:

$$U_{ref} = \begin{cases} 0 & t < 0.5 \\ 10 & 0.5 \leq t < 2.8 \\ 0 & 2.8 \leq t < 4.5 \\ -10 & 4.5 \leq t < 6 \\ 0 & t \geq 6 \end{cases} \tag{40}$$

Simulations are started from static equilibrium, to avoid instabilities in the simulation process. Table 2 presents the set of initial values used that define a static equilibrium. In the case of the perturbation method, the pressure p_3 and its derivative $\dot{p}_3$ are excluded from the initial conditions.

4. Results

The proposed method is evaluated with the described example. The seven second work cycle is simulated, and the relative performance is compared to the method proposed in [12]. Since the case example is implemented in Matlab environment, the absolute CPU-times are not shown here. However, the implementation was performed such that comparable times are obtained. The slight differences in the results that were introduced by the use of the singular perturbation method are highlighted.

An overview of the work cycle is given in Fig. 4, wherein the cylinder length during the simulations with both methods is depicted at the largest converged time steps, that is 2 ms for the lumped fluid method and 8 ms for the perturbation method. As can be seen, practically identical results were obtained for the mechanism. Fig. 5, in turn, presents the pressure p_1 in a detail upon valve opening at the same step sizes. It is clear, that the steady-state solution is practically the same between the two hydraulic descriptions, whereas differences occur at the transient phase.

To further illustrate the effects introduced by the use of the singular perturbation method, the volume flow Q_{31} just after the valve opening is depicted in Fig. 6. In addition to the largest converged step sizes, the figure also shows the largest step size with the perturbation method that produced a smooth solution for the studied case, i.e., a solution with $\Delta t = 5$ ms. Slight fluctuation in the lumped fluid solution can be seen at the $28\times$ magnification. This fluctuation is captured by the integrator, and, as Fig. 7 demonstrates, hence necessitates a larger number of iterations than the proposed method. Similar to the pressures presented in Fig. 5, no significant differences are visible in the steady-state solution.

It must be noted that the accuracy of the correction factors γ_- and γ_+ has a significant effect on the accuracy of them results obtained. Without these factors, error is introduced to the computed volume flow Q_{31}, which propagates to the computed derivative of pressure $\dot{p}_1$ of Eq. (27). No clear rule to obtain values for the correction factor was observed, so they were tuned manually. These results agree to the work of Kiani et al. [18].

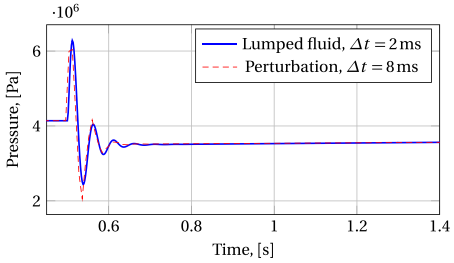


Fig. 5. Pressure p_1 during and after opening of the valve.

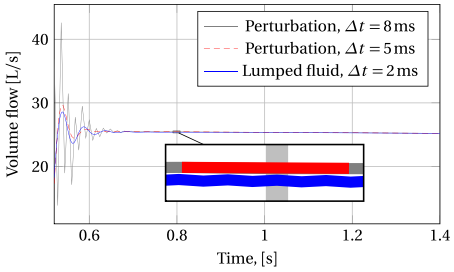


Fig. 6. Volume flow Q_{31} with the lumped fluid and the perturbation method after opening of the valve.

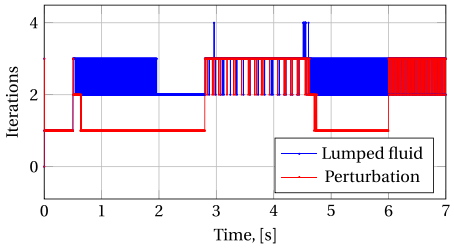


Fig. 7. Number of iterations with the 2 ms time step simulations.

Table 3
Number of iterations and relative CPU-times.

	Iterations - lumped fluid		Iterations - perturbation		CPU-time	
Δt	Avg.	Max.	Avg.	Max.	SP/LL	SP/LL(2 ms)
0.5 ms	1.76	4	1.62	4	0.83	2.34
1 ms	2.26	4	1.76	4	0.71	1.24
2 ms	2.52	4	1.81	3	0.66	0.66
3 ms	fail	fail	1.83	4	–	0.42
5 ms	fail	fail	1.85	3	–	0.27
8 ms	fail	fail	1.90	4	–	0.18

Regarding the efficiency and applicability for real-time applications, Table 3 shows the average and maximum number of iterations required for the convergence, alongside the relative CPU-time between the methods. As can be seen, the lumped fluid method fails to converge under the determined limit of 100 iterations at the experimented time steps larger than 2 ms, whereas convergence properties of the perturbed system do not change meaningfully as the time step increases. Fig. 7, that depicts the number of iterations at 2 ms time step simulations, illustrates the reason for the smaller average number

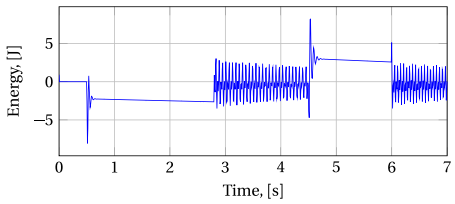


Fig. 8. Energy balance with the perturbation method, at $\Delta t = 8$ ms.

of iterations. As can be seen, the proposed method yields a smaller number of iterations when the valve is fully opened, whereas otherwise the number of iterations is approximately the same or lesser than with the lumped fluid method.

The last two columns in Table 3 give the observed relative CPU-times, computed by dividing the elapsed time with the singular perturbation approach (SP) by that with the lumped fluid method (LL). In the table, the SP/LL column therefore shows the CPU-time relationship at the current step size, and SP/LL(2 ms) is the relative efficiency of the proposed method relative to the fastest simulation with the lumped fluid method. In the former, at 0.5 ms time step the efficiency gain can be explained by the 17% smaller problem size, whereas at the larger step sizes the lower number of iterations take effect. The latter column, in turn, demonstrates the main potential of the singular perturbation method, as the computational time is significantly less than for the fastest converged solution with the lumped fluid method. These results suggest, that use of the singular perturbation method improves the real-time applicability of the method proposed in [12] as it allows larger time steps in cases wherein small volumes exist and it decreases the problem size.

The convergence rate is affected by the penalty factor from Eq. (11) and the integration tolerances. The latter, which are absolute tolerances, are set to 1×10^{-7} for the position-level variables and 1×10^{2} for the pressures. Both criteria need to be fulfilled for iteration to stop. Due to the numerical stiffness introduced by the hydraulics, penalty factor, which typically takes values from 1×10^{7} to 1×10^{9}, had to be increased to 1×10^{11} with both the proposed and the lumped fluid method for good convergence properties to be obtained. This results agrees to [12], and suggests that, while use of the perturbation method allows use of a larger step size and decreases the numerical stiffness, a large penalty factor is still required for good convergence properties to be obtained.

Finally, to check the energy conservation properties of the proposed method, the energy balance of the system at the largest converged time step, 8 ms, is presented in Fig. 8. As can be seen, the energy balance is most of the time less than ± 5 J and even at maximum about 8.1 J. Considered that the actuator work of about 1400 J, these figures are at an acceptable level of accuracy.

5. Conclusions

In this paper, a method for monolithic simulation of multibody and hydraulic dynamics using a penalty-based semi-recursive method and a perturbation method is proposed. The proposed method is compared to monolithic simulation in which the lumped fluid method is coupled with the semi-recursive method. To provide a general approach, a numerically obtained tangent matrix is used in the integration scheme, contrary to the original proposition of the multibody method.

Results indicate that use of the perturbation method in the monolithic simulation of multibody and hydraulic dynamics seems to hold potential, since, when compared to the lumped fluid method, it allows larger step sizes to be taken when small volumes are involved. In addition, again relative to the lumped fluid method, it decreases the computational load by removing the high frequency components of the solution that are introduced by the small volumes. Irrespective of the small scale differences introduced in the solution, the two methods yield practically the same solution. However, more research is needed to find the optimal value for the correction factor required by the perturbation method.

Declarations of interest

None.

Acknowledgments

The authors acknowledge support received for this project from SIM Platform at Lappeenranta University of Technology.

References

[1] J. Nokka, J.-H. Montonen, E. Bin Baharudin, P. Immonen, A. Rouvinen, L. Laurila, T. Lindh, A. Mikkola, J. Sopanen, J. Pyrhönen, Multi-body simulation based development environment for hybrid working machines, Int. Rev. Mode. Simult. 8 (4) (2015) 466–476.
[2] F. González, M. González, A. Mikkola, Efficient coupling of multibody software with numerical computing environments and block diagram simulators, Multibody Syst. Dyn. 24 (3) (2010) 237–253.

[3] A. Peiret, F. González, J. Kövecses, M. Teichmann, Multibody system dynamics interface modelling for stable multirate co-simulation of multiphysics systems, Mech. Mach. Theory 127 (2018) 52–72.
[4] C.W. Gear, D.R. Wells, Multirate linear multistep methods, BIT Numer. Math. 24 (4) (1984) 484–502.
[5] A. Cardona, M. Geradin, Modeling of a hydraulic actuator in flexible machine dynamics simulation, Mech. Mach. Theory 25 (2) (1990) 193–207.
[6] M.E. Baharudin, A. Rouvinen, P. Korkealaakso, A. Mikkola, Real-time multibody application for tree harvester truck simulator, Proc. Institut Mech. Eng. Part K 228 (2) (2014) 182–198.
[7] G. Sun, J. Liu, Dynamic responses of hydraulic crane during luffing motion, Mech. Mach. Theory 41 (11) (2006) 1273–1288.
[8] M.A. Naya, J. Cuadrado, D. Dopico, U. Lugris, An efficient unified method for the combined simulation of multibody and hydraulic dynamics: Comparison with simplified and Co-integration approaches, Arch. Mech. Eng. 58 (2) (2011) 223–243.
[9] E. Bayo, R. Ledesma, Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics, Nonlinear Dyn. 9 (1-2) (1996) 113–130.
[10] J. Cuadrado, J. Cardenal, P. Morer, E. Bayo, Intelligent Simulation of Multibody Dynamics: Space-State and Descriptor Methods in Sequential and Parallel Computing Environments, Multibody Syst. Dyn. 4 (1) (2000) 55–73.
[11] J. Watton, Fluid Power Systems: Modeling, Simulation, Analog and Microcomputer Control, Prentice-Hall, Inc., 1989.
[12] J. Rahikainen, A. Mikkola, J. Sopanen, J. Gerstmayr, Combined semi-recursive formulation and lumped fluid method for monolithic simulation of multibody and hydraulic dynamics, Multibody Syst. Dyn. (2018).
[13] J. Cuadrado, D. Dopico, M. Gonzalez, M.A. Naya, A combined penalty and recursive real-time formulation for multibody dynamics, J. Mech. Des. Trans. ASME 126 (4) (2004) 602–608.
[14] A. Callejo, S.H.K. Narayanan, J.G. de Jalón J., B. Norris, Performance of automatic differentiation tools in the dynamic simulation of multibody systems, Adv. Eng. Softw. 73 (2014) 35–44.
[15] A.F. Hidalgo, J. García De Jalón, Real-Time Dynamic Simulations of Large Road Vehicles Using Dense, Sparse, and Parallelization Techniques, J. Comput. Nonlinear Dyn. 10 (3) (2015).
[16] D.E. Bowns, L.M. Wang, The Digital Computation of Pressures in Hydraulic Pipes with Small Volume using an Iterative Technique, Proc. Institut. Mech. Eng. Part C 204 (1) (1990) 29–36.
[17] R. Piché, A. Ellman, Numerical Integration of Fluid Power Circuit Models Using Two-Stage Semi-Implicit Runge-Kutta Methods, Proc. Institut. Mech. Eng. Part C 208 (3) (1994) 167–175.
[18] M. Kiani, A. Mikkola, P. Jalali, Numerical treatment of singularity in hydraulic circuits using singular perturbation theory, IEEE/ASME Trans. Mechatron. (2018). Submitted for publication
[19] A.N. Tikhonov, Systems of differential equations containing small parameters in the derivatives, Matematicheskii sbornik 73 (3) (1952) 575–586.
[20] L. Wang, W.J. Book, J.D. Huggins, Application of singular perturbation theory to hydraulic pump controlled systems, IEEE/ASME Trans. Mechatron. 17 (2) (2012) 251–259.
[21] L. Noethen, S. Walcher, Tikhonovs theorem and quasi-steady state, Discrete Continuous Dyn. Syst. 16 (3) (2011) 945–961.
[22] J. Cuadrado, D. Dopico, M.A. Naya, M. Gonzalez, Real-time multibody dynamics and applications, in: M. Arnold, W. Schiehlen (Eds.), Simulation Techniques for Applied Dynamics, Springer Vienna, 2009, pp. 247–311.
[23] D. Dopico, F. González, J. Cuadrado, J. Kövecses, Determination of holonomic and nonholonomic constraint reactions in an index-3 augmented lagrangian formulation with velocity and acceleration projections, J. Comput. Nonlinear Dyn. 9 (4) (2014) 041006.
[24] K.E. Brenan, S.L. Campbell, L.R. Petzold, The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations, Elsevier, 1989.
[25] H.M. Handroos, M.J. Vilenius, Flexible semi-empirical models for hydraulic flow control valves, J. Mech. Trans. Autom. Des. 113 (3) (1991) 232–238.

Publication III

Rahikainen, J., González, F., Naya, M.A.,
An automated methodology to select functional co-simulation configurations

Reprinted from
Multibody System Dynamics
Online first, September 2019

Multibody Syst Dyn
https://doi.org/10.1007/s11044-019-09696-y

An automated methodology to select functional co-simulation configurations

Jarkko Rahikainen[1] · **Francisco González**[2] · **Miguel Ángel Naya**[2]

Received: 12 November 2018 / Accepted: 9 August 2019

Abstract The development of machinery often requires system-level analysis, in which non-mechanical subsystems, such as hydraulics, need to be considered. Co-simulation allows analysts to divide a problem into subsystems and use tailored software solutions to deal individually with their respective dynamics. On the other hand, these subsystems must be coupled at particular instants in time, called communication points, through the exchange of coupling variables. Between communication points, each subsystem solver carries out the integration of its states without interacting with its environment. This may cause the integration to become unstable, especially when non-iterative co-simulation is used. The co-simulation configuration, i.e., the parameters and simulation options selected by the analyst, such as the way to handle the coupling variables or the choice of subsystem solvers, is often a critical factor regarding co-simulation stability. In practice it is difficult to anticipate which selection is the most appropriate for a particular problem, especially if some inputs come from external sources, such as human operators, and cannot be determined beforehand. We put forward a methodology to automatically determine a stable and computationally efficient configuration for Jacobi-scheme co-simulation. The method uses energy residuals to gain insight into co-simulation stability. The relation between energy residual and communication step-size is exploited to monitor co-simulation accuracy during a series of tests in which the external inputs are replaced with predetermined input functions. The method was tested with hydraulically actuated mechanical examples. Results indicate that the proposed method can be used to find stable and accurate configurations for co-simulation applications.

J. Rahikainen
jarkko.rahikainen@lut.fi

F. González
f.gonzalez@udc.es

M.Á. Naya
miguel.naya@udc.es

1 Department of Mechanical Engineering, LUT University, Skinnarilankatu 34, Lappeenranta, Finland

2 Mechanical Engineering Laboratory, University of A Coruña, Mendizábal s/n, Ferrol, Spain

Published online: 10 September 2019

Keywords Co-simulation · Multiphysics · Multibody system dynamics · Hydraulic dynamics

1 Introduction

The use of multibody-based simulation tools has enabled engineers to iteratively design, validate, and re-design new products with reduced prototyping costs. This approach has also been extended to real-time environments [23]. Growing product complexity and the increasing importance of system-level simulation have resulted in renewed research interest in the efficient and accurate coupling of dynamical system solver tools.

A straightforward way to couple the simulation of multiple dynamical systems is the strongly coupled or monolithic approach [22, 27, 29] that yields a single set of equations to be integrated. The benefits of simple and accurate coupling, however, are often overshadowed in practical applications by the loss of modularity. For this reason, co-simulation approaches [2, 3, 15, 35] are used in a large number of multidisciplinary simulations. In co-simulation setups the application under study is divided into several subsystems and a different numerical solver is assigned to each of them. This makes it possible to select simulation tools that specifically fit the nature of each subsystem and allows one to tune their parameters independently. The information exchanged between solver tools is limited to a reduced set of input and output quantities, i.e., *coupling variables*, and takes place at discrete instants in time. In the *macro-step* between these communication points the integration of each subsystem proceeds independently, without knowledge of the internals of other system components. This is an attractive feature when coupling solvers and models from different vendors in industrial applications, as the implementation details of each software tool remain unknown to other system components, and so intellectual property is protected.

Following a co-simulation approach, however, poses multiple challenges regarding the stability of the numerical integration and the accuracy of the results. Coupling errors inherently occur as a consequence of discrete-time information exchange and, for that reason, accuracy and stability cannot always be guaranteed. Iterative approaches have been proposed to alleviate this problem [20, 32], but in some cases, such as demanding real-time environments, this may not be a feasible option and, thus, non-iterative schemes need to be used. Moreover, some simulation packages and models do not permit to retake an integration step once it has been completed, which prevents the use of iterative co-simulation schemes if such tools are to be used. When computational efficiency is an issue, the execution of the numerical integration of the subsystems should take place in parallel, following the well-known Jacobi scheme. This means that in each macro-step $[t_0, t_0 + H]$ all the subsystems should perform their numerical integration only with the input variables available at t_0, without knowledge of the results delivered by other subsystems at time $t_0 + H$.

The use of non-iterative schemes demands additional effort to guarantee the accuracy and stability of the results. This often requires some numerical treatment of the coupling variables. In some cases, a simple polynomial extrapolation of the input values of a subsystem may increase the accuracy of the results and allow the use of larger communication step-sizes. However, as shown in [18], an optimal solution valid for any system at all times cannot be found, in general.

The topic of enhancing the stability of non-iterative co-simulation schemes in a systematic way has been addressed several times in the literature. It is possible to deal with this issue from different points of view. Extrapolation of the coupling variables can be used to

make the integration process more stable [9], especially in multi-rate co-simulation environments [14]. As noted in [18], the selection of the extrapolation method often has a critical and strongly case-dependent impact on the accuracy and robustness of the co-simulation process. The performance of extrapolation methods for co-simulation has often been assessed using linear test problems [1, 10]; it is not straightforward, however, to generalize the conclusions obtained with linear systems to more complex applications of industrial interest, whose behavior is frequently highly nonlinear. The methodology introduced in [4] represents a practical way to choose an extrapolation method when simulating complex systems: the degree of the extrapolation polynomial is selected online during the numerical integration process, as a function of the previously known system behavior and a prediction of its immediate evolution. System energy can also be used as a relevant source of information about the stability and accuracy of a co-simulation process [16]. Maintaining the energy balance at the co-simulation interface is the basis for the nearly energy-preserving coupling element introduced in [7]. In [28], in turn, the concepts of power bond and energy residual were used to adaptively select the communication step-size in non-iterative co-simulation schemes. The online selection of macro step-size was also addressed in [6], using frequency-domain analysis instead of energy residuals. Moreover, if one or more of the components of the co-simulation environment are mechanical systems, and some information is available about their internal configuration, other methods can be used as well to achieve a more stable integration procedure. These include using the partial derivatives of the subsystem states with respect to the coupling variables to model more accurately the behavior of mechanical constraints [21, 31] and building reduced-order interface models of the mechanical subsystems to provide more accurate estimations of their evolution between communication points [25].

Despite recent developments and the wide variety of available coupling techniques, polynomial extrapolation is likely to remain a relevant method in the information exchange in co-simulation setups due to its generality and simple implementation. Polynomial extrapolation can be used even when information about the internals of the subsystems is completely unavailable, and lends itself well to multi-rate co-simulation setups and subsystems with nonmatching time-grids. On the other hand, selecting an adequate extrapolation method for a particular co-simulation task is still a challenging and time-consuming endeavor in many cases. Relevant settings in the co-simulation environment, such as the integrator method used in the subsystems, may not be tunable, because their implementation can be hidden from third-party software. Even if they are, their selection is often conducted by trial and error, with little or no information about their effect on the overall accuracy of the simulation. Either way, it is difficult to predict from a theoretical standpoint which extrapolation methods would suit best the co-simulation problem at hand. This is even more challenging if some of the simulation inputs come from external sources, which is the case in Human/Hardware-in-the-Loop (HiL) and System-in-the-Loop (SiTL) applications, where the co-simulation environment interacts with physical components or human operators. Such inputs cannot be anticipated in many cases, and they have a critical effect on simulation stability. For this reason, an automated methodology to select a functional co-simulation configuration, based on a set of pre-determined tests, can represent a valuable tool to co-simulation analysts [5].

In this paper we put forward a methodology to determine a stable and efficient configuration for non-iterative Jacobi scheme co-simulation setups, especially aimed at multiphysics problems in the context of simulation of machinery. The models used in this field often receive inputs that are difficult to predict and feature frequent discontinuities in their dynamics, e.g., the above-mentioned HiL/SiTL simulators and test benches. The subsystems in such applications are commonly solved using relatively simple integration formulas and

constant step-sizes, in order to keep the elapsed time in computations short and predictable. A major concern when using these setups is preventing the numerical integration from becoming unstable during execution [33]. Because co-simulation can be used to deal with a very wide range of applications, these conditions may differ from the ones encountered in other co-simulation environments, where it is possible to theoretically determine acceptable error bounds for a given problem, or to modify the integration step-size to match the changing system dynamics [30]. In some cases, the introduction of errors at the coupling interface will give rise to latency instead of instability, which requires a different treatment [34].

The method described in this paper relies on the evaluation of energy residuals at the co-simulation interface, which are used as indicators of the stability of the numerical integration. Subsystem inputs are replaced during the initialization phase with a series of standard test functions and the communication step-size H is gradually increased until the stability limit is reached. Determining a priori an optimal set of integration parameters and configuration settings for each particular co-simulation application is not feasible in most cases [18], partly because the actual inputs of the numerical simulation are often unknown and the system behavior is highly dependent on these. However, the proposed method provides information to determine a stable co-simulation configuration in a simple manner with little additional effort for the user. The viability of the algorithm was verified in the co-simulation of benchmark problems composed of mechanical and hydraulic subsystems.

2 Methods

This section introduces the co-simulation methodology used in this research and the error estimation via conservation laws, and its application in non-iterative co-simulation. For the sake of clarity the coupling of only two subsystems is considered in this section, although the presented methods could be applied, as it is shown with a case-example, to co-simulation environments with more than two co-simulation units.

2.1 Co-simulation setup

A non-iterative, parallelizable Jacobi-scheme co-simulation approach was selected for the purposes of this work. This scheme is often found in applications that require computational effort to be kept to a minimum, such as real-time environments like heavy machinery simulators, and HiL and SiTL platforms. A generic co-simulation manager tool, intended to couple several subsystems in multi-rate co-simulation applications, was implemented to test the methods described in this paper. The scope of this research is limited to subsystems with fixed integration step-sizes and matching communication time grids, although with multi-rate integration. The proposed method, however, could also be used with variable-step integrators.

Figure 1 shows the model used in this research, namely two subsystems, 1 and 2, coupled in a co-simulation setup through a co-simulation manager. Each subsystem has its own set of internal states, not disclosed to the rest of the simulation environment. The integration of these internal states is carried out with integrator $\int_1$ and step-size h_1 for subsystem 1, and $\int_2$ and h_2 for subsystem 2. The communication between 1 and 2 takes place only at discrete points in time, separated by macro time steps of duration H. At communication points, the subsystems provide their outputs $\mathbf{y}_1$ and $\mathbf{y}_2$ to the co-simulation manager and receive their inputs $\mathbf{u}_1$ and $\mathbf{u}_2$. Subsystem 2, moreover, receives a set of external inputs $\mathbf{w}_2$. These may represent interactions with elements that are not contained in the computational environment, e.g., inputs provided by a human operator.

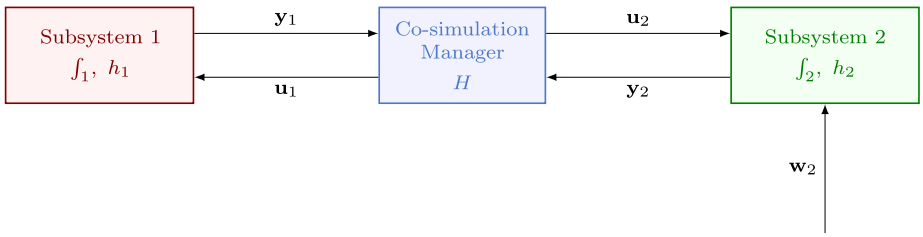


Fig. 1 Two subsystems in a co-simulation setup with external inputs $\mathbf{w}_2$

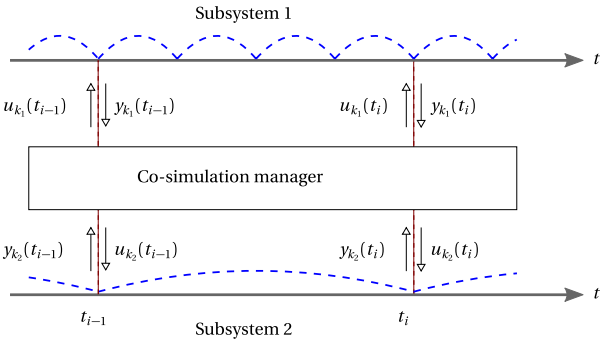


Fig. 2 Power bonds in a Jacobi-scheme co-simulation

2.2 Power bonds

Multiple error estimation methods have been applied to co-simulation setups [15]. In practice, however, the available options are limited by the application requirements, such as no rollback and fixed step-sizes in real-time cases. A possible solution for this problem was recently introduced by Sadjina et al. in [28], where a conservation law based approach was proposed for a non-iterative adaptive step-size control and error estimation. The method is based on the concept of power bonds and it allows one to monitor the spurious energy created or lost in the signal extrapolation at the co-simulation interface, and requires only little domain knowledge about the co-simulation units. As the underlying theory of power bonds, and bond graphs in general, has been extensively presented in the literature [8, 24], only the topics relevant to this work are presented here.

A power bond is defined by a pair of variables such as velocity and force, often referred to as flow and effort in the literature, whose product is a physical power, and it connects two systems via their power ports, i.e., parts of the system where power flows in and out. In the context of co-simulation, this pair of variables corresponds to the inputs and outputs exchanged between the subsystems, i.e., the coupling variables. To illustrate this, consider the progress of Jacobi-scheme co-simulation between two communication time steps shown in Fig. 2. At communication point $i-1$ subsystems 1 and 2 receive inputs $u_{k_1}(t_{i-1})$ and $u_{k_2}(t_{i-1})$, respectively. As explained previously, both subsystems then integrate their internal states independently until the next communication point i, when they send outputs $y_{k_1}(t_i)$ and $y_{k_2}(t_i)$, computed based on their previous inputs $u_k(t_{i-1})$. Here, the subscript k refers to power bond k, and, in general, u and y can be vectors of power bond variables, as will be explained in Sect. 2.3.

As inputs u are only known at the communication points, subsystems that use shorter internal time steps $h < H$ need to extrapolate their inputs until the next update instant. In the

case shown in Fig. 2, subsystem 1 needs extrapolated inputs u_{k_1} at every internal micro time step. If only the input value u is known, but not its derivatives, Lagrange polynomials or least squares fitting can be used to carry out this extrapolation. If the input derivatives $\dot{u}$, $\ddot{u}$, ..., etc. are also available, the inputs can also be integrated over the communication time step. Both input extrapolation and integration can be carried out either by subsystem 1 or by the co-simulation manager; in the latter case, the subsystem needs to establish a communication with the manager every time that it takes a new internal time step.

In any case, the concept of power bonds can be used to monitor the flow of energy between the systems. Subsystems 1 and 2 exchange energy via power ports k_1 and k_2, correspondingly, and thus create a power bond k between the systems. From the Subsystem 1 point of view, the total transmitted power P_{k_1} at a communication time t can be computed as

$$P_{k_1}(t) = \tilde{u}_{k_1}(t) y_{k_1}(t), \tag{1}$$

where $\tilde{u}_{k_1}(t)$ is the extrapolated value of the input $u_{k_1}(t)$ at the end of the communication time step, and $y_{k_1}(t)$ is the output. Correspondingly, the power transmitted by Subsystem 2, P_{k_2} takes the form

$$P_{k_2}(t) = \tilde{u}_{k_2}(t) y_{k_2}(t), \tag{2}$$

where $\tilde{u}_{k_2}(t)$ and $y_{k_2}(t)$ are, again, the inputs and outputs, respectively.

Ideally, no power should be lost at the coupling interface between the subsystems. Thus, the following balance should hold at any communication point:

$$-(P_{k_1} + P_{k_2}) = 0. \tag{3}$$

However, as the extrapolation error is an inherent issue in non-iterative co-simulation due to the independent integration of the subsystems, Eq. (3) will be violated during the simulation, and, therefore

$$-(P_{k_1} + P_{k_2}) \neq 0. \tag{4}$$

Since the transmitted energy is the power integrated over time, this violation of power leads to the accumulation of errors in the system energy.

2.3 Energy residuals

The violation of energy conservation that unavoidably occurs in non-iterative co-simulation interfaces can be exploited to monitor the simulation accuracy. As presented in [28], a residual power δP can be defined for the power bond k as

$$\delta P_k = -(P_{k_1} + P_{k_2}). \tag{5}$$

The dot product can be utilized to conveniently compute the residual power for the whole system. For power bonds $[k \ l \ \ldots]$, the inputs can be grouped into a vector $\tilde{\mathbf{u}} = [\tilde{u}_{k_1} \ \tilde{u}_{k_2} \ \tilde{u}_{l_1} \ \tilde{u}_{l_2} \ \ldots]^{\mathrm{T}}$, and the outputs, similarly, $\mathbf{y} = [y_{k_1} \ y_{k_2} \ y_{l_1} \ y_{l_2} \ \ldots]^{\mathrm{T}}$. Then the residual power for the whole system can be expressed as

$$\delta P = -\tilde{\mathbf{u}} \cdot \mathbf{y} \tag{6}$$

where $\cdot$ stands for a dot product between two vectors.

In [28], zero-order hold extrapolation was assumed for the input values $\mathbf{u}$, and, thus, following the notation used in Fig. 2, the residual power δP_k at communication time step i can be written as

$$\delta P_k(t_i) = -\mathbf{u}_k(t_{i-1}) \cdot \mathbf{y}_k(t_i). \tag{7}$$

On the other hand, if we assume that an extrapolated input is available or can be computed in the co-simulation interface, a more accurate value for the residual power at communication time step i can obtained:

$$\delta P_k(t_i) = -\tilde{\mathbf{u}}_k(t_i) \cdot \mathbf{y}_k(t_i). \tag{8}$$

The corresponding energy residual can now be defined [28] by an integral

$$\delta E_k(t_i) = \int_{t_{i-1}}^{t_i} \delta P_k(t)\, dt, \tag{9}$$

which gives the energy incorrectly added to the coupled system as a result of the extrapolation errors during the time step $t_{i-1} \rightarrow t_i$. These local energy fluxes will inevitably alter the energy balance of the coupled system, and, therefore, deteriorate the co-simulation accuracy.

Numerical approximation can be used to evaluate the integral in Eq. (9). For a constant extrapolation $\tilde{\mathbf{u}}_k(t_i) = \mathbf{u}_k(t_{i-1})$ the rectangle quadrature rule can be used, resulting in

$$\delta E_k(t_i) \approx \delta P_k(t_i) H(t_i), \tag{10}$$

where $H(t_i)$ is the communication step-size. For higher-order extrapolation, higher-order quadrature rules should be used, to prevent integration errors from becoming of the same order as the accumulated residuals. In the case of linear extrapolation, the energy residual can be approximated with the trapezoidal rule

$$\delta E_k(t_i) \approx -\frac{H}{2}\big(\delta P_k(t_{i-1}) + \delta P_k(t_i)\big). \tag{11}$$

For $m = 2$, in turn, Simpson's rule might be appropriate. However, since this formula would require one to know the outputs $\mathbf{y}_k(t_{i-1} + 0.5H)$, which may not be available, in this work trapezoidal rule is used for the extrapolation orders $m > 1$.

2.4 Co-simulation configuration search

The above presented concept of energy residual [28] allows us now to monitor the co-simulation accuracy. The energy residual is, however, only a scalar value that is not normalized, thus, it may be difficult to interpret on its own. While it is obvious that a relatively low amount of spurious energy with respect to the total transmitted energy accumulated or dissipated at the coupling process indicates stable and accurate co-simulation, the exact point at which the co-simulation accuracy degrades to insufficient or the stability is lost is much less clear. In addition, the residual energy is dependent on the extrapolation method used in the information exchange, which further complicates the accuracy estimation. For this reason, in [28] the energy residual was utilized to develop a scalar error indicator, which requires tuning for each power bond, and an adaptive step-size control for non-iterative co-simulation. Here, a different direction is taken to methodologically evaluate the co-simulation accuracy under different configurations, based on the energy residual.

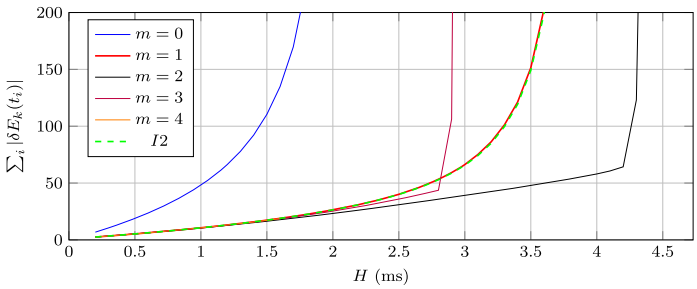


Fig. 3 Relation between energy residual E_k and communication step-size H

The definition of co-simulation accuracy, however, also needs to be discussed in this context. The energy residuals do measure the accuracy of the simulation, but only at the co-simulation interface. Assuming that the internal integrators of the subsystems do not introduce any additional errors, the energy residuals directly define the system-level energy error. However, in practice this will not be the case, since the subsystem integrators will introduce their own error into the system-level solution. In this work, it is assumed that the internal integrators are accurate enough for the purposes of the analyst, and therefore the accuracy depends only on the co-simulation configuration. Since the scope of the work is in the area of HiL/SiTL simulators, it is assumed that numerically stable solutions also have sufficient accuracy.

The basis of the proposed approach is illustrated in Fig. 3, where the dependency between energy residual and communication step-size for a nonlinear case example described in Sect. 3.1.1 is presented. In the figure, the energy residual is plotted for polynomial extrapolations of orders 0 to 4 ($m = 0$ to $m = 4$) and integration of the system inputs ($I2$). As can be seen, when the communication step-size is reduced, the residuals appear to approach zero. In the figure, the extrapolation methods other than $m = 0$ yield practically the same value for the linear approximation of the residual by the trapezoidal rule of Eq. (11) at small communication step-sizes, and, thus only one line is visible. In theory, the depicted errors in the energy residuals should be $\mathcal{O}(H^{m+1})$ [28]. As the figure indicates, this property is lost when the simulation goes unstable, i.e., $\sum_i |\delta E_k(t_i)| \gg \mathcal{O}(H^{m+1})$. When H is increased the energy residuals are approximately linearly dependent on the communication step-size at first, until either exponential growth is seen, i.e., the accuracy gradually degrades, or the residual suddenly rises to a high value, i.e., the simulation goes unstable. This behavior is further demonstrated in Sect. 4.

In this work, the observed relation between the communication step-size and the accumulated energy residual over all the macro time steps is taken as an indicator of a stable and accurate co-simulation; it is assumed that the co-simulation is stable and accurate as long as the relation between the accumulated energy residual and communication step-size is approximately linear. This criterion allows us to methodologically evaluate the accuracy of co-simulation under different configurations in a simple manner, with modifications to the co-simulation scheme only implemented in the co-simulation manager. As the number of possible configurations can be too large to experiment manually, an automated procedure, which is described in Sect. 2.4.1, is used to parse through the possible set of configurations. The produced information can be used by an analyst to find a suitable compromise between accuracy and efficiency for the case under study.

However, for practical reasons it would be useful or even required, e.g., when the external inputs $\mathbf{w}$ are not known, to perform the evaluation in advance without performing the full

simulation for each possible configuration. For this reason, and to provide an example of use-case for the method, an application for the methodology is put forward to find a suitable configuration for a co-simulation case in which the external inputs are not know. In practice, the automated procedure uses a set of pre-defined tests for the external inputs **w** when they are unknown.

2.4.1 Automated accuracy evaluation

The entire process of testing the different configurations with several communication step-sizes can be easily automated. In this section an algorithm for the automatic testing procedure, illustrated in Fig. 4, is proposed. Since the external inputs **w** are not, in general, known in advance, they are here replaced with a series of pre-defined tests, described in Sect. 2.4.2 in detail, to examine the system response under different excitations. To make the overall process efficient, the simulation time for each test is defined to be relatively short, i.e., only long enough to verify the response of each test. In the presented example that will follow, a simulation time of 1 s is used for step, ramp, and impulse test, and for the sinusoidal tests the length of one period is used. As information exchange at communication points adds overhead to the co-simulation process, the configuration that achieves the largest communication step-size can often be considered the most efficient for the system. While it would also be possible to tune the internal step-sizes of the co-simulation units, in this work the subsystem with the largest internal step-size is assumed to share the communication step-size, i.e., $h = H$, and in subsystems with $h < H$, the internal step-size is kept constant.

Admittedly, the assumption of $h = H$ in the slow subsystem leads to internal integration errors being scaled with H. While the use of smaller internal step-sizes would better decouple the errors that occur in the co-simulation interface and in the subsystem integrators, in practical co-simulation environments this would decrease the overall efficiency of the simulation, as the higher number of integration steps of slow subsystems would increase the total computational load of the problem. As the proposed test methodology aims to reproduce actual operating conditions, the assumption $h = H$ in the slow subsystem can be justified. Note that the procedure itself is valid for any value of h inside the subsystems.

In the first step, an initial communication step size H_{ini} and an increment H_{inc} are selected for the system. The initial communication step size should be selected small enough for a stable solution to be found, i.e., for the energy residual to be in the linear region. It may take more than one attempt to find suitable H_{ini} and H_{inc} for the system, if no knowledge about the system behavior is available. Secondly, a configuration for the co-simulation is selected and applied. In this paper, this includes the selection of extrapolation methods and options that are available for the co-simulation manager and the integrator methods used in the subsystems; other configuration environments can have different tunable options as part of their configuration space. The selection of the test input **w** for the system takes place at this point as well. In the third step, a simulation is run with the given configuration. Upon completion, if the simulation had been run for the first time with the current configuration, i.e., $H = H_{\text{ini}}$, the communication step-size is incremented and the simulation is run again. If the second simulation run succeeds, the linear relation between communication step-size and energy residual is checked. If the latest simulation is still in the linear region, the communication step-size is incremented, $H = H + H_{\text{inc}}$ and the simulation is run again. If not, or if the simulation failed, the configuration and the largest successful communication step-size are saved, a new configuration set is selected and the process starts again. Finally, after the configuration set space is exhausted, the obtained results are parsed for the user.

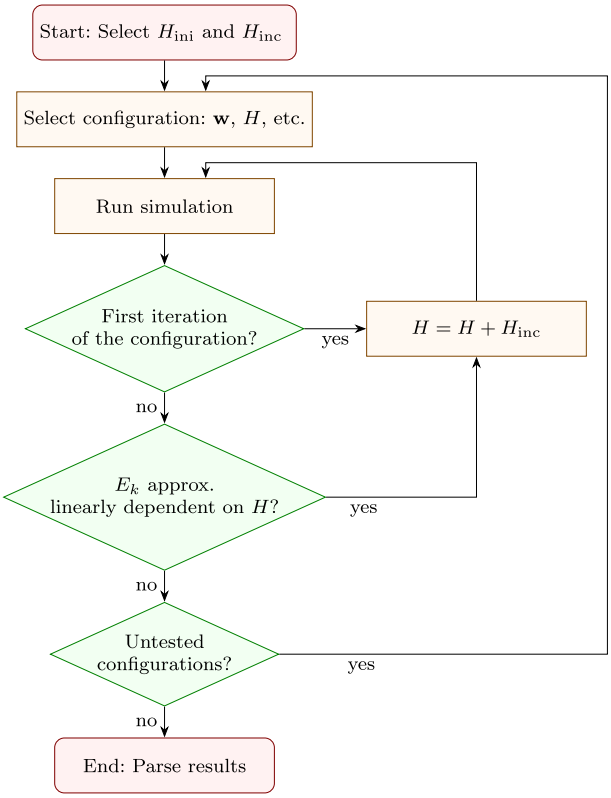


Fig. 4 Flowchart of the automated accuracy evaluation

2.4.2 *External input tests*

To assess the system behavior under different inputs, four typical functions, namely step, ramp, impulse, and sinusoidal are used.

In the following, v_0 is the initial value of the input, and v_{min} and v_{max} are the minimum and maximum values that it can take. In addition, the tests are assumed to have a length of 1 second. The step function, here denoted by w_1, is defined piece-wise as follows:

$$w_1(t) = \begin{cases} v_0, & t < 0.5, \\ v_{\text{max}}, & t \geq 0.5, \end{cases} \tag{12}$$

where t is the simulation time. It is assumed that the subsystem inputs can be discontinuous, which is often the case if discretely sampled digital controllers are used to feed inputs to the system. The function in Eq. (12) can easily be modified to obtain a continuous approximation of a step function if required.

A ramp function, w_2 is also defined piece-wise, as follows:

$$w_2(t) = \begin{cases} v_0, & t < 0.5, \\ v_0 + 2(t - 0.5)v_{\text{max}}, & t \geq 0.5, \end{cases} \tag{13}$$

where the input reaches its maximum value at the end of the simulation time of 1 second. An impulse function is defined as follows:

$$w_3(t) = \begin{cases} v_{\max}, & 0.5 - \frac{d}{2} \leq t < 0.5 + \frac{d}{2}, \\ v_0, & \text{otherwise}, \end{cases} \tag{14}$$

where d is the width of the signal determined by user. Determining the pulse width requires some knowledge of the system, but it is not unreasonable to assume that the analyst responsible for the co-simulation has enough information for an educated guess.

Finally, a sinusoidal function is defined as

$$w_4(t) = v_0 + a \sin(\omega t), \tag{15}$$

where a is the user-defined amplitude of the signal and ω is the input frequency. Again, a reasonable level of system knowledge is required to set suitable values for these parameters, but it can be assumed that, for instance, the maximum and minimum values and the frequency limits of the input signals are known.

3 Examples

Two case examples were implemented to evaluate the proposed method. A 2-D model of a single-actuator crane was used first to test the method in single-input co-simulation scenarios. A second actuator was later added to this model to evaluate cases with multiple inputs. In this section, both examples are presented together with the hydraulics model used in this research. The different co-simulation configurations tested are subsequently described.

3.1 Multibody models and algorithms

The mechanical systems used as test problems were modeled as multibody systems composed of rigid links, using a set of generalized coordinates $\mathbf{q}$ subjected to a set of kinematic constraints $\boldsymbol{\Phi} = \mathbf{0}$.

3.1.1 Single-actuator model

A 2-D model of a hydraulically actuated crane is shown in Fig. 5. A similar model was described in [22] and used as test problem in [25].

Link 1 is a rod of length L and distributed mass m. Link 2 has length L_h and is considered to be massless. Two point masses m_p and m_h are placed at points Q and R. The system moves under gravity effects and is actuated with a hydraulic piston that connects points B and P. The values of the system properties used in the numerical experiments are summarized in Table 1. The generalized coordinates used to model this system are the x and y global coordinates of points P and R, and angle θ_1: $\mathbf{q} = [x_\mathsf{P}, y_\mathsf{P}, \theta_1, x_\mathsf{R}, y_\mathsf{R}]^\mathrm{T}$. The system has two degrees of freedom; accordingly, three kinematic constraints are imposed on coordinates $\mathbf{q}$

$$\boldsymbol{\Phi} = \begin{bmatrix} L\cos\theta_1 - 2x_\mathsf{P} \\ L\sin\theta_1 - 2y_\mathsf{P} \\ (x_\mathsf{R} - L\cos\theta_1)^2 + (y_\mathsf{R} - L\sin\theta_1)^2 - L_\mathrm{h}^2 \end{bmatrix} = \mathbf{0}. \tag{16}$$

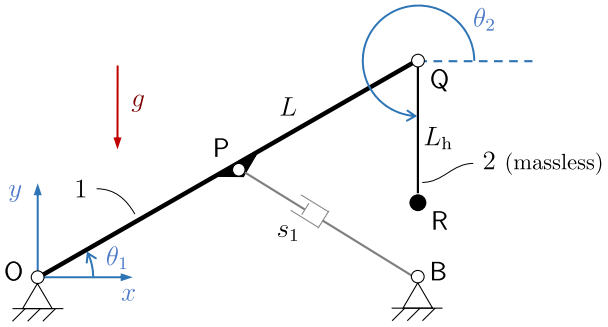


Fig. 5 Single-actuated planar model of a hydraulic crane

Table 1 Mechanical parameters of the single-actuated model

Parameter	Symbol	Value
Length of link 1	L	1.0 m
Length of link 2	L_h	0.5 m
Mass of link 1	m	200 kg
Point mass at Q	m_p	250 kg
Point mass at R	m_h	100 kg
Coordinates of fixed point B	$(x_\mathrm{B}, y_\mathrm{B})$	$(\sqrt{3}/2, 0)$ m
Initial angle, link 1	$(\theta_1)_0$	$\pi/6$ rad
Initial angle, link 2	$(\theta_2)_0$	$3\pi/2$ rad
Gravity	g	-9.81 m/s^2

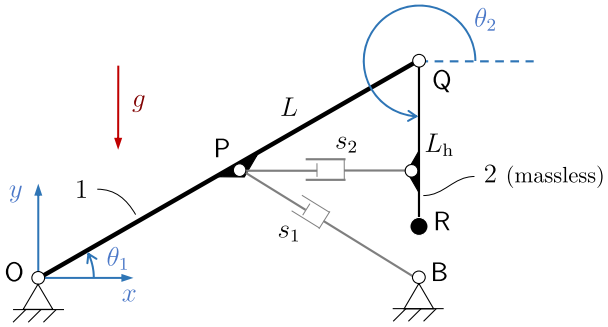


Fig. 6 Double-actuated planar model of a hydraulic crane

3.1.2 Two-actuator model

Adding a second hydraulic actuator to the mechanical system in Sect. 3.1.1 we obtain the system in Fig. 6. The mechanical properties of the system are those reported in Table 1. Both actuators are identical; the second one acts between point P and the middle point of link 2. The generalized coordinates $\mathbf{q}$ and kinematic constraints $\boldsymbol{\Phi} = \mathbf{0}$ used to model this example are the same ones used in Sect. 3.1.1 for the single actuator model.

3.1.3 Multibody system dynamics algorithms

The dynamics of the examples in Sects. 3.1.1 and 3.1.2 was solved by means of two augmented Lagrangian algorithms described in [11]. The trapezoidal rule (TR) was used as

integration formula. The first algorithm is an index-1 algorithm (I1AL), in which the system accelerations $\ddot{\mathbf{q}}$ are used as primary integration variables. The dynamics is solved via the following iterative process:

$$\begin{aligned}&\left(\mathbf{M}+\boldsymbol{\Phi}_{\mathbf{q}}{}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{q}}\right)\ddot{\mathbf{q}}_j+\boldsymbol{\Phi}_{\mathbf{q}}{}^{\mathrm{T}}\boldsymbol{\lambda}_j^*\\&\quad=\mathbf{Q}-\boldsymbol{\Phi}_{\mathbf{q}}{}^{\mathrm{T}}\alpha\left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}+\dot{\boldsymbol{\Phi}}_t+2\xi\omega\dot{\boldsymbol{\Phi}}+\omega^2\boldsymbol{\Phi}\right),\end{aligned}\tag{17a}$$

$$\boldsymbol{\lambda}_{j+1}^*=\boldsymbol{\lambda}_j^*+\alpha\left(\ddot{\boldsymbol{\Phi}}+2\xi\omega\dot{\boldsymbol{\Phi}}+\omega^2\boldsymbol{\Phi}\right),\tag{17b}$$

where j stands for the iteration number. In a problem with n generalized coordinates and m kinematic constraints, term $\mathbf{M}$ is the $n\times n$ system mass matrix, $\mathbf{Q}$ stands for the $n\times 1$ generalized applied forces term, $\boldsymbol{\Phi}_{\mathbf{q}}$ is the $m\times n$ Jacobian matrix of the constraints, $\boldsymbol{\Phi}_t=\partial\boldsymbol{\Phi}/\partial t$, and $\boldsymbol{\lambda}^*$ represents the m Lagrange multipliers of the method. Term α is a scalar penalty factor, and ξ and ω are scalar parameters that play a role similar to those used in Baumgarte stabilization.

The second method is an index-3 augmented Lagrangian algorithm (I3AL) that uses the generalized coordinates $\mathbf{q}$ as primary variables. The dynamics equations (17a), (17b) are combined with the trapezoidal rule expressions

$$\begin{aligned}\dot{\mathbf{q}}_{i+1}&=\frac{2}{h}\mathbf{q}_{i+1}-\widehat{\dot{\mathbf{q}}}_i;\quad\text{where }\widehat{\dot{\mathbf{q}}}_i=\frac{2}{h}\mathbf{q}_i+\dot{\mathbf{q}}_i,\\\ddot{\mathbf{q}}_{i+1}&=\frac{4}{h^2}\mathbf{q}_{i+1}-\widehat{\ddot{\mathbf{q}}}_i;\quad\text{where }\widehat{\ddot{\mathbf{q}}}_i=\frac{4}{h^2}\mathbf{q}_i+\frac{4}{h}\dot{\mathbf{q}}_i+\ddot{\mathbf{q}}_i,\end{aligned}\tag{18}$$

where h is the integration step-size, and subscript i stands for the integration time step, to obtain a system of nonlinear equations at time step $i+1$ in the form

$$\mathbf{g}(\mathbf{q},\dot{\mathbf{q}})=\mathbf{0},\tag{19}$$

which is solved by means of Newton–Raphson iteration. Upon convergence of the generalized coordinates $\mathbf{q}$, the generalized velocities and accelerations $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are projected upon the constraint manifold to ensure the satisfaction of the derivatives of the kinematic constraints. The method is described in detail in [12] and [13]. Both I1AL and I3AL methods have displayed a robust and efficient performance in the simulation of challenging multibody dynamics problems, e.g., [17].

3.2 Hydraulic model

The hydraulic circuit used to command the actuators was represented using a simple model. The schematics of the circuit are presented in Fig. 7 and its physical parameters can be found in Table 2. The cylinder is connected to a pump and a tank, both of which are simplified as constant pressure sources, and the flow is controlled by a $4/3$ directional valve. In addition, at the piston side the flow is restricted with a throttle valve. This model was originally described in [27] and later used with minor modifications in [26]; it is used in this research with different parameters and with the addition of end dampers to the cylinder. The modeling approaches have not been altered for this work. In the case study, the cylinder is connected to the manipulators described at Sects. 3.1.1 and 3.1.2 such that the piston side is attached to point B, and the rod side to point P.

Lumped fluid theory [36] is applied to compute the pressure rates in the hydraulic volumes. This approach splits the circuit into volumes, in which the pressures can be assumed to

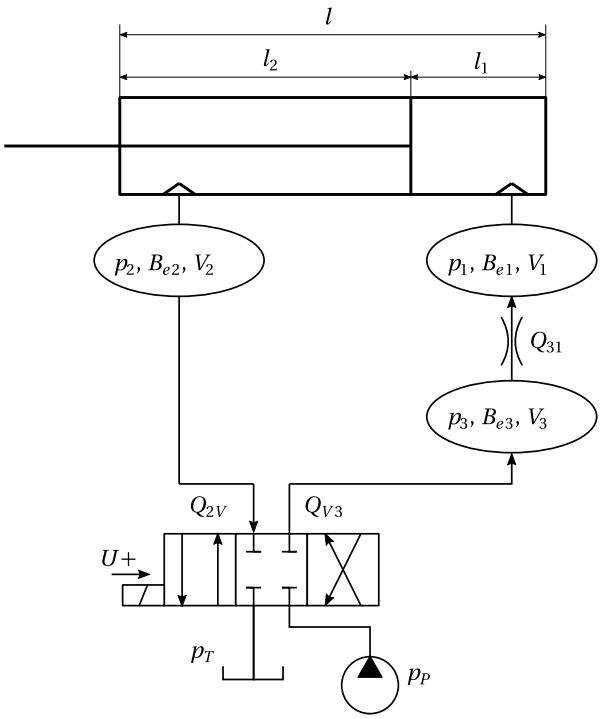


Fig. 7 A hydraulic circuit with three volumes

be equally distributed, i.e., phenomena such as pressure waves are ignored. For the pressure rates in the volumes V_1, V_2, and V_3, this approach yields the following expressions:

$$\dot{p}_1 = \frac{B_{e1}}{V_1}(Q_{31} - A_1\dot{s}), \tag{20}$$

$$\dot{p}_2 = \frac{B_{e2}}{V_2}(A_2\dot{s} - Q_{2V}), \tag{21}$$

$$\dot{p}_3 = \frac{B_{e3}}{V_3}(Q_{V3} - Q_{31}), \tag{22}$$

where B_{e1}, B_{e2}, and B_{e3} are the effective bulk moduli of the volumes, A_1 and A_2 are the piston-side and rod-side areas, respectively, of the cylinder, $\dot{s}$ is the cylinder velocity, and, finally, Q_{31}, Q_{2V}, and Q_{V3} are the volume flows over valves. The positive directions of the volume flows are depicted in Fig. 7. The pump and tank pressures, p_P and p_T, correspondingly, are assumed to be constant.

The effective bulk moduli take the oil and chamber compressibility into account, and can be expressed for the volumes as

$$B_{e1} = \left(B_o^{-1} + \left(\frac{V_1 B_c}{A_1 l_1}\right)^{-1} + \left(\frac{V_1 B_h}{V_{h1}}\right)^{-1}\right)^{-1}, \tag{23}$$

$$B_{e2} = \left(B_o^{-1} + \left(\frac{V_2 B_c}{A_2 l_1}\right)^{-1} + \left(\frac{V_2 B_h}{V_{h2}}\right)^{-1}\right)^{-1}, \tag{24}$$

Table 2 Parameters of the example circuit

Parameter	Symbol	Value
Length of the piston	l	0.3 m
Initial piston-side length	l_{1_0}	0.05 m
Initial rod-side length	l_{2_0}	0.25 m
End damper length	l_d	0.008 m
End damper spring coefficient	k_d	1×10^7 N/m
End damper damping coefficient	c_d	5×10^3 Ns/m
Initial actuator length	s_0	0.5 m
Fluid density	ρ	850 kg/m^3
Oil bulk modulus	B_o	1500 MPa
Cylinder bulk modulus	B_c	31 500 MPa
Hose bulk modulus	B_h	150 MPa
Volume of hose 1	V_{h1}	3.14×10^{-5} m^3
Volume of hose 2	V_{h2}	7.85×10^{-5} m^3
Volume of hose 3	V_{h3}	7.85×10^{-7} m^3
Diameter of piston	d_1	80 mm
Diameter of piston rod	d_2	35 mm
Tank pressure	p_T	0.1 MPa
Pump pressure	p_P	7.6 MPa
Cylinder efficiency	η	0.88
Throttle discharge coefficient	C_d	0.8
Semi-empirical coefficient	C_v	1.069×10^{-8} m^3/sV$\sqrt{\text{Pa}}$
$-45°$ phase shift frequency	f_{-45}	35 Hz

$$B_{e3} = \left(B_o^{-1} + B_h^{-1}\right)^{-1}, \tag{25}$$

where B_o, B_c, and B_h are the bulk moduli of the oil, the cylinder, and the hoses, correspondingly.

3.2.1 Valve models

The system valves are modeled following a semi-empirical approach [19], which allows for a simple and accurate model to be built even if all the internal details are not known. For a simple throttle valve the volume flows can be written as follows:

$$Q = \begin{cases} C_t\sqrt{|\Delta p|}, & \Delta p > 0, \\ 0, & \Delta p = 0, \\ -C_t\sqrt{|\Delta p|}, & \Delta p < 0, \end{cases} \tag{26}$$

where C_t is the semi-empirical flow rate coefficient and Δp is pressure difference over the valve. While the semi-empirical coefficient is usually computed from empirical data, here we obtain it from $C_t = C_d A_t \sqrt{\frac{2}{\rho}}$, an equation for valves with simple geometry where C_d is the flow discharge coefficient and A_t is the throttle area.

For the directional valve the following equations can be used:

$$Q = \begin{cases} C_v U \sqrt{|\Delta p|}, & \Delta p > 0, \\ 0, & \Delta p = 0, \\ -C_v U \sqrt{|\Delta p|}, & \Delta p < 0, \end{cases} \tag{27}$$

where C_v is the semi-empirical coefficient of the directional valve and U is the feedback signal that corresponds to spool position. The coefficient C_v can be determined from empirical data with $C_v = \frac{Q}{U\sqrt{\Delta p}}$ when the volume flow is known at one operational point. To avoid numerical problems with the solution of Eqs. (26) and (27) near zero pressure difference, these models are replaced with a linear model when $|\Delta p| < 2$ bar.

To take the valve dynamics into account, the spool position is obtained from the solution of a first-order differential equation

$$\dot{U} = \frac{U_{\text{ref}} - U}{\tau}, \tag{28}$$

where the time constant of the valve, τ, is computed as

$$\tau = \frac{1}{2\pi f_{-45}}, \tag{29}$$

and U_{ref} is the user-defined reference signal fed to the valve and, here, it can take values from -10 V to 10 V, zero being the closed position. In Eq. (29), f_{-45} is the frequency at $-45°$ phase shift, read from the Bode diagram of the valve.

3.2.2 Cylinder force

The cylinder force can be computed based on the pressure difference over the piston, and the velocity-dependent friction term. In addition, the end dampers can be taken into account in the force computation. Thus, the force can be expressed as

$$F = p_1 A_1 - p_2 A_2 - c\dot{s} + F_d, \tag{30}$$

where c is the viscous coefficient and F_d is the end damper force. In this case, the end dampers are modeled as a spring–damper system:

$$F_d = \begin{cases} k_d(l_d - l_1) - c_d\dot{s} & l_1 \le l_d, \\ -k_d(l_d - l_2) - c_d\dot{s} & l_2 \le l_d, \\ 0 & \text{otherwise}, \end{cases} \tag{31}$$

where l_d is length of the end damper, and k_d and c_d are the spring and damping coefficients of the damper, respectively.

3.3 Co-simulation configuration

The implemented co-simulation manager is used to couple the above described models by means of a Jacobi-scheme, and also to compute the energy residuals at the interface. The available variables for the information exchange are, in the current implementation, the cylinder length, rate, and acceleration, s, $\dot{s}$, and $\ddot{s}$, from the mechanical subsystem, and

Table 3 Parameter space

Option	Mech.	Hyd
Extrapolation	$m = 0$	Orders 0–4
Integration of exchanged variables	–	s and $\dot{s}$
Integrator	I3AL/I1AL	TR/FWE
Step-size	$h_\mathrm{m} = H$	$h_\mathrm{h} = 0.1$ ms $< H$

the actuator force F from the hydraulics. In addition, the co-simulation manager is responsible for evaluating the extrapolated values of these values, thus allowing use of Eq. (8) in the process of energy residual computation.

Table 3 shows the different co-simulation configuration options used in this study. The mechanical subsystem shares the communication step-size of the co-simulation, i.e., $h_m = H$. The hydraulic subsystem, in turn, has a shorter internal step-size of $h_\mathrm{h} = 0.1$ ms and therefore requires the input variables, in this case the cylinder length s and velocity $\dot{s}$, to be extrapolated. Polynomial extrapolation of orders 0–4, i.e., $m = 0$ to $m = 4$, and also integration of the variables based on their derivatives, i.e., $I2$ for the integration of s and $\dot{s}$, were implemented to this end. The extrapolation and integration are done on the basis of the variables exchanged at the communication points, extrapolation being performed by the means of Lagrange polynomials within the co-simulation manager.

The dynamics of the multibody systems was solved using the I1AL and I3AL augmented Lagrangian algorithms described in Sect. 3.1.3, in combination with the trapezoidal rule (TR) integration formula, the integrated variables being, correspondingly, either the accelerations $\ddot{\mathbf{q}}$ or positions $\mathbf{q}$. The hydraulics equations were integrated using either the trapezoidal rule or the forward Euler (FWE) formula, and the integrated variables, in turn, were the pressures $\dot{p}_1$, $\dot{p}_2$, and $\dot{p}_3$ from Equations (20)–(22). However, it must be noted that in this study the step-size for the integration of the hydraulics was kept constant and set to 0.1 ms, which was small enough to keep the integration stable with both integrators. Accordingly, changing the hydraulics integration formula had little impact on the co-simulation performance.

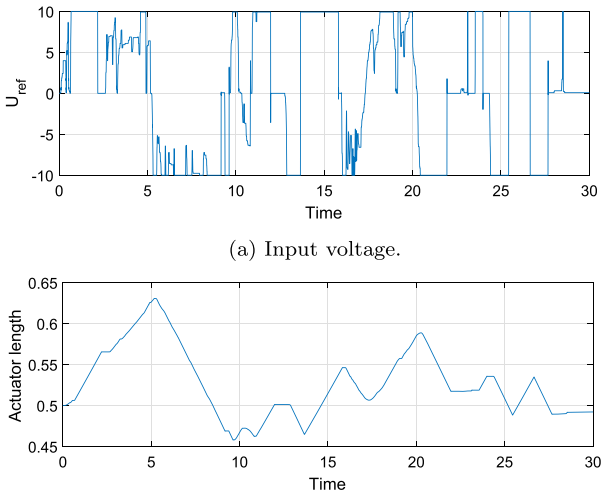
3.4 Configuration of the test functions

As the external input of the described hydraulic circuit takes values from −10 V to 10 V, 0 V being the initial position, configuration of the inputs is a rather straightforward process. The step and ramp functions defined in Sect. 2.4.2 can be directly used. As the coupled system can be expected to be quite slow, the width of the impulse is set to 0.4 s to give time for the system to react.

Regarding the sinusoidal input, its amplitude in this case can be set according to minimum and maximum values of the input voltage. Frequencies, in turn, are, in this case, set according to the time constant of the valve. Thus, 35 Hz and 70 Hz, i.e., the frequency at −45° phase shift and its second order, correspondingly, are used, as well as 17.5 Hz.

4 Results

The validity of the co-simulation accuracy and stability criteria discussed in Sect. 2.4 was confirmed comparing the results delivered by a series of simulations, in which some configurations complied with the criteria and others were outside the region assumed to be stable



(a) Input voltage.

(b) Actuator length.

Fig. 8 The joystick input applied to the directional valve, and the resultant actuator length

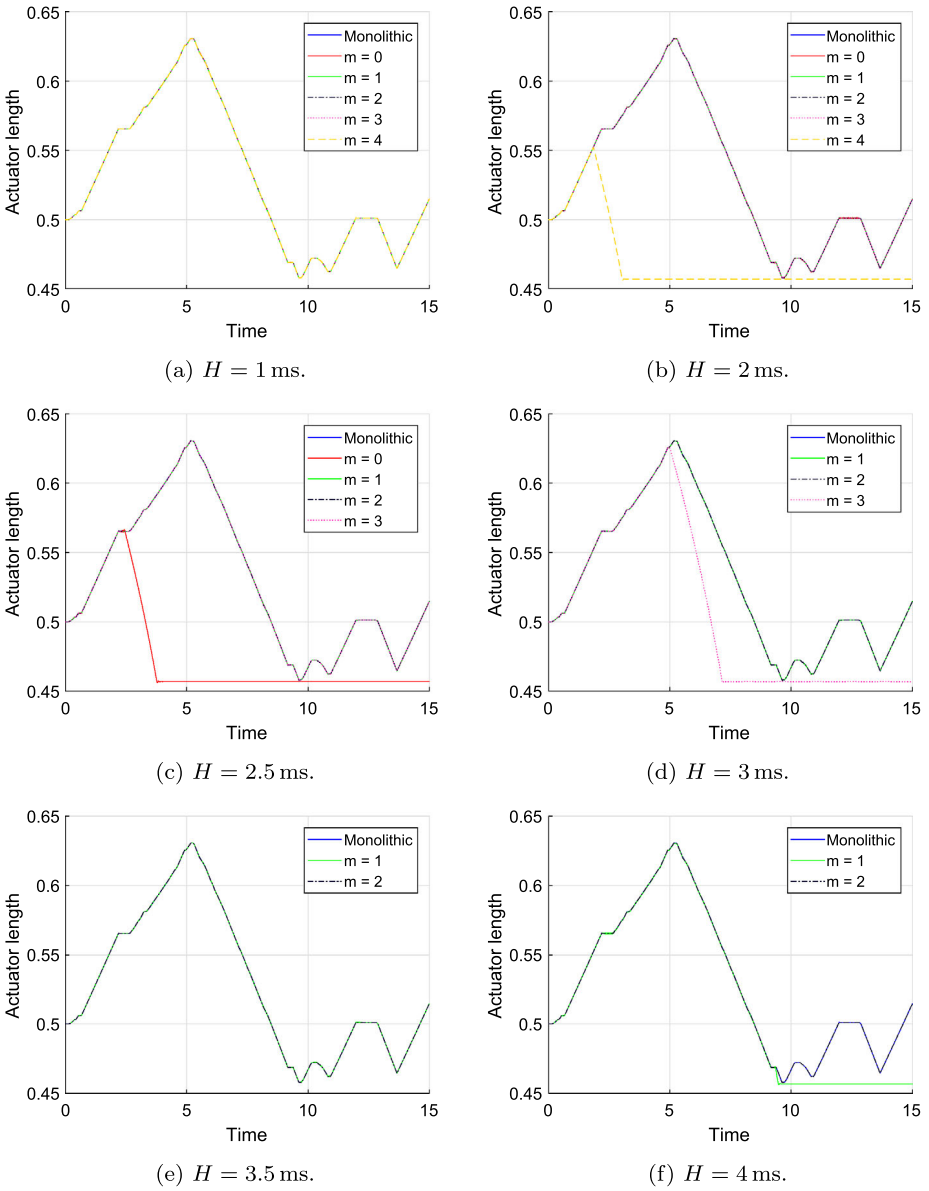
in Fig. 3. In addition, the use of test functions for the external inputs was evaluated. The single actuator model, which contains one external input, was first used for both cases, and, secondly, the double actuator model with multiple external inputs was examined. A user-generated joystick input was introduced in each case as reference signal U_{ref} to the directional valve in order to evaluate the criteria and the use of the test functions. Results are compared, when applicable, to a reference solution obtained with a monolithic scheme.

4.1 Single actuator model

Figure 8 shows the actuator length of the single-actuator model with the user-generated input. This input is used to examine the dependency between the accumulation of E_k and H, and it is generated such that the end dampers, which are represented by a contact model, are hit during the simulation. The figure is obtained with $H = 0.2$ ms and second-order extrapolation with I1AL and forward Euler integrators, which, as will be shown next, can be considered a stable and accurate configuration for the system. For this model and input the graph of E_k as a function of H is already presented in Fig. 3.

Figure 9 compares the co-simulation traces between the simulations that are in the approximately linear region and the non-linear region, according to Fig. 3. In the figures, the actuator length s, as obtained from the mechanical system, is depicted for different values of H during the first half of the simulation cycle. As can be seen, after a certain limit, depending on the extrapolation order, an increase in the communication step-size quickly yields incorrect results, and, thus in Fig. 9f, only $m = 2$ remains stable. Compared to Fig. 3 in Sect. 2.4, it is evident that the unstable solutions lie in the highly nonlinear region. As Figs. 9d and 9e demonstrate, however, the increase in error shown for $m = 1$ in Fig. 3 is barely evident in the position-level solution. This is better illustrated in Fig. 10, where the corresponding velocity-level solutions are depicted.

Figures 9 and 10 support the statement that an approximately linear relation, demonstrated in Sect. 2 with Fig. 3, between the energy residual accumulation and the communication step-size indicates an accurate and stable co-simulation.



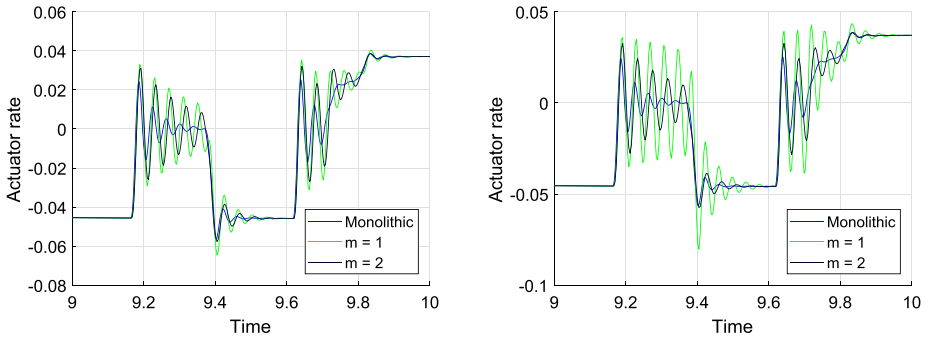
(a) $H = 1\,\text{ms}$. (b) $H = 2\,\text{ms}$.

(c) $H = 2.5\,\text{ms}$. (d) $H = 3\,\text{ms}$.

(e) $H = 3.5\,\text{ms}$. (f) $H = 4\,\text{ms}$.

Fig. 9 Stability and accuracy issues associated with the increase of the communication step-size

Next, the external input of the hydraulic subsystem was replaced with the pre-defined test inputs to determine a stable configuration a priori. The results obtained for the single-actuator model with the algorithm described in Sect. 2.4.1 are given in Table 4. In the table, the subscript of H refers to the communication step-size increment of the search algorithm; in all cases H_{ini} was set to 1 ms. As can be seen, the selected given integrator for the hydraulics and algorithm for the multibody dynamics were the same in all cases. The extrapolation method and communication step-size, in turn, were dependent on the external input.



(a) $H = 3\,\text{ms}$. (b) $H = 3.5\,\text{ms}$.

Fig. 10 A detail of the velocity-level solution

Table 4 Results of the pre-defined test for the single-actuator model

Test function	Extrapolation	$\int_m$	$\int_h$	$H_{0.2\,\text{ms}}$
Step	$m = 2$	I1AL	Euler	4.0 ms
Ramp	$m = 2$	I1AL	Euler	5.0 ms
Impulse	$m = 2$	I1AL	Euler	3.6 ms
Sinusoidal 17.5 Hz	$m = 2$	I1AL	Euler	2.6 ms
Sinusoidal 35 Hz	$m = 3$	I1AL	Euler	2.8 ms
Sinusoidal 70 Hz	$m = 2$	I1AL	Euler	4.4 ms

Regarding the obtained communication step-sizes and extrapolation methods, a comparison to the behavior depicted in Figs. 3 and 9 suggests that a stable and accurate configuration can be found with the method, given that the most pessimistic result of those shown in the table, i.e., the configuration with the smallest H, is taken as a final solution. In more detail, the table shows for the ramp and 70 Hz sinusoidal input test communication step-sizes of 5.0 ms and 4.4 ms, correspondingly, with $m = 2$. Figure 3, in turn, demonstrates unstable behavior at these communication step-sizes. A possible explanation for this result is that while the user-defined input hit the end-dampers of the cylinder, and thus created a demanding operation point for the simulation, these points were not reached during the short pre-defined tests. Alternatively, it is also possible that these tests, i.e., ramp and 70 Hz sinusoidal inputs, are not demanding enough for the step-size limits to be found.

Table 4 also shows that the step-size H and the extrapolation method selected by the configuration algorithm differ depending on the input signal used. This result agrees with [18], where it was shown that even for a simple model no all-encompassing optimal co-simulation method can be found, whereas here it is shown that even for exactly the same model a single optimal solution may not exist.

4.2 Double actuator model

In Sect. 4.1, a single subsystem with a single external input was considered. To evaluate the use of the criteria and the automated procedure with more than one subsystem, the double-actuator model is studied here. As described in Sect. 3.1.2, both hydraulic subsystems take one external input, and again, these inputs, resultant actuator lengths of which are depicted

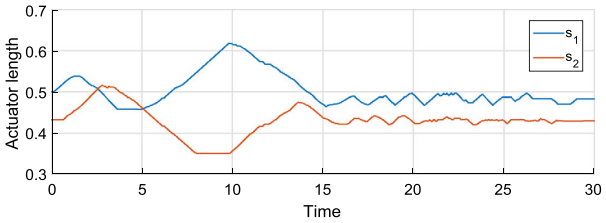


Fig. 11 Actuator lengths obtained with the joystick input

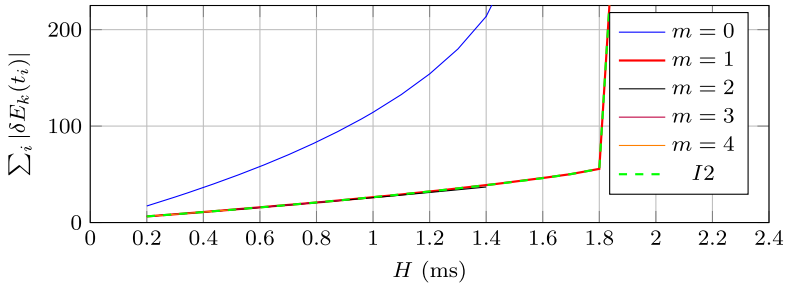
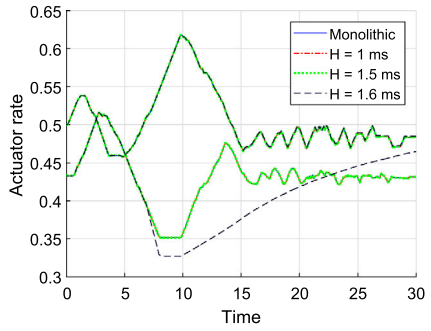


Fig. 12 Dependency between the energy residual and communication step-size with the double-actuator model

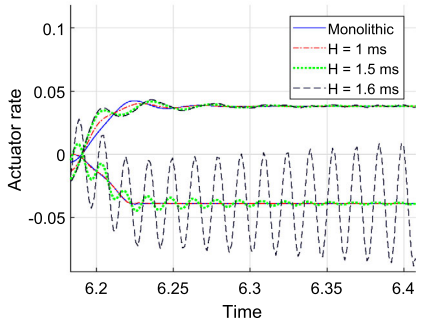
in Fig. 11, are defined by user. The figure is obtained with $H = 1$ ms and second-order extrapolation, which, as will be show next, is inside the stable region.

The dependency between the energy residual and the communication step-size with different extrapolation methods for the work cycle of Fig. 11 is depicted in Fig. 12. A behavior similar to the single-actuator model can be seen, although in this case the achieved step-sizes are smaller due to the indirect coupling between the hydraulic subsystems via the mechanical system. Again, the extrapolation methods other than $m = 0$ yield almost identical results at small communication step-sizes, and, thus only one line can be seen. The first-order extrapolation and integration of s and $\dot{s}$ also yield almost the same residuals. In all cases the energy residual grows at first almost linearly with respect to the communication step-size, until the simulation becomes unstable. In Fig. 13 the actuator positions and velocities for the two cylinders with zeroth-order hold and integration of s and $\dot{s}$ are illustrated. As can be seen, at certain point, depending on the used configuration, the simulation quickly becomes unstable. It is noteworthy that the error may not be clearly visible in the position-level solution, whereas the velocity-level solution shows discrepancies, as shown in the figure. Since the energy residual is computed from the velocities and forces, the erroneous behavior is, however, visible in $\sum_i |\delta E_k(t_i)|$.

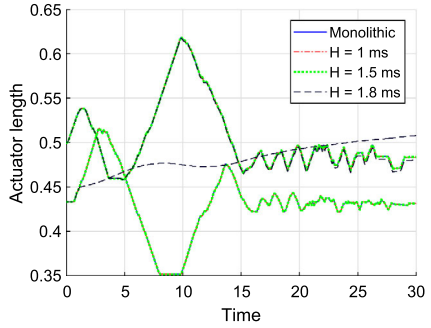
Again, the external inputs, i.e., the control signals U_{ref} of the hydraulic subsystems were replaced with the pre-defined tests. To limit the number of possible configurations, the same input functions were used for both subsystems. Results of the procedure are displayed in Table 5. In this case, the H_{ini} was set to 0.5 ms, and the increment also to 0.5 ms. Similar to the single-actuator model, the integrators for hydraulic systems and the formulation for the mechanical system yielded by the configuration algorithm were found to be the same for all test cases. Regarding the extrapolation methods and communication step-sizes, in this case it also seems that a stable configuration can be found, the most pessimistic result from the table being taken as the configuration for the system. While this result may not be the most efficient possible configuration, it seems to be, as Fig. 12 shows, in a stable region also for a more demanding simulation.



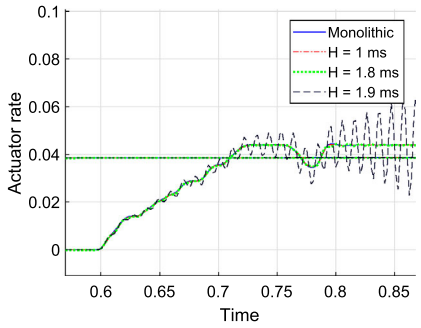
(a) Cylinder lengths with $m = 0$.



(b) Cylinder rates at the moment of instability with $m = 0$.



(c) Cylinder lengths with integration of s and $\dot{s}$.



(d) Cylinder rates at the moment of instability with integration of s and $\dot{s}$.

Fig. 13 Positions and elongation rates of the actuators

Table 5 Co-simulation options delivered by the configuration algorithm for each test case for the double-actuator model

Test function	Extrapolation	$\int_m$	$\int_h$	$H_{0.2\,\mathrm{ms}}$
Step	$I2$	I1AL	Euler	1.6 ms
Ramp	$m = 3$	I1AL	Euler	2.0 ms
Impulse	$I2$	I1AL	Euler	1.6 ms
Sinusoidal 17.5 Hz	$I2$	I1AL	Euler	1.8 ms
Sinusoidal 35 Hz	$m = 2$	I1AL	Euler	1.8 ms
Sinusoidal 70 Hz	$m = 3$	I1AL	Euler	1.8 ms

5 Discussion

The proposed configuration method discussed in this paper intends to provide a stable initial configuration for a Jacobi-scheme co-simulation in which some external inputs are unknown beforehand. Discontinuities in system dynamics need to be addressed, as they are often detrimental to system stability. In the demonstrated examples, the hydraulic circuit contains multiple potential sources of discontinuities, the end dampers being the most obvious as they represent a contact model. Should the spring and damping coefficients not be properly tuned, an otherwise stable simulation might fail, since the maximum communication step-size in this region might be significantly shorter than in other operation points. In addition to

the end dampers, the valve model may become numerically problematic if the linearization of the volume flows is neglected, or if the valve dynamics given in Eq. (28) are not included.

The need for careful modeling of the potentially discontinuous operating points holds true also in the context of the proposed algorithm, since, as is the case with the demonstrated examples, these discontinuities may not be encountered when using test functions as inputs. In these cases the algorithm might fail to provide a useful configuration, if a smooth solution cannot be obtained. This, however, is not a fault of the algorithm itself, but a consequence of the fixed communication step-size and the blind evaluation of states in Jacobi-scheme co-simulation.

In addition to the modeling aspects, attention must be paid to the internal solvers of the subsystems. Regarding this, an interesting observation about the integrators can be made from Tables 4 and 5, as the integrators chosen by the algorithm were the same in all cases. The hydraulic system is numerically stiff, and therefore a small internal integration step-size is required for the forward Euler formula to provide correct simulation results. In this work, it was assumed that the step-size used for the hydraulics was small enough to guarantee that the integration proceeded in an accurate and stable way.

6 Conclusions

In this paper a method for the automatic configuration of Jacobi-scheme co-simulation setups in which some inputs are provided by external sources has been put forward. The method is based on the relation between energy residuals of a stable simulation and the communication step-size at the co-simulation interface; this relation is here exploited to monitor co-simulation accuracy and stability. During the initialization phase of the co-simulation setup, the proposed method replaces the external inputs with standard test functions. A battery of short tests is run during this initialization stage to determine a combination of co-simulation parameters, such as extrapolation method, subsystem integrators, and communication step-size, that can be used to conduct the actual simulation of the system. The energy residual is employed as an indicator to achieve an efficient and stable set of parameters.

The initialization method was tested in the simulation of two mechanical systems with hydraulic actuation. The results indicate that the approximately linear region in the relation between communication step-size and energy residual is the region in which a stable and accurate solution can be found. Thus, the algorithm put forward in this paper seems to be able to provide a stable initial configuration for co-simulation environments in an automated fashion. The use of standardized test functions as a replacement for unknown external system inputs during the initialization phase enables the selection of a set of co-simulation parameters, such as the communication step-size and extrapolation method, to keep the numerical integration stable in an efficient way. As it is the case with co-simulation in general, carefully constructed subsystem models are required for reliable results to be obtained.

Acknowledgements Open access funding provided by LUT University. The second author acknowledges the support of the Ministry of Economy of Spain through the Ramón y Cajal research program, contract no. RYC-2016-20222. This work has been partially financed by the Spanish Ministry of Economy and Competitiveness (MINECO) and EU-ERDF funds under the project 'Técnicas de co-simulación en tiempo real para bancos de ensayo en automoción' (TRA2017-86488-R).

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Andersson, C.: Methods and tools for co-simulation of dynamic systems with the Functional Mock-up Interface. Ph.D. thesis, Lund University (2016)
2. Antunes, P., Magalhães, H., Ambrósio, J., Pombo, J., Costa, J.: A co-simulation approach to the wheel–rail contact with flexible railway track. Multibody Syst. Dyn. (2018). https://doi.org/10.1007/s11044-018-09646-0
3. Arnold, M., Burgermeister, B., Führer, C., Hippmann, G., Rill, G.: Numerical methods in vehicle system dynamics: state of the art and current developments. Veh. Syst. Dyn. **49**(7), 1159–1207 (2011). https://doi.org/10.1080/00423114.2011.582953
4. Ben Khaled-El Feki, A., Duval, L., Faure, C., Simon, D., Gaid, M.B.: CHOPtrey: contextual online polynomial extrapolation for enhanced multi-core co-simulation of complex systems. Simulation **93**(3) 185–200 (2017). https://doi.org/10.1177/0037549716684026
5. Benedikt, M., Holzinger, F.R.: Automated configuration for non-iterative co-simulation. In: 2016 17th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE), pp. 1–7 (2016). https://doi.org/10.1109/EuroSimE.2016.7463355
6. Benedikt, M., Watzenig, D., Hofer, A.: Modelling and analysis of the non-iterative coupling process for co-simulation. Math. Comput. Model. Dyn. Syst. **19**(5), 451–470 (2013). https://doi.org/10.1080/13873954.2013.784340
7. Benedikt, M., Watzenig, D., Zehetner, J., Hofer, A.: A nearly energy-preserving coupling element for holistic weak-coupled system co-simulations. In: NAFEMS World Congress 2013, Salzburg, Austria (2013)
8. Breedveld, P.: Port-Based Modelling of Multidomain Physical Systems in Terms of Bond Graphs pp. 141–190. Springer, Vienna (2009). https://doi.org/10.1007/978-3-211-89548-1_4
9. Burger, M., Steidel, S.: Local extrapolation and linear-implicit stabilization in a parallel coupling scheme. In: IUTAM Symposium on Solver-Coupling and Co-Simulation, pp. 43–56. Springer, Berlin (2019). https://doi.org/10.1007/978-3-030-14883-6_3
10. Busch, M.: Continuous approximation techniques for co-simulation methods: analysis of numerical stability and local error. J. Appl. Math. Mech./Z. Angew. Math. Mech. **96**(9), 1061–1081 (2016). https://doi.org/10.1002/zamm.201500196
11. Cuadrado, J., Cardenal, J., Bayo, E.: Modeling and simulation methods for efficient real-time simulation of multibody dynamics. Multibody Syst. Dyn. **1**(3), 259–280 (1997). https://doi.org/10.1023/A:1009754006096
12. Cuadrado, J., Cardenal, J., Morer, P., Bayo, E.: Intelligent simulation of multibody dynamics: space–state and descriptor methods in sequential and parallel computing environments. Multibody Syst. Dyn. **4**(1), 55–73 (2000). https://doi.org/10.1023/A:1009824327480
13. Dopico, D., González, F., Cuadrado, J., Kövecses, J.: Determination of holonomic and nonholonomic constraint reactions in an index-3 augmented Lagrangian formulation with velocity and acceleration projections. J. Comput. Nonlinear Dyn. **9**(4), 041006 (2014). https://doi.org/10.1115/1.4027671
14. Gear, C.W., Wells, D.R.: Multirate linear multistep methods. BIT Numer. Math. **24**(4), 484–502 (1984). https://doi.org/10.1007/BF01934907
15. Gomes, C., Thule, C., Broman, D., Larsen, P.G., Vangheluwe, H.: Co-simulation: a survey. ACM Comput. Surv. **51**(3) 49:1–49:33 (2018). https://doi.org/10.1145/3179993
16. González, F., Arbatani, S., Mohtat, A., Kövecses, J.: Energy-leak monitoring and correction to enhance stability in the co-simulation of mechanical systems. Mech. Mach. Theory **131**, 172–188 (2019). https://doi.org/10.1016/j.mechmachtheory.2018.09.007
17. González, F., Dopico, D., Pastorino, R., Cuadrado, J.: Behaviour of augmented Lagrangian and Hamiltonian methods for multibody dynamics in the proximity of singular configurations. Nonlinear Dyn. **85**(3), 1491–1508 (2016). https://doi.org/10.1007/s11071-016-2774-5
18. González, F., Naya, M.A., Luaces, A., González, M.: On the effect of multi-rate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. Multibody Syst. Dyn. **25**(4), 461–483 (2011). https://doi.org/10.1007/s11044-010-9234-7

19. Handroos, H.M., Vilenius, M.J.: Flexible semi-empirical models for hydraulic flow control valves. J. Mech. Transm. Autom. Des. **113**(3), 232–238 (1991). https://doi.org/10.1115/1.2912774
20. Kübler, R., Schiehlen, W.: Modular simulation in multibody system dynamics. Multibody Syst. Dyn. **4**(2), 107–127 (2000). https://doi.org/10.1023/A:1009810318420
21. Meyer, T., Li, P., Lu, D., Schweizer, B.: Implicit co-simulation method for constraint coupling with improved stability behavior. Multibody Syst. Dyn. **44**(2), 135–161 (2018). https://doi.org/10.1007/s11044-018-9632-9
22. Naya, M.A., Cuadrado, J., Dopico, D., Lugris, U.: An efficient unified method for the combined simulation of multibody and hydraulic dynamics: comparison with simplified and co-integration approaches. Arch. Mech. Eng. **58**(2), 223–243 (2011). https://doi.org/10.2478/v10180-011-0016-4
23. Nokka, J., Laurila, L., Pyrhönen, J.: Virtual simulation-based underground loader hybridization study—comparative fuel consumption and productivity analysis. Int. Rev. Model. Simul. (IREMOS) **10**, 222 (2017). https://doi.org/10.15866/iremos.v10i4.12130
24. Paynter, H.: Analysis and Design of Engineering Systems; Class Notes for MIT Course (1961)
25. Peiret, A., González, F., Kövecses, J., Teichmann, M.: Multibody system dynamics interface modelling for stable multirate co-simulation of multiphysics systems. Mech. Mach. Theory **127**, 52–72 (2018). https://doi.org/10.1016/j.mechmachtheory.2018.04.016
26. Rahikainen, J., Kiani, M., Sopanen, J., Jalali, P., Mikkola, A.: Computationally efficient approach for simulation of multibody and hydraulic dynamics. Mech. Mach. Theory **130**, 435–446 (2018). https://doi.org/10.1016/j.mechmachtheory.2018.08.023
27. Rahikainen, J., Mikkola, A., Sopanen, J., Gerstmayr, J.: Combined semi-recursive formulation and lumped fluid method for monolithic simulation of multibody and hydraulic dynamics. Multibody Syst. Dyn. **44**(3), 293–311 (2018). https://doi.org/10.1007/s11044-018-9631-x
28. Sadjina, S., Kyllingstad, L.T., Skjong, S., Pedersen, E.: Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation. Eng. Comput. **33**(3), 607–620 (2017). https://doi.org/10.1007/s00366-016-0492-8
29. Samin, J.C., Brüls, O., Collard, J.F., Sass, L., Fisette, P.: Multiphysics modeling and optimization of mechatronic multibody systems. Multibody Syst. Dyn. **18**(3), 345–373 (2007). https://doi.org/10.1007/s11044-007-9076-0
30. Schierz, T., Arnold, M., Clauss, C.: Co-simulation with communication step size control in an FMI compatible master algorithm. In: Proceedings of the 9th International MODELICA Conference, Munich, Germany (2012). https://doi.org/10.3384/ecp12076205
31. Schweizer, B., Li, P., Lu, D.: Explicit and implicit cosimulation methods: stability and convergence analysis for different solver coupling approaches. J. Comput. Nonlinear Dyn. **10**(5), 051007 (2015). https://doi.org/10.1115/1.4028503
32. Schweizer, B., Li, P., Lu, D.: Implicit co-simulation methods: stability and convergence analysis for solver coupling approaches with algebraic constraints. J. Appl. Math. Mech./Z. Angew. Math. Mech. **96**(8), 986–1012 (2016). https://doi.org/10.1002/zamm.201400087
33. Skjong, S., Pedersen, E.: On the numerical stability in dynamical distributed simulations. Math. Comput. Simul. **163**, 183–203 (2019). https://doi.org/10.1016/j.matcom.2019.02.018
34. Stettinger, G., Benedikt, M., Tranninger, M., Horn, M., Zehetner, J.: Recursive FIR-filter design for fault-tolerant real-time co-simulation. In: 2017 25th Mediterranean Conference on Control and Automation (MED), Valletta, Malta (2017). https://doi.org/10.1109/MED.2017.7984160
35. Vaculín, O., Krüger, W.R., Valášek, M.: Overview of coupling of multibody and control engineering tools. Veh. Syst. Dyn. **41**(5), 415–429 (2004). https://doi.org/10.1080/00423110412331300363
36. Watton, J.: Fluid Power Systems: Modeling, Simulation, Analog and Microcomputer Control. Prentice Hall, New York (1989)

Publication IV

Rahikainen, J., González, F., Naya, M.A., Sopanen, J., Mikkola, A.,
On the co-simulation of multibody systems and hydraulic dynamics

Submitted to
Multibody System Dynamics

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

843. BUZUKU, SHQIPE. Enhancement of decision-making in complex organizations: A systems engineering approach. 2019. Diss.

844. SHCHERBACHEVA, ANNA. Agent-based modelling for epidemiological applications. 2019. Diss.

845. YLIJOKI, OSSI. Big data - towards data-driven business. 2019. Diss.

846. KOISTINEN, KATARIINA. Actors in sustainability transitions. 2019. Diss.

847. GRADOV, DMITRY. Experimentally validated numerical modelling of reacting multiphase flows in stirred tank reactors. 2019. Diss.

848. ALMPANOPOULOU, ARGYRO. Knowledge ecosystem formation: an institutional and organisational perspective. 2019. Diss.

849. AMELI, ALIREZA. Supercritical CO2 numerical modelling and turbomachinery design. 2019. Diss.

850. RENEV, IVAN. Automation of the conceptual design process in construction industry using ideas generation techniques. 2019. Diss.

851. AVRAMENKO, ANNA. CFD-based optimization for wind turbine locations in a wind park. 2019. Diss.

852. RISSANEN, TOMMI. Perspectives on business model experimentation in internationalizing high-tech companies. 2019. Diss.

853. HASSANZADEH, AIDIN. Advanced techniques for unsupervised classification of remote sensing hyperspectral images. 2019. Diss.

854. POPOVIC, TAMARA. Quantitative indicators of social sustainability applicable in process systems engineering. 2019. Diss.

855. RAMASAMY, DEEPIKA. Selective recovery of rare earth elements from diluted aqueous streams using N- and O –coordination ligand grafted organic-inorganic hybrid composites. 2019. Diss.

856. IFTEKHAR, SIDRA. Synthesis of hybrid bio-nanocomposites and their application for the removal of rare earth elements from synthetic wastewater. 2019. Diss.

857. HUIKURI, MARKO. Modelling and disturbance compensation of a permanent magnet linear motor with a discontinuous track 2019. Diss.

858. AALTO, MIKA. Agent-based modeling as part of biomass supply system research. 2019. Diss.

859. IVANOVA, TATYANA. Atomic layer deposition of catalytic materials for environmental protection. 2019. Diss.

860. SOKOLOV, ALEXANDER. Pulsed corona discharge for wastewater treatment and modification of organic materials. 2019. Diss.

861. DOSHI, BHAIRAVI. Towards a sustainable valorisation of spilled oil by establishing a green chemistry between a surface active moiety of chitosan and oils. 2019. Diss.

862. KHADIJEH, NEKOUEIAN. Modification of carbon-based electrodes using metal nanostructures: Application to voltammetric determination of some pharmaceutical and biological compounds. 2019. Diss.

863. HANSKI, JYRI. Supporting strategic asset management in complex and uncertain decision contexts. 2019. Diss.

864. OTRA-AHO, VILLE. A project management office as a project organization's strategizing tool. 2019. Diss.

865. HILTUNEN, SALLA. Hydrothermal stability of microfibrillated cellulose. 2019. Diss.

866. GURUNG, KHUM. Membrane bioreactor for the removal of emerging contaminants from municipal wastewater and its viability of integrating advanced oxidation processes. 2019. Diss.

867. AWAN, USAMA. Inter-firm relationship leading towards social sustainability in export manufacturing firms. 2019. Diss.

868. SAVCHENKO, DMITRII. Testing microservice applications. 2019. Diss.

869. KARHU, MIIKKA. On weldability of thick section austenitic stainless steel using laser processes. 2019. Diss.

870. KUPARINEN, KATJA. Transforming the chemical pulp industry – From an emitter to a source of negative CO2 emissions. 2019. Diss.

871. HUJALA, ELINA. Quantification of large steam bubble oscillations and chugging using image analysis. 2019. Diss.

872. ZHIDCHENKO, VICTOR. Methods for lifecycle support of hydraulically actuated mobile working machines using IoT and digital twin concepts. 2019. Diss.

873. EGOROV, DMITRY. Ferrite permanent magnet hysteresis loss in rotating electrical machinery. 2019. Diss.

874. PALMER, CAROLIN. Psychological aspects of entrepreneurship – How personality and cognitive abilities influence leadership. 2019. Diss.

875. TALÁSEK, TOMÁS. The linguistic approximation of fuzzy models outputs. 2019. Diss.

876. LAHDENPERÄ, ESKO. Mass transfer modeling in slow-release dissolution and in reactive extraction using experimental verification. 2019. Diss.

877. GRÜNENWALD, STEFAN. High power fiber laser welding of thick section materials - Process performance and weld properties. 2019. Diss.

878. NARAYANAN, ARUN. Renewable-energy-based single and community microgrids integrated with electricity markets. 2019. Diss.

879. JAATINEN, PEKKO. Design and control of a permanent magnet bearingless machine. 2019. Diss.

880. HILTUNEN, JANI. Improving the DC-DC power conversion efficiency in a solid oxide fuel cell system. 2019. Diss.

ISBN 978-952-335-446-3
ISBN 978-952-335-447-0 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491
Lappeenranta 2019