Research paper

Multibody system dynamics interface modelling for stable multirate co-simulation of multiphysics systems

Albert Peiret [a,*], Francisco González [a], József Kövecses [a], Marek Teichmann [b]

[a] *Department of Mechanical Engineering and Centre for Intelligent Machines, McGill University, 817 Sherbrooke St. West, Montréal, Québec H3A 0C3, Canada*
[b] *CM Labs Simulations, 645 Wellington, Montréal, Québec H3C 1T2, Canada*

ABSTRACT

Many industrial applications benefit from predictive computer simulation to reduce costs and time, and shorten product development cycle. Computational multibody system dynamics formalisms and software tools have proved to be particularly useful in the simulation of machinery and mechanical systems. Nowadays, however, the complexity of the applications under study often makes it necessary to consider the interaction of mechanical systems with other components of different nature, physical behaviour, and time scale, such as hydraulics or electronics. Co-simulation is an increasingly important approach to formulate and solve the dynamics of these multiphysics setups. In these, modelling techniques and solvers that are tailored to the requirements of each subsystem execute in parallel and are coupled via the exchange of a limited number of inputs and outputs at certain communication times. Co-simulation has clear potential in the modelling of complex engineering systems. On the other hand, there are also challenges. The use of co-simulation may compromise the stability of the numerical solution, especially when non-iterative coupling schemes are used.

In this work, we introduce a modelling technique to improve the dynamic interfacing of mechanical systems in co-simulation setups, based on a reduced representation of multibody systems. This reduced order model is used to obtain a physically meaningful prediction of the evolution of the multibody subsystem dynamics that enables the improvement of the solution of other subsystems. The technique is illustrated in the co-simulation of some examples that include both mechanical and hydraulic components. Results show that dynamic interfaces based on reduced models can be used to improve the stability of non-iterative co-simulation schemes in multiphysics engineering systems, enabling the use of larger communication step-sizes.

© 2018 Published by Elsevier Ltd.

1. Introduction

Computer simulation of mechanical systems has reported significant benefits to industry during the last decades, shortening product development cycles and reducing the costs associated with testing, validation, and re-design of new products. In particular, advances in multibody system (MBS) dynamics research have enabled the use of realistic models of complex,

---

* Corresponding author.
E-mail addresses: albert.peiret@mail.mcgill.ca (A. Peiret), franglez@cim.mcgill.ca (F. González), jozsef.kovecses@mcgill.ca (J. Kövecses), marek@cm-labs.com (M. Teichmann).

large-scale mechanical systems in demanding simulation environments, e.g., those that require real-time execution. Nowadays, various methods exist for the effective simulation and analysis of multibody systems [1,2], and a large array of software tools have been developed to carry out this task. However, there is increasing practical demand to couple multibody simulation to models of subsystems of other physical domains, such as hydraulics and electronics. The individual subsystems in such multiphysics settings can have quite different properties, and may require the use of different solvers and time scales.

The modelling and simulation of multiphysics systems can be approached in several ways. A possibility is to define an all-encompassing set of equations that describes the response of every element in the system [3]. This technique is sometimes referred to as *monolithic simulation*. Another increasingly important option is the coupling of different domain-specific solvers in a *co-simulation* setup. This makes it possible to use modelling and solution techniques tailored to the requirements and time-scale of each subsystem. The numerical integration of the subsystems is coordinated via the exchange of input and output quantities at certain communication points in time. Such an approach makes it possible to use domain-specific tools to model each component; moreover, these can often be executed in parallel, reducing the wall-clock time elapsed in computations. Multirate integration can also be used in this case, adjusting the step-size in each subsystem to its particular time-scale [4]. The simulation of a vast range of multiphysics systems can be addressed this way. To achieve this, a large number of partition methods, communication strategies, and time-stepping approaches can be found in the literature [5,6].

In spite of its advantages, co-simulation poses a series of challenges that need to be addressed to obtain reliable and robust code execution. The co-simulation of dynamic models requires common interface definitions to facilitate the exchange of information between subsystems. This issue has been tackled with the introduction of standards like the Functional Mock-up Interface (FMI) [7]. Another major problem stems from the fact that solvers in a co-simulation environment receive information from other subsystems in discrete time. Inter-solver information exchange occurs only at discrete communication points, at which subsystems are synchronized; the time interval between two communication points is usually referred to as *macro time-step*. This inherently introduces coupling errors and discontinuities in the system dynamics, sometimes making it difficult to guarantee that the integration proceeds in an accurate or stable way [8]. This problem can be alleviated adopting iterative coupling schemes [9]. However, these may become too time-consuming for some applications, such as those that require real-time execution; besides, they cannot be used with certain simulation tools that do not allow for subsystem resetting.

When non-iterative co-simulation is used, additional steps have to be taken to ensure that the obtained results are accurate and the coupled integration process remains stable [10]. Adaptive stabilization strategies, that extract information from subsystem dynamics to improve the communication procedures, are gaining importance nowadays [11]. Possible ways to do this include adjusting the integration step-sizes as a function of the system instantaneous frequency [12,13] or energy level [14], introducing adaptive damping to dissipate the excess energy generated at the interface [15], and interpolating or extrapolating the system inputs to minimize the effect of discrete-time input exchange. The simplest way to handle the inputs that a subsystem receives at the beginning of a macro time-step is to consider that they remain constant until the next communication point. This extrapolation is known as zero-order hold (ZOH). Higher order polynomials [16,17] or smoothing techniques [18,19] can also be used to extrapolate or predict unknown values of the input data. Nevertheless, the above mentioned coupling techniques can be made more accurate if some additional information about the internal dynamics of each subsystem is available at each macro time-step. For example, stabilized coupling approaches were derived in [20–22] based on the availability of the partial derivatives of the subsystem states with respect to the coupling variables at the interface.

In this work, we propose to use a *reduced order model* to characterize the dynamics of a multibody subsystem at the interface that connects it to the rest of components in a multiphysics co-simulation setup. Model order reduction is used in a variety of applications to decrease the size of the system under study while preserving its most representative dynamic behaviour. This is often done for efficiency reasons, e.g., to shorten simulation times when dealing with large finite element models. Reduced order models are also used to this end in the context of multibody system dynamics, usually when dealing with models that account for the structural properties of the system [23,24]. A review and comparison of existing model order reduction methods for structures and large flexible systems can be found in [25,26]. In this paper, we also use a reduced order model but this is conceptually different from the aforementioned models. Our reduced model replaces the original full model during a macro time-step in a co-simulation setup, during which it communicates to another subsystem. In many cases, this other subsystem has different properties, i.e., smaller time scales and faster dynamics. This is often the case of hydraulics elements and microelectronics controllers. Accordingly, their integration requires shorter step-sizes, and more than one integration step takes place within a macro time-step. The reduced model is updated at the beginning of each macro time-step and is subsequently used to obtain information about the evolution of the mechanical subsystem until the next communication point occurs. During this interval, the integration of the reduced model proceeds at a faster rate than that of the original full multibody model, closer to the time scale of the other subsystem in the co-simulation. We will refer to this reduced model as *reduced interface model* (RIM) in this paper.

The RIM reduces the computational complexity of the full model in two ways. First, in most cases it has fewer degrees of freedom than the original model due to the projection of the system dynamics to the interface subspace; moreover, if the original modelling was carried out using dependent coordinates, then the kinematic constraint equations are no longer present in the RIM, which can be expressed as a system of ordinary differential equations. Second, the mass matrix and

force term in the RIM are kept constant between communication points in the co-simulation setup, which decreases the computational cost.

A commonly used approach to generate reduced order models is the extraction of representative motion modes from the dynamic response of the full order system, e.g., [24,27]. Another possibility is directly linearizing the dynamics equations of the multibody system [28]. Here, we follow a different concept. The RIM that represents the original multibody system is obtained from the characterization of the interface between the mechanical subsystem and the other subsystem, which has a different physical nature, e.g., hydraulics. This interface from the point of view of the multibody system can be characterized with the set of interface velocity components that represent the connection to the other subsystem. These interface velocities can define a subspace in the dynamic model, which can be interpreted in a way like the subspace of constrained motion discussed in [29]. The dynamics of the whole multibody system can then be reformulated and decomposed so that the dynamics associated with the interface subspace is explicitly factored out. This gives rise to the concept of the RIM used here. In this paper, we show that a RIM developed this way can be used to improve the stability of non-iterative co-simulation setups that include multibody systems. It also makes it possible to use larger communication step-sizes when it is used as interface between a full multibody model and other subsystems with faster dynamics, e.g., hydraulic components. We illustrate the effectiveness of the method with the simulation of hydraulically actuated mechanical systems.

2. Dynamic interface modelling

Let us assume that a multibody system $\mathcal{M}$ can be described with a set of n_q independent generalized coordinates $\mathbf{q}$ and n independent generalized velocities $\mathbf{v}$, related to the time derivatives of the coordinates, $\dot{\mathbf{q}}$, by transformation

$$\dot{\mathbf{q}} = \mathbf{N}\mathbf{v} \tag{1}$$

where $\mathbf{N} = \mathbf{N}(\mathbf{q})$ is an $n_q \times n$ transformation matrix. The multibody system $\mathcal{M}$ can also interact with a subsystem $\mathcal{S}$, which, at the same time, can be constituted by several elements. In this case, the dynamics equations of the multibody system can be written as

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{c} = \mathbf{f} + \mathbf{f}_i \tag{2}$$

where $\mathbf{M}$ is the $n \times n$ mass matrix of the system, $\mathbf{c}$ is the Coriolis and centrifugal effects term, $\mathbf{f}_i$ contains the interaction forces between subsystem $\mathcal{S}$ and the multibody system through their common coupling interface, and $\mathbf{f}$ represents every other force applied on $\mathcal{M}$.

The interaction forces $\mathbf{f}_i$ affect the dynamics of the multibody system $\mathcal{M}$ in a subspace that can be parametrized by the interface velocities

$$\mathbf{w}_i = \mathbf{A}_i\mathbf{v} \tag{3}$$

where $\mathbf{w}_i$ are the p interface velocity components that parametrize the dynamics of the multibody system $\mathcal{M}$ at the interface and $\mathbf{A}_i$ is a $p \times n$ Jacobian matrix. The generalized forces transmitted at the interface, $\mathbf{f}_i$, can be expressed according to this local parametrization as

$$\mathbf{f}_i = \mathbf{A}_i^\mathrm{T}\boldsymbol{\lambda}_i \tag{4}$$

where $\boldsymbol{\lambda}_i$ is a $p \times 1$ array that contains the forces transmitted at the interface.

A full transformation of the generalized velocities of the system can be performed to decouple the interface dynamics, by introducing an additional set of velocities so that the new system velocity is defined as [30]

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_i \\ \mathbf{w}_a \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i \\ \mathbf{B} \end{bmatrix}\mathbf{v} \tag{5}$$

where $\mathbf{w}_a$ is an $(n - p) \times 1$ array of generalized velocities that complement $\mathbf{w}_i$ to a full set and describe the motion that is compatible with the interface velocities $\mathbf{w}_i$, and $\mathbf{B}$ is the corresponding $(n - p) \times n$ Jacobian matrix. We will term $\mathbf{w}_a$ admissible velocities.

It is always possible to select a set of admissible velocities $\mathbf{w}_a$ that satisfy the condition $\mathbf{A}_i\mathbf{M}^{-1}\mathbf{B}^\mathrm{T} = \mathbf{0}$ to achieve decoupling, and in such a case the dynamics equations (2) can be transformed into [30,31]

$$\begin{bmatrix} \left(\mathbf{A}_i\mathbf{M}^{-1}\mathbf{A}_i^\mathrm{T}\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \left(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^\mathrm{T}\right)^{-1} \end{bmatrix}\begin{bmatrix} \dot{\mathbf{w}}_i \\ \dot{\mathbf{w}}_a \end{bmatrix} + \begin{bmatrix} \mathbf{z}_i \\ \mathbf{z}_a \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_i \\ \boldsymbol{\tau}_a \end{bmatrix} + \begin{bmatrix} \boldsymbol{\lambda}_i \\ \mathbf{0} \end{bmatrix} \tag{6}$$

where

$$\mathbf{z}_i = \left(\mathbf{A}_i\mathbf{M}^{-1}\mathbf{A}_i^\mathrm{T}\right)^{-1}\mathbf{A}_i\mathbf{M}^{-1}\mathbf{c} - \left(\mathbf{A}_i\mathbf{M}^{-1}\mathbf{A}_i^\mathrm{T}\right)^{-1}\dot{\mathbf{A}}_i\mathbf{v} \tag{7}$$

$$\mathbf{z}_a = \left(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^\mathrm{T}\right)^{-1}\mathbf{B}\mathbf{M}^{-1}\mathbf{c} - \left(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^\mathrm{T}\right)^{-1}\dot{\mathbf{B}}\mathbf{v} \tag{8}$$

are the decoupled Coriolis and centrifugal terms, and the generalized force terms can be given by

$$\boldsymbol{\tau}_i = \left(\mathbf{A}_i\mathbf{M}^{-1}\mathbf{A}_i^\mathrm{T}\right)^{-1}\mathbf{A}_i\mathbf{M}^{-1}\mathbf{f} \tag{9}$$
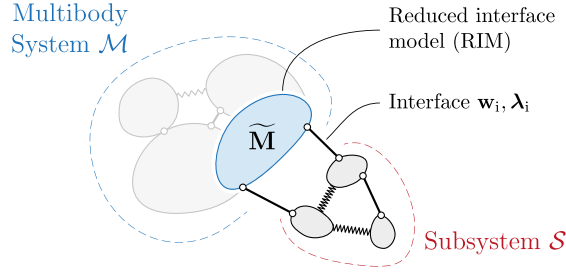
Fig. 1. Interfacing a multibody system $\mathcal{M}$ to a subsystem $\mathcal{S}$ via a RIM.

$$\boldsymbol{\tau}_{\mathrm{a}} = \left(\mathbf{BM}^{-1}\mathbf{B}^{\mathrm{T}}\right)^{-1}\mathbf{BM}^{-1}\mathbf{f} \tag{10}$$

Equation (6) provides a representation of the multibody system in which the dynamics associated with the interaction subspace is explicitly factored out. These interface dynamics equations can be written as

$$\widetilde{\mathbf{M}}\dot{\mathbf{w}}_{\mathrm{i}} = \widetilde{\mathbf{f}} + \boldsymbol{\lambda}_{\mathrm{i}} \tag{11}$$

where

$$\widetilde{\mathbf{M}} = \left(\mathbf{A}_{\mathrm{i}}\mathbf{M}^{-1}\mathbf{A}_{\mathrm{i}}^{\mathrm{T}}\right)^{-1} \tag{12}$$

is the effective mass matrix and

$$\widetilde{\mathbf{f}} = \boldsymbol{\tau}_{\mathrm{i}} - \mathbf{z}_{\mathrm{i}} = \left(\mathbf{A}_{\mathrm{i}}\mathbf{M}^{-1}\mathbf{A}_{\mathrm{i}}^{\mathrm{T}}\right)^{-1}\left(\mathbf{A}_{\mathrm{i}}\mathbf{M}^{-1}(\mathbf{f}-\mathbf{c}) + \dot{\mathbf{A}}_{\mathrm{i}}\mathbf{v}\right) \tag{13}$$

is the effective force term. Equation (11) describes the dynamics of the reduced interface model (RIM) that represents the mechanical system $\mathcal{M}$ in terms of the local parametrization of the interface to the other subsystems. Fig. 1 conceptually illustrates the use of a RIM to represent the dynamics of the multibody system $\mathcal{M}$ in the interaction with $\mathcal{S}$.

It is important to mention that the interface velocities $\mathbf{w}_{\mathrm{i}}$ define the coupling between the RIM and $\mathcal{S}$. Therefore, the number of elements in $\mathcal{S}$ that interact with the multibody system or their complexity are not relevant if all the interface velocities are considered together. On the other hand, the p interface velocities can also be considered independently, so that a single degree of freedom RIM would represent the interface for each interface velocity component. This would result in a series of p decoupled RIMs, as opposed to considering the a single RIM that represents the dynamics of the multibody system that couples the interface velocities, which is reflected by the off-diagonal elements of the effective mass matrix (12).

2.1. Extension to systems modelled with dependent coordinates

We consider now the case in which the multibody system is modelled with n dependent generalized velocities $\mathbf{u}$ related by m kinematic constraints, expressed at the velocity level as

$$\mathbf{w}_{\mathrm{c}} = \mathbf{Du} \tag{14}$$

where $\mathbf{D}$ is the $m \times n$ constraint Jacobian matrix and $\mathbf{w}_{\mathrm{c}}$ represents the m constrained velocities that are given by kinematic constraints, often such as $\mathbf{w}_{\mathrm{c}} = \mathbf{0}$ [31]. In this case, the system dynamics in Eq. (2) must be rewritten to include a term $\mathbf{g}_{\mathrm{c}} = \mathbf{D}^{\mathrm{T}}\boldsymbol{\lambda}$ to account for the constraint forces as

$$\mathbf{H}\dot{\mathbf{u}} + \mathbf{d} = \mathbf{g} + \mathbf{g}_{\mathrm{i}} + \mathbf{g}_{\mathrm{c}} = \mathbf{g} + \mathbf{D}_{\mathrm{i}}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{i}} + \mathbf{D}^{\mathrm{T}}\boldsymbol{\lambda} \tag{15}$$

where $\mathbf{H}$, $\mathbf{d}$, $\mathbf{g}_{\mathrm{i}}$, and $\mathbf{g}$ are the $n \times n$ mass matrix, the Coriolis and centrifugal effects term, the interface forces, and the generalized forces expressed in terms of the set of velocities $\mathbf{u}$, respectively. Term $\boldsymbol{\lambda}$ contains the m reactions in the subspace of constrained motion, and $\mathbf{D}_{\mathrm{i}}$ is the Jacobian matrix that relates the interface and the generalized dependent velocities

$$\mathbf{w}_{\mathrm{i}} = \mathbf{D}_{\mathrm{i}}\mathbf{u} \tag{16}$$

The expressions of the effective mass matrix and force term in this case are given by

$$\widetilde{\mathbf{M}} = \left(\mathbf{D}_{\mathrm{i}}(\mathbf{I}-\mathbf{P}_{\mathrm{c}})\mathbf{H}^{-1}\mathbf{D}_{\mathrm{i}}^{\mathrm{T}}\right)^{-1} \tag{17}$$

$$\widetilde{\mathbf{f}} = \widetilde{\mathbf{M}}\left(\mathbf{D}_{\mathrm{i}}(\mathbf{I}-\mathbf{P}_{\mathrm{c}})\mathbf{H}^{-1}(\mathbf{g}-\mathbf{d}) + \dot{\mathbf{D}}_{\mathrm{i}}\mathbf{u} + \mathbf{D}_{\mathrm{i}}\mathbf{P}_{\mathrm{c}}\dot{\mathbf{u}}\right) \tag{18}$$

Fig. 2. Multibody system ($\mathcal{M}$) and subsystem ($\mathcal{S}$) coupled in a co-simulation setup.
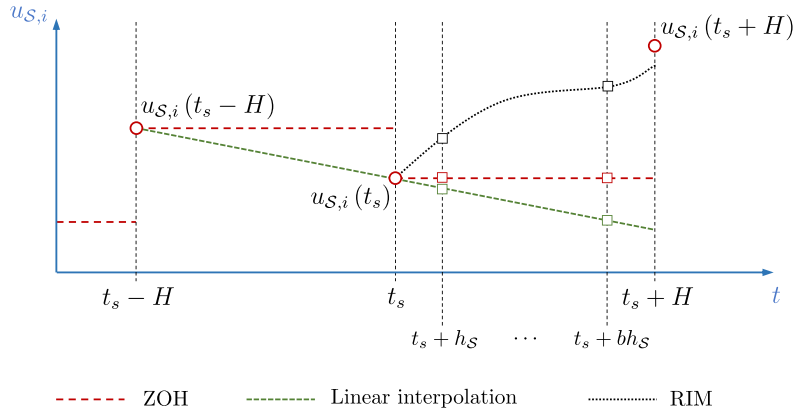

Fig. 3. Illustration of the effect of different extrapolation methods on an input variable $u_{\mathcal{S},i}$ of subsystem $\mathcal{S}$.

where **I** is the $n \times n$ identity matrix, and $\mathbf{P}_c$ is the projector matrix onto the subspace of constrained motion [31], which can be given as

$$\mathbf{P}_c = \mathbf{H}^{-1}\mathbf{D}^{\mathrm{T}}\left(\mathbf{D}\mathbf{H}^{-1}\mathbf{D}^{\mathrm{T}}\right)^{-1}\mathbf{D} \tag{19}$$

This form of the projector matrix $\mathbf{P}_c$ requires the Jacobian of the constraints **D** to have full row rank, i.e., that all the kinematic constraints imposed on the mechanical system are linearly independent. However, the projector matrix can also be determined when this is not the case, making use of the Moore-Penrose generalized inverse for instance [29,31]. The details of the derivation of Eqs. (17) and (18) can be found in Appendix A.

3. Reduced interface model in co-simulation setups

We consider now the case in which the multibody system $\mathcal{M}$ is interfaced in a non-iterative co-simulation scheme to another system $\mathcal{S}$, which can be of a non-mechanical nature (see Fig. 2).

The dynamics of the multibody subsystem are given by Eq. (2) or (15); these equations can be integrated using any of the methods available in the literature with an integration step-size $h_{\mathcal{M}}$. The subsystem $\mathcal{S}$, on the other hand, has its own dynamic formulation described by a function **h** of the system state **x**, integrated with a different method and step-size $h_{\mathcal{S}}$. In this case, it is reasonable to assume that $h_{\mathcal{S}} < h_{\mathcal{M}}$. The communication between the two subsystems takes place via a co-simulation interface, through which information in the form of inputs **u** and outputs **y** is exchanged at the beginning of each macro time-step, of size H. The subsystems do not receive any information from each other until the following communication point. For this reason, the input values $\mathbf{u}_{\mathcal{S}}$ may need to be extrapolated in some way within the macro time-step. The simplest way to do this is assuming that they remain constant; this approach is known as *zero order hold* (ZOH). Other extrapolation methods, e.g., polynomial-based ones, can be used as well.

Fig. 3 illustrates the effect of the chosen extrapolation on the evaluation of subsystem inputs. At communication time t_s, the multibody system provides the output $\mathbf{y}_{\mathcal{M}}$ to the co-simulation manager, which is transferred as an input $\mathbf{u}_{\mathcal{S}}$ to the subsystem $\mathcal{S}$. This value will not be updated until the macro time-step is completed, at time $t = t_s + H$. If a ZOH is used, the value of $\mathbf{u}_{\mathcal{S}}(t) = \mathbf{y}_{\mathcal{M}}(t_s)$ is considered constant in this interval. A linear extrapolation approach will evaluate $\mathbf{u}_{\mathcal{S}}$ as a linear function of time, based on the last two available values of the input. With these methods, however, the values of $\mathbf{u}_{\mathcal{S}}$ required to integrate $\mathcal{S}$ at $t = t_s + kh_{\mathcal{S}}$, where $k = 1, 2, \ldots, b$, are obtained without taking into consideration the dynamics of the subsystem $\mathcal{M}$. As a consequence, the use of one or another extrapolation method may or may not be beneficial for the accuracy of the simulation, depending on the proximity of the predicted values to the actual system behaviour. A way to deal with this issue is attempting to select the most suitable extrapolation strategy based on the observed evolution of the input values available so far [32]. We propose here an alternative possibility based on the RIM described in Section 2. This model can be used to provide a dynamics-based prediction of these inputs until the next communication point. The values of $\mathbf{u}_{\mathcal{S}}$ within the macro time-step are obtained based on the integration of a RIM of the multibody system. This is expected to result in more accurate values of the inputs at the intermediate integration steps $t = t_s + kh_{\mathcal{S}}$ and a reduction of the deviations between the predicted and the actually received inputs at the next communication point, $t_s + H$.

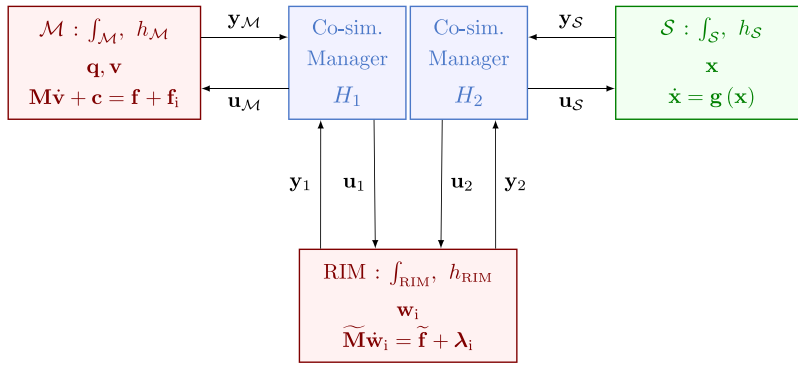**Fig. 4.** Multibody system $\mathcal{M}$ and subsystem $\mathcal{S}$ coupled in a co-simulation setup via a RIM of $\mathcal{M}$.
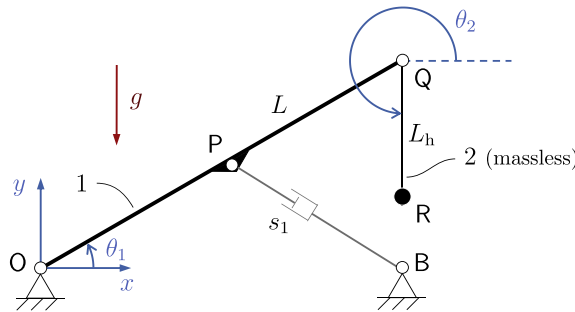

Fig. 5. Single-actuated planar model of a hydraulic crane.

The introduction of a RIM of subsystem $\mathcal{M}$ in the co-simulation leads to the scheme shown in Fig. 4, a multirate integration algorithm in which two communication step-sizes, H_1 and H_2, are employed. With this approach, subsystem $\mathcal{S}$ exchanges information through the co-simulation manager only with the RIM. Both the RIM and $\mathcal{S}$ are integrated simultaneously and synchronized every H_2 seconds. The full multibody system model, on the other hand, is integrated at a slower rate and synchronized with the RIM with a step-size $H_1 > H_2$. It must be noted that the output $\mathbf{y}_{\mathcal{M}}$, and subsequently the inputs of the RIM $\mathbf{u}_1$, must contain all the information necessary to evaluate the effective mass matrix $\widetilde{\mathbf{M}}$ and the force vector $\widetilde{\mathbf{f}}$ required by the RIM dynamics.

4. Examples

The ability of the RIM-based scheme described in Section 3 to improve the stability of multiphysics co-simulation will be demonstrated through the study of test problems composed of both mechanical and hydraulics components.

4.1. Planar model with one hydraulic actuator

The first example used here is a planar model of a crane that features a hydraulic actuator, a multibody system with two degrees of freedom shown in Fig. 5. A similar model was employed by Naya et al. [33] to compare the efficiency and accuracy of monolithic and solver coupling methods in the simulation of mechanical systems interfaced to hydraulic actuators.

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 configuration of the system can be described by the two angles θ_1 and θ_2 as independent generalized coordinates, $\mathbf{q} = [\theta_1, \theta_2]^T$. The transformation matrix in Eq. (1) can be considered as the identity matrix and $\mathbf{v} = \dot{\mathbf{q}}$. The dynamics equations in independent coordinates can then be written as

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{c} = \mathbf{f} + \mathbf{f}_h \tag{20}$$

where $\mathbf{f}_h$ contains the generalized forces exerted by the hydraulic actuator on the multibody system.

Table 1
Mechanical parameters of the single-actuated model.

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_B, y_B)	$\left(\sqrt{3}/2, 0\right)$ 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
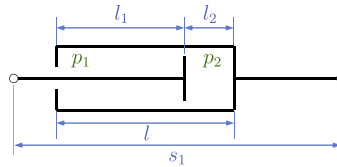
Fig. 6. Schematic of the hydraulic actuator.

The system can also be described by a set of dependent generalized coordinates $\mathbf{x} = [x_P, y_P, \theta_1, x_R, y_R]^T$, related by three kinematic constraint equations $\mathbf{\Phi}(\mathbf{x}) = \mathbf{0}$. The dynamics equations can be written as

$$\mathbf{H}\dot{\mathbf{u}} + \mathbf{d} = \mathbf{g} + \mathbf{g}_h + \mathbf{D}^T\boldsymbol{\lambda} \tag{21a}$$

$$\mathbf{\Phi} = \begin{bmatrix} L\cos\theta_1 - 2x_P \\ L\sin\theta_1 - 2y_P \\ (x_R - L\cos\theta_1)^2 + (y_R - L\sin\theta_1)^2 - L_h^2 \end{bmatrix} = \mathbf{0} \tag{21b}$$

where $\mathbf{u} = \dot{\mathbf{x}}$, $\mathbf{g}_h$ contains the hydraulic force, and $\mathbf{D} = \frac{\partial\mathbf{\Phi}}{\partial\mathbf{x}}$ is the 3×5 Jacobian matrix of the kinematic constraints $\mathbf{Du} = \mathbf{0}$.

The multibody-hydraulics interface in this example can be characterized with only one velocity, namely the rate $w_i = \dot{s}_1$. This velocity component is related to the generalized velocities as

$$\dot{s}_1 = \begin{bmatrix} \dfrac{L(x_B\sin\theta_1 - y_B\cos\theta_1)}{2s_1} & 0 \end{bmatrix}\mathbf{v} = \mathbf{A}_i\mathbf{v} \tag{22}$$

$$\dot{s}_1 = \begin{bmatrix} \dfrac{x_P - x_B}{s_1} & \dfrac{y_P - y_B}{s_1} & 0 & 0 & 0 \end{bmatrix}\dot{\mathbf{x}} = \mathbf{D}_i\dot{\mathbf{x}} \tag{23}$$

Jacobian matrices $\mathbf{A}_i$ and $\mathbf{D}_i$ are used to evaluate the effective mass and force terms in Eqs. (12)–(13) and (17)–(18), respectively.

The magnitude of the hydraulic force exerted by the actuator can be evaluated as

$$f_h = (p_2 - p_1)a_p - c\dot{s}_1 \tag{24}$$

where p_1 and p_2 are the fluid pressures within the cylinder and a_p is the total piston area. A viscous friction model with coefficient c was used to represent internal dissipation in the actuator, a simplified representation of which is shown in Fig. 6.

The dynamics of the hydraulic system can be described with the following set of first order, ordinary differential equations [33]

$$\dot{p}_1 = \frac{\beta_1}{a_p l_1}\left[a_p\dot{s}_1 + a_i c_d\sqrt{\frac{2(p_P - p_1)}{\rho}}\delta_{P1} - a_o c_d\sqrt{\frac{2(p_1 - p_T)}{\rho}}\delta_{T1}\right] \tag{25}$$

$$\dot{p}_2 = \frac{\beta_2}{a_p l_2}\left[-a_p\dot{s}_1 + a_o c_d\sqrt{\frac{2(p_P - p_2)}{\rho}}\delta_{P2} - a_i c_d\sqrt{\frac{2(p_2 - p_T)}{\rho}}\delta_{T2}\right] \tag{26}$$

where l_1 and l_2 are the variable lengths of the chambers on each side of the piston, a_i and a_o are the variable valve areas that connect these cylinder chambers to the pump and the tank in the hydraulics system, c_d is the discharge coefficient of the valves, ρ stands for the fluid density, p_P and p_T are the hydraulic pressure at the pump and the tank respectively. Coefficients δ_{P1}, δ_{P2}, δ_{T1}, and δ_{T2} are 0 when the quantity inside the square root that precedes them is negative and 1

Table 2
Hydraulic parameters.

Piston area	a_p	$65 \cdot 10^{-4}$ m^2
Cylinder length	l	0.442 m
Friction coefficient	c	10^5 Ns/m
Valve discharge coefficient	c_d	0.67
Fluid density	ρ	850 kg/m^3
Hydraulic pressure at the pump	p_P	7.6 MPa
Hydraulic pressure at the tank	p_T	0.1 MPa
Compressibility coefficient	a	$6.53 \cdot 10^{-10}$ Pa
Compressibility coefficient	b	$-1.19 \cdot 10^{-18}$
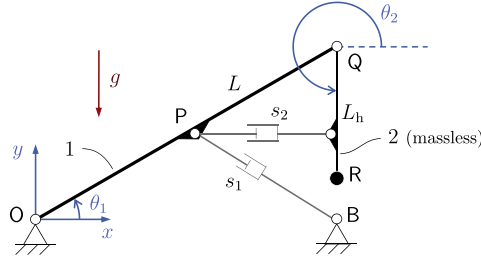
Fig. 7. Double-actuated planar model of a hydraulic crane.

otherwise. Terms β_1 and β_2 stand for the bulk modulus in each cylinder chamber, and they are evaluated as a function of the fluid pressure [34]

$$\beta_i = \frac{1 + ap_i + bp_i^2}{a + 2bp_i}, \quad i = 1, 2 \tag{27}$$

where a and b are constants for the fluid. Assuming that the two cylinder chambers have equal volume at the starting time of the simulation, chamber lengths l_1 and l_2 are given by

$$\begin{aligned} l_1 &= 0.5l + s_{1,0} - s_1 \\ l_2 &= 0.5l + s_1 - s_{1,0} \end{aligned} \tag{28}$$

where $s_{1,0}$ is the initial length of the actuator. Valve areas a_i and a_o have m^2 units and are obtained as

$$\begin{aligned} a_i &= 5 \cdot 10^{-4}\kappa \\ a_o &= 5 \cdot 10^{-4}(1 - \kappa) \end{aligned} \tag{29}$$

In Eq. (29), $\kappa \in [0, 1]$ is the valve control parameter or spool displacement, i.e., the kinematic input that controls the motion of the piston. The hydraulic subsystem parameters for this problem are shown in Table 2.

The manoeuvre simulated with this mechanical-hydraulic system consisted in a 10-s motion during which the value of κ was kinematically guided and given as a function of time

$$\kappa = \begin{cases} \kappa_0, & t \leq t_a \\ \kappa_0 - 0.01(t - t_a)/t_r, & t_a < t \leq t_a + t_r \\ \kappa_0 - 0.01, & t_a + t_r < t \leq t_b \\ \kappa_0 - 0.01 + 0.03(t - t_b)/(2t_r), & t_b < t \leq t_b + 2t_r \\ \kappa_0 + 0.02, & t_b + 2t_r < t \end{cases} \tag{30}$$

where $t_a = 2$ s and $t_b = 6$ s, κ_0 is the initial valve displacement, and t_r is a time constant that controls the rate of change of κ during transitions; its value was adjusted to $t_r = 1$ ms in this study. If κ_0 is adjusted to keep the system in static equilibrium, such a control law will result in a stationary state for $t \leq 2$ s, the extension of the actuator for 2 s $< t \leq 6$ s, and then its contraction until $t = 10$ s.

4.1.1. Extension to systems with several actuators

The use of the RIM to describe the multibody dynamics at the interface with the hydraulics is also straightforward in the case of systems with several actuators. Fig. 7 shows the model described in Section 4.1 with a second hydraulic actuator added between point P and the middle point of link 2; the properties of this extra actuator are the same as those of the first one.

The consideration of the second actuator requires the redefinition of the interface velocities $\mathbf{w}_i$, which can now be formulated as $\mathbf{w}_i = [\dot{s}_1, \dot{s}_2]^T$, where s_2 is the time-varying length of the second actuator. Considering again modelling using both independent and dependent coordinates, analogous to Eqs. (22) and (23), we can express the interface velocities as

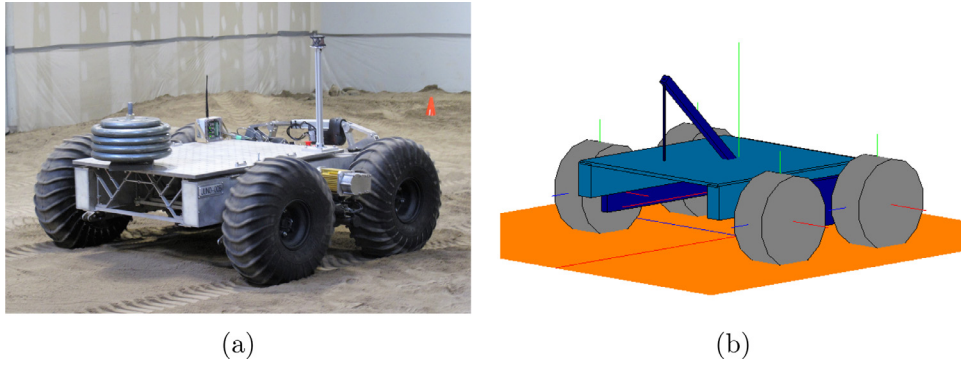$$\mathbf{w}_i = \mathbf{A}_i \mathbf{v} \tag{31}$$

Fig. 8. (a) Juno rover and (b) its multibody model featuring a two-link manipulator.

or

$$\mathbf{w}_i = \mathbf{D}_i \dot{\mathbf{x}} \tag{32}$$

where

$$\mathbf{A}_i = \begin{bmatrix} \dfrac{L(x_B \sin\theta_1 - y_B \cos\theta_1)}{2s_1} & 0 \\ \dfrac{-LL_h \sin\theta_1 - \theta_2}{4s_2} & \dfrac{LL_h \sin\theta_1 - \theta_2}{4s_2} \end{bmatrix} \tag{33}$$

$$\mathbf{D}_i = \begin{bmatrix} \dfrac{x_P - x_B}{s_1} & \dfrac{y_P - y_B}{s_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{x_R}{4s_2} & \dfrac{y_R}{4s_2} \end{bmatrix} \tag{34}$$

A 10-second manoeuvre was simulated with this example as well. The first cylinder received the same actuation law described in Eq. (30), while the spool displacement for the second actuator was given by

$$\kappa_2 = \begin{cases} \kappa_{2,0} - 0.0025t, & t \le t_a \\ \kappa_{2,0} - 0.005, & t > t_a \end{cases} \tag{35}$$

where $\kappa_{2,0}$ is the initial valve displacement that keeps the system in static equilibrium.

4.2. Off-road vehicle with robotic manipulator

The second test problem used was a modified version of the Juno rover [35], a wheeled robot prototype for off-road exploration, developed by Neptec, shown in Fig. 8. Juno is a 300 kg vehicle, designed to serve as a mobile platform for scientific equipment in unstructured environments. Its four wheels are connected to two side bogies, which in turn are linked to the main vehicle body via revolute joints. A differential mechanism ensures that the pitch angle of the chassis is equal to the average of those of the two side bogies. The two wheels on each side of the rover are powered by the same electrical motor; the turning motion of the vehicle is obtained through skid-steering.

In this work, a two-link manipulator with two hydraulic actuators was added to the model of Juno. The links, of length $L_{c1} = 0.9$ m and $L_{c2} = 0.5$ m and uniformly distributed mass $m_{c1} = 15$ kg and $m_{c2} = 25$ kg, were connected to each other and to the rover chassis via revolute joints. The hydraulic actuators had the same properties as those described in Section 4.1, except for the cylinder length, which was decreased to $l = 0.3$ m. A scheme of the mechanism is shown in Fig. 9; in this figure, $e = 0.1$ m. The resultant mechanical system was described using seven coordinates per body at the configuration level, namely the global x, y, and z coordinates of its centre of mass (c.o.m.) plus four Euler–Rodrigues parameters to represent its rotation. At the velocity level, the x, y, and z global components of the centre of mass velocity and the angular velocity of each body were used. The system has 13 degrees of freedom; accordingly, the dynamics equations consisted in a system of 54 ordinary differential equations plus 41 algebraic equations to enforce the kinematic constraints. A detailed description of the system configuration and parameters can be found in Appendix B.

The mechanical and hydraulic subsystems in the resulting model were coupled using direct ZOH and RIM-based co-simulation. A 2.5-second manoeuvre, consisting in operating the manipulator during a point turn and the subsequent straight motion of the vehicle was simulated to assess the behaviour of both coupling approaches. Between $t = 0.1$ s and $t = 0.8$ s a point-turning motion was achieved by applying a torque $\tau_p = 75$ Nm to the wheels on the port side of Juno, and $\tau_s = -75$ Nm to the wheels on the starboard side. For $t > 1.6$ s, all wheels received a positive torque $\tau = 75$ Nm, which
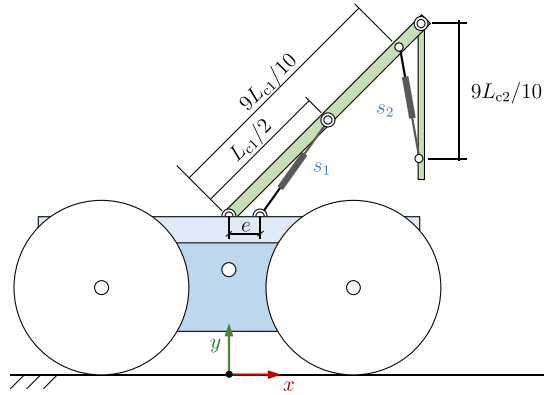
Fig. 9. Schematic of the two-link manipulator mounted on the Juno rover.

resulted in a straight forward motion of the vehicle. The interaction between the wheels and the terrain was modelled using the terramechanics relations described in [36] and summarized in Appendix B.

The manipulator was actuated by regulating the valve control parameters of its actuators, κ_{c1} and κ_{c2}, according to the following laws:

$$\kappa_{c1} = \begin{cases} \kappa_{01}, & t \leq t_{a,c} \\ \kappa_{01} + 0.005(t - t_{a,c})/(t_{b,c} - t_{a,c}), & t_{a,c} \leq t \leq t_{b,c} \\ \kappa_{01} + 0.005, & t \geq t_{b,c} \end{cases} \tag{36}$$

$$\kappa_{c2} = \begin{cases} \kappa_{02} - 0.01t/t_{a,c}, & t \leq t_{a,c} \\ \kappa_{02} - 0.01, & t \geq t_{a,c} \end{cases} \tag{37}$$

where κ_{01} and κ_{02} are the initial values of the valve parameters, $t_{a,c} = 1$ s, and $t_{b,c} = 1.6$ s.

4.3. Simulation methodology

The examples introduced in Sections 4.1 and 4.2 were simulated using two non-iterative co-simulation schemes. The first one was a ZOH direct co-simulation approach like the one in Fig. 2. Then, a RIM-based co-simulation scheme was tested, according to the description in Fig. 4. In both cases, the output of the multibody system, $\mathbf{y}_\mathcal{M}$ included the actuators displacements $\mathbf{s} = [s_1, s_2]^T$ and velocities $\dot{\mathbf{s}} = [\dot{s}_1, \dot{s}_2]^T = \dot{\mathbf{w}}_i$. The hydraulics outputs were the actuator forces, $f_{h,1}$ and $f_{h,2}$. In RIM-based coupling, $\mathbf{y}_\mathcal{M}$ included terms $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{f}}$ as well.

In direct co-simulation, the macro step-size H was made equal to the integration step-size of the multibody system, $H = h_\mathcal{M}$, in order to reduce the number of simulation parameters. When a RIM was used, the step-sizes were adjusted to $H_1 = h_\mathcal{M}$ and $H_2 = h_\mathcal{S} = h_{RIM}$. Moreover, $h_\mathcal{M}$ was adjusted to be an exact multiple of the hydraulics step-size $h_\mathcal{S}$. All the subsystems were integrated using a symplectic, semi-implicit Euler formula

$$\mathbf{q}^{k+1} = \mathbf{q}^k + h_\mathcal{M}\mathbf{N}(\mathbf{q}^k)\mathbf{v}^{k+1}; \qquad \mathbf{v}^{k+1} = \mathbf{v}^k + h_\mathcal{M}\dot{\mathbf{v}}^k \tag{38}$$

$$\mathbf{s}^{k+1} = \mathbf{s}^k + h_{RIM}\dot{\mathbf{s}}^{k+1}; \qquad \dot{\mathbf{s}}^{k+1} = \dot{\mathbf{s}}^k + h_{RIM}\ddot{\mathbf{s}}^k \tag{39}$$

$$\mathbf{p}^{k+1} = \mathbf{p}^k + h_\mathcal{S}\dot{\mathbf{p}}^k \tag{40}$$

where superscript k stands for the values at time t_k and $\mathbf{p}$ contains the hydraulic pressures, which define the internal state of the hydraulic subsystem $\mathcal{S}$.

Regardless of the selected co-simulation scheme, an initialization stage is always necessary to ensure the consistency of the internal states in the subsystems. In the examples described in Section 4, the initial positions and velocities of the elements in the mechanical system were known, but this was not the case for the hydraulic pressures, which needed to be determined before proceeding with the numerical integration. Their values where evaluated solving the following system of nonlinear equations in an iterative Newton–Raphson fashion

$$\begin{bmatrix} f_{h,1} - f_{s,1}^{eq} \\ f_{h,2} - f_{s,2}^{eq} \\ \dot{\mathbf{p}} \end{bmatrix} = \mathbf{0} \tag{41}$$

where $f_{s,1}^{\text{eq}}$ and $f_{s,2}^{\text{eq}}$ are the forces along s_1 and s_2 required for the initial static equilibrium of the mechanical system.

The direct ZOH co-simulation process is summarized in Algorithm 1, where $t_\mathcal{M}$ and $t_\mathcal{S}$ stand for the internal timestamps

---

Algorithm 1 Direct ZOH co-simulation.

1: **function** INITIALIZE
2:   Establish initial configuration in multibody system
3:   Evaluate required equilibrium forces $f_{s,1}^{\text{eq}}$, $f_{s,2}^{\text{eq}}$
4:   Evaluate equilibrium configuration in hydraulic subsystem, Eq. (41)
5:   Verify initial equilibrium conditions, $\dot{\mathbf{p}} = \mathbf{0}$, $\dot{\mathbf{v}} = \mathbf{0}$
6:   $t = 0$, $t_\mathcal{M} = 0$, $t_\mathcal{S} = 0$
7: **end**
8: **function** SIMULATION LOOP
9: *while* $t < t_{\text{end}}$:
10:   Synchronize systems, receive $\mathbf{y}_\mathcal{M}$, $\mathbf{y}_\mathcal{S}$, send $\mathbf{u}_\mathcal{M}$, $\mathbf{u}_\mathcal{S}$
11:   Integrate subsystems until $t_\mathcal{M} \geq t$ and $t_\mathcal{S} \geq t$
12:   $t = t + H$
13: *end*
14: **end**

---

of the multibody and the hydraulics subsystems, respectively.

The RIM co-simulation approach is described by Algorithm 2. Here, we distinguish two internal times for each

---

Algorithm 2 RIM-based co-simulation.

1: **function** INITIALIZE
2:   Establish initial configuration in multibody system
3:   Evaluate required equilibrium forces $f_{s,1}^{\text{eq}}$, $f_{s,2}^{\text{eq}}$
4:   Initialize RIM, $\mathbf{s}_0$
5:   Evaluate equilibrium configuration in hydraulic subsystem, Eq. (41)
6:   Verify initial equilibrium conditions, $\dot{\mathbf{p}} = \mathbf{0}$, $\dot{\mathbf{v}} = \mathbf{0}$, $\ddot{\mathbf{s}} = \mathbf{0}$
7:   $t_1 = 0$, $t_2 = 0$, $t_\mathcal{S} = 0$, $t_\mathcal{M} = 0$
8:   Synchronize all subsystems
9: **end**
10: **function** SIMULATION LOOP
11: *while* $t_1 < t_{\text{end}}$:
12:   Synchronize $\mathcal{M}$-RIM, receive $\mathbf{y}_\mathcal{M}$, $\mathbf{y}_1$, send $\mathbf{u}_\mathcal{M}$, $\mathbf{u}_1$
13:   Parallel execution:
14:     - Integrate $\mathcal{M}$ until $t_\mathcal{M} \geq t_1$
15:     - **function** NESTED SIMULATION LOOP
16:       *while* $t_2 < t_1$:
17:         Synchronize Hydraulics-RIM, receive $\mathbf{y}_\mathcal{S}$, $\mathbf{y}_2$, send $\mathbf{u}_\mathcal{S}$, $\mathbf{u}_2$
18:         Integrate Hydraulics and RIM until $t_\mathcal{S} \geq t_1$ and $t_{\text{RIM}} \geq t_1$
19:         $t_2 = t_2 + H_2$
20:       *end*
21:     **end**
22:   $t_1 = t_1 + H_1$
23: *end*
24: **end**

---

co-simulation manager: t_1 for the interface between RIM and the multibody model, and t_2 for the interface between RIM and hydraulics.

The solution of the dynamics equations of the mechanical subsystem required different approaches depending on the nature of the simulated problems. The system dynamics expressed with independent generalized coordinates, Eq. (20) could be directly solved to obtain the accelerations of the angle variables, $\dot{\mathbf{v}}$. However, if dependent coordinates were used, then a system of differential algebraic equations (DAEs) had to be solved, e.g., the one defined by Eqs. (21a) and (21b). In this case, an index-1 augmented Lagrangian algorithm [37] was used to solve the dynamics and obtain the system accelerations and constraint reactions. For the Juno rover in Section 4.2, configuration- and velocity-level projections were used to enhance the satisfaction of the kinematic constraints.

In the case of the single-actuated model, the co-simulation results obtained with Algorithms 1 and 2 were compared to those delivered by a monolithic formulation that solved together the dynamics of the multibody system and the hydraulics

Table 3
Initial static equilibrium values for the hydraulic actuators.

Actuator	κ	p_1 (MPa)	p_2 (MPa)
1	0.45435	3.17	4.53
2	0.5	3.85	3.85

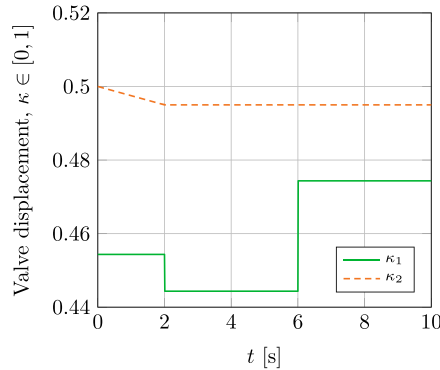**Fig. 10.** Spool displacements of the hydraulic actuators.

[33]. The monolithic method formulates the system dynamics equations as

$$\mathbf{H}\dot{\mathbf{u}} + \mathbf{d} = \mathbf{g} + \mathbf{g}_h + \mathbf{D}^T\boldsymbol{\lambda}$$
$$\dot{\mathbf{p}} = \mathbf{h}(\mathbf{p}, \mathbf{x}, \mathbf{u}) \tag{42}$$

where $\mathbf{h}$ stands for the expressions in Eqs. (25) and (26). Introducing the trapezoidal rule integration formula into Eq. (42), a dynamic equilibrium is established at time step $k + 1$. This equilibrium can be expressed as a system of nonlinear equations and solved to obtain the system coordinates $\mathbf{x}$ and hydraulic pressures $\mathbf{p}$ at time step $k + 1$ by means of Newton–Raphson iteration. This approach has shown good accuracy and efficiency properties and was used as reference to verify the correctness of the results delivered by the co-simulation methods.

5. Results

The results obtained in the simulation of the examples in Section 4 are summarized next. All numerical experiments were performed on an Intel Core i7-3770 computer at 3.40 GHz with 8.00 GB RAM, running Windows 10 Enterprise. The system models were implemented and executed in MATLAB R2017a.

5.1. Hydraulic manipulator model

The determination of the initial equilibrium configuration for the model resulted in the values of the spool displacements and hydraulic pressures shown in Table 3. With these values, the time-history of the spool displacements κ_1 and κ_2 are shown in Fig. 10.

5.1.1. Single-actuated model

The motion of the single-actuated model in Section 4.1 was simulated using the direct ZOH co-simulation and the RIM-based co-simulation scheme. Results were compared to those obtained with the monolithic formulation mentioned in Section 4.3 integrated with a step-size $h = 0.5$ ms, which was used as reference. The step-sizes for both co-simulation schemes were set to $h_{\mathcal{M}} = 10$ ms and $h_{\mathcal{S}} = 0.2$ ms. In all cases there were no significant differences between the co-simulation results obtained with the sets of independent and dependent coordinates.

Fig. 11 shows the actuator length s_1 during the manoeuvre with the single-actuated model. The three simulation methods, monolithic, direct ZOH co-simulation, and RIM-based co-simulation delivered very similar results. Fig. 11b highlights that the differences between the predicted values of s_1 by each approach remained below 2 mm. Moreover, it also shows that direct co-simulation leads to an oscillatory behaviour of the solution after the change in the input variable κ_1 at $t = 6$ s. This oscillation is also present, although less noticeable, in the response of the RIM co-simulation.

Fig. 12 shows the horizontal component of the position and velocity vectors of point R during motion, x_R and $\dot{x}_R$, respectively. These confirm that the differences between the results obtained with the different methods did not affect significantly the resulting motion of the model. In particular, the difference in the predicted x-coordinate of point R between direct ZOH
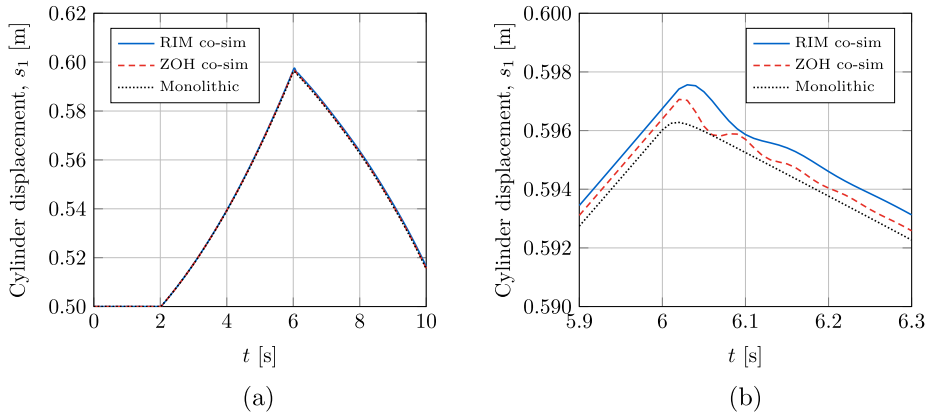
Fig. 11. (a) Actuator length s_1 during motion of the single-actuated model, evaluated with the monolithic approach, RIM co-simulation, and ZOH co-simulation; (b) detail view during the maximum displacement interval of the manoeuvre; with time-step sizes $h_{\mathcal{M}} = 10$ ms, $h_{\mathcal{S}} = 0.2$ ms, and $h = 0.5$ ms for the monolithic.
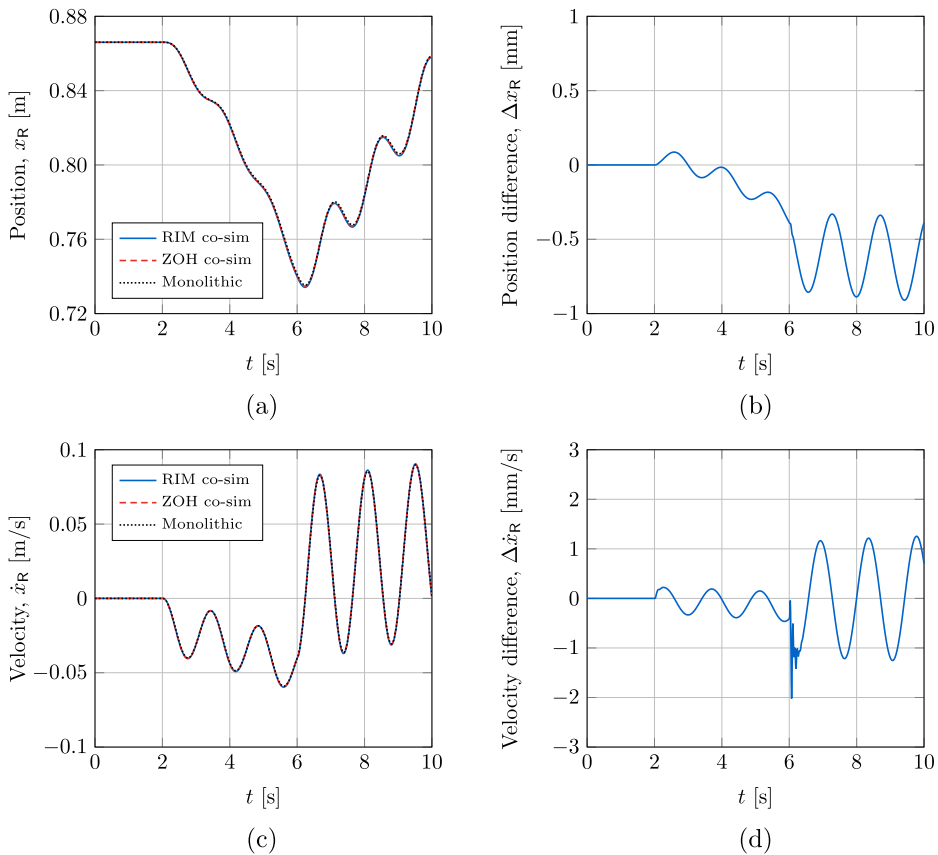
Fig. 12. (a) x-coordinate and (c) x-component of the velocity of point R during the manoeuvre with the single-actuated model; and difference of (b) x_R and (d) $\dot{x}_R$ obtained with direct ZOH and RIM-based co-simulation; with time-step sizes $h_{\mathcal{M}} = 10$ ms, $h_{\mathcal{S}} = 0.2$ ms, and $h = 0.5$ ms for the monolithic.

and RIM-based co-simulations remained smaller than 1 mm, i.e., less than 0.8% of the total motion amplitude. Regarding the velocity $\dot{x}_R$, the difference between the two co-simulation approaches (ZOH and RIM) was always below 2 mm/s.

Even though all the methods resulted in similar responses in the motion of the system, significant differences can be noticed between direct ZOH and RIM-based co-simulations regarding hydraulic pressures and actuator forces, especially around $t = 2$ s and $t = 6$ s, the time instants at which input κ changes abruptly. Fig. 13 shows the pressure p_1 in the actuator, and the actuator force f_h; significant differences between the three simulation methods can be noticed during these transients. While the maximum value reached after the change in κ_1 was similar for the three methods, the pressure p_1 obtained with
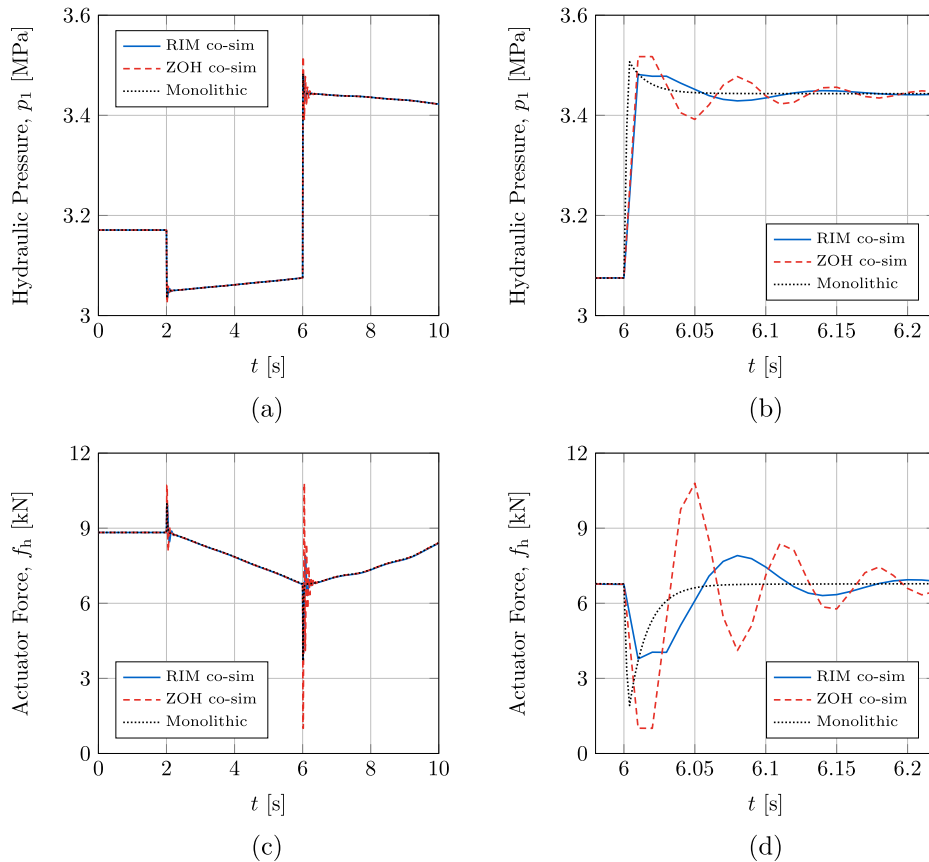
Fig. 13. (a) pressure p_1 and (c) actuator force f_h during the manoeuvre with the single-actuated model; and detail view of (b) the pressure and (d) the force at $t = 6$ s; with time-step sizes $h_\mathcal{M} = 10$ ms, $h_\mathcal{S} = 0.2$ ms, and $h = 0.5$ ms for the monolithic.

direct co-simulation was the highest; more importantly, an oscillatory variation of this pressure can be observed in Fig. 13b. The RIM-based co-simulation resulted in an oscillatory response as well, although more damped. In addition, the sudden change in κ_1 gave rise to an impact-like variation of the actuation. The monolithic approach, in which multibody dynamics and hydraulic pressures are strongly coupled, quickly dampened the peak force, keeping the simulation stable. Co-simulation methods, which do not feature such a strong coupling between variables, resulted in oscillatory actuator forces, that were eventually damped at the end of a transient period. Using a RIM in this case decreased the severity of these oscillations, by providing a more physically meaningful way to evaluate the inputs of the hydraulic subsystem.

The oscillations in the actuator force introduced by the co-simulation methods may cause the numerical integration to become unstable. As shown in Fig. 14a, increasing the multibody integration step-size $h_\mathcal{M}$ up to 15 ms and, accordingly, the macro communication step-size H_1, resulted in the loss of stability of the direct ZOH co-simulation integration process. The use of a RIM as interface made it possible to increase this step-size beyond $h_\mathcal{M} = 25$ ms. Although larger values of $h_\mathcal{M}$ decreased the precision of the simulation, the solutions can still be considered correct and the integration remained stable throughout the manoeuvre.

5.1.2. Double-actuated model

The model with two hydraulic actuators described in Section 4.1.1 was simulated using the two co-simulation approaches (direct ZOH and RIM-based) and several step-sizes $h_\mathcal{M}$ for the multibody system, while keeping the hydraulics step-size constant, $h_\mathcal{S} = 0.2$ ms. The actuator force $f_{h,2}$ is shown in Fig. 15, where step-size $h_\mathcal{M} = 3$ ms was used for both approaches. The direct co-simulation exhibited a noticeable oscillatory behaviour after $t = 6$ s, compared to the RIM-based co-simulation. Although both methods presented a force peak after changes in the valve spool displacements, the RIM-based method was able to damp the oscillations more efficiently.

Fig. 16 shows the actuator displacement s_1 for both co-simulation approaches. The direct ZOH co-simulation was not able to keep the equilibrium configuration stable and failed for step-sizes larger than 3 ms, whereas with the proposed RIM-based approach the system simulation remained stable for step-sizes as large as 20 ms. The RIM coupled the simulation of the two hydraulic cylinders through the effective mass of the interface, as shown in Eq. (11) and, therefore, it described more accurately the interface dynamics in between macro time-steps. In the actual mechanical system, the dynamics of the
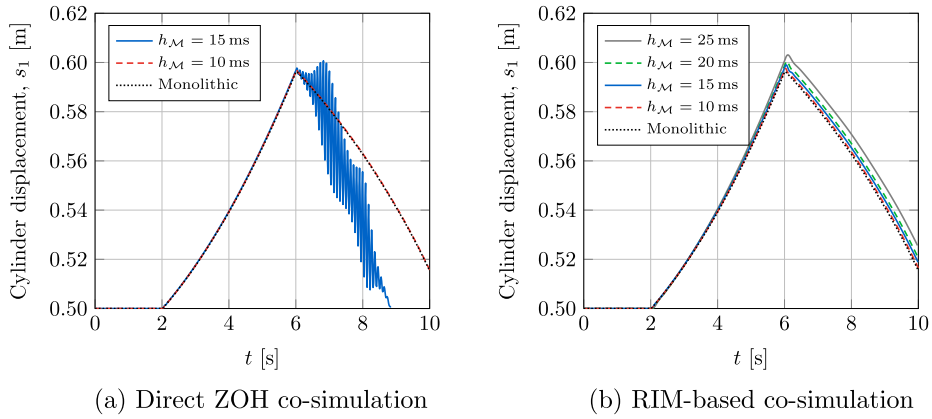
(a) Direct ZOH co-simulation

(b) RIM-based co-simulation

Fig. 14. Actuator displacement s_1 in the single-actuated model for $h_S = 0.2$ ms, different values of $h_\mathcal{M}$, and $h = 0.5$ ms for the monolithic simulation.

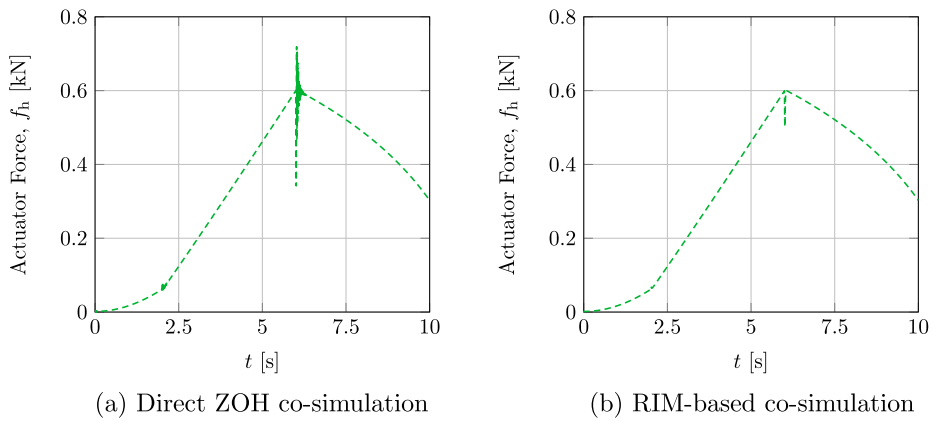(a) Direct ZOH co-simulation

(b) RIM-based co-simulation

Fig. 15. Hydraulic actuator force $f_{h,2}$ of the double-actuated model using direct ZOH and RIM-based co-simulation with step sizes of $h_S = 0.2$ ms and $h_\mathcal{M} = 3$ ms.
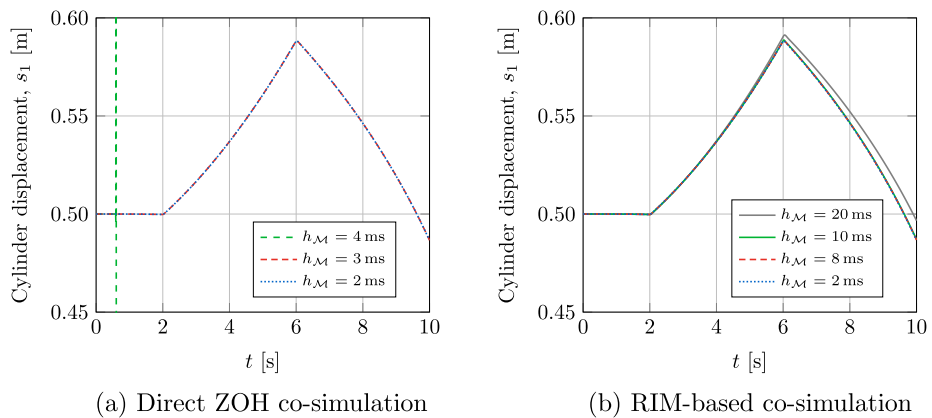

(a) Direct ZOH co-simulation

(b) RIM-based co-simulation

Fig. 16. Actuator displacement s_1 in the double-actuated model for $h_S = 0.2$ ms and different values of $h_\mathcal{M}$.

two hydraulic actuators are coupled through the multibody system. When direct co-simulation is used instead, the effect of one actuator on the mechanical system could not be perceived by the other actuator one until the end of each macro time-step. The integration of the hydraulics differential equations (25) and (26) proceeds separately for each hydraulic cylinder for the duration of the macro step. As a consequence, each of the two actuators evolved as if they were not really interacting, which resulted in a communication delay and the consequent loss of stability. The RIM-based approach accounts for this dynamic coupling among the actuators, and is able to handle situations where the direct co-simulation fails.

Table 4
Initial static equilibrium values of the spool displacement of the hydraulic actuators of the manipulator mounted on the Juno rover.

Actuator	κ	p_1 (MPa)	p_2 (MPa)
1	0.5126	4.04	3.66
2	0.5	3.85	3.85
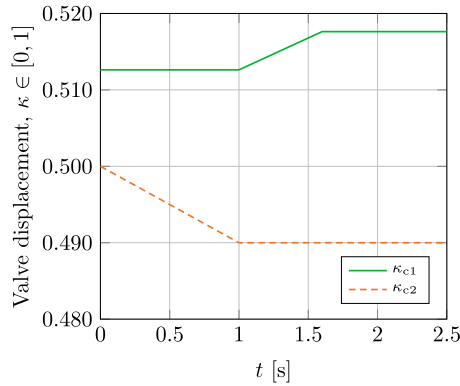
Fig. 17. Spool displacements during the manoeuvre with the Juno rover.

(a) Direct ZOH co-simulation

(b) RIM-based co-simulation
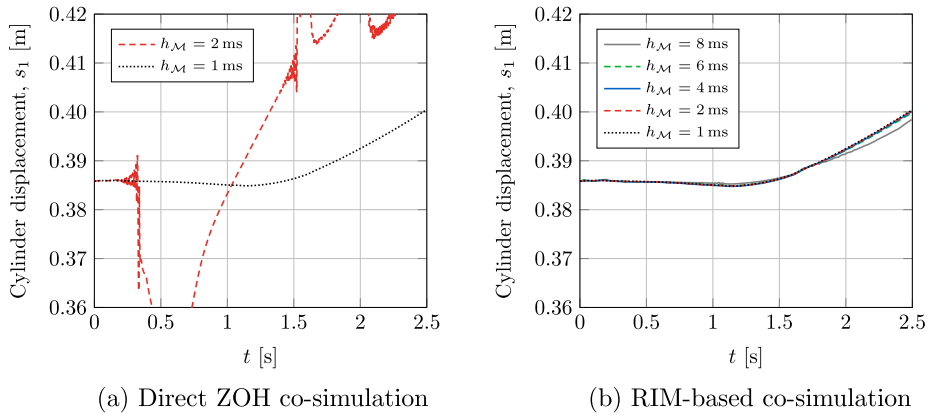
Fig. 18. Actuator displacement s_1 during the manoeuvre with the Juno rover for $h_S = 0.2$ ms and different values of h_M.

5.2. Off-road vehicle with robotic manipulator

The initial static equilibrium of the Juno rover manipulator assembly requires the spool displacements and hydraulic pressures included in Table 4. The angles of the first and second rod with respect to the horizontal x axis were set to $\theta_1 = 45°$ and $\theta_2 = -90°$.

With the initial values of κ shown in Table 4, the time-history of the spool displacements κ_{c1} and κ_{c2} during the manoeuvre with the Juno rover is that shown in Fig. 17.

It must be noted that the hydraulic manipulator mounted on the Juno rover turned out to be a stiff problem, because of the mutual interaction between the dynamics of the hydraulic system and the forces developed at the wheel/terrain interface, given by the terramechanics relations. This demanded the use of comparatively smaller step-sizes in the integration of the mechanical system. Fig. 18 shows the displacement of the first actuator, s_1, during the manoeuvre with the rover for different values of the step-size h_M used to integrate the multibody system. Both co-simulation approaches, direct ZOH and RIM-based, delivered practically the same motion for $h_M = 1$ ms. However, h_M could not be increased beyond 2 ms with the ZOH communication without rendering the simulation unstable. The RIM-based co-simulation, on the other hand, enabled the use of h_M integration step-sizes up to 6 ms, while still delivering reasonably accurate results.

Table 5 shows the elapsed times in the simulation of manoeuvre with both co-simulation methods, for different step-sizes h_M. The reported times were obtained averaging the times elapsed in five runs of the program. The results shown correspond to the total CPU time required by the simulation, as the execution of the code was not parallelized, i.e., the

Table 5
Elapsed times in the simulation
of the Juno rover manoeuvre.

$h_{\mathcal{M}}$	ZOH	RIM
1 ms	15.32 s	20.76 s
2 ms	(Failed)	11.03 s
3 ms	(Failed)	7.64 s
4 ms	(Failed)	6.17 s
5 ms	(Failed)	5.28 s

subsystems dynamics were evaluated sequentially in a single processor. It can be observed that using a RIM introduces a computational overhead, derived from the need to evaluate the effective mass and force terms in Eqs. (17) and (18), the integration of the RIM dynamics, and the additional communication steps required to synchronize the RIM and the hydraulics blocks in Fig. 4. Nonetheless, results also showed that the RIM-based scheme was able to achieve shorter execution times by employing larger step-sizes $h_{\mathcal{M}}$ in the solution of the dynamics of the multibody system.

6. Conclusions

Non-iterative co-simulation of multiphysics models is a promising but challenging approach to study the behaviour of complex engineering systems. With this technique, the exchange of information between subsystems takes place only at discrete synchronization points and this may give rise to the instability of the numerical integration process. When a multibody system is included in the simulation, a meaningful reduced interface model (RIM) representation can be used to interface it to other subsystems, improving the overall stability and dynamics behaviour. The RIM proposed here is based on the characterization of the multibody interface, which can be described with the selection of a representative parametrization. Unlike other reduction techniques, which obtain reduced order models from the extraction of motion modes from a previously known dynamic response of the full order model, our methodology is based on expressing the multibody system dynamics in terms of the interface velocities. The resulting RIM is defined by the effective mass matrix $\widetilde{\mathbf{M}}$ and force vector $\tilde{\mathbf{f}}$ terms, respectively. This representation of the multibody system can then be used as communication interface between the full model and other subsystems in the co-simulation setup.

The ability of the proposed RIM to enhance the stability of non-iterative co-simulation of multiphysics systems was assessed with several examples that included both mechanical and hydraulic components. Results confirmed that stability and dynamic behaviour were improved using the RIM, and showed that coupling subsystem solvers through a RIM of the multibody system can lead to a more stable integration process. The use of RIM representations of the multibody system made the selection of larger communication step-sizes possible in the co-simulation algorithm, reducing the total computation time required to carry out the simulation.

Acknowledgements

Appendix A. Effective mass matrix and force term expressions

Using the velocity transformation in Eq. (14), it is possible to decouple the system dynamics in Eq. (15) as it was done in Eq. (6). The dynamics in the subspace of constrained motion can accordingly be expressed as

$$\begin{aligned}
&\left(\mathbf{D}\mathbf{H}^{-1}\mathbf{D}^{\mathrm{T}}\right)^{-1}\dot{\mathbf{w}}_{\mathrm{c}} = \\
&\left(\mathbf{D}\mathbf{H}^{-1}\mathbf{D}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D}\mathbf{H}^{-1}\left(\mathbf{g}-\mathbf{d}+\mathbf{D}_{\mathrm{i}}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{i}}\right)+\dot{\mathbf{D}}_{\mathrm{i}}\mathbf{u}+\mathbf{D}\mathbf{H}^{-1}\mathbf{D}^{\mathrm{T}}\boldsymbol{\lambda}\right)
\end{aligned} \tag{A.1}$$

Assuming that $\mathbf{D}\mathbf{M}^{-1}\mathbf{D}^{\mathrm{T}}$ is a regular matrix, the expression of the constraint reactions as a function of $\boldsymbol{\lambda}_{\mathrm{i}}$ can be obtained from Eq. (A.1) as

$$\boldsymbol{\lambda} = \left(\mathbf{D}\mathbf{H}^{-1}\mathbf{D}^{\mathrm{T}}\right)^{-1}\left(\dot{\mathbf{w}}_{\mathrm{c}}-\dot{\mathbf{D}}\mathbf{u}-\mathbf{D}\mathbf{H}^{-1}\left(\mathbf{g}-\mathbf{d}+\mathbf{D}_{\mathrm{i}}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{i}}\right)\right) \tag{A.2}$$

The constrained system dynamics in Eq. (15)

$$\mathbf{H}\dot{\mathbf{u}}+\mathbf{d} = \mathbf{g}+\mathbf{D}_{\mathrm{i}}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{i}}+\mathbf{D}^{\mathrm{T}}\boldsymbol{\lambda} \tag{A.3}$$

can also be transformed and decoupled using the velocity transformation in Eq. (5) to obtain

$$\left(\mathbf{D}_{\mathrm{i}}\mathbf{H}^{-1}\mathbf{D}_{\mathrm{i}}^{\mathrm{T}}\right)^{-1}\dot{\mathbf{w}}_{\mathrm{i}} = \left(\mathbf{D}_{\mathrm{i}}\mathbf{H}^{-1}\mathbf{D}_{\mathrm{i}}^{\mathrm{T}}\right)^{-1}\left(\mathbf{g}-\mathbf{d}+\mathbf{D}^{\mathrm{T}}\boldsymbol{\lambda}+\dot{\mathbf{D}}_{\mathrm{i}}\mathbf{u}\right)+\boldsymbol{\lambda}_{\mathrm{i}} \tag{A.4}$$
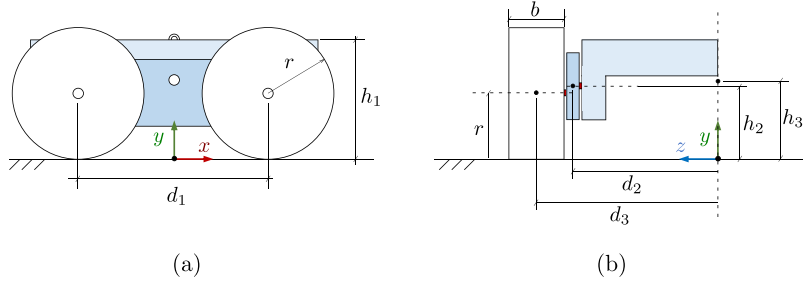
Fig. B1. Juno rover side (a) and front (b) views.

Substituting the expression of $\boldsymbol{\lambda}_i$ given by Eq. (A.2) in Eq. (A.4), the acceleration of the interaction interface becomes

$$
\begin{aligned}
\dot{\mathbf{w}}_i ={}& \mathbf{D}_i\mathbf{H}^{-1}\big(\mathbf{I} - \mathbf{D}^T\big(\mathbf{D}\mathbf{H}^{-1}\mathbf{D}^T\big)^{-1}\mathbf{D}\mathbf{H}^{-1}\big)(\mathbf{g} - \mathbf{d}) \\
& + \mathbf{D}_i\mathbf{H}^{-1}\big(\mathbf{I} - \mathbf{D}^T\big(\mathbf{D}\mathbf{H}^{-1}\mathbf{D}^T\big)^{-1}\mathbf{D}\mathbf{H}^{-1}\big)\mathbf{D}_i^T\boldsymbol{\lambda}_i \\
& + \dot{\mathbf{D}}_i\mathbf{u} + \mathbf{D}_i\mathbf{H}^{-1}\mathbf{D}^T\big(\mathbf{D}\mathbf{H}^{-1}\mathbf{D}^T\big)^{-1}\big(\dot{\mathbf{w}}_c - \dot{\mathbf{D}}\mathbf{u}\big)
\end{aligned}
\tag{A.5}
$$

The projector matrix onto the subspace of constrained motion, $\mathbf{P}_c$, is obtained as [31]

$$
\mathbf{P}_c = \mathbf{H}^{-1}\mathbf{D}^T\big(\mathbf{D}\mathbf{H}^{-1}\mathbf{D}^T\big)^{-1}\mathbf{D}
\tag{A.6}
$$

and the accelerations in the subspace of admissible motion can be written in terms of the system generalized velocities and accelerations by differentiating Eq. (14) with respect to time

$$
\dot{\mathbf{w}}_c = \mathbf{D}\dot{\mathbf{u}} + \dot{\mathbf{D}}\mathbf{u}
\tag{A.7}
$$

Substituting Eqs. (A.6) and (A.7) in Eq. (A.5)

$$
\dot{\mathbf{w}}_i = \mathbf{D}_i(\mathbf{I} - \mathbf{P}_c)\mathbf{H}^{-1}(\mathbf{g} - \mathbf{d}) + \mathbf{D}_i(\mathbf{I} - \mathbf{P}_c)\mathbf{H}^{-1}\mathbf{D}_i^T\boldsymbol{\lambda}_i + \dot{\mathbf{D}}_i\mathbf{u} + \mathbf{D}_i\mathbf{P}_c\dot{\mathbf{u}}
\tag{A.8}
$$

Equation (A.8) can be cast into the form

$$
\widetilde{\mathbf{M}}_c\dot{\mathbf{w}}_i = \widetilde{\mathbf{f}}_c + \boldsymbol{\lambda}_i
\tag{A.9}
$$

where

$$
\widetilde{\mathbf{M}}_c = \big(\mathbf{D}_i(\mathbf{I} - \mathbf{P}_c)\mathbf{H}^{-1}\mathbf{D}_i^T\big)^{-1}
\tag{A.10}
$$

$$
\widetilde{\mathbf{f}}_c = \widetilde{\mathbf{M}}_c\big(\mathbf{D}_i(\mathbf{I} - \mathbf{P}_c)\mathbf{H}^{-1}(\mathbf{g} - \mathbf{d}) + \dot{\mathbf{D}}_i\mathbf{u} + \mathbf{D}_i\mathbf{P}_c\dot{\mathbf{u}}\big)
\tag{A.11}
$$

are the effective mass matrix and force term of the mechanical system. Note that, when the system is modelled with independent coordinates no kinematic constraints are imposed on the system. In this case, the projector is the null matrix, and the effective mass matrix and force term become

$$
\widetilde{\mathbf{M}} = \big(\mathbf{D}_i\mathbf{H}^{-1}\mathbf{D}_i^T\big)^{-1}
\tag{A.12}
$$

$$
\widetilde{\mathbf{f}} = \widetilde{\mathbf{M}}\big(\mathbf{D}_i\mathbf{H}^{-1}(\mathbf{g} - \mathbf{d}) + \dot{\mathbf{D}}_i\mathbf{u}\big)
\tag{A.13}
$$

which are equivalent to the terms in Eqs. (12) and (13).

Appendix B. Computational model of the Juno rover

The Juno rover most relevant dimensions are shown in Fig. B.1. The numerical values of these dimensions are reported in Table B.1.

The mass and inertia properties of the different components used to model the Juno rover are summarized in Table B.2. The inertia moments are provided at the centre of mass and expressed in the local axes of each component. The centre of mass of each component was assumed to be located at its geometrical centre.

The forces and torques generated at the wheel-terrain interface were obtained using the terramechanics relations reported in [36], as a function of the system state, i.e., its generalized coordinates and velocities. With this approach, a set of terrain parameters, shown in Table B.3, is required to characterize the terrain behaviour. The selected parameters are representative of dry sand terrains.

Fig. B.2 shows the forces and torques at the interface and the magnitudes that are necessary to evaluate them using the terramechanics model. The load on the wheel is W, and its driving torque T_a. The wheel is moving with angular velocity ω and its centre translates with velocity v, contained in the wheel plane.

Table B1
Juno rover dimensions.

Wheel radius	r	0.2795 m
Wheel width	b	0.2340 m
x-distance between wheel centres	d_1	0.8350 m
z-distance between bogie centres	d_2	1.2560 m
z-distance between wheel centres	d_3	1.5420 m
Height to ground of top surface of main body	h_1	0.4335 m
Height to ground of c.o.m. of bogie	h_2	0.3045 m
Height to ground of c.o.m. of main body	h_3	0.3295 m

Table B2
Mass and inertia properties of Juno model components.

Component	mass [kg]	I_x [kgm^2]	I_y [kgm^2]	I_z [kgm^2]
Wheel	10.0	0.7812	0.3906	0.3906
Bogie	70.0	9.16	18.32	9.16
Chassis	140.0	9.16	8.396	9.16
Manipulator, link 1	15.0	$31.25 \cdot 10^{-4}$	$56.25 \cdot 10^{-3}$	$56.25 \cdot 10^{-3}$
Manipulator, link 2	25.0	$2.08 \cdot 10^{-4}$	$10.42 \cdot 10^{-3}$	$10.42 \cdot 10^{-3}$

Table B3
Terrain parameters used to characterize the wheel-terrain interaction.

Sinkage exponent	n	1.0
Cohesion stress	c	220 N/m^2
Friction angle	ϕ	33.1°
Cohesive pressure-sinkage modulus	k_c	1.4 kN/m^2
Frictional pressure-sinkage modulus	k_ϕ	2000 kN/m^2
Shear deformation modulus	K	0.015 m
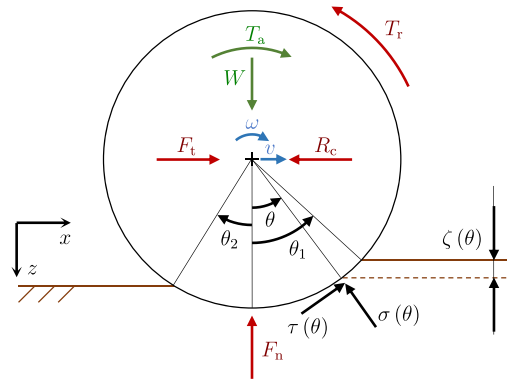
Fig. B2. Terrain reactions at the wheel-soil interface.

The normal stress, $\sigma(\theta)$, at the interface is obtained as

$$\sigma(\theta) = \left(\frac{k_c}{b} + k_\phi\right)\zeta(\theta)^n \tag{B.1}$$

where b is the wheel width, ζ is the wheel sinkage, and θ is the angle that locates the contact point with respect to the vertical direction. The shear stress at the interface can be determined as

$$\tau(\theta) = (c + \sigma(\theta)\tan\phi)\left(1 - e^{-r(\theta_1 - \theta - (1-s)(\sin\theta_1 - \sin\theta))/K}\right) \tag{B.2}$$

where θ_1 is the entry angle of the wheel, r is the wheel radius, and s stands for wheel slip, defined in terms of the angular velocity ω of the wheel and the translational velocity of its centre v as $s = (r\omega - v)/r\omega$.

The normal load exerted by the terrain on the wheel, F_n is evaluated as

$$F_n = rb\left(\int_{\theta_2}^{\theta_1}\tau(\theta)\sin\theta\,\mathrm{d}\theta + \int_{\theta_2}^{\theta_1}\sigma(\theta)\cos\theta\,\mathrm{d}\theta\right) + c_z\dot{\zeta} \tag{B.3}$$

where θ_2 is the exit angle and c_z is a damping coefficient to provide the terrain model with a dissipative behaviour, set to 10% of the terrain stiffness. The traction force F_t and rolling resistance are given by

$$F_t = rb \int_{\theta_2}^{\theta_1} \tau(\theta) \cos\theta \, d\theta \tag{B.4}$$

$$R_c = rb \int_{\theta_2}^{\theta_1} \sigma(\theta) \sin\theta \, d\theta \tag{B.5}$$

and the resisting torque can be obtained as

$$T_r = r^2 b \int_{\theta_2}^{\theta_1} \tau(\theta) \, d\theta \tag{B.6}$$

The lateral force was assumed to be zero in this research.

References

[1] J. García de Jalón, E. Bayo, Kinematic and Dynamic Simulation of Multibody Systems. The Real–Time Challenge, Springer–Verlag, New York, USA, 1994, doi:10.1007/978-1-4612-2600-0.

[2] O.A. Bauchau, Flexible Multibody Dynamics, Springer, Dordrecht, The Netherlands, 2011, doi:10.1007/978-94-007-0335-3.

[3] J.C. Samin, O. Brüls, J.F. Collard, L. Sass, P. Fisette, Multiphysics modeling and optimization of mechatronic multibody systems, Multibody Syst. Dyn. 18 (3) (2007) 345–373, doi:10.1007/s11044-007-9076-0.

[4] C.W. Gear, D.R. Wells, Multirate linear multistep methods, BIT Numer. Math. 24 (4) (1984) 484–502, doi:10.1007/BF01934907.

[5] C.A. Felippa, K. Park, C. Farhat, Partitioned analysis of coupled mechanical systems, Comput. Methods Appl. Mech. Eng. 190 (24–25) (2001) 3247–3270, doi:10.1016/s0045-7825(00)00391-1.

[6] O. Vaculín, W.R. Krüger, M. Valášek, Overview of coupling of multibody and control engineering tools, Veh. Syst. Dyn. 41 (5) (2004) 415–429, doi:10.1080/00423110412331300363.

[7] T. Blockwitz, M. Otter, J. Akesson, M. Arnold, C. Clauss, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson, A. Viel, Functional mockup interface 2.0: the standard for tool independent exchange of simulation models, in: Proceedings of the 9th International MODELICA Conference,Munich, Germany, 2012, pp. 173–184, doi:10.3384/ecp12076173.

[8] C. Andersson, Methods and Tools for Co-Simulation of Dynamic Systems with the Functional Mock-up Interface, Lund University, 2016 Ph.D. thesis.

[9] R. Kübler, W. Schiehlen, Modular simulation in multibody system dynamics, Multibody Syst. Dyn. 4 (2) (2000) 107–127, doi:10.1023/A:1009810318420.

[10] M. Benedikt, A. Hofer, Guidelines for the application of a coupling method for non-iterative co-simulation, in: 2013 8th EUROSIM Congress on Modelling and Simulation (EUROSIM), Cardiff, UK, 2013, pp. 244–249, doi:10.1109/EUROSIM.2013.52.

[11] C. Gomes, B. Legat, R.M. Jungers, H. Vangheluwe, Stable adaptive co-simulation: a switched systems approach, in: IUTAM Symposium on Solver Coupling and Co-Simulation, Darmstadt, Germany, 2017.

[12] M. Benedikt, D. Watzenig, A. Hofer, Modelling and analysis of the non-iterative coupling process for co-simulation, Math. Comput. Model Dyn. Syst. 19 (5) (2013) 451–470, doi:10.1080/13873954.2013.784340.

[13] M. Benedikt, F.R. Holzinger, Automated configuration for non-iterative co-simulation, in: Proceedings of the 17th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE), Montpellier, France, 2016, doi:10.1109/EuroSimE.2016.7463355.

[14] S. Sadjina, L.T. Kyllingstad, S. Skjong, E. Pedersen, Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation, Eng. Comput. 33 (3) (2017) 607–620, doi:10.1007/s00366-016-0492-8.

[15] A. Mohtat, S. Arbatani, J. Kövecses, Enhancing the stability of co-simulation via an energy-leak monitoring and dissipation framework, in: IUTAM Symposium on Solver Coupling and Co-Simulation, Darmstadt, Germany, 2017.

[16] M. Arnold, Simulation Techniques for Applied Dynamics, Numerical methods for simulation in applied dynamics, Springer, pp. 191–246. doi:10.1007/978-3-211-89548-1.

[17] F. González, M.A. Naya, A. Luaces, M. González, On the effect of multi-rate co-simulation techniques in the efficiency and accuracy of multibody system dynamics, Multibody Syst. Dyn. 25 (4) (2011) 461–483, doi:10.1007/s11044-010-9234-7.

[18] O. Oberschelp, H. Vöcking, Multirate simulation of mechatronic systems, in: Proceedings of the IEEE International Conference on Mechatronics 2004, Istanbul, 2004, pp. 404–409.

[19] M. Busch, Continuous approximation techniques for co-simulation methods: analysis of numerical stability and local error, ZAMM - J. Appl. Math. Mech. / Zeitschrift für Angewandte Mathematik und Mechanik 96 (9) (2016) 1061–1081, doi:10.1002/zamm.201500196.

[20] B. Schweizer, P. Li, D. Lu, Explicit and implicit cosimulation methods: stability and convergence analysis for different solver coupling approaches, J. Comput. Nonlinear Dyn. 10 (5) (2015) 051007,1–12, doi:10.1115/1.4028503.

[21] B. Schweizer, D. Lu, Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints, Multibody Syst. Dyn. 34 (2) (2015) 129–161, doi:10.1007/s11044-014-9422-y.

[22] B. Schweizer, P. Li, D. Lu, Co-simulation method for solver coupling with algebraic constraints incorporating relaxation techniques, Multibody Syst. Dyn. 36 (1) (2016) 1–36, doi:10.1007/s11044-015-9464-9.

[23] R. Masoudi, T. Uchida, J. McPhee, Reduction of multibody dynamic models in automotive systems using the proper orthogonal decomposition, J. Comput. Nonlinear Dyn. 10 (3) (2015) 031007, doi:10.1115/1.4029390.

[24] D. Stadlmayr, W. Witteveen, W. Steiner, A generalized constraint reduction method for reduced order MBS models, Multibody Syst. Dyn. 41 (3) (2017) 259–274, doi:10.1007/s11044-016-9557-0.

[25] B. Besselink, U. Tabak, A. Lutowska, N. Van De Wouw, H. Nijmeijer, D. Rixen, M. Hochstenbach, W. Schilders, A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control, J. Sound Vib. 332 (19) (2013) 4403–4422, doi:10.1016/j.jsv.2013.03.025.

[26] P. Koutsovasilis, M. Beitelschmidt, Comparison of model reduction techniques for large mechanical systems, Multibody Syst. Dyn. 20 (2) (2008) 111–128, doi:10.1007/s11044-008-9116-4.

[27] K. Luo, H. Hu, C. Liu, Q. Tian, Model order reduction for dynamic simulation of a flexible multibody system via absolute nodal coordinate formulation, Comput. Methods Appl. Mech. Eng. 324 (2017) 573–594, doi:10.1016/j.cma.2017.06.029.

[28] F. González, P. Masarati, J. Cuadrado, M.A. Naya, Assessment of linearization approaches for multibody dynamics formulations, J. Comput. Nonlinear Dyn. 12 (4) (2017) 041009, doi:10.1115/1.4035410.

[29] J. Kövecses, Dynamics of mechanical systems and the generalized free–body diagram – part II: imposition of constraints, J. Appl. Mech. 75 (6, paper 061013) (2008) 1–8, doi:10.1115/1.2965373.

[30] F. González, J. Kövecses, J.M. Font-Llagunes, Load assessment and analysis of impacts in multibody systems, Multibody Syst. Dyn. 38 (1) (2016) 1–19, doi:10.1007/s11044-015-9485-4.

[31] J. Kövecses, Dynamics of mechanical systems and the generalized free–body diagram – part I: general formulation, J. Appl. Mech. 75 (6, paper 061012) (2008) 1–12, doi:10.1115/1.2965372.

[32] A. Ben Khaled-El Feki, L. Duval, C. Faure, D. Simon, M.B. Gaid, CHOPtrey: contextual online polynomial extrapolation for enhanced multi-core co-simulation of complex systems, Simulation 93 (3) (2017) 185–200, doi:10.1177/0037549716684026.

[33] 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, doi:10.2478/v10180-011-0016-4.

[34] A. Cardona, M. Geradin, Modeling of a hydraulic actuator in flexible machine dynamics simulation, Mech. Mach. Theory 25 (2) (1990) 193–207, doi:10.1016/0094-114X(90)90121-Y.

[35] B. Ghotbi, F. González, J. Kövecses, J. Angeles, A novel concept for analysis and performance evaluation of wheeled rovers, Mech. Mach. Theory 83 (2015) 137–151, doi:10.1016/j.mechmachtheory.2014.08.017.

[36] B. Ghotbi, F. González, J. Kövecses, J. Angeles, Mobility evaluation of wheeled robots on soft terrain: effect of internal force distribution, Mech. Mach. Theory 100 (2016) 259–282, doi:10.1016/j.mechmachtheory.2016.02.005.

[37] E. Bayo, R. Ledesma, Augmented Lagrangian and mass–orthogonal projection methods for constrained multibody dynamics, Nonlinear Dyn. 9 (1–2) (1996) 113–130, doi:10.1007/BF01833296.