

Comparing double-step and penalty-based semirecursive formulations for hydraulically actuated multibody systems in a monolithic approach

Jaiswal Suraj, Rahikainen Jarkko, Khadim Qasim, Sopanen Jussi, Mikkola Aki

This is a Publisher's version of a publication
published by Springer Nature
in Multibody System Dynamics

DOI: 10.1007/s11044-020-09776-4

Copyright of the original publication:

© The Author(s) 2021

Please cite the publication as follows:

Jaiswal, S., Rahikainen, J., Khadim, Q. et al. Comparing double-step and penalty-based semirecursive formulations for hydraulically actuated multibody systems in a monolithic approach. *Multibody Syst Dyn* 52, 169–191 (2021). <https://doi.org/10.1007/s11044-020-09776-4>

**This is a parallel published version of an original publication.
This version can differ from the original published article.**



Comparing double-step and penalty-based semirecursive formulations for hydraulically actuated multibody systems in a monolithic approach

Suraj Jaiswal¹ · Jarkko Rahikainen¹ · Qasim Khadim¹ ·
Jussi Sopanen¹ · Aki Mikkola¹

Received: 18 March 2020 / Accepted: 26 December 2020 / Published online: 21 January 2021
© The Author(s) 2021

Abstract The simulation of mechanical systems often requires modeling of systems of other physical nature, such as hydraulics. In such systems, the numerical stiffness introduced by the hydraulics can become a significant aspect to consider in the modeling, as it can negatively effect to the computational efficiency. The hydraulic system can be described by using the lumped fluid theory. In this approach, a pressure can be integrated from a differential equation in which effective bulk modulus is divided by a volume size. This representation can lead to numerical stiffness as a consequence of which time integration of a hydraulically driven system becomes cumbersome. In this regard, the used multibody formulation plays an important role, as there are many different procedures for the constraint enforcement and different sets of coordinates to choose from. This paper introduces the double-step semirecursive approach and compares it with a penalty-based semirecursive approach in case of coupled multibody and hydraulic dynamics within the monolithic framework. To this end, hydraulically actuated four-bar and quick-return mechanisms are analyzed as case studies. The two approaches are compared in terms of the work cycle, energy balance, constraint violation, and numerical efficiency of the mechanisms. It is concluded that the penalty-based semirecursive approach has a number of advantages compared with the double-step semirecursive approach, which is in accordance with the literature.

✉ S. Jaiswal
suraj.jaiswal@lut.fi

J. Rahikainen
jarkko.rahikainen@lut.fi

Q. Khadim
qasim.khadim@lut.fi

J. Sopanen
jussi.sopanen@lut.fi

A. Mikkola
aki.mikkola@lut.fi

¹ Department of Mechanical Engineering, Lappeenranta University of Technology, Yliopistonkatu 34, 53850 Lappeenranta, Finland

Keywords Multibody system dynamics · Double-step semirecursive formulation · Penalty-based semirecursive formulation · Hydraulic actuators · Monolithic simulation

1 Introduction

The computer simulation of dynamic systems has proven its value in product development, from early prototyping phase to user training, and emerging digital twins and artificial intelligence applications. In practice, the modeling of mechanisms can be carried out by using multibody system dynamics in which equations of motion describe a force equilibrium for the mechanical system under investigation. The use of multibody system dynamics also allows describing actuator models such as hydraulics or electric drives.

The multibody methods can be, in general, categorized to two main groups according to the selected coordinates [25]. Firstly, there is the group of formulations based on global coordinates, where the absolute positions, velocities, and accelerations of each body are used. In the second case, relative coordinates are employed. Should high computational efficiency be required, as is the case in real-time applications, such as those presented in [22, 23], the use of the relative coordinates is often considered to be an appropriate approach. The selection of an approach is case-dependent, as demonstrated in [17]. The computational efficiency can be significantly effected by implementation details [18], and, among others, the use of the automated differentiation tools [6], and sparse and parallelization techniques [20].

Within the family of methods based on the relative coordinates, the semirecursive approach is often used. In the semirecursive approach, closed loops need to be handled by employing constraint equations. Constraint equations, in turn, can be accounted in the semirecursive approach in many different ways, such as by using the Lagrange multiplier method, the penalty-based approach as proposed by Cuadrado et al. [10] or the double-step approach, which is using a coordinate partitioning as proposed by Rodríguez et al. [33]. The latter two approaches are originated from Featherstone's articulated inertia method [13]. The penalty-based approach utilizes the index-3 augmented Lagrangian formulation with projections [3, 9] to enforce the constraints. After their original introductions, both methods have also been successfully used in practical applications, such as in real-time vehicle simulations [20, 29]. In this study, the double-step approach based on coordinate partitioning method [24, 33] is referred as the double-step semirecursive formulation, and the penalty-based augmented Lagrangian approach [9, 10] is referred as the penalty-based semirecursive formulation.

Regarding the solution of the multiphysics problem that system-level simulations require, two main approaches exist in the literature. A straightforward one is the monolithic approach, or sometimes referred as unified scheme, where a single set of differential equations is formed and integrated forward in time as a whole. In an alternative approach, namely cosimulation approach, in turn, the system is split into two or more subsystems that are each integrated separately. In this approach, the required variables, such as the state variables, are communicated in predetermined time intervals. Multiple instances of both approaches can be found from the literature. Cosimulation, due to the discrete time information exchange and the resulting coupling error, has especially been under keen research interest in recent years [16, 36]. The studies include important aspects, such as the multirate cosimulation [18], cosimulation configuration [4, 30], and energy-based coupling error minimization [4, 34, 35]. Monolithic schemes, in turn, have been under less active development since the simple coupling requires less research effort. Nevertheless, in [12] a multiphysics model

was derived for a semiactive car suspension. Naya et al. [28], in turn, presented a monolithic method for a coupled multibody and hydraulic simulation based on the index-3 augmented Lagrangian [3], followed later in [32] and [31], where the index-3 semirecursive formulation [10] was used.

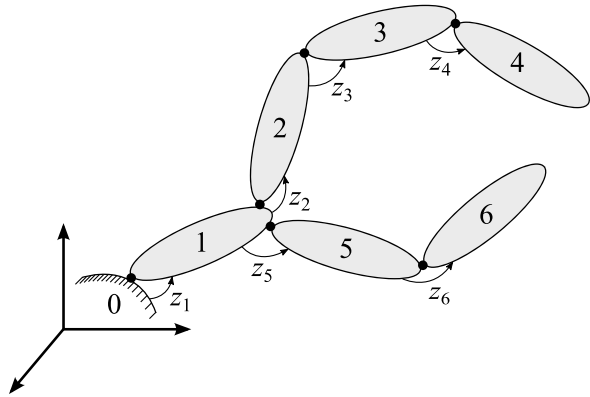
In the case of a coupled multibody and hydraulic dynamics, which is a common case in mobile working machinery, one of the key aspects to consider is the numerical stiffness introduced by the hydraulics. This problem can be alleviated by proper selection of a multibody approach. While these aspects have been discussed in some sources, such as in [32], few comparisons between the available methods in this context exist in the literature. In the context of the absolute nodal coordinate formulation, in turn, a study in this direction has been done by Matikainen et al. [27], where coordinate partitioning, Lagrange multiplier method with Baumgarte's stabilization, and penalty formulation were used for constraint enforcement. The results indicated best performance for the Lagrange multiplier method, closely followed by the penalty-based approach, while less efficient solution was sought with the coordinate partitioning method.

The objective of this paper is to introduce the double-step semirecursive formulation [24] and compare it with the penalty-based semirecursive formulation [10, 32] in the framework of hydraulically driven systems. A monolithic scheme for the coupled simulation of the double-step semirecursive formulation and hydraulic systems is introduced in this study. As explained earlier, the modeling of hydraulic actuators often leads to numerically stiff systems. In this study, the hydraulic system will be described by using the lumped fluid theory. While variable-step integrators often provide more efficient solutions, especially with stiff systems when compared with fixed step-size solutions, since the author's interests lie within the field of real-time simulation, only fixed step-sizes are considered in this work. The coupled systems are referred after the name of the multibody formulations used, such as the double-step semirecursive approach and the penalty-based semirecursive approach. As case studies, hydraulically actuated four-bar and quick-return mechanisms are illustrated. Using the numerical examples, the two approaches are compared based on the work cycle, energy balance, constraint violation, and numerical efficiency of the mechanisms.

2 Semirecursive multibody formulations

The dynamics of a constrained mechanical system can be described by using a multibody system dynamics approach. In the semirecursive formulations, the dynamics of the open-loop multibody systems are formulated in relative joint coordinates, which are independent. In the case of closed-loop multibody systems, in turn, the relative joint coordinates are not independent and cut-joint method is often used to open the loop. In this study, the cut-joint constraints are incorporated by using the coordinate partitioning method [24, 33], referred as the double-step semirecursive formulation, and by using the penalty-based augmented Lagrangian method [2, 9], referred as the penalty-based semirecursive formulation. Since the system is hydraulically actuated, the internal dynamics of the hydraulics are computed and the resultant force, as well as the stroke and stroke velocity, is used to combine the hydraulics to the multibody equations of motion. Therefore, in this study, the constraints are assumed scleronomic. Should kinematic constraints be needed, both methods can easily be extended to rheonomous systems, as shown in the literature [10, 24].

Fig. 1 An example of an open-loop system



2.1 Semirecursive formulation for an open-loop system

In the semirecursive formulation, a system is considered as an open-loop multibody system with N_b bodies. The following Cartesian velocity \mathbf{Z}_j and acceleration $\dot{\mathbf{Z}}_j$ are used to describe the absolute velocity and acceleration of each body [24]

$$\mathbf{Z}_j \equiv \begin{bmatrix} \dot{\mathbf{s}}_j \\ \dot{\boldsymbol{\omega}}_j \end{bmatrix}, \quad \dot{\mathbf{Z}}_j \equiv \begin{bmatrix} \ddot{\mathbf{s}}_j \\ \ddot{\boldsymbol{\omega}}_j \end{bmatrix}, \tag{1}$$

where $\dot{\mathbf{s}}_j$ and $\ddot{\mathbf{s}}_j$ are, respectively, the velocity and acceleration of the point of the body j , which at that particular time is coincident with the origin of the inertial reference frame, and $\dot{\boldsymbol{\omega}}_j$ and $\ddot{\boldsymbol{\omega}}_j$ are the angular velocity and angular acceleration, respectively, of the body j . In this approach, the kinematics of the open-loop system is calculated in a recursive form, either from the base to the leaves, as performed in this study, or from leaves to base, by applying the classical kinematic relations as in [11]. Figure 1 shows an example of an open-loop system. In general case, the position of the system can be described by using the relative joint coordinates $\mathbf{z} = [z_1, z_2, \dots, z_{N_b}]^T$.

The absolute velocity \mathbf{Z}_j and acceleration $\dot{\mathbf{Z}}_j$ for body $j \in [1, N_b]$ can be recursively expressed in terms of the previous bodies as [24]

$$\mathbf{Z}_j = \mathbf{Z}_{j-1} + \mathbf{b}_j \dot{z}_j, \tag{2}$$

$$\dot{\mathbf{Z}}_j = \dot{\mathbf{Z}}_{j-1} + \mathbf{b}_j \ddot{z}_j + \mathbf{d}_j, \tag{3}$$

where the scalars \dot{z}_j and \ddot{z}_j are the first and second time derivatives, respectively, of the relative joint coordinate z_j , and the vectors \mathbf{b}_j and \mathbf{d}_j depend on the type of joint [11] that connects the bodies $j - 1$ and j . Note that the indexes $j - 1$ and j may not be successive, as the system may branch.

The absolute velocities $\mathbf{Z} \in \mathbb{R}^{6N_b}$ and accelerations $\dot{\mathbf{Z}} \in \mathbb{R}^{6N_b}$ of the open-loop system can respectively be expressed in the matrix forms as $\mathbf{Z} = [\mathbf{Z}_1^T, \mathbf{Z}_2^T, \dots, \mathbf{Z}_{N_b}^T]^T$ and $\dot{\mathbf{Z}} = [\dot{\mathbf{Z}}_1^T, \dot{\mathbf{Z}}_2^T, \dots, \dot{\mathbf{Z}}_{N_b}^T]^T$. A velocity transformation matrix $\mathbf{R} \in \mathbb{R}^{6N_b \times N_b}$ that maps the absolute velocities into a set of relative joint velocities can be written as [11, 24]

$$\mathbf{Z} = \mathbf{R}\dot{\mathbf{z}} = \mathbf{TR}_d\dot{\mathbf{z}}, \tag{4}$$

$$\dot{\mathbf{Z}} = \mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} = \mathbf{TR}_d\ddot{\mathbf{z}} + \mathbf{TR}_d\dot{\mathbf{z}}, \tag{5}$$

where $\dot{\mathbf{z}} \in \mathbb{R}^{N_b}$ is the vector of relative joint velocities, $\mathbf{T} \in \mathbb{R}^{6N_b \times 6N_b}$ is the path matrix that represents the topology of the open-loop system, and $\mathbf{R}_d \in \mathbb{R}^{6N_b \times N_b}$ is a diagonal matrix whose elements are the vectors \mathbf{b}_j arranged in an ascending order. The path matrix \mathbf{T} is a lower block-triangular matrix [11] whose elements from diagonal to left are (6×6) identity matrices \mathbf{I}_6 , if the corresponding body is in between the considered body and the root of the system. Note that the term $\dot{\mathbf{R}}\dot{\mathbf{z}}$ in Eq. (5) can be expressed in terms of the vectors \mathbf{d}_j by using Eq. (3) [11].

The absolute velocity \mathbf{Y}_j and acceleration $\dot{\mathbf{Y}}_j$ of the center of mass of body j can be written as

$$\mathbf{Y}_j = \begin{bmatrix} \dot{\mathbf{g}}_j \\ \boldsymbol{\omega}_j \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & -\tilde{\mathbf{g}}_j \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{s}}_j \\ \boldsymbol{\omega}_j \end{bmatrix} = \mathbf{D}_j \mathbf{Z}_j, \tag{6}$$

and

$$\dot{\mathbf{Y}}_j = \begin{bmatrix} \ddot{\mathbf{g}}_j \\ \dot{\boldsymbol{\omega}}_j \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & -\tilde{\mathbf{g}}_j \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{s}}_j \\ \dot{\boldsymbol{\omega}}_j \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{\omega}}_j \tilde{\boldsymbol{\omega}}_j \mathbf{g}_j \\ \mathbf{0} \end{bmatrix} = \mathbf{D}_j \dot{\mathbf{Z}}_j + \mathbf{e}_j, \tag{7}$$

where \mathbf{g}_j is the position vector of the center of mass of body j with respect to the inertial reference frame, \mathbf{I}_3 is a (3×3) identity matrix, and $\dot{\mathbf{g}}_j$, $\ddot{\mathbf{g}}_j$, and $\tilde{\mathbf{g}}_j$, respectively, are the first and second time derivative, and the skew-symmetric matrix of the position vector \mathbf{g}_j . By using Eqs. (6) and (7), the virtual power of the inertia and external forces acting on the open-loop system can be written as [24]

$$\sum_{j=1}^{N_b} \mathbf{Y}_j^{*\text{T}} (\mathbf{M}_j \dot{\mathbf{Y}}_j - \mathbf{Q}_j) = \sum_{j=1}^{N_b} \mathbf{Z}_j^{*\text{T}} (\bar{\mathbf{M}}_j \dot{\mathbf{Z}}_j - \bar{\mathbf{Q}}_j) = 0, \tag{8}$$

where the virtual velocities are denoted with an asterisk (*), and the matrices \mathbf{M}_j , $\bar{\mathbf{M}}_j$, and $\bar{\mathbf{Q}}_j$ can be written as

$$\mathbf{M}_j = \begin{bmatrix} m_j \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_j \end{bmatrix}, \quad \bar{\mathbf{M}}_j = \begin{bmatrix} m_j \mathbf{I}_3 & -m_j \tilde{\mathbf{g}}_j \\ m_j \tilde{\mathbf{g}}_j & \mathbf{J}_j - m_j \tilde{\mathbf{g}}_j \tilde{\mathbf{g}}_j \end{bmatrix}, \tag{9}$$

and

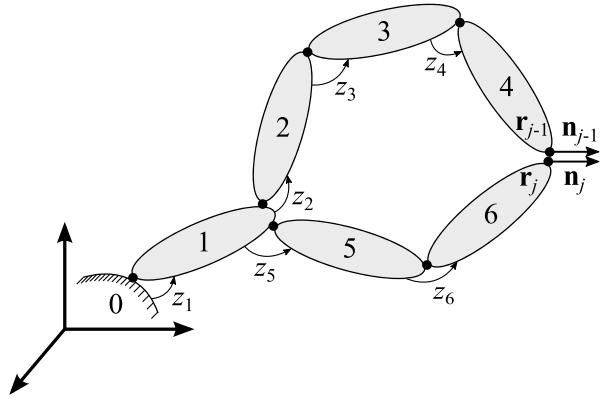
$$\bar{\mathbf{Q}}_j = \begin{bmatrix} \mathbf{f}_j - \tilde{\boldsymbol{\omega}}_j (\tilde{\boldsymbol{\omega}}_j m_j \mathbf{g}_j) \\ \boldsymbol{\tau}_j - \tilde{\boldsymbol{\omega}}_j \mathbf{J}_j \boldsymbol{\omega}_j + \tilde{\mathbf{g}}_j (\mathbf{f}_j - \tilde{\boldsymbol{\omega}}_j (\tilde{\boldsymbol{\omega}}_j m_j \mathbf{g}_j)) \end{bmatrix}, \tag{10}$$

where m_j is the mass of body j and \mathbf{J}_j is the inertia tensor of body j . The inertia tensor \mathbf{J}_j can be written as $\mathbf{J}_j = \mathbf{A}_j^{\text{T}} \bar{\mathbf{J}}_j \mathbf{A}_j$, where $\bar{\mathbf{J}}_j$ is the constant inertia tensor in the body reference frame of body j and \mathbf{J}_j is expressed in the inertial reference frame. Note that both \mathbf{J}_j and $\bar{\mathbf{J}}_j$ are defined with respect to the center of mass of body j . In Eq. (10), \mathbf{f}_j is the vector of external forces applied on body j and $\boldsymbol{\tau}_j$ is the vector of external moments with respect to the center of mass of body j . By substituting Eqs. (4) and (5) in Eq. (8), a set of ordinary differential equations that describes the motion of the open-loop system can be written as [24]

$$\mathbf{R}_d^{\text{T}} \mathbf{T}^{\text{T}} \bar{\mathbf{M}} \mathbf{T} \mathbf{R}_d \dot{\mathbf{z}} = \mathbf{R}_d^{\text{T}} \mathbf{T}^{\text{T}} (\bar{\mathbf{Q}} - \bar{\mathbf{M}} \mathbf{T} \mathbf{R}_d \dot{\mathbf{z}}), \tag{11}$$

where $\bar{\mathbf{M}} \in \mathbb{R}^{6N_b \times 6N_b}$ is a diagonal matrix that consists of the mass matrices $\bar{\mathbf{M}}_j$, and $\bar{\mathbf{Q}} \in \mathbb{R}^{6N_b}$ is a column vector that consists of the force vectors $\bar{\mathbf{Q}}_j$. Equation (11) can be rewritten

Fig. 2 An example of a closed-loop system



as

$$M'\ddot{z} = Q', \tag{12}$$

where $M' = R_d^T T^T \bar{M} T R_d$ and $Q' = R_d^T T^T (\bar{Q} - \bar{M} T \dot{R}_d \dot{z})$.

2.2 Double-step semirecursive formulation

In the double-step semirecursive formulation, the constraints are introduced by using the coordinate partitioning method [24]. This method can be used in frameworks of the explicit and implicit integrators. A set of m constraint equations $\Phi = 0$ are used for the closure of an open-loop. For the sake of simplicity, the constraint equations are assumed holonomic and scleronomic. To account for rheonomic constraints, the reader is referred to [24]. The constraint equations $\Phi = 0$ can be expressed in terms of the relative joint coordinates as $\Phi(z) = 0$. Figure 2 shows an example of a closed-loop system.

By using the coordinate partitioning method, the dependent velocities can be written in terms of the system's degrees of freedom f as [25]

$$\dot{z}^d = -(\Phi_z^d)^{-1} \Phi_z^i \dot{z}^i, \tag{13}$$

where $\Phi_z^d \in \mathbb{R}^{m \times m}$ and $\Phi_z^i \in \mathbb{R}^{m \times f}$ are, respectively, the dependent and independent columns of Jacobian matrix Φ_z , and $\dot{z}^d \in \mathbb{R}^m$ and $\dot{z}^i \in \mathbb{R}^f$ are, respectively, the dependent and independent relative joint velocities. It is assumed that neither redundant constraints nor singular configurations exist, which guarantees that the inverse of Φ_z^d can be found. A velocity transformation matrix $R_z \in \mathbb{R}^{N_b \times f}$ is introduced to transform the dependent relative joint velocities into independent ones as [25]

$$\begin{bmatrix} \dot{z}^d \\ \dot{z}^i \end{bmatrix} = \begin{bmatrix} -(\Phi_z^d)^{-1} \Phi_z^i \\ \mathbf{I} \end{bmatrix} \dot{z}^i \equiv R_z \dot{z}^i. \tag{14}$$

Similarly, accelerations can be written as

$$\ddot{z} = R_z \ddot{z}^i + \dot{R}_z \dot{z}^i, \tag{15}$$

where $\ddot{z}^i \in \mathbb{R}^f$ are the independent relative joint accelerations. In this study, the independent relative joint coordinates are identified by using the Gaussian elimination with full

pivoting to the Jacobian matrix Φ_z . Note that Gaussian elimination with full pivoting is a lower–upper (LU) matrix factorization technique, where the rows and columns of a matrix are interchanged to use the largest element (in absolute value) in the matrix as the pivot [21]. Accordingly, the $(N_b - m)$ columns of the Jacobian matrix Φ_z , where the m pivots do not appear, determines the independent relative joint coordinates [15, 25]. Fiset and Vaneghem utilized the same technique in the identification of dependent and independent coordinates [14]. This can be considered as a relative drawback of the double-step semirecursive formulation because the penalty-based semirecursive formulation utilizes the full set of coordinates. By substituting Eqs. (14) and (15) into Eq. (11), the dynamic equations of the closed-loop system can be written as [7]

$$\mathbf{R}_z^T \mathbf{M}' \mathbf{R}_z \ddot{\mathbf{z}}^i = \mathbf{R}_z^T \mathbf{R}_d^T (\mathbf{T}^T \bar{\mathbf{Q}} - \mathbf{T}^T \bar{\mathbf{M}} \mathbf{D}), \tag{16}$$

where $\mathbf{D} \equiv \mathbf{TR}_d \begin{bmatrix} -(\Phi_z^d)^{-1} (\dot{\Phi}_z \dot{\mathbf{z}}) \\ \mathbf{0} \end{bmatrix} + \mathbf{TR}_d \dot{\mathbf{z}}$ are the absolute accelerations, when vector $\ddot{\mathbf{z}}$ in Eq. (5) is set to zero. Equation (16) can be expressed in a simplified form as

$$\mathbf{M}'' \ddot{\mathbf{z}}^i = \mathbf{Q}'', \tag{17}$$

where $\mathbf{M}'' = \mathbf{R}_z^T \mathbf{M}' \mathbf{R}_z$ and $\mathbf{Q}'' = \mathbf{R}_z^T \mathbf{R}_d^T (\mathbf{T}^T \bar{\mathbf{Q}} - \mathbf{T}^T \bar{\mathbf{M}} \mathbf{D})$. Note that the relation between \mathbf{Q}' and \mathbf{Q}'' can be written as $\mathbf{Q}'' = \mathbf{R}_z^T \mathbf{Q}' - \mathbf{R}_z^T \mathbf{M}' \mathbf{R}_z \dot{\mathbf{R}}_z \dot{\mathbf{z}}^i$.

2.3 Penalty-based semirecursive formulation

In the penalty-based semirecursive formulation, the constraints are introduced by using the penalty-based augmented Lagrangian method [10]. In this formulation, the time integration scheme is carried out by using the trapezoidal rule. The loop-closure constraints, a set of m constraint equations $\Phi = \mathbf{0}$, are introduced in Eq. (12) with a penalty method similar to the index-3 augmented Lagrangian with projections to satisfy the constraints on velocity and acceleration levels [3, 9]. The equations of motion for the closed-loop system can be written as

$$\mathbf{M}' \ddot{\mathbf{z}} + \Phi_z^T \alpha \Phi + \Phi_z^T \lambda = \mathbf{Q}', \tag{18}$$

where Φ_z is the Jacobian matrix of $\Phi(\mathbf{z}) = \mathbf{0}$, α is the penalty factor that can be set the same for all constraints, and λ is the vector of iterated Lagrange multipliers. In this method, these multipliers are obtained at each time-step k as

$$\lambda_k^{(h+1)} = \lambda_k^{(h)} + \alpha \Phi_k^{(h+1)}, \tag{19}$$

where h is the iteration step. The value of $\lambda_k^{(0)}$ is the final value of λ_{k-1} , calculated in the previous time-step [10]. As mentioned earlier, the system is integrated by using an implicit single-step trapezoidal scheme [10]. In this approach, the relative joint velocities $\dot{\mathbf{z}}$ and accelerations $\ddot{\mathbf{z}}$ are corrected by using the mass-damping-stiffness-orthogonal projections as [3, 9]

$$\left[\mathbf{W} + \frac{\Delta t^2}{4} \Phi_z^T \alpha \Phi_z \right] \dot{\mathbf{z}} = \mathbf{W} \dot{\mathbf{z}}', \tag{20}$$

$$\left[\mathbf{W} + \frac{\Delta t^2}{4} \Phi_z^T \alpha \Phi_z \right] \ddot{\mathbf{z}} = \mathbf{W} \ddot{\mathbf{z}}' - \frac{\Delta t^2}{4} \Phi_z^T \alpha (\dot{\Phi}_z \dot{\mathbf{z}}), \tag{21}$$

where $\dot{\mathbf{z}}'$ and $\ddot{\mathbf{z}}'$ are, respectively, the relative joint velocities and accelerations obtained from the Newton–Raphson iteration, and $\mathbf{W} = \mathbf{M} + \frac{\Delta t}{2} \mathbf{C} + \frac{\Delta t^2}{4} \mathbf{K}$, where \mathbf{C} and \mathbf{K} represent the damping and stiffness contributions in the system. Note that in Eqs. (20) and (21), time-dependent constraint terms are not incorporated because the constraints are assumed scleronomous. To account for rheonomic constraints, the reader is referred to [10, 32].

3 Hydraulic actuators

In this study, the hydraulic pressures within a hydraulic circuit is computed by using the lumped fluid theory [37]. By using this approach, a hydraulic circuit is divided into discrete volumes where pressures are assumed to be equally distributed. The effect of acoustic waves is thus assumed to be insignificant. In a hydraulic volume V_k , the pressure p_k can be computed as

$$\dot{p}_k = \frac{B_{ek}}{V_k} \sum_{s=1}^{n_f} Q_{ks}, \tag{22}$$

where Q_{ks} is the sum of incoming and outgoing flows associated with the volume V_k , n_f is the total number of volume flows going in or out of the volume V_k , and B_{ek} is the effective bulk modulus associated to the volume V_k . The effective bulk modulus can be written as

$$B_{ek} = \left(\frac{1}{B_{oil}} + \sum_{s=1}^{n_c} \frac{V_s}{V_k B_s} \right)^{-1}, \tag{23}$$

where B_{oil} is the bulk modulus of oil, n_c is the total number of subvolumes V_s that form the volume V_k , and B_s is the corresponding bulk modulus of the volume V_s .

3.1 Valves

In this study, the valves are described by using a semiempirical modeling method [19]. By using the semiempirical modeling approach, the volume flow rate Q_t through a simple throttle valve can be written as

$$Q_t = C_{v_t} \operatorname{sgn}(\Delta p) \sqrt{|\Delta p|}, \tag{24}$$

where Δp is the pressure difference over the valve, $\operatorname{sgn}(\Delta p)$ is the sign function that determines the sign of Δp , and C_{v_t} is the semiempirical flow rate coefficient of the throttle valve that can be calculated as

$$C_{v_t} = C_d A_t \sqrt{\frac{2}{\rho}}, \tag{25}$$

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

Similarly, the volume flow rate Q_d through a directional control valve can be written as

$$Q_d = C_{v_d} U \operatorname{sgn}(\Delta p) \sqrt{|\Delta p|}, \tag{26}$$

where C_{v_d} is the semiempirical flow rate constant of the valve procured from the manufacturer catalogues, and U is the relative poppet/spool position. If the pressure difference is

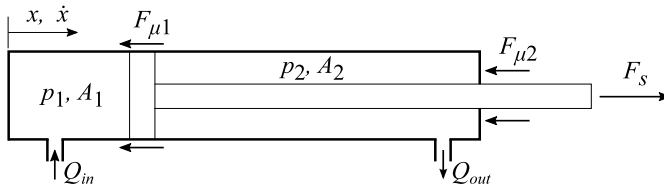


Fig. 3 Schematic figure of a hydraulic cylinder

less than 2 bar, the volume flow is assumed to be laminar, and Eqs. (24) and (26) are modified so that the volume flow and the pressure difference follows a linear relation. Equation (26) is complemented by the following first order differential equation that describes a spool position

$$\dot{U} = \frac{U_{ref} - U}{\tau}, \tag{27}$$

where U_{ref} is the reference voltage signal for the reference spool position, and τ is the time constant, which can be obtained from the Bode-diagram of the valve that describes the dynamics of valve spool.

3.2 Cylinders

The volume flow produced due to the motion of a hydraulic cylinder (shown in Fig. 3) can be written as

$$Q_{in} = \dot{x}A_1, \quad Q_{out} = \dot{x}A_2, \tag{28}$$

where Q_{in} and Q_{out} are, respectively, the volume flow rate going inside and coming out of the cylinder, \dot{x} is the piston velocity, and A_1 and A_2 are, respectively, the areas on the piston and piston-rod side of the cylinder. The force F_s produced by the cylinder can be written in terms of its dimensions and chambers pressure as

$$F_s = p_1A_1 - p_2A_2 - F_\mu, \tag{29}$$

where p_1 and p_2 are, respectively, the pressure on the piston and piston-rod side that can be calculated by using Eq. (22), and F_μ is the total friction force caused by sealing.

4 Coupling of multibody formulations and hydraulic actuators

In this section, the multibody formulations described in Sect. 2 are extended to incorporate the dynamics of the hydraulic actuators described in Sect. 3 in a monolithic approach. The coupling of the double-step semirecursive formulation with the lumped fluid theory is inspired from [28] and [32]. Whereas the coupling of the penalty-based semirecursive formulation with the lumped fluid theory was already carried out in [32]. The force vector \mathbf{Q} in Eqs. (17) and (18) is incremented with the pressure variation equations, leading to the combined system of equations as follows

$$\left. \begin{aligned} \mathbf{M}''\ddot{\mathbf{z}} &= \mathbf{Q}''(\mathbf{z}, \dot{\mathbf{z}}, \mathbf{p}) \\ \dot{\mathbf{p}} &= \mathbf{h}(\mathbf{p}, \mathbf{z}, \dot{\mathbf{z}}) \end{aligned} \right\} \text{(double-step semirecursive approach)}, \tag{30}$$

$$\left. \begin{aligned} \mathbf{M}'\ddot{\mathbf{z}} + \Phi_z^T \alpha \Phi + \Phi_z^T \lambda &= \mathbf{Q}'(\mathbf{z}, \dot{\mathbf{z}}, \mathbf{p}) \\ \lambda^{(h+1)} &= \lambda^{(h)} + \alpha \Phi^{(h+1)} \\ \dot{\mathbf{p}} &= \mathbf{h}(\mathbf{p}, \mathbf{z}, \dot{\mathbf{z}}) \end{aligned} \right\} \text{(penalty-based semirecursive approach),} \tag{31}$$

where \mathbf{p} is a vector of the pressures in the hydraulic subsystem and $\mathbf{h}(\mathbf{p}, \mathbf{z}, \dot{\mathbf{z}})$ are the pressure variation equations. It is assumed that the dependency of both \mathbf{Q}' and function \mathbf{h} with respect to $\mathbf{z}, \dot{\mathbf{z}}$, and \mathbf{p} are known.

In this study, both the approaches are integrated by using an implicit single-step trapezoidal rule [26], which is second order and A-stable method. While the trapezoidal rule was often used in structural dynamics, it was, however, seldom used in multibody system dynamics until the study by Bayo et al. [1]. In the study [1], it was agreed that the trapezoidal rule will lead to poor convergence characteristics when applied to multibody system dynamics in a similar way as other multistep integrators, that is, by considering the accelerations as primary variables. However, Bayo et al. [1] and Cuadrado et al. [8–10] demonstrated that the trapezoidal rule performs very satisfactorily when it is combined directly with the equations of motion by taking the positions as the primary variables, as shown below. Similarly, for the hydraulic subsystem, pressures are taken as the primary variables, as shown in [28, 32]. Note that this study is more inclined to use the above approaches for real-time applications, such as [23], in future studies. Therefore, a single-step integration method is preferred that can use the same computational cost in each integration step [1]. Furthermore, even though the explicit, multistep integrators can be inexpensive and accurate, however, they do not demonstrate good stability, which is a limiting factor for real-time integration [1], especially for stiff systems. Thus, an implicit method is used. Moreover, A-stability is crucial for a numerically stiff system [1], such as presented in this study.

In the double-step semirecursive approach, the trapezoidal rule can be written as

$$\left. \begin{aligned} \mathbf{z}_{k+1}^i &= \mathbf{z}_k^i + \frac{\Delta t}{2} (\dot{\mathbf{z}}_k^i + \dot{\mathbf{z}}_{k+1}^i) \\ \dot{\mathbf{z}}_{k+1}^i &= \dot{\mathbf{z}}_k^i + \frac{\Delta t}{2} (\ddot{\mathbf{z}}_k^i + \ddot{\mathbf{z}}_{k+1}^i) \\ \mathbf{p}_{k+1} &= \mathbf{p}_k + \frac{\Delta t}{2} (\dot{\mathbf{p}}_k + \dot{\mathbf{p}}_{k+1}) \end{aligned} \right\}, \tag{32}$$

where Δt is the time-step, \mathbf{z}_k^i are the independent relative joint coordinates, $\dot{\mathbf{z}}_k^i$ are the independent relative joint velocities, $\ddot{\mathbf{z}}_k^i$ are the independent relative joint accelerations, and \mathbf{p}_k and $\dot{\mathbf{p}}_{k+1}$ are, respectively, the pressures and pressure derivatives. Equation (32) can be rewritten by considering \mathbf{z}_{k+1}^i and \mathbf{p}_{k+1} as the primary variables and, respectively, solving for $\dot{\mathbf{z}}_{k+1}^i$, $\ddot{\mathbf{z}}_{k+1}^i$, and $\dot{\mathbf{p}}_{k+1}$ at time-step $(k + 1)$ as

$$\left. \begin{aligned} \dot{\mathbf{z}}_{k+1}^i &= \frac{2}{\Delta t} \mathbf{z}_{k+1}^i + \hat{\mathbf{z}}_k^i \\ \ddot{\mathbf{z}}_{k+1}^i &= \frac{4}{\Delta t^2} \mathbf{z}_{k+1}^i + \hat{\mathbf{z}}_k^i \\ \dot{\mathbf{p}}_{k+1} &= \frac{2}{\Delta t} \mathbf{p}_{k+1} + \hat{\mathbf{p}}_k \end{aligned} \right\}, \tag{33}$$

where

$$\left. \begin{aligned} \hat{\mathbf{z}}_k^i &= - \left(\frac{2}{\Delta t} \mathbf{z}_k^i + \dot{\mathbf{z}}_k^i \right) \\ \hat{\mathbf{z}}_k^i &= - \left(\frac{4}{\Delta t^2} \mathbf{z}_k^i + \frac{4}{\Delta t} \dot{\mathbf{z}}_k^i + \ddot{\mathbf{z}}_k^i \right) \\ \hat{\mathbf{p}}_k &= - \left(\frac{2}{\Delta t} \mathbf{p}_k + \dot{\mathbf{p}}_k \right) \end{aligned} \right\} . \tag{34}$$

Note that in the double-step semirecursive approach, the rules $\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \dot{\mathbf{z}}_k^i \Delta t + \frac{1}{2} \ddot{\mathbf{z}}_k^i \Delta t^2$ and $\mathbf{p}_{k+1} = \mathbf{p}_k + \dot{\mathbf{p}}_k \Delta t$ are applied to \mathbf{z}_{k+1}^i and \mathbf{p}_{k+1} , respectively. In the double-step semirecursive approach, given \mathbf{z}_{k+1}^i , the dependent relative joint coordinates \mathbf{z}_{k+1}^d are obtained by iteratively solving the loop-closure constraint equations $\Phi(\mathbf{z}) = \mathbf{0}$, which are highly nonlinear [11]. In this study, the Newton–Raphson method is used to solve the loop-closure position problem and convergence is achieved by providing a reliable estimate for \mathbf{z}_{k+1}^d by using information from the previous time-step as [15] $\mathbf{z}_{k+1}^d = \mathbf{z}_k^d + \dot{\mathbf{z}}_k^d \Delta t + \frac{1}{2} \ddot{\mathbf{z}}_k^d \Delta t^2$. The dependent relative joint velocities $\dot{\mathbf{z}}_{k+1}^d$ and accelerations $\ddot{\mathbf{z}}_{k+1}^d$ are computed from Eqs. (14) and (15), respectively. In the penalty-based semirecursive approach, the full set of relative joint coordinates \mathbf{z}_{k+1} , velocities $\dot{\mathbf{z}}_{k+1}$, and accelerations $\ddot{\mathbf{z}}_{k+1}$ are used in the above discussion of Eqs. (32), (33), and (34), instead of \mathbf{z}_{k+1}^i , $\dot{\mathbf{z}}_{k+1}^i$, and $\ddot{\mathbf{z}}_{k+1}^i$. The dynamic equilibrium, established at time-step $(k + 1)$, for both the approaches can be written as

$$\left. \begin{aligned} \mathbf{M}'' \mathbf{z}_{k+1}^i - \frac{\Delta t^2}{4} \mathbf{Q}''_{k+1} + \frac{\Delta t^2}{4} \mathbf{M}'' \hat{\mathbf{z}}_k^i &= \mathbf{0} \\ \frac{\Delta t}{2} \mathbf{p}_{k+1} - \frac{\Delta t^2}{4} \mathbf{h}_{k+1} + \frac{\Delta t^2}{4} \hat{\mathbf{p}}_k &= \mathbf{0} \end{aligned} \right\} \text{(double-step semirecursive approach),} \tag{35}$$

$$\left. \begin{aligned} \mathbf{M}' \mathbf{z}_{k+1} + \frac{\Delta t^2}{4} \Phi_{\mathbf{z}_{k+1}}^T (\alpha \Phi_{k+1} + \lambda_{k+1}) - \frac{\Delta t^2}{4} \mathbf{Q}'_{k+1} + \frac{\Delta t^2}{4} \mathbf{M}' \hat{\mathbf{z}}_k &= \mathbf{0} \\ \lambda_{k+1}^{(h+1)} &= \lambda_{k+1}^{(h)} + \alpha \Phi_{k+1}^{(h+1)} \\ \frac{\Delta t}{2} \mathbf{p}_{k+1} - \frac{\Delta t^2}{4} \mathbf{h}_{k+1} + \frac{\Delta t^2}{4} \hat{\mathbf{p}}_k &= \mathbf{0} \end{aligned} \right\} \tag{36}$$

(penalty-based semirecursive approach).

Equations (35) and (36) are a nonlinear system of equations that can be denoted as $\mathbf{f}(\mathbf{x}_{k+1}) = \mathbf{0}$, where $\mathbf{x} = \left[(\mathbf{z}^T, \mathbf{p}^T)^T \right]$ for the double-step semirecursive approach and $\mathbf{x} = \left[\mathbf{z}^T, \mathbf{p}^T \right]^T$ for the penalty-based semirecursive approach. Such a nonlinear system of equations can be iteratively solved by employing the Newton–Raphson method as

$$\left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right]_{k+1}^{(h)} \Delta \mathbf{x}_{k+1}^{(h)} = - [\mathbf{f}(\mathbf{x})]_{k+1}^{(h)} . \tag{37}$$

The residual vector $[\mathbf{f}(\mathbf{x})]_{k+1}^{(h)}$ can be written as

$$[\mathbf{f}(\mathbf{x})]_{k+1}^{(h)} = \frac{\Delta t^2}{4} \left[\begin{array}{c} \mathbf{M}'' \hat{\mathbf{z}}_k^i - \mathbf{Q}'' \\ \hat{\mathbf{p}} - \mathbf{h} \end{array} \right]_{k+1}^{(h)} \text{ (double-step semirecursive approach),} \tag{38}$$

$$[\mathbf{f}(\mathbf{x})]_{k+1}^{(h)} = \frac{\Delta t^2}{4} \left[\mathbf{M}\ddot{\mathbf{z}} + \Phi_z^T \alpha \Phi + \Phi_z^T \lambda - \mathbf{Q}' \right]_{k+1}^{(h)} \tag{39}$$

(penalty-based semirecursive approach),

where λ is obtained as shown in Eq. (19). The approximated tangent matrix $[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}]_{k+1}^{(h)}$ can be obtained numerically by using the forward differentiation rule

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0 + \epsilon) - f(x_0)}{\epsilon}, \tag{40}$$

where ϵ is the differentiation increment. To avoid ill-conditioning of the tangent matrix, ϵ is computed as in [32], namely

$$\epsilon = 1 \times 10^{-8} \max(1 \times 10^{-2}, |x_0|), \tag{41}$$

where 1×10^{-2} limits the minimum value for the differentiation increment to 1×10^{-10} . Equation (41) is a modification of a method presented in [5]. In the penalty-based semirecursive approach, $\dot{\mathbf{z}}$ and $\ddot{\mathbf{z}}$ are corrected by using the mass-damping-stiffness-orthogonal projections [3, 9] as shown in Eqs. (20) and (21).

5 The case studies of hydraulically actuated four-bar and quick-return mechanisms

In this study, a hydraulically actuated four-bar mechanism, as shown in Fig. 4, and a hydraulically actuated quick-return mechanism, as shown in Fig. 5, are used for a comparative study between the two multibody formulations in a numerically stiff coupled environment. The numerical stiffness in the coupled environment is introduced by the hydraulic subsystem. The mechanisms are modeled, first by using the double-step semirecursive formulation, and later by using the penalty-based semirecursive formulation, as explained in Sect. 2. For the planar system in Fig. 4, three joint coordinates are used in the modeling of the structure and two loop-closure constraints are used for a cut-joint (revolute joint) at point *E*. Whereas for the planar system in Fig. 5, five joint coordinates are used in the modeling of the structure and four loop-closure constraints are used for two cut-joints (translational joints) at points *J* and *M*. Both the mechanisms have one degree of freedom.

In Fig. 4, bodies 1, 2, and 3 are assumed as rectangular beams, whose lengths and masses are $L_1 = 9$ m, $L_2 = \sqrt{2}$ m, and $L_3 = 2$ m, and $m_1 = 225$ kg, $m_2 = 35$ kg, and $m_3 = 50$ kg, respectively. The locations of points *E*, *G*, and *C* in the inertial reference frame are $[0, -1, 0]^T$ m, $[0, -2, 0]^T$ m, and $[\frac{L_1}{3}, 0, 0]^T$ m, respectively. Point *F* is located at the center of mass of body-3. In Fig. 4, the relative joint coordinates at points *O*, *C*, and *D* are respectively represented by z_1, z_2 and z_3 that define the orientation of the respective bodies in the inertial reference frame, *XYZ*. Their initial values are considered as $0^\circ, -135^\circ$, and -45° , respectively. To avoid instabilities in the integration process, the initial values of the relative joint velocities are considered $0^\circ/s$.

In Fig. 5, bodies 1, 2, and 4 are assumed as rectangular beams and bodies 3 and 5 are assumed as cuboid, whose lengths and masses are $L_1 = 5$ m, $L_2 = 1.2$ m, $L_3 = 0.3$ m, $L_4 = 1$ m, and $L_5 = 0.3$ m, and $m_1 = 25$ kg, $m_2 = 12$ kg, $m_3 = 3$ kg, $m_4 = 100$ kg, and $m_5 = 30$ kg, respectively. The locations of points *I*, *K*, and *Q* in the inertial reference frame are $[0, 5, 0]^T$ m, $[0, 3, 0]^T$ m, and $[-0.96, 3, 0]^T$ m, respectively. Points *N* and *L* are

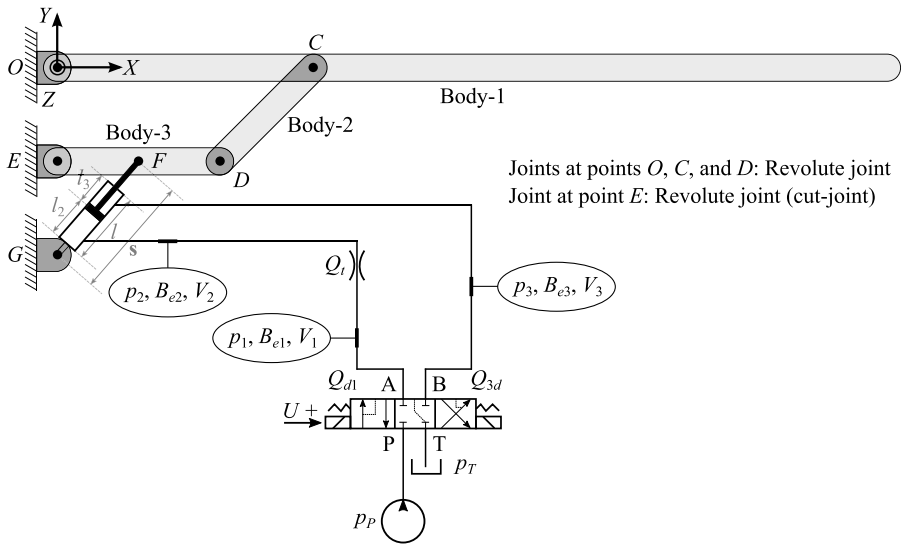


Fig. 4 A four-bar mechanism actuated by a double-acting hydraulic cylinder

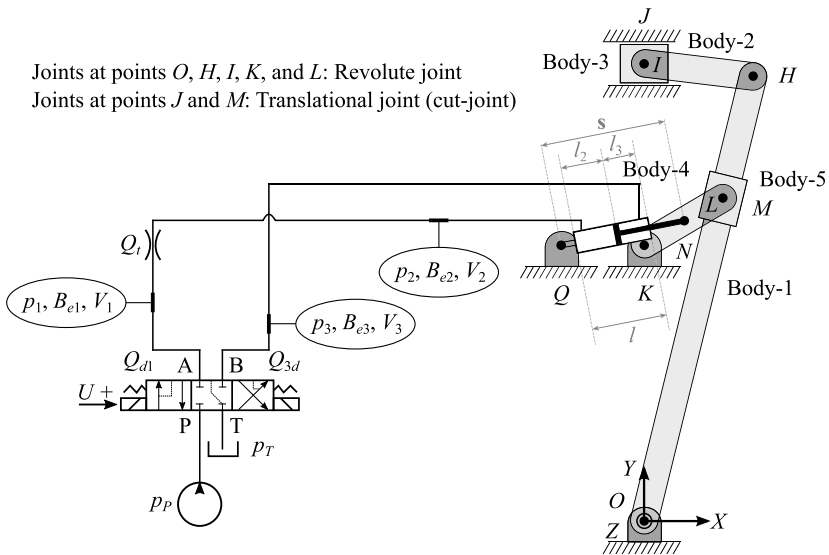


Fig. 5 A quick-return mechanism actuated by a double-acting hydraulic cylinder

located at the center of mass of bodies 4 and 5, respectively. In Fig. 5, the relative joint coordinates at points O , H , I , K , and L are respectively represented by z_1, z_2, z_3, z_4 , and z_5 that define the orientation of the respective bodies in the inertial reference frame, XYZ . Their initial values are considered as $76.22^\circ, 96.89^\circ, 6.89^\circ, 30.59^\circ$, and 45.63° , respectively. The mass moment of inertia of a rectangular beam and a cuboid are considered as $\frac{mL^2}{12}$ and $\frac{mL^2}{6}$,

respectively, where m is the mass and L is the length. The gravity is assumed to act in the negative Y direction, whose value is $g = 9.81 \text{ m/s}^2$.

Both mechanisms are actuated by using the hydraulic actuators, as shown in Figs. 4 and 5. For simplicity, identical hydraulic actuators are used in both mechanisms. In this study, a simple hydraulic circuit is accounted, which consists of a pump, with a constant pressure source p_P , a directional control valve, with a control signal U , a throttle valve, a double-acting hydraulic cylinder, connecting hoses, and a tank, with a constant pressure p_T . The control volumes, V_1 , V_2 , and V_3 , used in the modeling of the hydraulic circuit are marked in Fig. 4. The pressure in the respective control volumes are p_1 , p_2 , and p_3 ; and their respective effective bulk modulus are B_{e1} , B_{e2} , and B_{e3} , that are calculated by using Eq. (23). For simplicity, the hydraulic circuit is assumed ideal, that is, the leakage is neglected.

In the hydraulic subsystem, the control volumes V_1 , V_2 , and V_3 are calculated as

$$\left. \begin{aligned} V_1 &= V_{h_1} \\ V_2 &= V_{h_2} + A_2 l_2 \\ V_3 &= V_{h_3} + A_3 l_3 \end{aligned} \right\}, \tag{42}$$

where V_{h_1} , V_{h_2} , and V_{h_3} are the volumes of the respective hoses; A_2 and A_3 are, respectively, the areas of the piston side and the piston-rod side within a cylinder; and l_2 and l_3 are, respectively, the lengths of the chambers of the cylinder, piston and piston-rod side. The length of the hydraulic cylinder is l such that $l_2 + l_3 = l$; and the variable chamber lengths, l_2 and l_3 , are calculated as

$$\left. \begin{aligned} l_2 &= l_{2_0} + s_0 - |s| \\ l_3 &= l_{3_0} - s_0 + |s| \end{aligned} \right\}, \tag{43}$$

where $|s|$ is the actuator length of the hydraulic cylinder (see Figs. 4 and 5); s_0 is the actuator length at $t = 0$; and $l_{2_0} = s_0 - l$ and $l_{3_0} = l - l_{2_0}$ are, respectively, the length of the piston and piston-rod side of the cylinder at $t = 0$. The differential equations of the pressures p_1 , p_2 , and p_3 are computed based on Eq. (22) as

$$\left. \begin{aligned} \dot{p}_1 &= \frac{B_{e1}}{V_1} (Q_{d1} - Q_t) \\ \dot{p}_2 &= \frac{B_{e2}}{V_2} (Q_t - A_2 \dot{s}) \\ \dot{p}_3 &= \frac{B_{e3}}{V_3} (A_3 \dot{s} - Q_{3d}) \end{aligned} \right\}, \tag{44}$$

where the volume flow rates Q_{d1} and Q_{3d} are calculated from Eq. (26), the volume flow rate Q_t is calculated from Eq. (24), and \dot{s} is the actuator velocity. The actuator length $|s|$ and actuator velocity \dot{s} of the hydraulic cylinder are computed as a function of the relative joint coordinates. For example, in Fig. 4, the upper end of the hydraulic cylinder is attached to body-3 at point F , while, the lower end is attached to ground at point G , such that $\mathbf{r}_G = [0, -2, 0]^T \text{ m}$. Therefore, \mathbf{s} and \dot{s} for the four-bar mechanism can be computed as

$$\left. \begin{aligned} \mathbf{s} &= \mathbf{r}_F - \mathbf{r}_G \\ \dot{s} &= \frac{d|\mathbf{s}|}{dt} = \dot{\mathbf{s}} \cdot \frac{\mathbf{s}}{|\mathbf{s}|} = \dot{\mathbf{r}}_F \cdot \frac{\mathbf{s}}{|\mathbf{s}|} \end{aligned} \right\}, \tag{45}$$

Table 1 Parameters of the hydraulic circuit

Parameter	Symbol	Value
Pressure of the pump	p_P	7.6 MPa
Pressure of the tank (atmospheric pressure)	p_T	0.1 MPa
Semiempirical flow rate constant of the directional control valve	C_{vd}	$2.138 \times 10^{-8} \text{ m}^3/\text{s} \sqrt{\text{Pa}}$
Volume of the hose (section-1)	V_{h1}	$4.71 \times 10^{-5} \text{ m}^3$
Volume of the hose (section-2)	V_{h2}	$3.14 \times 10^{-5} \text{ m}^3$
Volume of the hose (section-3)	V_{h3}	$7.85 \times 10^{-5} \text{ m}^3$
Area of the throttle valve	A_t	$2.83 \times 10^{-5} \text{ m}^2$
Flow discharge coefficient of the throttle valve	C_d	0.8
Density of the oil	ρ	850 kg/m ³
Bulk modulus of the hoses	B_h	550 MPa
Bulk modulus of the oil	B_{oil}	1500 MPa
Bulk modulus of the hydraulic cylinder	B_c	31500 MPa
Efficiency of the cylinder	η	0.88
Diameter of the piston	d_2	80 mm
Diameter of the piston-rod	d_3	35 mm
Length of the cylinder/piston	l	1.1 m
Initial actuator length	s_0	$\sqrt{2}$ m

where the position \mathbf{r}_F and velocity $\dot{\mathbf{r}}_F$ are calculated by applying the classical kinematic relations as in [11, 22]. For simplicity, the force F_s produced by the hydraulic cylinder in Eq. (29) is expressed in the form of Eq. (10) as $\mathbf{F}_s = \left[\frac{s_X}{|s|} F_s, \frac{s_Y}{|s|} F_s, \frac{s_Z}{|s|} F_s \right]^T$, where s_X , s_Y , and s_Z are the components of vector \mathbf{s} along the axes of the inertial reference frame. The initial value of the force F_{s_0} produced by the hydraulic cylinder is calculated from the static configurations, as shown in Figs. 4 and 5. For example, in case of four-bar mechanism, $F_{s_0} = \sqrt{2}g(3m_1 + 2m_2 + m_3)$. In the static configuration, the initial value of pressure p_1 is equal to the initial value of pressure p_2 , which can be calculated based on Eq. (29) as $p_2 = (F_{s_0} + p_3 A_3) / A_2$. Note that the friction is neglected in static configuration and the initial value of pressure p_3 is assumed 3.5 MPa. The directional control valve, parameter U in Eq. (27), controls the movement of the cylinder through volume flows and is actuated for 10 s by using the following reference voltage signal U_{ref} as

$$\text{(Four-bar mechanism)} \quad U_{ref} = \begin{cases} 0, & t < 1 \text{ s}, \quad 2.5 \text{ s} \leq t < 5 \text{ s}, \quad t \leq 10, \\ 10, & 1 \text{ s} \leq t < 2.5 \text{ s}, \\ -10, & 5 \text{ s} \leq t < 8 \text{ s}, \end{cases} \quad (46)$$

$$\text{(Quick-return mechanism)} \quad U_{ref} = \begin{cases} 0, & t < 1 \text{ s}, \quad 4 \text{ s} \leq t < 6.5 \text{ s}, \quad t \leq 10, \\ -10, & 1 \text{ s} \leq t < 4 \text{ s}, \\ 10, & 6.5 \text{ s} \leq t < 8 \text{ s}, \end{cases} \quad (47)$$

where t is the simulation run time. The parameters of the hydraulic circuit are shown in Table 1.

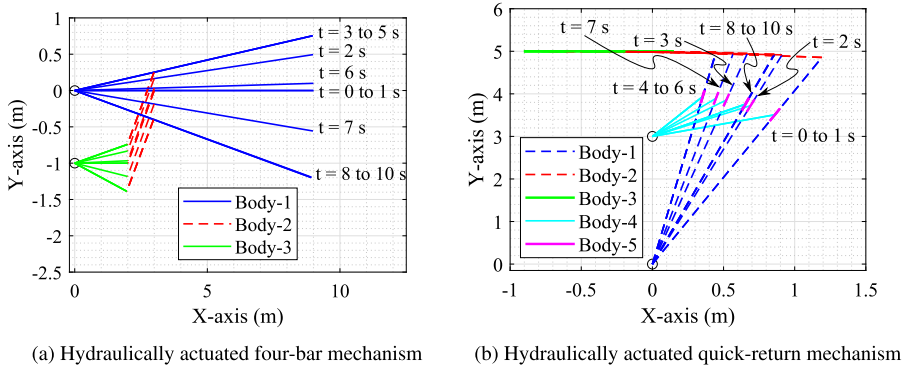


Fig. 6 The positions of the mechanisms at every second during the simulation run

The set of variables used to solve the combined system of equations with the proposed integration scheme (Sect. 4) are

$$\left. \begin{aligned}
 \mathbf{x} &= \left[(\mathbf{z}^i)^T, \mathbf{p}^T \right]^T \text{ (double-step semirecursive approach)} \\
 \mathbf{x} &= \left[\mathbf{z}^T, \mathbf{p}^T \right]^T \text{ (penalty-based semirecursive approach)}
 \end{aligned} \right\}. \tag{48}$$

Note that, in the double-step semirecursive approach (Eq. (48)), \mathbf{z}^i is identified by using the Gaussian elimination with full pivoting to the Jacobian matrix $\Phi_{\mathbf{z}}$. In this study, the error tolerance is considered as 1×10^{-7} rad for positions and 1×10^{-2} Pa for pressures. The voltage that corresponds to the spool position is integrated by using the trapezoidal rule and its error tolerance is considered as 1×10^{-7} V. Furthermore, the penalty factor α in Eq. (18) is considered as 1×10^{11} . Note that in the penalty-based semirecursive approach, the penalty term is analogous to a spring constant by considering that there is a spring attached to the cut-joint location to fulfill the constraints. Due to the high numerical stiffness introduced by the hydraulics, a high penalty term (spring constant) is used. In this study, both the approaches are implemented in the Matlab environment.

6 Results and discussion

This section presents the simulation results of the hydraulically actuated four-bar and quick-return mechanisms presented in the previous section. Figures 6a and 6b show the simulation frames of the four-bar and quick-return mechanisms, respectively, presenting the position of the bodies at different instants of time. Here, the two approaches, namely, the double-step semirecursive approach and the penalty-based semirecursive approach, are compared based on the simulation work cycle, energy balance, constraint violation, and numerical efficiency.

6.1 The work cycle

In the four-bar mechanism, the hydraulic cylinder lifts the structure between 1–2.5 s and lowers it down between 5–8 s. Whereas in the quick-return mechanism, the hydraulic cylinder pulls the structure between 1–4 s and pushes it between 6.5–8 s. In the subsequent

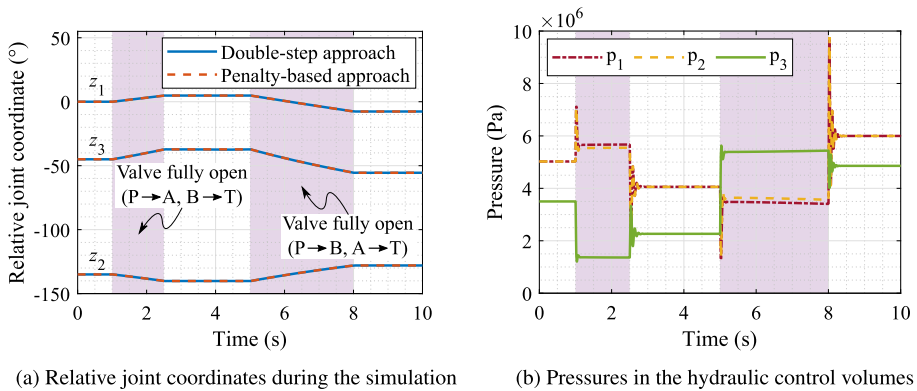


Fig. 7 The relative joint coordinates and pressures for the four-bar mechanism

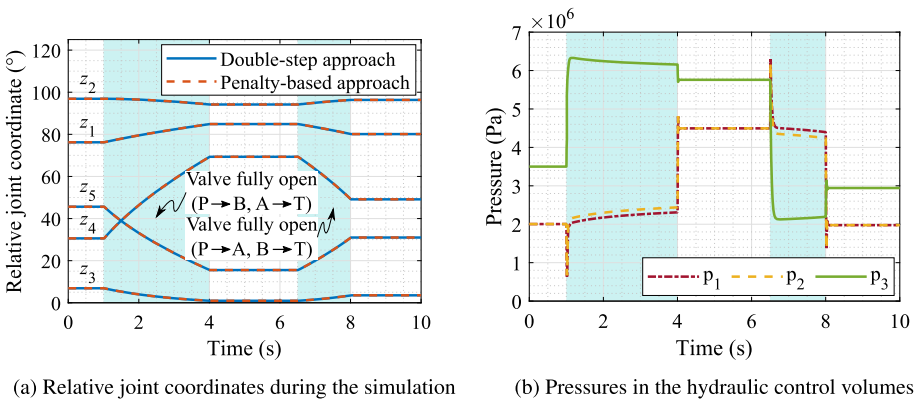


Fig. 8 The relative joint coordinates and pressures for the quick-return mechanism

plots, the regions between the opening and closing of the directional control valve are highlighted in purple for the four-bar mechanism and in cyan for the quick-return mechanism. For both the mechanisms, the relative joint coordinates showed a good agreement in the two approaches, as shown in Figs. 7a and 8a. Thus, the solutions of both the approaches are accurate with respect to each other. The pressures in the hydraulic control volumes are shown in Figs. 7b and 8b, respectively, for the four-bar and quick-return mechanisms. Note that the pressures are identical in both the approaches and the choice of the multibody formulation does not affect the results of the hydraulics.

In the double-step approach, the independent joint coordinate is identified by using the Gaussian elimination with full pivoting on the Jacobian matrix of the constraints. For the presented work cycle, the independent coordinates are identified as z_3 for the four-bar mechanism and as z_4 for the quick-return mechanism, throughout the simulation. In the double-step approach, if the independent and dependent coordinates are not identified adequately, then it can lead to the numerical problems during the integration because of the poor conditioned matrices. Therefore, the adequate identification of the independent and dependent coordinates is considered a relative drawback of the double-step approach compared with the penalty-based approach because the latter uses the full set of coordinates.

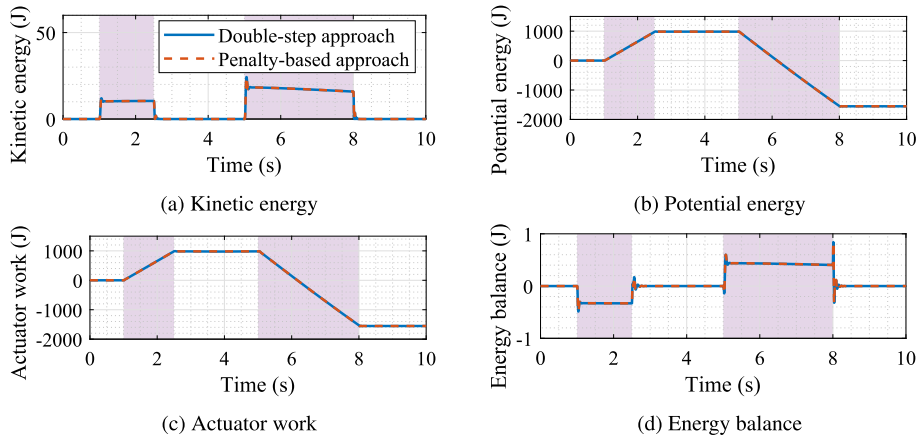


Fig. 9 A comparison of energies for the four-bar mechanism with a time-step of 1 ms

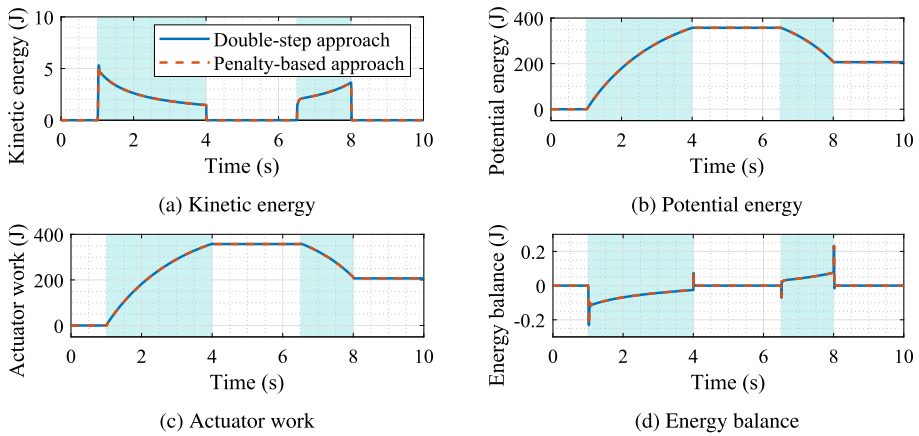


Fig. 10 A comparison of energies for the quick-return mechanism with a time-step of 1 ms

6.2 Energy balance

The energy balances in both the approaches are compared by analyzing the kinetic energy, potential energy, and work done by the actuator. The energy comparison of the four-bar mechanism is shown in Fig. 9 and of the quick-return mechanism is shown in Fig. 10. The energy balance in the double-step and penalty-based approaches showed a good agreement with each other for both the mechanisms. The peak energy drift in case of the four-bar mechanism is 0.84 J, which is 0.09% of the maximum actuator work, 982.95 J, and it occurred at the closing of the valve around 8 s. Whereas in case of the quick-return mechanism, the peak energy drift is 0.23 J, that is, 0.06% of the maximum actuator work, 357.47 J, and it also occurred at the closing of the valve around 8 s. An analogy can be drawn between the hydraulic system and a stiff spring, which supports the structure of the mechanisms.

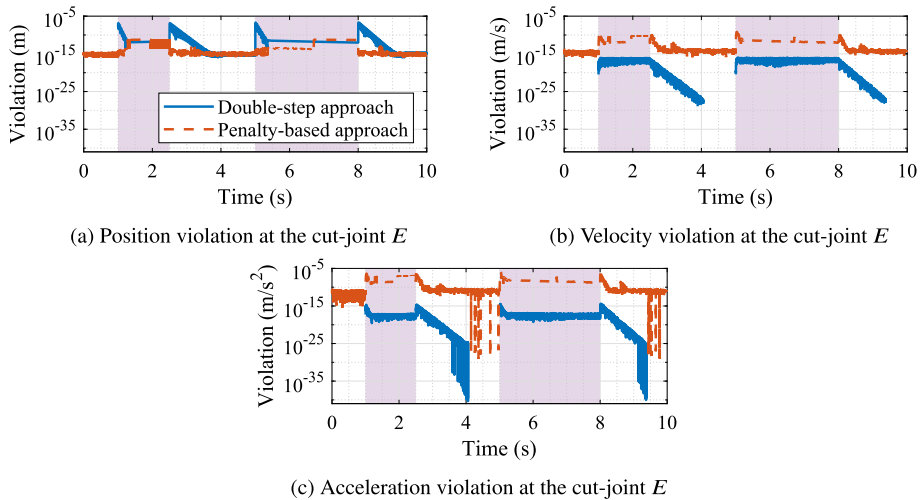


Fig. 11 Magnitude of constraint violations for the four-bar mechanism with a time-step of 1 ms

6.3 Constraint violation

The basic difference between the double-step and penalty-based approaches is the way they handle constraints in their multibody formulation. Thus, it is worth showing a comparison of the constraint violations in both the approaches for the two mechanisms as shown in Figs. 11 and 12. Note that in the quick-return mechanism, two cut-joints are used in the modeling, but for demonstration purposes, the results for only one of the cut-joint are presented. In both the approaches, the constraints are fulfilled with good accuracy for both mechanisms. Thus, the robustness of the multibody formulations, as explained in the literature [10, 24], is preserved in the application of a monolithic simulation of coupled multibody and hydraulic systems. Furthermore, the double-step approach fulfills constraints to the level of machine precision, or the precision specified, and it maintains a good error control. Therefore, this can be considered as the relative advantage of the double-step approach compared with the penalty-based approach because the latter approach is relatively more relaxed on the fulfillment of constraints.

In the double-step approach, for a possible mapping between independent and dependent relative joint coordinates, there is an assumption that redundant constraints and singular configurations do not exist. This assumption is a relative disadvantage of the double-step approach compared with the penalty-based approach because the latter approach can handle redundant constraints and can deliver accurate solutions in the vicinity of singular configurations [15].

6.4 Numerical efficiency

The numerical efficiencies of both the approaches are compared in Figs. 13 and 14, for the four-bar and quick-return mechanisms, respectively. In the four-bar mechanism, the average and maximum iterations, and the total integration time for the double-step approach are 1.39, 3, and 27.45 s and for the penalty-based approach are 1.56, 4, 21.54 s. Whereas in the quick-return mechanism, the average and maximum iterations, and the total integration time for the double-step approach are 1.49, 3, and 29.03 s and for the penalty-based approach are 1.57, 6,

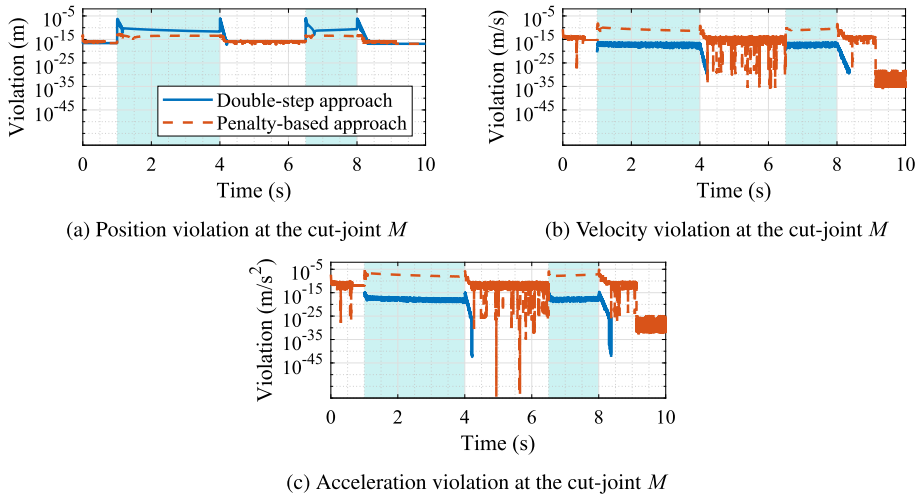


Fig. 12 Magnitude of constraint violations for the quick-return mechanism with a time-step of 1 ms

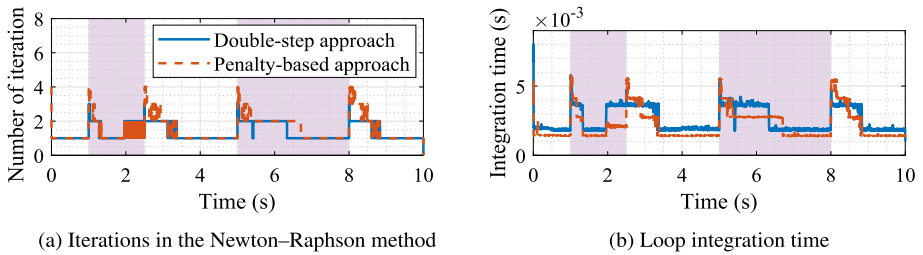


Fig. 13 A comparison of numerical efficiencies for the four-bar mechanism with a time-step of 1 ms

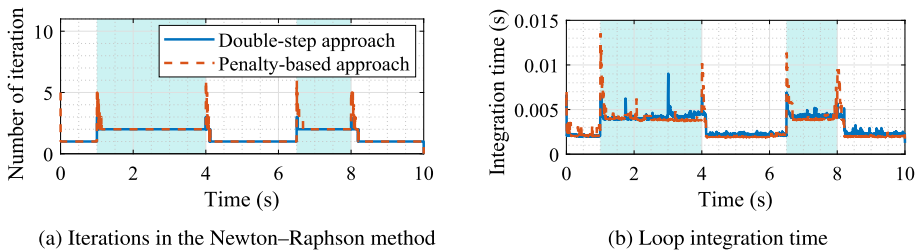


Fig. 14 A comparison of numerical efficiencies for the quick-return mechanism with a time-step of 1 ms

and 28.47 s. The maximum number of iterations occurred during the opening and closing of the directional control valve. Even though the number of iterations are lower in the double-step approach, its integration time is greater compared with the penalty-based approach, which is in accordance with the literature [15, 27]. The poor numerical efficiency of the double-step approach is attributed to the iterative solution of the dependent joint coordinates by using the Newton–Raphson method.

Note that the examples are implemented in the Matlab environment and for this reason, the result about the integration time may be unreliable because of the implementation details. However, both approaches are carefully implemented such that they have the same number of function calls in each iteration. This was achieved by symbolic precomputation and function generation of the matrix product, as in [32]. Furthermore, both approaches can be potential candidates for real-time applications by implementing them in a lower level language such as C++ or Fortran and by limiting the number of iterations to a lower value.

7 Conclusion

This study introduced the double-step semirecursive formulation and compared it with the penalty-based semirecursive formulation, in a numerically stiff coupled environment. The numerical stiffness was introduced by a hydraulic system. A monolithic scheme for the coupled simulation of the double-step semirecursive formulation and hydraulic systems was introduced in this study. To this end, the hydraulic system was described by using the lumped fluid theory. As case studies, hydraulically actuated four-bar and quick return mechanisms were modeled to compare the double-step semirecursive approach with the penalty-based semirecursive approach. The two approaches were compared based on the work cycle, energy balance, constraint violation, and numerical efficiency of the mechanisms, and similar results were obtained in both the mechanisms.

The relative joint coordinates and energy balances showed a good agreement in both the approaches. In the double-step approach, the adequate identification of the independent and dependent joint coordinates was carried out by using the Gaussian elimination with full pivoting on the Jacobian matrix of the constraints. Its identification is considered a relative drawback compared with the penalty-based approach where the full set of coordinates were used. In both the approaches, the constraints were fulfilled with good accuracy. However, the double-step approach had an advantage of fulfilling the constraints to the level of machine precision, or the precision specified compared with the penalty-based approach. Nevertheless, the double-step approach had an assumption that the redundant constraints and singularity configuration do not exist. This is a relative disadvantage of the double-step approach because the penalty-based approach can handle redundant constraints and can provide accurate solutions in the vicinity of singular configurations. Furthermore, the double-step approach suffered from poor numerical efficiency, which was attributed to the iterative solution of the dependent joint coordinates by using the Newton–Raphson method. To improve the numerical efficiency of the double-step approach, it would be of utmost importance to enhance the iterative solution of dependent coordinates and this is left as a topic for future studies. In conclusion, the penalty-based semirecursive approach has a number of advantages over the double-step semirecursive approach, which is in accordance with the literature [15, 27].

For future studies, alternate multibody formulations can be coupled with hydraulic systems to study on the optimal formulation for simulating coupled multibody and hydraulic systems. Note that the selection of the integrator type is bound to have an effect on the results. In this study, an implicit integrator is used, whereas the double-step semirecursive formulation is usually integrated by using the fourth order Runge–Kutta method, that is, an explicit integrator. Therefore, different integrators can be implemented to study on the optimal integrator choice for such coupled systems. This study utilized the Matlab environment, therefore, to make a firm conclusion on the numerical efficiency of the two approaches, a large-scale, three-dimensional example needs to be investigated in programming languages, such as C++ or Fortran.

Acknowledgements This work was supported in part by the Business Finland [project: Service Business from Physics Based Digital Twins—DigiBuzz], and in part by the Academy of Finland under Grant #316106.

Funding Note Open Access funding provided by LUT University (previously Lappeenranta University of Technology (LUT)).

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bayo, E., Jalon, J.G.D., Avello, A., Cuadrado, J.: An efficient computational method for real time multibody dynamic simulation in fully cartesian coordinates. *Comput. Methods Appl. Mech. Eng.* **92**(3), 377–395 (1991)
2. Bayo, E., Jalon, J.G.D., Serna, M.A.: A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. *Comput. Methods Appl. Mech. Eng.* **71**(2), 183–195 (1988)
3. Bayo, E., Ledesma, R.: Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. *Nonlinear Dyn.* **9**(1–2), 113–130 (1996)
4. Benedikt, M., Holzinger, F.R.: 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, Montpellier, France*, pp. 1–7 (2016)
5. Brenan, K.E., Campbell, S.L., Petzold, L.R.: *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, vol. 14. SIAM, Philadelphia (1996)
6. Callejo, A., Narayanan, S.H.K., Jalon, J.G.D., Norris, B.: Performance of automatic differentiation tools in the dynamic simulation of multibody systems. *Adv. Eng. Softw.* **73**, 35–44 (2014)
7. Callejo, A., Pan, Y., Ricón, J.L., Kövecses, J., Jalon, J.G.D.: Comparison of semi-recursive and subsystem synthesis algorithms for the efficient simulation of multibody systems. *J. Comput. Nonlinear Dyn.* **12**(1), 0111020(1) (2017)
8. 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)
9. 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)
10. Cuadrado, J., Dopico, D., Gonzalez, M., Naya, M.A.: A combined penalty and recursive real-time formulation for multibody dynamics. *J. Mech. Des.* **126**(4), 602–608 (2004)
11. Cuadrado, J., Dopico, D., Naya, M.A., Gonzalez, M.: Real-time multibody dynamics and applications. In: *Simulation Techniques for Applied Dynamics*, vol. 507, pp. 247–311. Springer, Berlin (2008)
12. Docquier, N., Poncelet, A., Delannoy, M., Fiset, P.: Multiphysics modelling of multibody systems: application to car semi-active suspensions. *Veh. Syst. Dyn.* **48**(12), 1439–1460 (2010)
13. Featherstone, R.: The calculation of robot dynamics using articulated-body inertias. *Int. J. Robot. Res.* **2**(1), 13–30 (1983)
14. Fiset, P., Vaneghem, B.: Numerical integration of multibody system dynamic equations using the coordinate partitioning method in an implicit Newmark scheme. *Comput. Methods Appl. Mech. Eng.* **135**(1–2), 85–105 (1996)
15. Flores, P., Ambrosio, J., Claro, J.P., Lankarani, H.M.: *Kinematics and Dynamics of Multibody Systems with Imperfect Joints: Models and Case Studies*, vol. 34. Springer, Berlin (2008)

16. Gomes, C., Thule, C., Broman, D., Larsen, P.G., Vangheluwe, H.: Co-simulation: a survey. *ACM Comput. Surv.* **51**(3), 1–33 (2018)
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)
18. González, F., Naya, M.Á., Luaces, A., González, M.: On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. *Multibody Syst. Dyn.* **25**(4), 461–483 (2011)
19. Handroos, H.M., Vilenius, M.J.: Flexible semi-empirical models for hydraulic flow control valves. *J. Mech. Des.* **113**(3), 232–238 (1991)
20. Hidalgo, A.F., Jalon, J.G.D.: Real-time dynamic simulations of large road vehicles using dense, sparse, and parallelization techniques. *J. Comput. Nonlinear Dyn.* **10**(3), 031005 (2015)
21. Higham, N.J.: Gaussian elimination. *Wiley Interdiscip. Rev.: Comput. Stat.* **3**(3), 230–238 (2011)
22. Jaiswal, S., Islam, M., Hannola, L., Sopenan, J., Mikkola, A.: Gamification procedure based on real-time multibody simulation. *Int. Rev. Model. Simul.* **11**(5), 259–266 (2018)
23. Jaiswal, S., Korkealaakso, P., Åman, R., Sopenan, J., Mikkola, A.: Deformable terrain model for the real-time multibody simulation of a tractor with a hydraulically driven front-loader. *IEEE Access* **7**, 172694–172708 (2019)
24. Jalon, J.G.D., Alvarez, E., Ribera, F.A.D., Rodríguez, I., Funes, F.J.: A fast and simple semi-recursive formulation for multi-rigid-body systems. *Comput. Methods Appl. Sci.* **2**, 1–23 (2005)
25. Jalon, J.G.D., Bayo, E.: *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*. Springer, Berlin (1994)
26. Linge, S., Langtangen, H.P.: *Programming for Computations-MATLAB/Octave: A Gentle Introduction to Numerical Simulations with MATLAB/Octave*, vol. 14. Springer, Berlin (2016)
27. Matikainen, M.K., Von Hertzen, R., Mikkola, A., Gerstmayr, J.: Elimination of high frequencies in the absolute nodal coordinate formulation. *Proc. Inst. Mech. Eng., Proc., Part K, J. Multi-Body Dyn.* **224**(1), 103–116 (2010)
28. 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)
29. Pan, Y., He, Y., Mikkola, A.: Accurate real-time truck simulation via semi-recursive formulation and Adams–Bashforth–Moulton algorithm. *Acta Mech. Sin.* **35**(3), 641–652 (2019)
30. Rahikainen, J., González, F., Naya, M.Á.: An automated methodology to select functional co-simulation configurations. *Multibody Syst. Dyn.* **48**(1), 79–103 (2020)
31. Rahikainen, J., Kiani, M., Sopenan, J., Jalali, P., Mikkola, A.: Computationally efficient approach for simulation of multibody and hydraulic dynamics. *Mech. Mach. Theory* **130**, 435–446 (2018)
32. Rahikainen, J., Mikkola, A., Sopenan, 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)
33. Rodríguez, J.I., Jiménez, J.M., Funes, F.J., Jalon, J.G.D.: Recursive and residual algorithms for the efficient numerical integration of multibody systems. *Multibody Syst. Dyn.* **11**(4), 295–320 (2004)
34. 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)
35. Sadjina, S., Pedersen, E.: Energy conservation and coupling error reduction in non-iterative co-simulations. *Eng. Comput.* **36**, 1579–1587 (2020)
36. Schweiger, G., Gomes, C., Engel, G., Hafner, I., Schoeegl, J., Posch, A., Nouidui, T.: An empirical survey on co-simulation: promising standards, challenges and research needs. *Simul. Model. Pract. Theory* **95**, 148–163 (2019)
37. Watton, J.: *Fluid Power Systems: Modeling, Simulation, Analog, and Microcomputer Control*. Prentice Hall, New York (1989)