

A parallel Hamiltonian formulation for forward dynamics of closed-loop multibody systems

K. Chadaj¹  · P. Malczyk¹ · J. Frączek¹

Received: 19 July 2015 / Accepted: 21 July 2016 / Published online: 29 August 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract This paper presents a novel recursive divide-and-conquer formulation for the simulation of complex constrained multibody system dynamics based on Hamilton’s canonical equations (HDCA). The systems under consideration are subjected to holonomic, independent constraints and may include serial chains, tree chains, or closed-loop topologies. Although Hamilton’s canonical equations exhibit many advantageous features compared to their acceleration based counterparts, it appears that there is a lack of dedicated parallel algorithms for multi-rigid-body system dynamics based on the Hamiltonian formulation. The developed HDCA formulation leads to a two-stage procedure. In the first phase, the approach utilizes the divide and conquer scheme, i.e., a hierarchic assembly–disassembly process to traverse the multibody system topology in a binary tree manner. The purpose of this step is to evaluate the joint velocities and constraint force impulses. The process exhibits linear $O(n)$ (n – number of bodies) and logarithmic $O(\log_2 n)$ numerical cost, in serial and parallel implementations, respectively. The time derivatives of the total momenta are directly evaluated in the second parallelizable step of the algorithm. Sample closed-loop test cases indicate very small constraint violation errors at the position and velocity level as well as marginal energy drift without any additional form of constraint stabilization techniques involved in the solution process. The results are comparatively set against more standard acceleration based Featherstone’s DCA approach to indicate the performance of the HDCA algorithm.

Keywords Forward dynamics · Closed-loop systems · Hamilton’s canonical equations · Parallel algorithm · Divide and conquer · Joint coordinates

✉ K. Chadaj
kchadaj@meil.pw.edu.pl

P. Malczyk
pmalczyk@meil.pw.edu.pl

J. Frączek
jfraczek@meil.pw.edu.pl

¹ Division of Theory of Machines and Robots, Institute of Aeronautics and Applied Mechanics, Faculty of Power and Aeronautical Engineering, Warsaw University of Technology, Nowowiejska 24, 00-665 Warsaw, Poland

1 Introduction

In current research and industrial applications, there is a need for fast multibody (MBS) dynamics simulations. One can encounter applications like that in such areas as robotics, biomechanics, and automotive industry. The other places where one needs efficient MBS simulations can be found in various interdisciplinary applications like, e.g., molecular dynamics simulations [1, 2] or granular media physics [3]. The applications are more complex and demanding. They may come from different domains and more often require the exploitation of coupling techniques [4]. In this context, numerical formulations for MBS should provide stable, accurate and reliable results. This short literature review is mainly focused on efficient sequential and parallel algorithms for multi-rigid body dynamics. Also the Hamiltonian based formulations are recalled from the MBS simulations point of view.

Due to the advances in the robotics field, there has been a growing attention to the development of efficient, low order algorithms for the simulation of open-loop and closed-loop kinematic chain system dynamics [5–7]. It is especially worth noticing the analogies between the optimal filtering theory and robot dynamics [8] that gave foundations for various mass matrix factorizations of a manipulator. The idea of articulated body inertias (ABI) for solving the equations of motion for an n -link manipulator, originally introduced by Featherstone [9], expanded into various forms and flavors. A summary of ABI-like algorithms can be found in [10]. There are also recent advances in the field which can be found in different papers, e.g., [11–13]. The mature technology associated with recursive formulations gave basis for further development of efficient low order parallel algorithms.

With the advances in parallel computer architectures, researchers paid more attention to the development of parallel algorithms for multibody dynamics. This trend can be observed in early works in this field, e.g., [14–16]. In 1999, Featherstone [17] developed truly optimal-time, logarithmic order divide-and-conquer algorithm (DCA) for the dynamics of tree-like topologies as well as closed-loop multibody systems. Other authors [18] preferred the idea of a decomposition of multibody system into sub-chains. In [19], Critchley and Anderson explored the notion of recursive coordinate reduction. There are other works associated with parallel multibody dynamics simulations as well [3, 20, 21]. Recently, the divide-and-conquer based schemes [17] have attracted significant attention to the development of efficient parallel algorithms for large MBS, partially due to the fact that computationally powerful multicore processors or graphics processor units are cheaply available on the market. Various algorithms that are effectively based on the Featherstone's DCA are elaborated with myriad of extensions to, e.g., humanoid robot simulations [22], molecular dynamics simulations, real-time applications, discontinuous changes in the system topology [23], sensitivity analysis, constraint enforcement [24–27], nonholonomic constraints, flexible multibody systems, or generalized constraint treatment. Advances in the application of the DCA approach to large multibody system simulations can be found in [28].

On the other hand, Hamilton's canonical equations constitute an interesting alternative as a basis for elaborating low order formulations. As Hamilton's canonical equations are expressed in terms of velocities and momenta of the system and the possible constraints are usually imposed at the velocity level, many sources indicate profitable characteristics of the Hamiltonian approach from the numerical point of view. Various authors exploited this approach for real-time simulations of vehicles [29], in the analyses of MBS subjected to intermittent motion [30], and in the development of penalty methods for constrained mechanical system dynamics [31]. Other interesting Hamiltonian based low order algorithms can be found in [32], where a recursive $O(n)$ algorithm for multi-rigid-body dynamics has been presented. Initially, the formulations are elaborated for serial chains and subsequently

extended to deal with closed-loop topologies. Nevertheless, the approach is purely sequential since it is based on the idea of ABI. Surprisingly, the literature review reveals that it is difficult to find a fully parallel approach for multi-rigid body dynamics simulations based on the Hamiltonian approach.

In previous papers [33, 34], the authors proposed a Hamiltonian based divide-and-conquer algorithm (HDCA) for open-loop multi-rigid body dynamics. In this paper, the HDCA algorithm is significantly extended and generalized to deal with systems possessing closed-loop and coupled-loops topologies. The method solves Hamilton’s canonical equations for constrained multi-rigid-body system in a divide-and-conquer manner. While the state of the mechanical system is described in terms of joint coordinates and joint momenta, the absolute coordinates are used as well to clearly present the underlying relationships between the variables. The HDCA recursive procedure for closed-loop multibody systems allows one to evaluate joint velocities and time derivatives of joint momenta with the aid of velocity level constraint equations in an efficient and parallelizable manner.

This paper is divided into four sections. Following the Introduction with brief literature review, Sect. 2 introduces basic concepts and definitions exploited in the paper and demonstrates novel Hamiltonian based parallel formulation for multi-rigid-body system dynamics possessing closed-loop chains. Section 3 describes and discusses the numerical results that were gathered to indicate the accuracy and efficiency of the proposed formulation. Original contributions of the paper are summarized in the last section.

2 Algorithm formulation

2.1 Hamilton’s canonical equations

This section provides a detailed formulation of the HDCA algorithm for the dynamics of complex multibody systems by utilizing the Hamilton’s approach. Before we begin with the algorithm development, Hamilton’s canonical equations for constrained multi-rigid body systems are recalled in a brief manner. At first, the classical equations of motion in dependent canonical coordinates are formulated. Subsequently, more elaborated form of Hamilton’s equations is demonstrated through the exploitation of the augmented momenta.

The state of the multibody system having n degrees of freedom can be described by a set of first-order Hamilton’s differential equations in the form [35, 36]

$$\dot{\mathbf{q}}_n = \left(\frac{\partial \mathcal{H}(t, \mathbf{q}_n, \mathbf{p}_n)}{\partial \mathbf{p}_n} \right)^T, \tag{1}$$

$$\dot{\mathbf{p}}_n = - \left(\frac{\partial \mathcal{H}(t, \mathbf{q}_n, \mathbf{p}_n)}{\partial \mathbf{q}_n} \right)^T + \mathbf{Q}_n - \Phi_{\mathbf{q}}^T \lambda_m, \tag{2}$$

$$\Phi(t, \mathbf{q}_n) = \mathbf{0} \quad \Rightarrow \quad \dot{\Phi}(t, \mathbf{q}_n, \dot{\mathbf{q}}_n) = \Phi_{\mathbf{q}} \dot{\mathbf{q}}_n + \Phi_t = \mathbf{0}, \tag{3}$$

where $\mathbf{q}_n, \mathbf{p}_n$ are system’s canonical coordinates and momenta, \mathbf{Q}_n are external, applied and centrifugal forces, and Φ are m algebraic constraint equations, which have to be fulfilled and which are associated with the unknown Lagrange multipliers λ_m . The terms $\Phi_{\mathbf{q}}$ and Φ_t represent the partial derivatives of the constraint equations with respect to the generalized coordinates and time, respectively. The Hamiltonian function is the Legendre transformation of the Lagrangian function, and it is defined as

$$\mathcal{H}(t, \mathbf{q}_n, \mathbf{p}_n) = \dot{\mathbf{q}}_n^T \mathbf{p}_n - \mathcal{L}(t, \mathbf{q}_n, \dot{\mathbf{q}}_n). \tag{4}$$

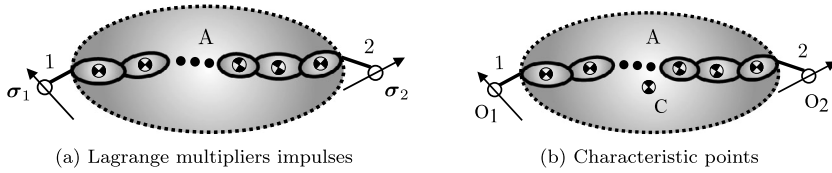


Fig. 1 A compound body *A* with two handles

We consider mechanical systems subjected to holonomic constraints which are assumed to be independent. The motion of the system is analyzed with respect to the inertial reference frame. Since the total kinetic energy T is a quadratic function of the generalized velocities, it can be expressed with the use of canonical momenta

$$T = \frac{1}{2} \dot{\mathbf{q}}_n^T \mathbf{M} \dot{\mathbf{q}}_n = \frac{1}{2} \dot{\mathbf{q}}_n^T \mathbf{p}_n, \tag{5}$$

where \mathbf{M} is the mass matrix of the system. As presented in [30], it is possible to reformulate Hamilton’s equations of motion to obtain a form which is more suitable for numerical implementations

$$\mathbf{P}_n = \mathbf{M} \dot{\mathbf{q}}_n - \Phi_{\mathbf{q}}^T \boldsymbol{\sigma}_m, \tag{6}$$

$$\dot{\mathbf{P}}_n = \mathbf{Q}_n - \dot{\Phi}_{\mathbf{q}}^T \boldsymbol{\sigma}_m, \tag{7}$$

where $\boldsymbol{\sigma}_m$ are the unknown constraint Lagrange multipliers impulses associated with the constraint equations at the velocity level, and \mathbf{P}_n are called augmented momenta vectors. One can combine (6) together with the velocity level constraint equations (3) to get a more concise matrix form

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_n \\ -\boldsymbol{\sigma}_m \end{bmatrix} = \begin{bmatrix} \mathbf{P}_n \\ -\Phi_t \end{bmatrix}. \tag{8}$$

Equations (7) and (8) describe the motion of a system through a set of $2n + m$ equations, which have to be solved and integrated at each time-step to determine the state of the system in the next time-instant. To be able to solve the differential–algebraic equations, one has to specify initial conditions for positions and momenta at joints, which have to be consistent with the imposed constraint equations. It is easier to specify initial conditions for position of a system. The initial conditions for momenta \mathbf{p}_n can be calculated using (6). In order to do that, one has to assume that at the initial time-step all constraint force impulses are known to be zero ($\boldsymbol{\sigma}_m = \mathbf{0}$), and no momentum is absorbed at constrained directions (see [30] for a more detailed discussion).

2.2 Joint velocities and constraint force impulses

Articulated momentum definition and the integral form of the equations of motion At an arbitrary point O , which is not coincident with a body’s center of mass C (see Fig. 1b), the spatial momentum vector for a general body can be defined in a matrix form as $\mathbf{P}_O \in \mathcal{R}^6$ [32]

$$\mathbf{P}_O = \mathbf{M}_O \mathbf{V}_O = \begin{bmatrix} m\mathbf{I} & -m\widetilde{\mathbf{l}}_{OC} \\ m\widetilde{\mathbf{l}}_{OC} & \mathbf{J}_O \end{bmatrix} \begin{bmatrix} \mathbf{v}_O \\ \boldsymbol{\omega} \end{bmatrix}. \tag{9}$$

All expressions in (9) are derived with respect to the local coordinate frame with the origin at point O and then expressed in the global reference frame. The term $\mathbf{M}_O \in \mathcal{R}^{6 \times 6}$ denotes a mass matrix of the body for a given vector \mathbf{l}_{OC} between point O and point C , while \mathbf{V}_O is a spatial velocity vector composed of translational \mathbf{v}_O and rotational $\boldsymbol{\omega}$ components. The term m is the mass of that body and \mathbf{J}_O is the moment of inertia with respect to the axis of local reference frame defined at point O and expressed in the global reference frame. The identity matrix is defined as $\mathbf{I} \in \mathcal{R}^3$ and the tilde symbol above a vector defines a skew-symmetric matrix associated with this vector. Let us consider a physical or compound rigid body A connected with the rest of the multibody system by arbitrary kinematic joints at the boundary points O_1 and O_2 as depicted in Fig. 1. From now on, let us consider such points as body's interfaces called handles, by means of which the body communicates with the rest of the system using both active and constraint force components. Particularly, the spatial articulated momentum vector $\bar{\mathbf{P}} \in \mathcal{R}^6$ of linear and angular momenta related to a handle can be defined as

$$\bar{\mathbf{P}} = \mathbf{D}\boldsymbol{\sigma} + \mathbf{H}\mathbf{p}, \tag{10}$$

where $\mathbf{H} \in \mathcal{R}^{6 \times n_f}$ represents the joint's motion subspace (n_f – number of degrees of freedom for the joint) and $\mathbf{D} \in \mathcal{R}^{6 \times (6-n_f)}$ is associated with the joint's constrained directions [24, 26]. We assume that both matrices \mathbf{H} and \mathbf{D} consist of ones and zeros. The following identities hold in the formulation: $\mathbf{H}^T \mathbf{H} = \mathbf{I}$, $\mathbf{D}^T \mathbf{D} = \mathbf{I}$. The matrices \mathbf{H} and \mathbf{D} are orthogonal to each other, which yields the condition

$$\mathbf{D}^T \mathbf{H} = \mathbf{0}. \tag{11}$$

Equation (10) assumes that the articulated momentum vector taken up from the system can be decomposed into the parts related to the active component $\mathbf{H}\mathbf{p}$ and constraint component $\mathbf{T} = \mathbf{D}\boldsymbol{\sigma}$. Vector $\mathbf{p} \in \mathcal{R}^{n_f \times 1}$ denotes the joint canonical momenta, whereas vector $\boldsymbol{\sigma} \in \mathcal{R}^{6-n_f}$ indicates constraint load impulses (Lagrange multipliers at the velocity level) at joints. The following condition holds for the Lagrange multipliers $\dot{\boldsymbol{\sigma}} = \boldsymbol{\lambda}$ [30]. For an exemplary body A (see Fig. 1) with two handles at points O_1 and O_2 , the integral form of the equations of motion can be formulated with the use of articulated momentum vectors as

$$\mathbf{M}_1^A \mathbf{V}_1^A = \bar{\mathbf{P}}_1^A + \mathbf{S}_{12}^A \bar{\mathbf{P}}_2^A \tag{12}$$

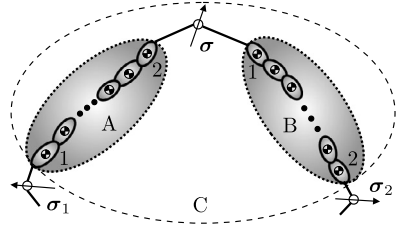
$$\mathbf{M}_2^A \mathbf{V}_2^A = \mathbf{S}_{21}^A \bar{\mathbf{P}}_1^A + \bar{\mathbf{P}}_2^A, \tag{13}$$

where the subscripts 1 or 2 indicate the handles O_1 and O_2 . The above equations make use of the shift matrices \mathbf{S}_{12}^A and \mathbf{S}_{21}^A , which are a handy tool for the transformation of six-dimensional momentum, force, and velocity vectors to other points of operation. If the position vector \mathbf{l}_{12} is measured from the point O_1 to O_2 , the shift matrix takes the form

$$\mathbf{S}_{12}^A = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \tilde{\mathbf{l}}_{12} & \mathbf{I} \end{bmatrix}_{6 \times 6}, \tag{14}$$

where $\mathbf{0} \in \mathcal{R}^{3 \times 3}$ is the null matrix of appropriate size and $\tilde{\mathbf{l}}_{12}$ is a skew-symmetric matrix. By performing minor modifications, (12)–(13) can be rearranged (by using (10)) into the more

Fig. 2 The assembly of bodies *A* and *B* into body *C*



favorable form

$$\mathbf{V}_1^A = \xi_{11}^A \mathbf{T}_1^A + \xi_{12}^A \mathbf{T}_2^A + \xi_{10}^A, \tag{15}$$

$$\mathbf{V}_2^A = \xi_{21}^A \mathbf{T}_1^A + \xi_{22}^A \mathbf{T}_2^A + \xi_{20}^A. \tag{16}$$

Equations (15)–(16) provide a direct relationship between the spatial velocity vectors $\mathbf{V}_1^A, \mathbf{V}_2^A$ of body *A* and constraint force impulses acting on the terminal bodies $\mathbf{T}_1^A, \mathbf{T}_2^A$. These equations form an integral form of the equations of motion for a general body *A* with two handles. The recursive assembly–disassembly procedure allows one to calculate all constraint force impulses and joint velocities in a divide-and-conquer manner. As a final remark, it is worth noticing that the matrix coefficients $\xi_{11}^A, \xi_{12}^A, \xi_{21}^A, \xi_{22}^A$ for the velocity level equations have the same form as the coefficients formulated in the original Featherstone’s paper (cp. [17]) for acceleration based equations. The coefficients include the system’s mass and inertia parameters. On the other hand, the quantities ξ_{10}^A, ξ_{20}^A are associated with joint’s canonical momenta and joint positions.

2.2.1 Assembly phase

This subsection formulates the integral form of the equations of motion for a compound body *C*. Let us define a set of bodies *A* and *B* interconnected by a kinematic joint as shown in Fig. 2. The integral form of the equations of motion for body *A* and body *B* can be written as

$$\mathbf{V}_1^A = \xi_{11}^A \mathbf{T}_1^A + \xi_{12}^A \mathbf{T}_2^A + \xi_{10}^A, \tag{17}$$

$$\mathbf{V}_2^A = \xi_{21}^A \mathbf{T}_1^A + \xi_{22}^A \mathbf{T}_2^A + \xi_{20}^A, \tag{18}$$

$$\mathbf{V}_1^B = \xi_{11}^B \mathbf{T}_1^B + \xi_{12}^B \mathbf{T}_2^B + \xi_{10}^B, \tag{19}$$

$$\mathbf{V}_2^B = \xi_{21}^B \mathbf{T}_1^B + \xi_{22}^B \mathbf{T}_2^B + \xi_{20}^B. \tag{20}$$

The goal of the hierarchic assembly procedure is to construct the integral form of the equations of motion for body *C* in the analogous form as in the case of body *A* and *B*, i.e.,

$$\mathbf{V}_1^C = \xi_{11}^C \mathbf{T}_1^C + \xi_{12}^C \mathbf{T}_2^C + \xi_{10}^C, \tag{21}$$

$$\mathbf{V}_2^C = \xi_{21}^C \mathbf{T}_1^C + \xi_{22}^C \mathbf{T}_2^C + \xi_{20}^C, \tag{22}$$

where the handles O_1 and O_2 of the compound body *C* are coincident with handle O_1 of body *A* and handle O_2 of body *B*, respectively. The same form of the equations enables us to utilize the divide-and-conquer paradigm to assembly the whole system into one super-body, whose velocities depend on constraint force impulses at the terminal handles. In other words,

the expressions $\xi_{11}^C, \xi_{12}^C, \dots, \xi_{20}^C$ for body C have to be evaluated in terms of the known quantities for bodies A and B . The procedure requires an intermediate calculations of the interbody force impulses \mathbf{T}_1^B and \mathbf{T}_2^A using the expressions for force impulses \mathbf{T}_1^A and \mathbf{T}_2^B . The presence of the interconnecting kinematic joint imposes the velocity level conditions given in the form

$$\mathbf{V}_1^B - \mathbf{V}_2^A = \mathbf{H}\dot{\mathbf{q}}, \tag{23}$$

where \mathbf{H} is the joint motion subspace that maps joint velocity vectors $\dot{\mathbf{q}}$ into the spatial velocity vectors. Let us substitute (18)–(19) into (23). By premultiplying the resulting conditions by \mathbf{D}^T , we obtain

$$\mathbf{D}^T (\xi_{11}^B \mathbf{T}_1^B + \xi_{12}^B \mathbf{T}_2^B + \xi_{10}^B - \xi_{21}^A \mathbf{T}_1^A - \xi_{22}^A \mathbf{T}_2^A - \xi_{20}^A) = \mathbf{0}. \tag{24}$$

Newton’s third law of motion has to be satisfied, thus the constraint force impulses have to satisfy the condition

$$\mathbf{T}_1^B = -\mathbf{T}_2^A = \mathbf{D}\boldsymbol{\sigma}, \tag{25}$$

which enables us to use relation (25) in (24) to express the Lagrange multipliers $\boldsymbol{\sigma}$ as a function of constraint force impulses \mathbf{T}_1^A and \mathbf{T}_2^B at the terminal bodies

$$\boldsymbol{\sigma} = -[\mathbf{D}^T (\xi_{11}^B + \xi_{22}^A)\mathbf{D}]^{-1} \mathbf{D}^T (\xi_{12}^B \mathbf{T}_2^B - \xi_{21}^A \mathbf{T}_1^A + \xi_{10}^B - \xi_{20}^A). \tag{26}$$

The inverse matrices appearing in (26) exist since both matrices ξ_{11}^B and ξ_{22}^A are symmetric and positive definite as long as the constraints imposed on the system are independent. To simplify the formulas, the following matrices are defined:

$$\mathbf{W} = \mathbf{DCD}^T = -\mathbf{D}[\mathbf{D}^T (\xi_{11}^B + \xi_{22}^A)\mathbf{D}]^{-1} \mathbf{D}^T, \tag{27}$$

$$\boldsymbol{\beta} = \mathbf{DCb} = \mathbf{DCD}^T (\xi_{10}^B - \xi_{20}^A). \tag{28}$$

Hence, (25) becomes

$$\mathbf{T}_1^B = \mathbf{W}\xi_{12}^B \mathbf{T}_2^B - \mathbf{W}\xi_{21}^A \mathbf{T}_1^A + \boldsymbol{\beta} = -\mathbf{T}_2^A, \tag{29}$$

and enables us the calculation of constraint force impulses at the interconnecting joint using constraint force impulses at handle O_1 of body A and handle O_2 of body B . The final step involves the substitution of (29) into (17) and (20)

$$\mathbf{V}_1^A = [\xi_{11}^A + \xi_{12}^A \mathbf{W}\xi_{21}^A] \mathbf{T}_1^A + [-\xi_{12}^A \mathbf{W}\xi_{12}^B] \mathbf{T}_2^B + [\xi_{10}^A - \xi_{12}^A \boldsymbol{\beta}], \tag{30}$$

$$\mathbf{V}_2^B = [-\xi_{21}^B \mathbf{W}\xi_{21}^A] \mathbf{T}_1^A + [\xi_{22}^B + \xi_{21}^B \mathbf{W}\xi_{12}^B] \mathbf{T}_2^B + [\xi_{20}^B + \xi_{21}^B \boldsymbol{\beta}]. \tag{31}$$

As the handles of the compound body C are coincident with the leftmost handle of body A , and with the rightmost handle of body B , respectively, the following equations are fulfilled:

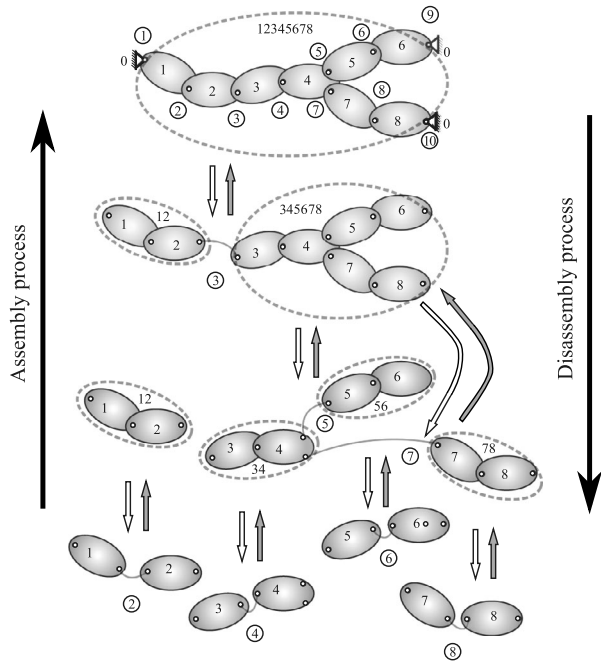
$$\mathbf{V}_1^C = \mathbf{V}_1^A \quad \text{and} \quad \mathbf{T}_1^C = \mathbf{T}_1^A, \tag{32}$$

$$\mathbf{V}_2^C = \mathbf{V}_2^B \quad \text{and} \quad \mathbf{T}_2^C = \mathbf{T}_2^B. \tag{33}$$

The comparison of (30)–(31) with (21)–(22) leads to the following recursive expressions:

$$\xi_{11}^C = \xi_{11}^A + \xi_{12}^A \mathbf{W}\xi_{21}^A, \quad \xi_{12}^C = -\xi_{12}^A \mathbf{W}\xi_{12}^B, \tag{34}$$

Fig. 3 Possible assembly–disassembly process for an exemplary multibody system with closed-loops. The process allows one to assemble the multibody system starting from individual bodies denoted by ellipses to build compound (or virtual) bodies. At the end of the assembly process, the whole MBS is constructed, which enables one to exploit the boundary conditions for the base body connections. In the disassembly phase, the process is reversed in order to decompose the system into individual bodies



$$\xi_{21}^C = -\xi_{21}^B \mathbf{W} \xi_{21}^A = (\xi_{12}^C)^T, \quad \xi_{22}^C = \xi_{22}^B + \xi_{21}^B \mathbf{W} \xi_{12}^B, \quad (35)$$

$$\xi_{10}^C = \xi_{10}^A - \xi_{12}^A \beta, \quad \xi_{20}^C = \xi_{20}^B + \xi_{21}^B \beta. \quad (36)$$

Equations (34)–(36) constitute the main formulas for the divide-and-conquer assembly–disassembly procedure, which enables us to construct the integral form of the equations of motion for the articulated body *C* consisting of bodies *A* and *B*. At this point it is possible to perform the hierarchic assembly of the multibody system based on the binary tree decomposition (see Fig. 3). The efficient evaluation of the binary tree associated with the multibody system is a separate task and it will not be considered in this paper. The step finishes when the whole multibody system is assembled into one super-body. The assembly phase is summarized in Pseudo-algorithm 1.

Base body connection Let us define the integral form of the equations of motion for the whole system represented by one super-body as

$$\mathbf{V}_1^C = \xi_{11}^C \mathbf{T}_1^C + \xi_{12}^C \mathbf{T}_2^C + \xi_{10}^C, \quad (37)$$

$$\mathbf{V}_2^C = \xi_{21}^C \mathbf{T}_1^C + \xi_{22}^C \mathbf{T}_2^C + \xi_{20}^C. \quad (38)$$

Constraint load impulses at point *O*₁ and point *O*₂ (provided that they exist) can be linked with the Lagrange multipliers σ_1 and σ_2 as

$$\mathbf{T}_1^C = \mathbf{D}_1 \sigma_1, \quad -\mathbf{T}_2^C = \mathbf{D}_2 \sigma_2. \quad (39)$$

On the other hand, the joint velocities can be expressed as

$$\dot{\mathbf{q}}_1 = \mathbf{H}_1^T \mathbf{V}_1^C, \quad -\dot{\mathbf{q}}_2 = \mathbf{H}_2^T \mathbf{V}_2^C. \quad (40)$$

Algorithm 1 Pseudo-algorithm for the assembly phase

Inputs:

- Coefficients ξ of bodies A and B ,
- External load vectors \mathbf{Q} of bodies A and B .

Procedure:

1. Compute matrices \mathbf{W} and β in (27) and (28).
2. Compute coefficients ξ for body C in (34)–(36).
3. Compute external load vectors \mathbf{Q}^C in (74).

Outputs:

- Coefficients ξ of body C ,
 - External load vectors \mathbf{Q} of body C .
-

At this point, three cases are possible:

- (a) The mechanism is free-floating. The boundary force impulses are equal to zero ($\mathbf{T}_1^C = \mathbf{0}$, $\mathbf{T}_2^C = \mathbf{0}$), and the expressions for the spatial velocities of terminal bodies simplify to

$$\mathbf{V}_1^C = \xi_{10}^C, \quad \mathbf{V}_2^C = \xi_{20}^C. \tag{41}$$

- (b) The mechanism has an open-loop or tree-like topology. The handle O_2 refers to the tip of the chain and the constraint force impulse \mathbf{T}_2^C is known to be zero. Thus, (37) can be reduced to the following form:

$$\mathbf{V}_1^C = \xi_{11}^C \mathbf{T}_1^C + \xi_{10}^C. \tag{42}$$

Let us premultiply (42) by \mathbf{D}_1^T . By using (39), one can resolve the relations for the unknown Lagrange multipliers σ_1 to get

$$\sigma_1 = -(\mathbf{D}_1^T \xi_{11}^C \mathbf{D}_1)^{-1} \mathbf{D}_1^T \xi_{10}^C = \mathbf{C}_1 \mathbf{D}_1^T \xi_{10}^C. \tag{43}$$

Since ξ_{11}^C remains symmetric and positive definite, there is no problem with finding the inverse. At this instant, the absolute velocity \mathbf{V}_1^C is determinable by using (42).

- (c) The mechanism has a closed-loop topology, and it is connected to the fixed base body by two kinematic joints at handle O_1 and handle O_2 . By inserting (39) into relations (37)–(38), we obtain an intermediate formula that can be premultiplied by \mathbf{D}_1^T and \mathbf{D}_2^T to get

$$\begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix}^T \begin{bmatrix} \xi_{11}^C & \xi_{12}^C \\ \xi_{21}^C & \xi_{22}^C \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ -\sigma_2 \end{bmatrix} + \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix}^T \begin{bmatrix} \xi_{10} \\ \xi_{20} \end{bmatrix} = \mathbf{0}. \tag{44}$$

Equations (44) enable us to calculate the unknown boundary Lagrange multipliers

$$\begin{bmatrix} \sigma_1 \\ -\sigma_2 \end{bmatrix} = -\left(\begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix}^T \begin{bmatrix} \xi_{11}^C & \xi_{12}^C \\ \xi_{21}^C & \xi_{22}^C \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix}^T \begin{bmatrix} \xi_{10} \\ \xi_{20} \end{bmatrix}. \tag{45}$$

Usually, the inverse matrices in (45) exist. However, there are some circumstances, in which the calculation of constraint load impulses is not so straightforward. The prob-

lems with matrix inversions may arise when the system passes near or through the singular configuration or/and the system is redundantly constrained. The redundancy results in problems of joint reactions solvability [37, 38]. Nevertheless, let us assume that the boundary loads impulses have been successfully evaluated at this phase. With the boundary Lagrange multipliers σ_1, σ_2 known, the velocities that connect the assembly with the fixed base-body can be computed as

$$\begin{bmatrix} \dot{\mathbf{q}}_1 \\ -\dot{\mathbf{q}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix}^T \begin{bmatrix} \xi_{11}^C & \xi_{12}^C \\ \xi_{21}^C & \xi_{22}^C \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ -\sigma_2 \end{bmatrix} + \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix}^T \begin{bmatrix} \xi_{10} \\ \xi_{20} \end{bmatrix}. \quad (46)$$

Analogously, the above scheme can be easily adapted to cope with more than two handles per body.

Disassembly phase As soon as the constraint force impulses \mathbf{T}_1^C and \mathbf{T}_2^C are available, the disassembly phase is started (cf. Fig. 3). The values of \mathbf{T}_1^C and \mathbf{T}_2^C can be transferred to the proceeding nodes of the binary tree to recursively compute all interconnecting constraint force impulses utilizing equation (29) and handles coincidence conditions (Eqs. (32)–(33)). When the interbody constraint force impulses are evaluated for all joints, the spatial velocities can be computed by using (15)–(16). Finally, it is possible to determine the joint velocities with the aid of projection onto the joints' motion subspaces (see Eq. (23)). The disassembly phase is summarized in Pseudo-algorithm 2.

Algorithm 2 Pseudo-algorithm for the disassembly phase

Inputs:

- Constraint force impulses \mathbf{T}^C of assembly C ,
- Articulated loads vectors \mathbf{Q}^C of assembly C .

Procedure:

1. Compute force impulses \mathbf{T}_1^A and \mathbf{T}_2^B in (32)–(33).
2. Compute force impulses \mathbf{T}_1^B and \mathbf{T}_2^A in (29).
3. Compute articulated loads $\overline{\mathbf{Q}}_1^A = \overline{\mathbf{Q}}_1^C$ and $\overline{\mathbf{Q}}_2^B = \overline{\mathbf{Q}}_2^C$ in (74)–(75).
4. Compute articulated loads $\overline{\mathbf{Q}}_1^B = \mathbf{Q}_1^B - \mathbf{S}_{12}^B \overline{\mathbf{Q}}_2^C = -\overline{\mathbf{Q}}_2^A$ in (76).

Outputs:

- Constraint force impulses \mathbf{T} of bodies A and B ,
 - Articulated loads vectors $\overline{\mathbf{Q}}$ of bodies A and B .
-

The computational steps described above are equivalent to the evaluation of the first set of Hamilton's canonical equations (1). The proposed method allows one to recursively calculate the derivatives of canonical coordinates without the need of determination of the system's Hamiltonian and its partial derivatives $\partial\mathcal{H}/\partial\mathbf{p}$.

2.3 Derivatives of canonical momenta

This part of the paper is devoted to the calculation of time derivatives of canonical momenta. Initially, the equations of motion for MBS are formulated in terms of absolute momenta.

Subsequently, the notion of articulated and accumulated momenta are exploited to form the equations of motion in a divide-and-conquer manner. Finally, the expressions for the time derivatives of canonical momenta are demonstrated.

Equations of motion for unconstrained body According to Newton’s law of motion, the derivative of the absolute momenta vector of unconstrained body is equal to the sum of external load vectors. The following equations of motion, expressed with respect to the centroidal reference frame, are initially used:

$$\dot{\mathbf{P}}_C = \mathbf{Q}_C. \tag{47}$$

To construct the equations of motion with respect to the local coordinate frame attached at point $O \neq C$, one has to define the shift matrix \mathbf{S}_{CO} as

$$\mathbf{P}_C = \mathbf{S}_{CO}\mathbf{P}_O, \quad \mathbf{Q}_C = \mathbf{S}_{CO}\mathbf{Q}_O. \tag{48}$$

The shift matrix \mathbf{S}_{CO} is defined analogously to (14). Let us note that the spatial momentum vectors transform in the same way as spatial force vectors. The following identities [10] are useful in further derivations:

$$(\mathbf{S}_{CO})^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \widetilde{\mathbf{I}}_{CO} & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\widetilde{\mathbf{I}}_{CO} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \widetilde{\mathbf{I}}_{OC} & \mathbf{I} \end{bmatrix} = \mathbf{S}_{OC}, \tag{49}$$

$$\mathbf{S}_{OC}\dot{\mathbf{S}}_{OC} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \widetilde{\mathbf{I}}_{OC} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \widetilde{\mathbf{I}}_{OC} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \widetilde{\mathbf{I}}_{OC} & \mathbf{0} \end{bmatrix} = \dot{\mathbf{S}}_{OC} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \widetilde{\mathbf{I}}_{OC} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \widetilde{\mathbf{I}}_{OC} & \mathbf{I} \end{bmatrix} = \dot{\mathbf{S}}_{OC}\mathbf{S}_{OC}. \tag{50}$$

Equations (47) and (48) can be combined to obtain the equation of motion with respect to the local coordinate frame whose origin is attached at point O

$$\dot{\mathbf{P}}_O + \dot{\mathbf{S}}_{CO}\mathbf{P}_O = \mathbf{Q}_O, \tag{51}$$

where \mathbf{Q}_O is a spatial external loads vector and \mathbf{P}_O is a spatial momentum vector related to the body at point O . A closer look at (51) reveals the fact that it is possible to conveniently simplify the expression $\dot{\mathbf{S}}_{CO}\mathbf{P}_O$ and receive

$$\dot{\mathbf{P}}_O + \dot{\mathbf{S}}_O\mathbf{P}_O = \mathbf{Q}_O, \tag{52}$$

where

$$\dot{\mathbf{S}}_O = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \widetilde{\mathbf{v}}_O & \mathbf{0} \end{bmatrix}_{6 \times 6}, \tag{53}$$

and $\dot{\mathbf{S}}_O$ consists of a skew-symmetric matrix of point O ’s translational velocity vector in inertial reference frame.

Equations of motion for constrained body Like in Sect. 2.2.1, let us consider two interacting bodies A and B (see Fig. 2). Let us assign again two handles per compound body by means of which the body interacts with other bodies in the system. The equations of motion

for such system can be written in the form

$$\dot{\mathbf{P}}_1^A + \dot{\mathbf{S}}_1^A \mathbf{P}_1^A = \mathbf{Q}_1^A + \mathbf{F}_1^A + \mathbf{S}_{12}^A \mathbf{F}_2^A, \tag{54}$$

$$\dot{\mathbf{P}}_1^B + \dot{\mathbf{S}}_1^B \mathbf{P}_1^B = \mathbf{Q}_1^B + \mathbf{F}_1^B + \mathbf{S}_{12}^B \mathbf{F}_2^B, \tag{55}$$

where index 1 refers to the leftmost handle, and index 2 refers to the rightmost handle of a body. The matrices $\dot{\mathbf{S}}_1^A$ and $\dot{\mathbf{S}}_1^B$ contain absolute, translational velocities of body *A* and body *B* expressed at handle 1 of the first and second body, respectively. The terms \mathbf{F}_1 and \mathbf{F}_2 denote reaction forces, which have to satisfy Newton’s third law $\mathbf{F}_1^B = -\mathbf{F}_2^A$. Newton’s third law of motion enables us to substitute \mathbf{F}_1^B from (55) into (54) to construct the expression for the coupled, articulated body

$$\dot{\mathbf{P}}_1^A + \mathbf{S}_{12}^A \dot{\mathbf{P}}_1^B + \dot{\mathbf{S}}_1^A \mathbf{P}_1^A + \dot{\mathbf{S}}_1^B \mathbf{P}_1^B = \mathbf{Q}_1^A + \mathbf{S}_{12}^A \mathbf{Q}_1^B + \mathbf{F}_1^A + \mathbf{S}_{12}^A \mathbf{S}_{12}^B \mathbf{F}_2^B. \tag{56}$$

The equation can be reconstructed to a more suitable form using the simplifying transformation $\dot{\mathbf{S}}_1^B = \dot{\mathbf{S}}_1^A + \dot{\mathbf{S}}_{12}^A$ as

$$\frac{d}{dt} (\mathbf{P}_1^A + \mathbf{S}_{12}^A \mathbf{P}_1^B) + \dot{\mathbf{S}}_1^A (\mathbf{P}_1^A + \mathbf{S}_{12}^A \mathbf{P}_1^B) = (\mathbf{Q}_1^A + \mathbf{S}_{12}^A \mathbf{Q}_1^B) + \mathbf{F}_1^A + \mathbf{S}_{12}^A \mathbf{S}_{12}^B \mathbf{F}_2^B. \tag{57}$$

Using the following substitutions

$$\mathbf{P}_1^C = \mathbf{P}_1^A + \mathbf{S}_{12}^A \mathbf{P}_1^B, \quad \mathbf{Q}_1^C = \mathbf{Q}_1^A + \mathbf{S}_{12}^A \mathbf{Q}_1^B, \quad \mathbf{S}_{12}^C = \mathbf{S}_{12}^A \mathbf{S}_{12}^B, \tag{58}$$

and including the appropriate handles coincidence relations

$$\mathbf{F}_1^C = \mathbf{F}_1^A, \quad \mathbf{F}_2^C = \mathbf{F}_2^B, \quad \dot{\mathbf{S}}_1^C = \dot{\mathbf{S}}_1^A, \tag{59}$$

(57) takes a more concise form, similar to the relations in (54)–(55)

$$\dot{\mathbf{P}}_1^C + \dot{\mathbf{S}}_1^C \mathbf{P}_1^C = \mathbf{Q}_1^C + \mathbf{F}_1^C + \mathbf{S}_{12}^C \mathbf{F}_2^C, \tag{60}$$

where \mathbf{P}_1^C and \mathbf{Q}_1^C are accumulated momenta and accumulated external loads. At this point it would be possible to perform the whole multibody system hierarchic assembly according to the binary tree decomposition. The purpose is to define the equation of motion of the whole system in terms of the boundary reaction forces. This step finishes when the base-body is reached.

Now, let us notice that the joint canonical momentum can be found as $\mathbf{p}_1 = \mathbf{H}_1^T \mathbf{P}_1^C$ [32]. Therefore, by exploiting (60), the time derivative of this expression yields

$$\dot{\mathbf{p}}_1 = \mathbf{H}_1^T (\mathbf{Q}_1^C - \dot{\mathbf{S}}_1^C \mathbf{P}_1^C + \mathbf{S}_{12}^C \mathbf{F}_2^C) + \dot{\mathbf{H}}_1^T \mathbf{P}_1^C. \tag{61}$$

The drawback of (61) lies in the presence of the unknown reaction force \mathbf{F}_2^C . Secondly, although (60) and (61) look promising, it would be more beneficial to express the equations of motion in terms of articulated momenta which at this stage are already available.

The common procedure in the modeling of constrained multibody systems possessing closed-loops is to virtually cut redundant joints to generate a tree-like system. This is done by introducing the unknown constraint forces at cut locations. The idea formally leads to an extended Lagrangian and extended Hamiltonian functions for the considered multibody

system, derivation of which is presented in [30] and briefly recalled in Sect. 1. Let us assume that the compound body C is virtually cut at handle O_2 , which means that

$$\overline{\mathbf{P}}_2^C = \mathbf{T}_2^C. \tag{62}$$

The comparison of the accumulated momenta definition (58) with the integral form of the equations of motion (compare (12)–(13)) yields

$$\mathbf{P}_1^C = \overline{\mathbf{P}}_1^C + \mathbf{S}_{12}^C \overline{\mathbf{P}}_2^C, \tag{63}$$

$$\mathbf{P}_2^C = \mathbf{S}_{21}^C \overline{\mathbf{P}}_1^C + \overline{\mathbf{P}}_2^C. \tag{64}$$

These expressions proclaim the fact that there is a clear dependency between accumulated and articulated momenta. Substituting (63) into (60) and exploiting (62) gives

$$\frac{d}{dt} \overline{\mathbf{P}}_1^C + \dot{\mathbf{S}}_1^C \overline{\mathbf{P}}_1^C + (\mathbf{S}_1^C \dot{\mathbf{S}}_{12}^C + \dot{\mathbf{S}}_1^C \mathbf{S}_{12}^C) \overline{\mathbf{P}}_2^C - \mathbf{S}_{12}^C \dot{\mathbf{D}}_2 \sigma_2 = \mathbf{Q}_1^C + \mathbf{F}_1^C, \tag{65}$$

which simplifies to

$$\frac{d}{dt} \overline{\mathbf{P}}_1^C + \dot{\mathbf{S}}_1^C \overline{\mathbf{P}}_1^C + \dot{\mathbf{S}}_2^C \overline{\mathbf{P}}_2^C - \mathbf{S}_{12}^C \dot{\mathbf{D}}_2 \sigma_2 = \mathbf{Q}_1^C + \mathbf{F}_1^C, \tag{66}$$

and allows eliminating reaction force \mathbf{F}_2^C with the derivative of (62). Let us premultiply (66) by the boundary joint’s motion subspace \mathbf{H}_1^T to remove the dependence on the constraint force $\mathbf{F}_1^C = \mathbf{D}_1 \dot{\sigma}_1$, yielding

$$\dot{\mathbf{p}}_1 = \mathbf{H}_1^T (\mathbf{Q}_1^C - \dot{\mathbf{S}}_1^C \overline{\mathbf{P}}_1^C - \dot{\mathbf{S}}_2^C \overline{\mathbf{P}}_2^C + \mathbf{S}_{12}^C \dot{\mathbf{D}}_2 \sigma_2) + \dot{\mathbf{H}}_1^T \overline{\mathbf{P}}_1^C. \tag{67}$$

Please note that the quantity $\dot{\mathbf{p}}_1$ is dependent on the known articulated momenta. Based on (62) and the assumption that there is a cut joint at handle O_2 , the value of the momentum’s derivative at handle O_2 is equal to zero, i.e.,

$$\dot{\mathbf{p}}_2 = \mathbf{H}_2^T \frac{d}{dt} \overline{\mathbf{P}}_2^C + \dot{\mathbf{H}}_2^T \overline{\mathbf{P}}_2^C = \mathbf{0}, \tag{68}$$

which is derived with the use of the identity in the form $\mathbf{H}_2^T \dot{\mathbf{D}}_2 = -\dot{\mathbf{H}}_2 \mathbf{D}_2$. Any joint in the system that opens the loop(s) can be chosen as a cut joint. It is assumed that the time derivative of such joint’s canonical momenta are equal to zero.

The above formalism can be extended to deal with any joint of constrained multibody system. The derivative of joint’s canonical momenta k related to any body K can be expressed in terms of already available articulated momenta as

$$\dot{\mathbf{p}}_k = \mathbf{H}_k^T (\mathbf{Q}_1^K - \dot{\mathbf{S}}_1^K \overline{\mathbf{P}}_1^K - \dot{\mathbf{S}}_c \overline{\mathbf{P}}_c + \mathbf{S}_{kc} \dot{\mathbf{D}}_c \sigma_c) + \dot{\mathbf{H}}_k^T \overline{\mathbf{P}}_1^K, \tag{69}$$

where subindex c designates a cut joint, $\overline{\mathbf{P}}_1^K$ and $\overline{\mathbf{P}}_c$ are the articulated momenta for joint k and cut joint c , respectively. The derivatives of shift matrices $\dot{\mathbf{S}}_1^K$ and $\dot{\mathbf{S}}_c$ contain the translational velocities of handle k and handle c computed with respect to the global, inertial reference frame. The application of this form of equations of motion enables us to include unknown reaction force at cut location c within the definition of articulated momentum vector. Equation (69) allows us to calculate the time derivatives of canonical momenta, by having a vector of accumulated external loads \mathbf{Q}_1^K , which is only unknown in this equation.

Articulated external loads. The purpose of this section is to demonstrate how to calculate the articulated external loads by using a divide and conquer strategy.

Let us define articulated external loads vectors $\overline{\mathbf{Q}}$ at handle O_1 and handle O_2 for bodies A and B , respectively (cp. Fig. 2). Let us also construct the conditions describing the relation between external loads vectors as

$$\mathbf{Q}_1^A = \overline{\mathbf{Q}}_1^A + \mathbf{S}_{12}^A \overline{\mathbf{Q}}_2^A, \tag{70}$$

$$\mathbf{Q}_1^B = \overline{\mathbf{Q}}_1^B + \mathbf{S}_{12}^B \overline{\mathbf{Q}}_2^B. \tag{71}$$

Load vectors \mathbf{Q}_1^A and \mathbf{Q}_1^B represent the external forces acting on bodies A and B , respectively, while $\overline{\mathbf{Q}}_1^A, \dots, \overline{\mathbf{Q}}_2^B$ refer to resultant external loads. As the articulated load vectors $\overline{\mathbf{Q}}_2^A$ and $\overline{\mathbf{Q}}_1^B$ are associated with a single joint, the forces have to satisfy Newton’s law

$$\overline{\mathbf{Q}}_1^B = -\overline{\mathbf{Q}}_2^A. \tag{72}$$

One can substitute (71) into (70) and utilize (72) to algorithmically assemble the forces for body A and B

$$\mathbf{Q}_1^A + \mathbf{S}_{12}^A \mathbf{Q}_1^B = \overline{\mathbf{Q}}_1^A + \mathbf{S}_{12}^A \mathbf{S}_{12}^B \overline{\mathbf{Q}}_2^B. \tag{73}$$

By applying the relations (58) and handles’ coincidence conditions for compound body C (compare with (32)–(33)), one can obtain the following final form:

$$\mathbf{Q}_1^C = \overline{\mathbf{Q}}_1^C + \mathbf{S}_{12}^C \overline{\mathbf{Q}}_2^C. \tag{74}$$

Equation (74) presents the relation between the accumulated and articulated external loads vectors. Considering the whole multibody system assembled into one super-body C , the articulated loads vector $\overline{\mathbf{Q}}_2^C$ is equal to zero as we assume that it concerns the cut joint (see (62)) of a closed-loop system. Thus, the base-body connection conditions can be taken into account to receive

$$\overline{\mathbf{Q}}_1^A = \mathbf{Q}_1^C = \mathbf{Q}_1^A + \mathbf{S}_{12}^A \mathbf{Q}_1^B. \tag{75}$$

By knowing the values of the articulated loads vectors for specified handles, by means of which the system C connects to the base-body, it is possible to perform a back-propagation phase by utilizing the equation

$$\overline{\mathbf{Q}}_1^B = \mathbf{Q}_1^B - \mathbf{S}_{12}^B \overline{\mathbf{Q}}_2^C = -\overline{\mathbf{Q}}_2^A. \tag{76}$$

The above procedure allows one to efficiently compute all necessary loads vectors in parallel achieving logarithmic $O(\log_2 n)$ computational cost. If the external loads do not depend on the velocity terms, the method can be included in the phase of evaluation of joint velocities. Otherwise, it is unavoidable computing the loads separately, or expressing variables \mathbf{Q}_1 and \mathbf{Q}_2 for all bodies in terms of canonical momenta.

Derivatives of canonical momenta By knowing all velocities, constraint force impulses, articulated momenta and articulated external loads, it is possible to compute the derivatives of canonical momenta for every joint k related to the body K using the following formula:

$$\dot{\mathbf{p}}_k = \mathbf{H}_k^T (\overline{\mathbf{Q}}_1^K - \dot{\mathbf{S}}_1^K \overline{\mathbf{P}}_1^K - \dot{\mathbf{S}}_c \overline{\mathbf{P}}_c + \mathbf{S}_{kc} \dot{\mathbf{D}}_c \sigma_c) + \dot{\mathbf{H}}_k^T \overline{\mathbf{P}}_1^K. \tag{77}$$

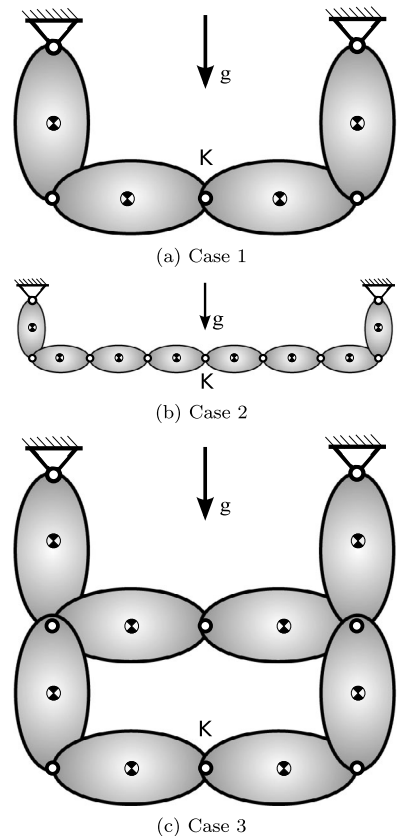
Equation (77) is equivalent to the second set of Hamilton’s canonical equations (2) for constrained multibody systems. There is no need to explicitly calculate the system’s Hamiltonian and its partial derivatives $\partial\mathcal{H}/\partial\mathbf{q}$.

3 Numerical examples

3.1 Illustrative examples

To verify the proposed HDCA algorithm, three examples of closed-loop mechanisms have been considered: single closed-loop with four bodies (case 1), single closed-loop with eight bodies (case 2), and double closed-loop with eight bodies (case 3). The mechanisms at their initial positions are illustrated in Fig. 4. The initial velocities are set to zero. Each body has a mass $m = 1$ kg, characteristic length of a body in the plane of motion $l = 1$ m, and a moment of inertia $\mathbf{J}_c = \text{diag}([1 \ 1 \ 1])$ with respect to the centroidal reference frame of a body. The bodies are interconnected by revolute joints. The gravity forces act in the vertical direction, where $g = 9.81$ m/s² is the acceleration due to gravity. The mechanisms should preserve the symmetry of the motion during the entire simulation. The results were competitively set against analogous DCA algorithm [17] that solves forward dynamics problem by using acceleration based Newton–Euler’s equations, and the closed-loops were enforced by the application of pseudo-inverses.

Fig. 4 Numerical test cases at initial positions



3.2 Accuracy results

The non-stiff Dormand–Prince one step solver implemented in MATLAB (*ode45*) was used with absolute and relative error tolerances set to $\delta = 10^{-6}$. Figure 5 presents the total energy preservation error as a function of the simulation time. It can be concluded from the plots that the HDCA algorithm preserves the energy better for all cases within the range of assumed integrator tolerances. Figures 6 and 7 demonstrate the vertical component of the position and translational velocity of point *K* for the first test case (see Fig. 4). The algorithm is able to reproduce the point's trajectory correctly. The same conclusion may be drawn for other test cases. Figure 8 demonstrates a comparison of the constraint violation errors at the velocity level. The HDCA algorithm combines the equations of motion with the velocity level algebraic constraints. On the other hand, the DCA formulation makes use of algebraic constraints at the acceleration level. There is a huge discrepancy between the results in favor of the Hamilton's divide-and-conquer method. In the case of HDCA, the constraint equations at the velocity level are satisfied within the numerical accuracy. This behavior has an impact on the constraints violation errors at the position level. The position constraint errors for the DCA approach diverge quadratically in terms of the simulation time. On the other hand, the drift made by the HDCA is negligible as indicated in Fig. 9.

3.3 Efficiency results

Three MATLAB solvers are chosen to comparatively test the efficiency of the HDCA and the DCA algorithms, i.e., *ode45*, *ode113*, and *ode23*. The Dormand–Prince one step solver (*ode45*) is suggested as a first try for most problems. The *ode113* method is a multistep and variable order Adams–Bashforth–Moulton PECE solver for non-stiff, but computationally intensive tasks. The *ode23* is an implementation of an explicit Runge–Kutta (2, 3) pair of Bogacki and Shampine. It may be more accurate than *ode45* in the presence of moderately stiff equations. The dynamics of the mechanisms (cases 1–3) presented in Fig. 4 are simulated for 10 seconds and various integration tolerances $\delta = \{10^{-3}, 10^{-4}, \dots, 10^{-9}\}$. It is assumed that the relative integrator tolerance is equal to the absolute tolerance. A series of numerical experiments have been performed to measure the relative performance of HDCA and DCA formulations. Three numerical test cases are taken into account as described in Sect. 3.1.

The total wall-clock time of 10-second simulations and the number of right-hand side function evaluations taken by the integrators are recorded in order to build the comparisons. Figure 10 shows a ratio of the wall-clock time taken to execute DCA algorithm to the wall-clock time taken to process HDCA for a given test case. These measures are presented as a function of the integrators' tolerances. On the other hand, Fig. 11 demonstrates the ratio of the right-hand side function evaluations associated with both algorithms in terms of the integrators' tolerances. There are several remarks that can be made based on the plots. The performance of HDCA is in most cases worse than DCA when *ode45* integration routine is used for calculations. One case for which HDCA performs better than DCA can be observed for high integrator tolerances (10^{-9}). The situation changes when multi-step *ode113* integrator routine is exploited for calculations.

This time HDCA has greater dominance over the acceleration based competitor measured either by using the wall-clock time or by counting the number of right-hand side function calls. It is worth noticing significant computational savings in terms of the wall-clock time for tight tolerances (i.e., 10^{-9}). When *ode23* is exploited, one could take the performance advantages for the test case 1 and 2, while HDCA performs a little bit worse for the test case 3.

Fig. 5 Total energy error

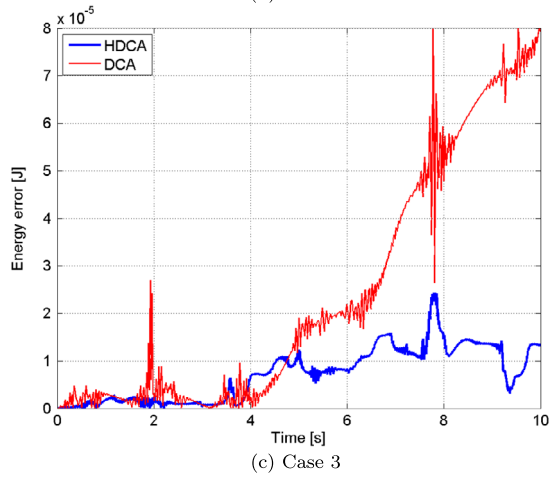
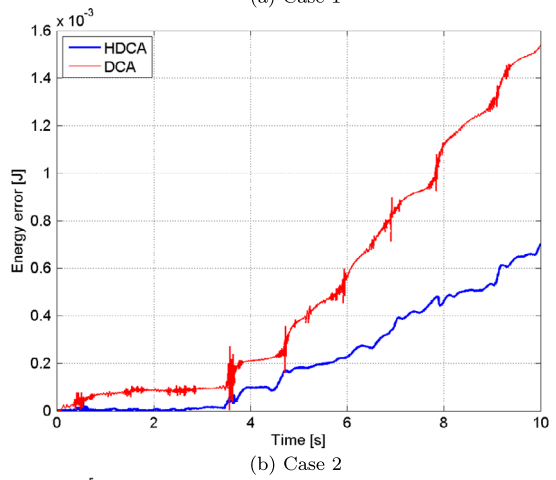
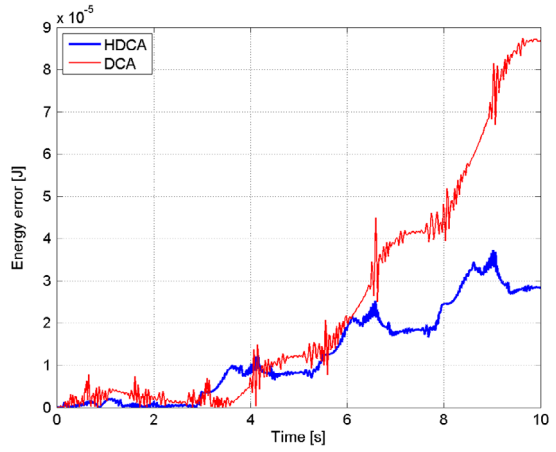


Fig. 6 *Y* position of the point *K* for case 1

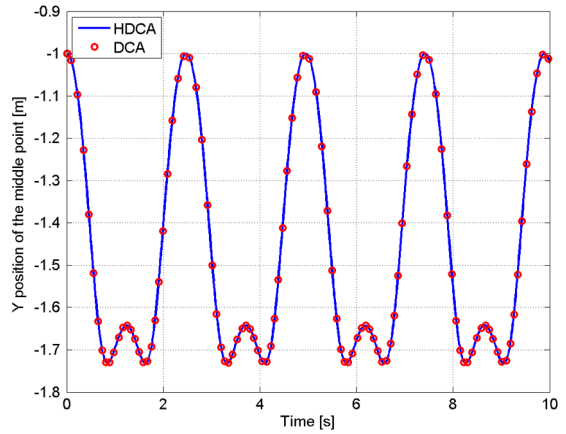


Fig. 7 *Y* velocity of the point *K* for case 1

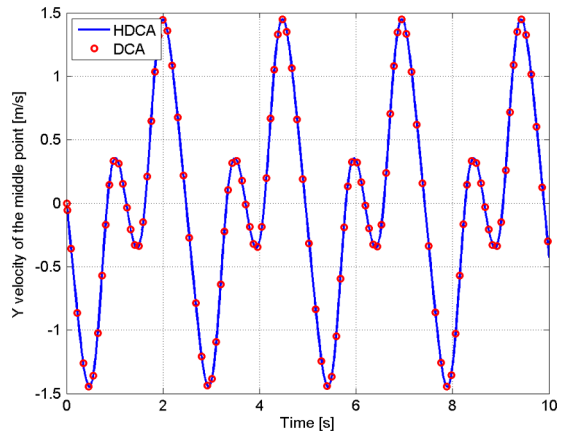
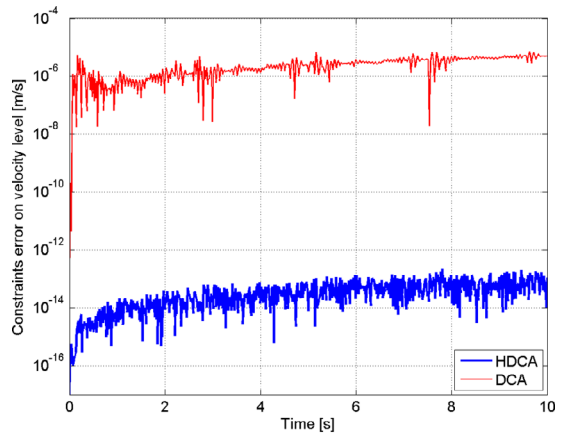


Table 1 Mean right-hand side function evaluation time in milliseconds

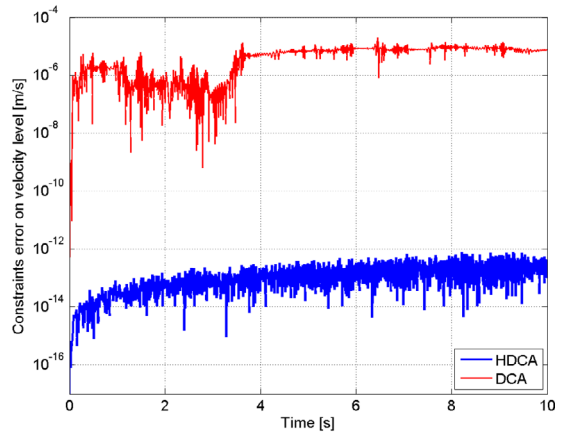
Algorithm	Case 1	Case 2	Case 3
HDCA	5.203	8.806	15.278
DCA	7.394	13.388	15.501

To complement the results, Table 1 demonstrates the mean wall-clock time taken for a single call of the right-hand side function. In our implementation, the HDCA approach requires less time to calculate the state variables for the test cases 1 and 2 than the analogous procedure for the DCA method. The mean time required to evaluate the right-hand side function is similar for both formulations when coupled-loop test case is considered. Table 2 presents the wall-clock time for the three test cases and three integrators used for the simulations. It turns out that *ode23* is much slower than other two integration methods. The fastest method solving Hamilton’s and Newton–Euler’s equations of motion was *ode113*. Based on the results in Tables 1 and 2, one could estimate the total number of right-hand side function calls for each simulation.

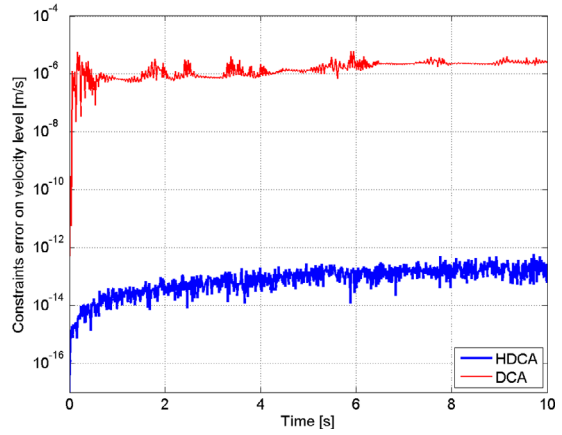
Fig. 8 Constraint violation errors of kinematic loop closure at the velocity level



(a) Case 1



(b) Case 2



(c) Case 3

Fig. 9 Constraint violation errors of kinematic loop closure at the position level

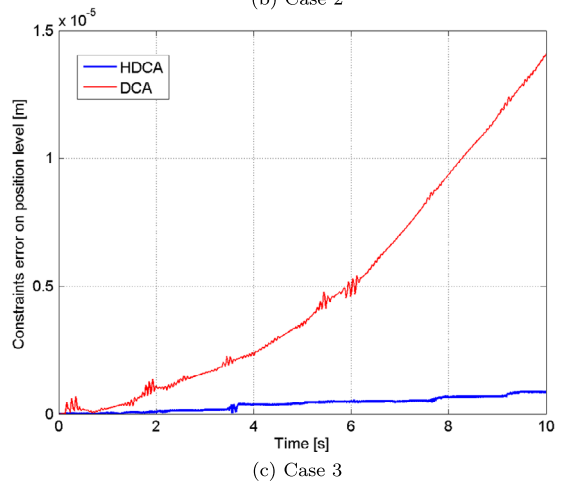
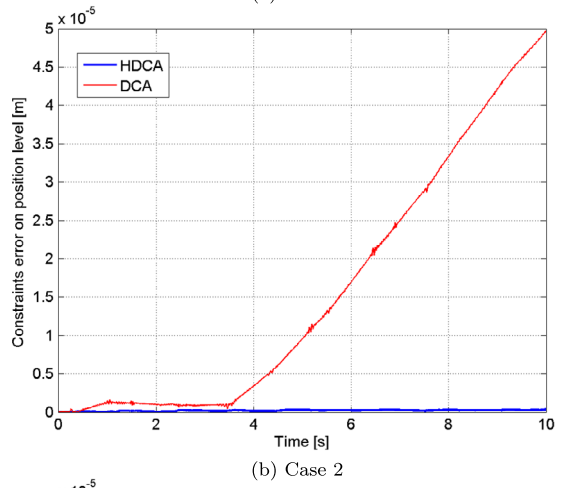
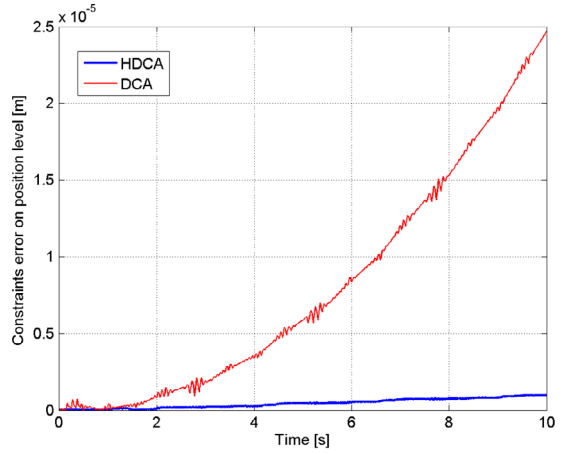
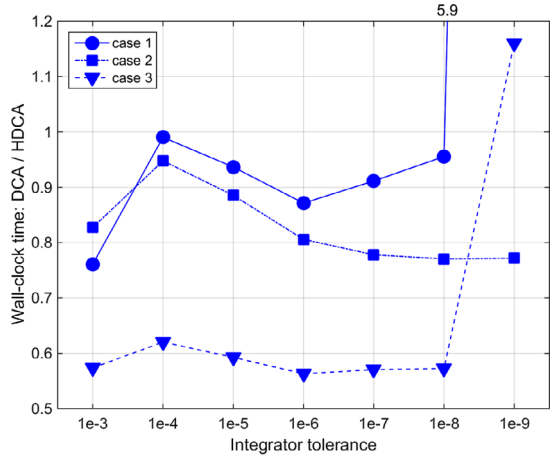
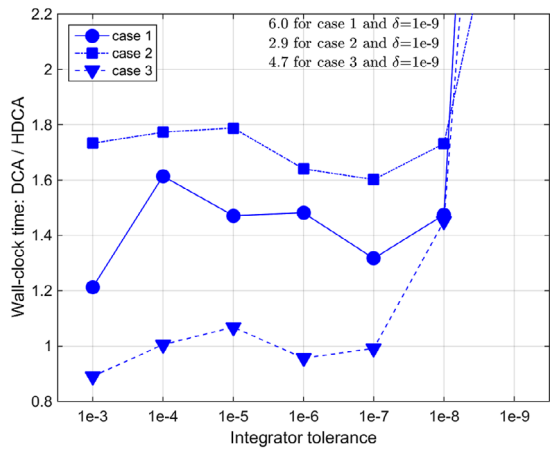


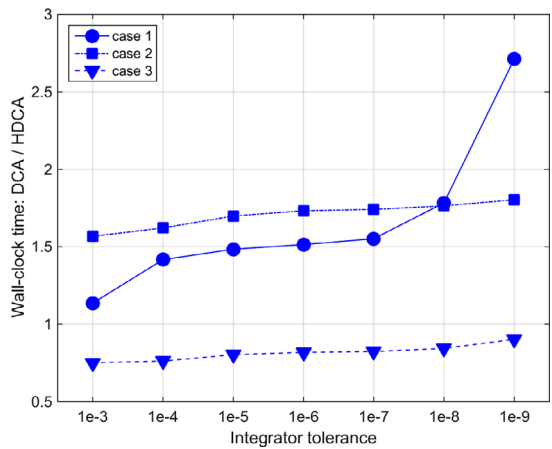
Fig. 10 The wall-clock time of 10 s simulation: DCA/HDCA



(a) *ode45*

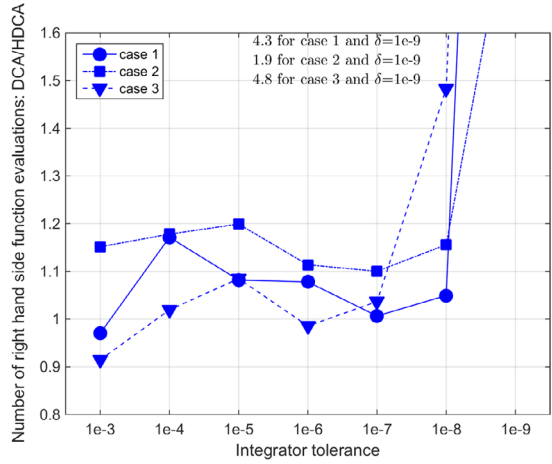


(b) *ode113*

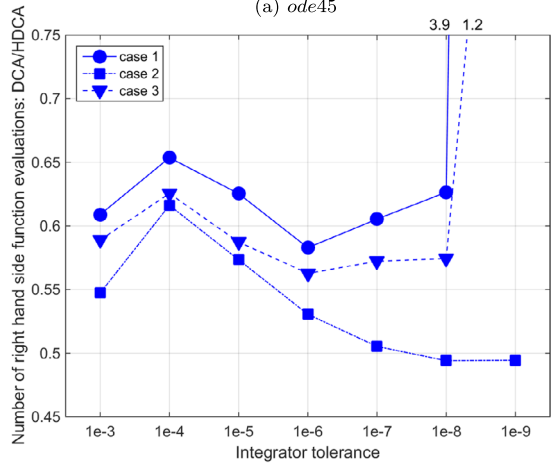


(c) *ode23*

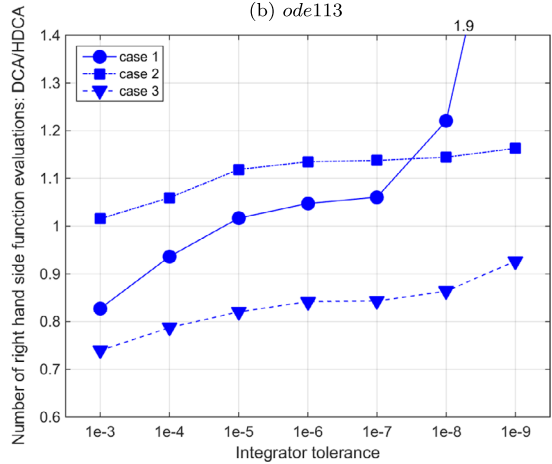
Fig. 11 Number of right-hand side function evaluations, DCA/HDCA



(a) *ode45*



(b) *ode113*



(c) *ode23*

Table 2 The wall-clock time of 10 s simulation (in seconds)

	δ	<i>ode45</i>			<i>ode113</i>			<i>ode23</i>		
		HDCA	DCA	κ	HDCA	DCA	κ	HDCA	DCA	κ
case 1	1e-3	2.72	2.07	0.76	1.62	1.97	1.21	2.84	3.22	1.14
	1e-4	2.91	2.88	0.99	1.86	3.00	1.61	4.40	6.24	1.42
	1e-5	4.51	4.23	0.94	2.49	3.66	1.47	8.88	13.18	1.48
	1e-6	6.83	5.95	0.87	3.16	4.68	1.48	18.85	28.54	1.51
	1e-7	10.06	9.17	0.91	4.11	5.41	1.32	39.86	61.85	1.55
	1e-8	15.49	14.80	0.96	4.78	7.05	1.47	85.25	152.04	1.78
	1e-9	24.46	144.94	5.93	5.76	34.34	5.96	183.88	498.54	2.71
case 2	1e-3	9.70	8.03	0.83	5.18	8.98	1.73	8.49	13.30	1.57
	1e-4	13.06	12.38	0.95	7.00	12.42	1.77	16.23	26.31	1.62
	1e-5	20.23	17.91	0.89	9.20	16.45	1.79	33.38	56.68	1.70
	1e-6	31.49	25.38	0.81	11.84	19.43	1.64	70.55	122.22	1.73
	1e-7	49.30	38.36	0.78	14.96	23.96	1.60	151.42	263.58	1.74
	1e-8	77.72	59.89	0.77	17.89	30.98	1.73	324.17	571.81	1.76
	1e-9	124.01	95.72	0.77	21.99	63.60	2.89	697.14	1257.17	1.80
case 3	1e-3	8.43	4.84	0.57	5.23	4.66	0.89	9.55	7.18	0.75
	1e-4	10.96	6.80	0.62	6.50	6.54	1.01	18.38	14.02	0.76
	1e-5	16.68	9.89	0.59	8.30	8.87	1.07	38.02	30.56	0.80
	1e-6	26.16	14.73	0.56	10.96	10.49	0.96	80.01	65.53	0.82
	1e-7	40.47	23.10	0.57	13.20	13.10	0.99	172.23	141.96	0.82
	1e-8	63.59	36.41	0.57	16.08	23.33	1.45	371.33	313.55	0.84
	1e-9	99.90	115.96	1.16	18.39	86.90	4.73	799.48	721.92	0.90

3.4 Discussion

The HDCA algorithm presented in this paper is the approach that enables one to simulate complex multi-rigid-body systems subjected to independent, holonomic constraints in an efficient and robust manner. Hamilton's canonical equations are set and solved in a divide-and-conquer framework which enables a full exploitation of parallel computing. Pseudo-algorithm 3 presents a list of detailed algorithmic steps that one has to perform in order to calculate the time derivatives of joint canonical coordinates. The numerical cost of sequential calculations remains linear in terms of the number of bodies n in the system. The method exhibits near logarithmic complexity for n threads available for computations. The parallelizable assembly–disassembly procedure is exploited at several different levels of the algorithm. Velocities, Lagrange multipliers, and time derivatives of canonical momenta are evaluated in a recursive, divide-and-conquer manner. There are two general strategies that can be utilized to parallelize the computations. The recursive HDCA method is connected to the binary tree associated with the topology of a system. Inherently, one can assume that each node of this tree establishes an independent instance of calculation. At each level of the binary tree, one could perform embarrassingly parallel computations. On the other hand, the whole tree could be divided into well-balanced subtrees to equally spread the computational load over each thread.

Algorithm 3 Pseudo-algorithm for HDCA

1. Compute coefficients ξ for each body in the system (15)–(16).
2. For each pair of bodies perform the assembly (see Algorithm 1).
3. Repeat step 2 until the whole system is assembled into a single body C .
4. Consider base body connection. Compute:
 - (a) Lagrange multipliers in (45),
 - (b) Constraint force impulses in (39),
 - (c) Articulated loads $\overline{\mathbf{Q}}_1^C = \mathbf{Q}_1^C$ in (74),
 - (d) Set $\mathbf{T}_2^C = 0$ and $\overline{\mathbf{Q}}_2^C = 0$.
5. Perform disassembly procedure (see Algorithm 2) for each of the assembled entities.
6. Calculate velocities \mathbf{V} for each body in (15)–(16).
7. Compute all joint velocities $\dot{\mathbf{q}}$ by projection of (23) onto joint's motion subspaces.
8. For each body determine $\overline{\mathbf{P}}_1$ (10) and $\dot{\mathbf{S}}_1$ in (53).
9. Identify cut joints and introduce constraint force impulses in (62).
10. Compute all derivatives of canonical momenta in (77).

There are several remarks that should be mentioned to properly analyze the numerical results gathered in Sect. 3. Firstly, the numerical examples examined in the paper are planar systems. Nevertheless, the HDCA algorithm can be successfully used to analyze the dynamics of spatial closed-loop systems. The method exploits spatial velocity as well as spatial momenta vectors. No simplifications have been made to limit the applicability of the formulation at any stage of the development. Moreover, the experience of the authors confirms that the HDCA method performs very well for fully three-dimensional multibody systems in terms of the total energy conservation and the constraint violation errors. Secondly, Sect. 3.2 shows that the HDCA algorithm is able to preserve the energy and constraint equations better than the DCA approach for considered closed-loop test cases. This property comes from the fact that HDCA takes the velocity level conditions as primary constraints. It is indicated that only the position constraints are violated, and the resulting errors are much smaller compared to the acceleration based DCA method. Thirdly, Sect. 3.3 demonstrates the performance of HDCA and the comparisons against DCA. The reader should be aware that the integrator absolute and relative tolerances are equal to each other in our implementations. The tolerances are equally assigned to the velocities vectors $\dot{\mathbf{q}}$ as well as to the momenta vectors \mathbf{p} . To compare both formulations more accurately, one should go into the details of the integration routines. This issue is beyond the scope of this paper.

In some applications, there is a need to calculate constraint forces. This issue is particularly important when design or control applications are considered. The HDCA algorithm presented in this paper allows one to calculate the constraint force impulses. The method needs some extensions to be able to determine the constraint forces. One method to calculate the constraint forces is to exploit the condition $\dot{\sigma} = \lambda$. The finite difference method may be employed with its own advantages and disadvantages. The other way is to add a separate computational module for the calculation of reaction forces. It should be noted that the additional cost of such procedure is associated only with the evaluation of the matrices ξ_{10} and ξ_{20} . The other coefficients are already available since they are calculated by the HDCA algorithm. The evaluation of constraint forces requires only a small fraction of the total cost of the assembly–disassembly procedure for the evaluation of velocities and constraint force impulses.

As the constraint forces are considered, one should mention additional property of the algorithm. The constraint force impulse multipliers σ are the primary quantities evaluated

by the HDCA method. In the case when a reaction force component λ_i never changes its sign during the motion, the Lagrange multiplier σ_i may eventually grow indefinitely with the simulation time (compare the condition $\dot{\sigma}_i = \lambda_i$). This may lead to numerical errors. Presumably, this issue could be circumvented by scaling the subspace of constrained directions \mathbf{D}_i that in turn would affect the value of the Lagrange multipliers σ_i . Obviously, such a scaling should not have any effect on the value of the constraint force impulses \mathbf{T}_i .

The HDCA algorithm is exploited for the dynamics simulation of systems possessing single kinematic loop and coupled kinematic loops. The idea demonstrated in the text can be generalized to cope with more complex topologies involving many coupled loops, especially when the number of handles per body is greater than two. The mentioned extensions require more detailed analysis of the topology of the system under consideration; however, the concept of the HDCA algorithm remains the same. Apart from the issues mentioned within this subsection, there are several other possibilities for the HDCA algorithm development. Undoubtedly, the adaptation of stabilization methods and symplectic integration procedures would further improve the HDCA algorithm's efficiency and accuracy.

4 Conclusions

In this paper, a novel HDCA algorithm is presented for the simulation of constrained multi-rigid body system dynamics possessing closed-loop topologies. The HDCA approach utilizes Hamilton's canonical equations of motion, which are recursively generated in a divide-and-conquer framework. The two- or multi-handle equations are formed for the spatial velocities of associated bodies and constraint load impulses at joints. Also the evaluation of the time derivatives of joint canonical momenta can be performed in a fully parallel manner. The comparison of the results with the analogous, acceleration-based divide-and-conquer formulation indicates the predominance of the HDCA method for the considered numerical examples in terms of the total energy conservation and constraint violation errors. Sample closed-loop test cases demonstrate excellent constraint satisfaction at the position and velocity level as well as marginal energy drift for various integration procedures used for calculations. The properties of the method makes the HDCA scheme worth considering in the applications to systems with more general topologies than those presented in the paper.

Acknowledgement This work has been supported by the National Science Centre under grant No. DEC-2012/07/B/ST8/03993.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Poursina, M., Bhalerao, K., Flores, S., Anderson, K., Laederach, A.: Strategies for articulated multibody-based adaptive coarse grain simulation of RNA. *Methods in enzymology. Methods Enzymol.* **487**, 73–98 (2014)
2. Malczyk, P., Frączek, J.: Molecular dynamics simulation of simple polymer chain formation using divide and conquer algorithm based on the augmented Lagrangian method. *J. Multi-Body Dyn.* **229**(2), 116–131 (2015)
3. Negrut, D., Serban, R., Mazhar, H., Heyn, T.: Parallel computing in multibody system dynamics: why, when, and how. *J. Comput. Nonlinear Dyn.* **9**(4), 041007 (2014)

4. Tomulik, P., Frączek, J.: Simulation of multibody systems with the use of coupling techniques: a case study. *Multibody Syst. Dyn.* **25**(2), 145–165 (2011)
5. Rodriguez, G.: Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics. *IEEE J. Robot. Autom.* **6**, 624–639 (1987)
6. Bae, D.S., Haug, E.J.: A recursive formulation for constrained mechanical system dynamics, part I: open loop systems. *Mech. Struct. Mach.* **15**, 359–382 (1987)
7. Anderson, K.: An order n formulation for the motion simulation of general multi-rigid-body constrained systems. *Comput. Struct.* **3**, 565–579 (1992)
8. Rodriguez, G., Jain, A., Kreutz-Delgado, K.: Spatial operator algebra for manipulator modelling and control. *Int. J. Robot. Res.* **10**(4), 371–381 (1991)
9. Featherstone, R.: The calculation of robot dynamics using articulated-body inertias. *Int. J. Robot. Res.* **2**, 13–30 (1983)
10. Jain, A.: Unified formulation of dynamics for serial rigid multibody systems. *J. Guid. Control Dyn.* **14**(3), 531–542 (1991)
11. Yamane, K., Nakamura, Y.: $O(N)$ forward dynamics computation of open kinematic chains based on the principle of virtual work. In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2824–2831 (2001)
12. Saha, K.S., Schiehlen, W.: Recursive kinematics and dynamics for parallel structured closed-loop multibody systems. *Mech. Struct. Mach.* **29**(2), 143–175 (2001)
13. Anderson, K., Critchley, J.: Improved 'Order- N ' performance algorithm for the simulation of constrained multi-rigid-body dynamic systems. *Multibody Syst. Dyn.* **9**, 185–225 (2003)
14. Lathrop, R.: Parallelism in manipulator dynamics. *Int. J. Robot. Res.* **4**(2), 80–102 (1985)
15. Bae, D.S., Kuhl, J.G., Haug, E.J.: A recursive formulation for constrained mechanical system dynamics, part 3. *Mech. Based Des. Struct. Mach.* **16**(2), 249–269 (1988)
16. Fijany, A., Sharf, I., D'Eleuterio, G.: Parallel $O(\log n)$ algorithms for computation of manipulator forward dynamics. *IEEE Trans. Robot. Autom.* **11**, 389–400 (1995)
17. Featherstone, R.: A divide-and-conquer articulated body algorithm for parallel $O(\log n)$ calculation of rigid body dynamics, parts 1 and 2. *Int. J. Robot. Res.* **18**, 867–892 (1999)
18. Fiset, P., Peterkenne, J.M.: Contribution to parallel and vector computation in multibody dynamics. *Parallel Comput.* **24**, 717–728 (1998)
19. Critchley, J., Anderson, K.: A parallel logarithmic order algorithm for general multibody system dynamics. *Multibody Syst. Dyn.* **12**, 75–93 (2004)
20. Malczyk, P., Frączek, J.: Cluster computing of mechanisms dynamics using recursive formulation. *Multibody Syst. Dyn.* **20**(2), 177–196 (2008)
21. González, F., Luaces, A., Lugiš, U., González, M.: Non-intrusive parallelization of multibody system dynamic simulations. *Comput. Mech.* **44**(4), 493–504 (2009)
22. Yamane, K., Nakamura, Y.: Comparative study on serial and parallel forward dynamics algorithms for kinematic chains. *Int. J. Robot. Res.* **28**(5), 622–629 (2009)
23. Mukherjee, R., Anderson, K.: Efficient methodology for multibody simulations with discontinuous changes in system definition. *Multibody Syst. Dyn.* **18**, 145–168 (2007)
24. Mukherjee, R., Anderson, K.: Orthogonal complement based divide-and-conquer algorithm for constrained multibody systems. *Nonlinear Dyn.* **48**, 199–215 (2007)
25. Malczyk, P., Frączek, J.: A divide and conquer algorithm for constrained multibody system dynamics based on augmented Lagrangian method with projections-based error correction. *Nonlinear Dyn.* **70**(1), 871 (2012). doi:[10.1007/s11071-012-0503-2](https://doi.org/10.1007/s11071-012-0503-2)
26. Khan, I., Anderson, K.: Performance investigation and constraint stabilization approach for the orthogonal complement-based divide-and-conquer algorithm. *Mech. Mach. Theory* **67**, 111–121 (2013)
27. Mukherjee, R., Malczyk, P.: Efficient approach for constraint enforcement in constrained multibody system dynamics. In: *Proc. of the ASME 2013 IDETC/CIE Conf. on Multibody Systems, Nonlinear Dynamics, and Control*, Portland, USA (2013)
28. Laffin, J.J., Anderson, K.S., Khan, I.M., Poursina, M.: Advances in the application of the divide-and-conquer algorithm to multibody system dynamics. *J. Comput. Nonlinear Dyn.* **9**(4), 041003 (2014)
29. Bae, D.S., Won, Y.S.: A Hamiltonian equation of motion for real-time vehicle simulation. *Adv. Des. Autom.* **2**, 151–157 (1990)
30. Lankarani, H., Nikravesh, P.: Application of the canonical equations of motion in problems of constrained multibody systems with intermittent motion. *Adv. Des. Autom.* **1**, 417–423 (1988)
31. Bayo, E., Avello, A.: Singularity free augmented Lagrangian algorithms for constrained multibody dynamics. *Nonlinear Dyn.* **5**, 247–255 (1994)
32. Naudet, J., Lefeber, D., Daerden, F., Terze, Z.: Forward dynamics of open-loop multibody mechanisms using an efficient recursive algorithm based on canonical momenta. *Multibody Syst. Dyn.* **10**(1), 45–59 (2003)

33. Chadaj, K., Malczyk, P., Frączek, J.: Efficient parallel formulation for dynamics simulation of large articulated robotic systems. In: Proc. of the 20th International Conference on Methods and Models in Automation and Robotics, Międzyzdroje, Poland (2015)
34. Chadaj, K., Malczyk, P., Frączek, J.: A parallel recursive Hamiltonian algorithm for forward dynamics of serial kinematic chains. *IEEE Trans. Robot.* (2016, conditionally accepted as regular paper)
35. Goldstein, H., Poole, C., Safko, J.: *Classical Mechanics*. Addison–Wesley, Reading (2001)
36. Leimkuhler, B., Reich, S.: *Simulating Hamiltonian Dynamics*. Cambridge Univ. Press, London (2005)
37. Wojtyra, M., Frączek, J.: Comparison of selected methods of handling redundant constraints in multi-body systems simulations. *J. Comput. Nonlinear Dyn.* **8**, 1–9 (2013)
38. Wojtyra, M., Frączek, J.: Solvability of reactions in rigid multibody systems with redundant nonholonomic constraints. *Multibody Syst. Dyn.* **30**(2), 153–171 (2013)