Multibody Syst Dyn (2018) 44:367–395
https://doi.org/10.1007/s11044-018-09642-4

Dynamic modeling of tree-type robotic systems by combining 3 × 3 rotation and 4 × 4 transformation matrices

A.M. Shafei[1] · **H.R. Shafei**[1]

Received: 13 July 2017 / Accepted: 13 September 2018 / Published online: 4 October 2018

Abstract The objective of this article is to present a systematic method of deriving the mathematical formulation for the oblique impact of a tree-type robotic manipulator. The dynamic response of this system (confined within a curved-wall environment) is expressed by two diverse models. A set of differential equations is employed to obtain the dynamic behavior of the system when it has no contact with any object in its environment (flying phase), and a set of algebraic equations is used to describe the collision of the system with the curved walls (impact phase). The Gibbs–Appell formulation in recursive form and the Newton's impact law are utilized to derive the governing equations of this robotic system for the flying and impact phases, respectively. The main innovation of this article is the development of an automatic approach based on the combination of 3×3 rotation and 4×4 transformation matrices. In fact, this is the first time the merits of 3×3 rotation matrices (i.e., improving the computational efficiency of the developed algorithm) have been merged with the capabilities of 4×4 transformation matrices (i.e., deriving more compact motion equations by combining rotations with translations). Finally, a case study involving a tree-type robotic system with 12 degrees of freedom has been simulated to show the efficiency of the proposed dynamic modeling.

Keywords 4×4 transformation matrix · 3×3 rotation matrix · Oblique impact · Tree-type · Gibbs–Appell

Nomenclature

$\overset{m}{\boldsymbol{A}}_i$	Joint transformation matrix
$\overset{m}{\boldsymbol{E}}_i$	Link transformation matrix
$\boldsymbol{I}_f$	Inertia matrix in the flight phase
$\boldsymbol{I}_i$	Inertia matrix in the impact phase
$\boldsymbol{J}$	Jacobian matrix
$\overset{m}{\boldsymbol{J}}'_i$	Mass moment of inertia per unit length of the ith link in branch m

✉ A.M. Shafei
shafei@uk.ac.ir

[1] Department of Mechanical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

$\overset{m}{Q}$	Differential element
${}^{j}_{m}\boldsymbol{R}_{i}$	Rotation matrix
S	Gibbs function
${}^{j}_{m}\boldsymbol{T}_{i}$	Transformation matrix
V	Gravitational potential energy
X_j	The position of O_1 in the ${}^{\text{ref}}X_j$ direction
${}^{\text{ref}}X_1\,{}^{\text{ref}}X_2\,{}^{\text{ref}}X_3$	Global coordinate system
e	Coefficient of restitution
$\boldsymbol{g}$	Gravitational field vector
$\overset{m}{l}_i$	Length of the ith link in branch m
${}^{\text{ref}}\boldsymbol{n}_j$	Normal vector at contact point
n	Number of rigid links
n_b	Number of branches
n_m	Number of links in branch m
$\overset{m}{q}_i$	Generalized coordinate of the ith joint in branch m
${}^{\text{ref}}\dot{\boldsymbol{r}}_{O_i}$	Absolute velocity of the ith joint in inertial reference frame
${}^{i}\boldsymbol{r}_{Q/O_i}$	Position of differential element $\overset{m}{Q}$ with respect to $\overset{m}{O}_i$
t^-/t^+	Time just before/after impact
$\overset{m}{x}_{i,1}\ \overset{m}{x}_{i,2}\ \overset{m}{x}_{i,3}$	Coordinate system attached to the beginning of the ith link in branch m
$\overset{m}{\hat{x}}_{i,1}\ \overset{m}{\hat{x}}_{i,2}\ \overset{m}{\hat{x}}_{i,3}$	Coordinate system attached to the end of the ith link in branch m
$\Re$	Remaining dynamic terms
α	Angle of attack
δf_i	Impulsive external force acting on the ith joint
$\overset{m}{\eta}_i$	Distance between differential element $\overset{m}{Q}$ and $\overset{m}{O}_i$
$\boldsymbol{\theta}$	Generalized coordinate vector
$\overset{m}{\mu}_i$	Mass per unit length of the ith link in branch m
$\overset{m}{\tau}_i$	Applied torque exerted on the ith joint in branch m
${}^{i}\overset{m}{\boldsymbol{\omega}}_i$	Angular velocity of the ith joint in branch m

1 Introduction

Developments in the field of robotics have led to the design and construction of robotic systems that imitate animals. These systems can be modeled by several open kinematic chains (branches); so, they are referred to as tree-type (branched) robotic systems. Depending on the assignment of these systems, their bases might be fixed or moving. Clearly, a moving base will allow a robot to operate in a much larger workspace. Moreover, these highly-mobile robotic systems are superior to their wheeled counterparts that move on level surfaces.

A branched robotic system with the 4-legged locomotion mechanism of horses was studied by Muybridge [1] about 50 years ago. Schiehlen [2] presented the dynamics of walking human skeletal models with tree-type configurations. He also expanded the motion equations of musculoskeletal models by using contractile and elastic elements. The concept of Decoupled Natural Orthogonal Complement (DeNOC) matrices was exploited by Shah et al. [3] to derive the governing equations of branched robotic manipulators. Vukobratovic

et al. [4] obtained the motion equations for a humanoid robotic system composed of a basic body and several branches. In their work, the humanoid model was not connected to the ground or to any object in its environment. For simulating large articulated mechanisms such as humanoids, a parallel computer algorithm that combined Divide and Conquer Algorithm (DCA) with Articulated Body Algorithm (ABA) was proposed by Bhalerao et al. [5]. The most comprehensive survey of the new developments in the dynamic modeling, designing and controlling of tree-type robotic systems can be found in [6]. Nonetheless, despite the great developments achieved in the mentioned works, the combined use of differential and algebraic equations for modeling branched robotic systems still suffers from high computational complexities, especially when the number of branches increases.

Regarding tree-type robotic systems with a floating base, it is necessary to study the effect of system collision with confining surfaces. Although a large volume of literature exists on the subject of impact [7–12], we have focused on research works that study the dynamics of impacts in multibody system dynamics. The differences between two well-known methods of modeling the impact phenomena, the generalized Poisson's impact law and the generalized Newton's impact law, were explained in detail by Gloker [13, 14]. A model based on the momentum balance method was presented by Yigit et al. [15] for the dynamics of a radially rotating beam with impact. Their proposed method can predict rigid body motions as well as elastic deformations before and after impacts [16]. In another study by the same authors, the Spring–Dashpot model was employed to simulate the impact dynamics of the abovementioned system [17]. This model is based on the Hertzian contact law and uses a nonlinear damping mechanism for impact modeling. A time-stepping algorithm for the contact modeling of multibody systems with many unilateral constraints was proposed by Förg et al. [18]. They employed an iterative algorithm based on the Gauss–Seidel relaxation scheme to handle multiple contacts. Shafei and Shafei [19] employed the Newton's impact law to incorporate the effects of impact into the motion equations of rigid open kinematic chains. They also considered the effects of impact in the mathematical modeling of multi-flexible-link robotic manipulators by using two distinct algorithms. In the first proposed algorithm, they employed 3×3 rotation matrices to reduce the computational load [20], and in the second, they exploited 4×4 transformation matrices in order to more compactly derive the motion equations [21]. However, none of the abovementioned works has addressed the motion kinematics of tree-type robotic systems with impact. Therefore, in the work of the same authors [22], the mathematical model of multi-branch robotic manipulators subjected to ground collision was derived in a symbolic form by using the Gibbs–Appell formulation and 3×3 rotation matrices. Nonetheless, despite using an efficient algorithm with the least computational load in their work, the developed equations suffer from lengthy formulations. Moreover, the impact in this work has been confined to flat surfaces and collision with curved walls has not been considered.

For control purposes, it is essential to have an exact dynamic model for tree-type robotic systems. As the number of branches in these systems increases, the volume of computations related to their motion equations goes up tremendously; and the situation is exacerbated by considering the dynamics of impact for them. As a consequence, the manual derivation of motion equations becomes impractical; and it is necessary to employ an automatic scheme to symbolically derive the motion equations. There are many recursive formulations that can be exploited for open and closed kinematic chains [23–38]; however, the emphasis of this paper is on the recursive Gibbs–Appell method, which has been used the least among the other approaches. In a study related to robotics, Mata et al. [39] showed that their proposed algorithm, which is based on Gibbs–Appell formulation and 3×3 rotational matrices, requires a smaller number of algebraic operations, compared to other methods. This algorithm was

effectively applied by Korayem and Shafei to the dynamic modeling of multi-flexible-link robotic manipulators [40, 41], mobile-base robotic manipulators [42, 43], robotic manipulators with revolute-prismatic joints [44, 45], and closed chain robotic systems [46]; and its improved computational efficiency relative to Lagrangian formulations was demonstrated. The main feature of dynamic algorithms based on 3×3 rotation matrices is their ability to improve computational efficiency by using fewer multiplications and additions [47]. However, these kinds of dynamic models suffer from lengthy formulations. So, the abovementioned researchers have also developed an algorithm based on 4×4 transformation matrices (in which rotations and translations are combined) and Gibbs–Appell methodology in order to derive the motion equations of tree-topology systems in more compact form, albeit at a higher computational cost [48].

As mentioned above, 3×3 rotation matrices improve the computational efficiency, while 4×4 transformation matrices derive more compact motion equations. In fact, the main contribution of this paper is to use a combination of 3×3 rotation and 4×4 transformation matrices in a symbolic and systematic way in order to reach a compromise between the computational complexity of a recursive algorithm and the lengthy formulas used for deriving the motion equations of tree-type robotic systems in collision with curved walls. So, the rest of this paper is organized as follows: In Sect. 2, the kinematics of the considered tree-type robotic system is described. The dynamics of this system in the flying phase, including its general inertia matrix and the general right hand side vector, is formulated in Sect. 3. The dynamics of the system in the impact phase is presented in Sect. 4. In Sect. 5 the computational complexity of the proposed algorithm is presented. A ten-link branched robotic system is simulated in Sect. 6 to demonstrate the efficiency of the proposed formulation. And finally in Sect. 7, the concluding remarks are summarized and the merits of the proposed method are highlighted.

2 Kinematics of the system

Here, the kinematic model of a tree-type robotic system consisting of multiple branches, and surrounded by curved surfaces, is presented. Each branch is also composed of an open kinematic chain of rigid links. Figure 1 displays the schematic of the branched system composed of 10 links ($n = 10$). In order to derive the kinematic formulations, the tree-type system should be systematically divided into n_b branches. Although the idea of dividing a tree-type robotic system into several branches can be found in previous research works [49–51], a new systematic method has been employed here, in which each branch starts from the first point (i.e., O_1) and terminates at one of the end points (i.e., O_5, O_8, O_9 or O_{11}). In this way, any branched system is assumed to be constructed of a certain number of open kinematic chains (serial robotic manipulators). So, for the tree-type system shown in Fig. 1, the number of branches is $n_b = 4$. If a specific link appears in more than one branch, it will be assumed as massless in all those branches. This is illustrated in Fig. 1 by drawing dashed lines for these massless (virtual) links.

Links $(i-1)$ and (i) in branch m, consisting of n_m real and virtual links, are depicted in Fig. 1. Two coordinate systems $(\overset{m}{x}_{i,1}\ \overset{m}{x}_{i,2}\ \overset{m}{x}_{i,3}, \overset{m}{\hat{x}}_{i,1}\ \overset{m}{\hat{x}}_{i,2}\ \overset{m}{\hat{x}}_{i,3})$ are attached to each link according to the following guidelines: the origin of $\overset{m}{x}_{i,1}\ \overset{m}{x}_{i,2}\ \overset{m}{x}_{i,3}$ coordinate system is attached to the beginning of link i in the mth branch; the $\overset{m}{x}_{i,1}$ axis is along the link length (from $\overset{m}{O}_i$ to $\overset{m}{O}_{i+1}$), and the $\overset{m}{x}_{i,3}$ axis is along the joint axis of this link. Also, the origin of $\overset{m}{\hat{x}}_{i,1}\ \overset{m}{\hat{x}}_{i,2}\ \overset{m}{\hat{x}}_{i,3}$ coordinate system is located at the end of link i in the mth branch and its orientation is

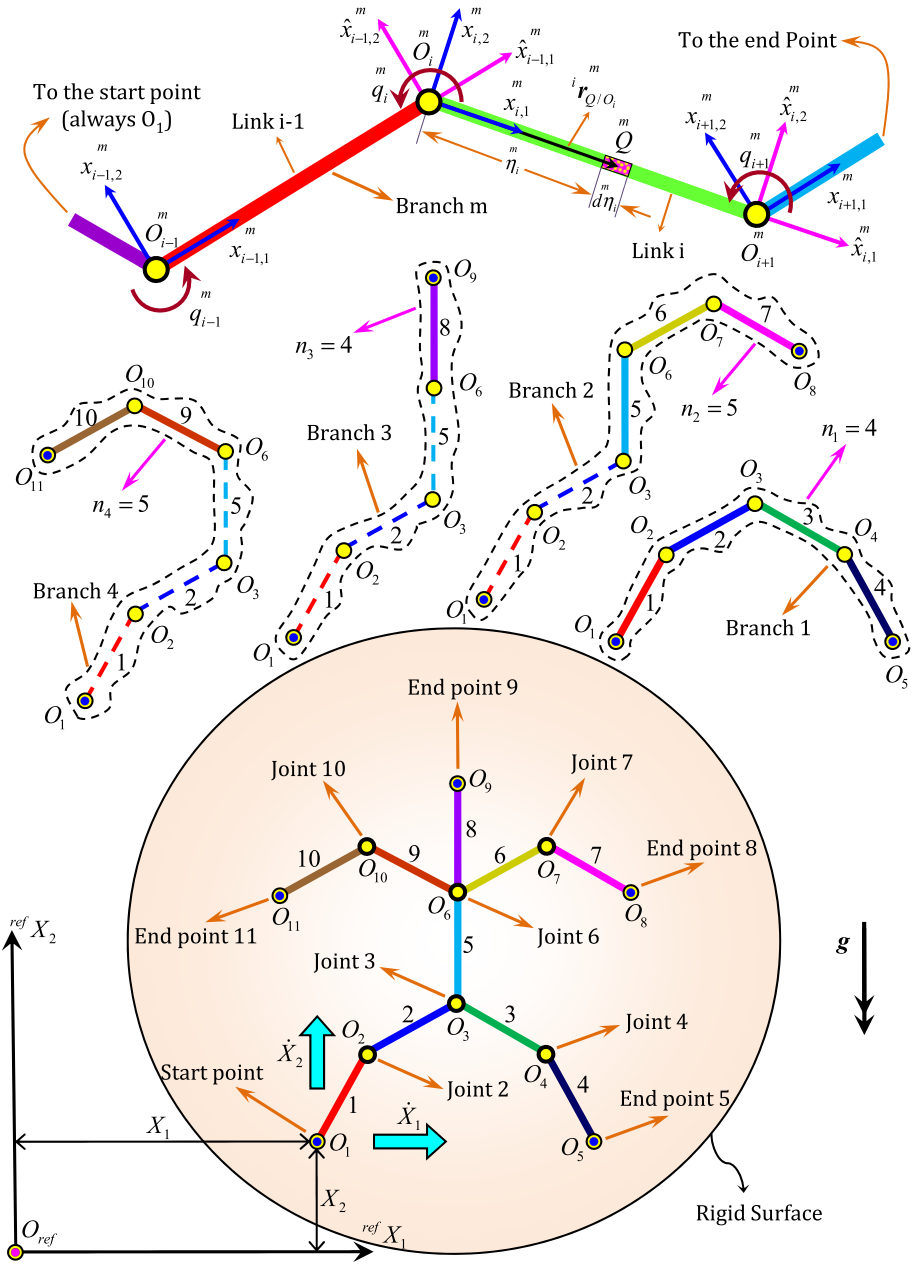

Fig. 1 A tree-type robotic system surrounded by a curved surface

exactly the same as that of the ${}^{m}x_{i,1}\,{}^{m}x_{i,2}\,{}^{m}x_{i,3}$ coordinate system. Moreover, ${}^{\text{ref}}X_1\,{}^{\text{ref}}X_2\,{}^{\text{ref}}X_3$ is the coordinate system which can be considered as the global reference frame. Here, it is assumed that the aforementioned robotic system has a floating base. So, the position and velocity of the start point (i.e., O_1) with respect to the global coordinate system are

respectively denoted by X_j and $\dot{X}_j$, where $j = 1, 2$. Thus, the system's degrees of freedom are expressed as $\mathrm{DOF} = n + 2$, where n degrees of freedom are related to the rotation of links (q_j), and the additional two degrees of freedom are associated with the position of O_1 with respect to the global coordinate system (X_j).

Figure 1 shows an arbitrary differential element $\overset{m}{Q}$ on link i of the mth branch. The position of this element with respect to the $\overset{m}{x}_{i,1}\,\overset{m}{x}_{i,2}\,\overset{m}{x}_{i,3}$ coordinate system is represented by ${}^{i}\overset{m}{\boldsymbol{r}}_{Q/O_i} = \left\{ \overset{m}{\eta}_i \; 0 \; 0 \; 1 \right\}^{\mathrm{T}}$ where $\overset{m}{\eta}_i$ is the distance between $\overset{m}{O}_i$ and differential element $\overset{m}{Q}$. This position vector can be expressed in any coordinate system j if the transformation matrix ${}^{j}\overset{m}{\boldsymbol{T}}_i$ is known, where

$$ {}^{j}\overset{m}{\boldsymbol{T}}_i = \left[\begin{array}{c|c} & {}^{j}\overset{m}{\boldsymbol{x}}_{j,1}^{\mathrm{T}} \cdot {}^{j}\overset{m}{\boldsymbol{r}}_{O_i/O_j} \\ {}^{j}\overset{m}{\boldsymbol{R}}_i & {}^{j}\overset{m}{\boldsymbol{x}}_{j,2}^{\mathrm{T}} \cdot {}^{j}\overset{m}{\boldsymbol{r}}_{O_i/O_j} \\ & {}^{j}\overset{m}{\boldsymbol{x}}_{j,3}^{\mathrm{T}} \cdot {}^{j}\overset{m}{\boldsymbol{r}}_{O_i/O_j} \\ \hline \boldsymbol{0}^{\mathrm{T}} & 1 \end{array}\right], \tag{1} $$

${}^{j}\overset{m}{\boldsymbol{R}}_i$ is a 3×3 rotation matrix which indicates the orientation of the $\overset{m}{x}_{i,1}\,\overset{m}{x}_{i,2}\,\overset{m}{x}_{i,3}$ coordinate system with respect to $\overset{m}{x}_{j,1}\,\overset{m}{x}_{j,2}\,\overset{m}{x}_{j,3}$, ${}^{j}\overset{m}{\boldsymbol{r}}_{O_i/O_j}$ is the 4×1 position vector of $\overset{m}{O}_i$ with respect to $\overset{m}{O}_j$, expressed in the jth body's local coordinate system of the mth branch ($\overset{m}{x}_{j,1}\,\overset{m}{x}_{j,2}\,\overset{m}{x}_{j,3}$); and ${}^{j}\overset{m}{\boldsymbol{x}}_{j,1} = \left\{ 1\,0\,0\,0 \right\}^{\mathrm{T}}$, ${}^{j}\overset{m}{\boldsymbol{x}}_{j,2} = \left\{ 0\,1\,0\,0 \right\}^{\mathrm{T}}$, ${}^{j}\overset{m}{\boldsymbol{x}}_{j,3} = \left\{ 0\,0\,1\,0 \right\}^{\mathrm{T}}$ and $\boldsymbol{0} = \left\{ 0\,0\,0 \right\}^{\mathrm{T}}$. Now, the position of $\overset{m}{Q}$ with respect to the global coordinate system can be defined as

$$ {}^{\mathrm{ref}}\overset{m}{\boldsymbol{r}}_{Q/O_{\mathrm{ref}}} = {}^{\mathrm{ref}}\overset{m}{\boldsymbol{T}}_i \, {}^{i}\overset{m}{\boldsymbol{r}}_{Q/O_i} = \overset{m}{\boldsymbol{T}}_i \, {}^{i}\overset{m}{\boldsymbol{r}}_{Q/O_i} . \tag{2} $$

In the above equation, $\overset{m}{\boldsymbol{T}}_i$ can be represented recursively as

$$ \overset{m}{\boldsymbol{T}}_i = \overset{m}{\boldsymbol{T}}_{i-1} \, \overset{m}{\boldsymbol{E}}_{i-1} \, \overset{m}{\boldsymbol{A}}_i \tag{3} $$

where $\overset{m}{\boldsymbol{A}}_i$ is defined as the joint's transformation matrix, which demonstrates the orientation of the $\overset{m}{x}_{i,1}\,\overset{m}{x}_{i,2}\,\overset{m}{x}_{i,3}$ coordinate system with respect to $\overset{m}{\hat{x}}_{i-1,1}\,\overset{m}{\hat{x}}_{i-1,2}\,\overset{m}{\hat{x}}_{i-1,3}$; and $\overset{m}{\boldsymbol{E}}_i$ is the link's transformation matrix, which shows the translation of $\overset{m}{\hat{x}}_{i,1}\,\overset{m}{\hat{x}}_{i,2}\,\overset{m}{\hat{x}}_{i,3}$ coordinate system with respect to $\overset{m}{x}_{i,1}\,\overset{m}{x}_{i,2}\,\overset{m}{x}_{i,3}$. These two matrices can be written as

$$ \overset{m}{\boldsymbol{A}}_i = \left[\begin{array}{cccc} \overset{m}{\boldsymbol{x}}_{i,1}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{x}}}_{i-1,1} & \overset{m}{\boldsymbol{x}}_{i,2}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{x}}}_{i-1,1} & \overset{m}{\boldsymbol{x}}_{i,3}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{x}}}_{i-1,1} & 0 \\ \overset{m}{\boldsymbol{x}}_{i,1}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{x}}}_{i-1,2} & \overset{m}{\boldsymbol{x}}_{i,2}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{x}}}_{i-1,2} & \overset{m}{\boldsymbol{x}}_{i,3}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{x}}}_{i-1,2} & 0 \\ \overset{m}{\boldsymbol{x}}_{i,1}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{x}}}_{i-1,3} & \overset{m}{\boldsymbol{x}}_{i,2}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{x}}}_{i-1,3} & \overset{m}{\boldsymbol{x}}_{i,3}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{x}}}_{i-1,3} & 0 \\ 0 & 0 & 0 & 1 \end{array}\right], \quad m = 1 \ldots n_b; \; i = 1 \ldots n_m; \tag{4} $$

$$\overset{m}{\boldsymbol{E}}_i = \begin{bmatrix} 1 & 0 & 0 & \overset{m}{l}_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad m = 1 \ldots n_b;\ i = 1 \ldots n_m. \tag{5}$$

In the above equation, $\overset{m}{l}_i$ is the length of link i in the mth branch. Since the approach established in this article is based on Gibbs–Appell methodology, it is necessary to obtain the absolute acceleration of differential element $\overset{m}{Q}$. This term can be obtained by twice differentiating Eq. (2) as follows:

$${}^{\text{ref}}\ddot{\boldsymbol{r}}_{Q/O_{\text{ref}}}^{m} = \ddot{\boldsymbol{T}}_i^{m\ i} \boldsymbol{r}_{Q/O_i}^{m}. \tag{6}$$

The first and second time derivatives of $\overset{m}{\boldsymbol{A}}_i$ and $\overset{m}{\boldsymbol{E}}_i$ exist in Eq. (6). These four terms can be represented as

$$\dot{\overset{m}{\boldsymbol{A}}}_i = \frac{\partial \overset{m}{\boldsymbol{A}}_i}{\partial \overset{m}{q}_i} \dot{\overset{m}{q}}_i = \overset{m}{\boldsymbol{U}}_{1i} \dot{\overset{m}{q}}_i, \tag{7}$$

$$\ddot{\overset{m}{\boldsymbol{A}}}_i = \frac{\partial^2 \overset{m}{\boldsymbol{A}}_i}{\partial \overset{m}{q}_i^2} \dot{\overset{m}{q}}_i^2 + \frac{\partial \overset{m}{\boldsymbol{A}}_i}{\partial \overset{m}{q}_i} \ddot{\overset{m}{q}}_i = \overset{m}{\boldsymbol{U}}_{2i} \dot{\overset{m}{q}}_i^2 + \overset{m}{\boldsymbol{U}}_{1i} \ddot{\overset{m}{q}}_i, \tag{8}$$

$$\dot{\overset{m}{\boldsymbol{E}}}_i = \ddot{\overset{m}{\boldsymbol{E}}}_i = [0]_{4\times 4}. \tag{9}$$

To insert the distance between the first point, O_1, and the origin of the reference coordinate system, O_{ref}, into the formulations, the $\overset{m}{E}_0$ transformation matrix is introduced as

$$\overset{m}{\boldsymbol{E}}_0 = \begin{bmatrix} 1 & 0 & 0 & \overset{m}{X}_1 \\ 0 & 1 & 0 & \overset{m}{X}_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Now, the first and second time derivatives of $\overset{m}{\boldsymbol{E}}_0$ can be expressed as

$$\dot{\overset{m}{\boldsymbol{E}}}_0 = \sum_{i=1}^{2} \frac{\partial \overset{m}{\boldsymbol{E}}_0}{\partial \overset{m}{X}_i} \dot{\overset{m}{X}}_i = \sum_{i=1}^{2} \overset{m}{\boldsymbol{U}}_{3i} \dot{\overset{m}{X}}_i, \tag{11}$$

$$\ddot{\overset{m}{\boldsymbol{E}}}_0 = \sum_{i=1}^{2} \overset{m}{\boldsymbol{U}}_{3i} \ddot{\overset{m}{X}}_i. \tag{12}$$

In the next section, Eq. (6) will be applied to determine the Gibbs function for the mth branch.

3 Dynamics of the system in the flying phase

During the flying phase, the considered tree-type robotic system is suspended in the air and has no contact with the curved walls. In this phase, the differential equations of motion for the flying phase can be obtained by the Gibbs–Appell formulation, in which the acceleration energy for each branch is calculated first, and then theses partial terms are added together to get the acceleration energy (Gibbs function) of the whole system (S):

$$S=\sum_{m=1}^{n_b}\sum_{i=1}^{n_m}\int_0^{\overset{m}{l_i}}\left[\frac{1}{2}\overset{m}{\mu_i}\,\mathrm{Tr}\left({}^{\mathrm{ref}}\overset{m}{\ddot{\boldsymbol{r}}}_{Q/O_{\mathrm{ref}}}\cdot{}^{\mathrm{ref}}\overset{m}{\ddot{\boldsymbol{r}}}^{\mathrm{T}}_{Q/O_{\mathrm{ref}}}\right)+\frac{1}{2}\,{}^{i}\overset{m}{\dot{\boldsymbol{\omega}}}_i^{\mathrm{T}}\cdot\overset{m}{\boldsymbol{J}'_i}\,{}^{i}\overset{m}{\dot{\boldsymbol{\omega}}}_i\right]d\overset{m}{\eta_i} \tag{13}$$

where $\mathrm{Tr}\{.\}$ is the trace operator; ${}^{i}\overset{m}{\boldsymbol{\omega}}_i$ is the 3×1 angular velocity of link i in the mth branch; whereas $\overset{m}{\mu_i}$ and $[\overset{m}{\boldsymbol{J}'_i}]_{3\times 3}$ are the mass per unit length and mass moment of inertia per unit length of this link, respectively. By substituting Eq. (6) into Eq. (13), we get

$$S=\sum_{m=1}^{n_b}\sum_{i=1}^{n_m}\mathrm{Tr}\left\{\overset{m}{\ddot{\boldsymbol{T}}_i}\,\overset{m}{\boldsymbol{B}_{1i}}\,\overset{m}{\ddot{\boldsymbol{T}}^{\mathrm{T}}_i}\right\}+\frac{1}{2}\,{}^{i}\overset{m}{\dot{\boldsymbol{\omega}}}_i^{\mathrm{T}}\cdot\overset{m}{\boldsymbol{B}_{2i}}\,{}^{i}\overset{m}{\dot{\boldsymbol{\omega}}}_i+\text{irrelevant terms} \tag{14}$$

where

$$[\overset{m}{\boldsymbol{B}_{1i}}]_{4\times 4}=\frac{1}{2}\int_0^{\overset{m}{l_i}}\overset{m}{\mu_i}\,{}^{i}\boldsymbol{r}\,\overset{m}{}_{Q/O_i}\,{}^{i}\boldsymbol{r}^{\mathrm{T}}_{Q/O_i}\,d\,\overset{m}{\eta_i}, \tag{15}$$

$$[\overset{m}{\boldsymbol{B}_{2i}}]_{3\times 3}=\int_0^{\overset{m}{l_i}}\overset{m}{\boldsymbol{J}'_i}\,d\,\overset{m}{\eta_i}\,. \tag{16}$$

In the Gibbs–Appell methodology, the equations of motion are extracted by differentiating the Gibbs function with respect to a set of independent generalized accelerations. So all the terms in acceleration energy that are not functions of generalized accelerations can be considered as "irrelevant terms" and omitted. In this paper, the angular accelerations of links (i.e., $\overset{m}{\ddot{q}_j}$) and the translational accelerations of O_1 (i.e., $\overset{m}{\ddot{X}_j}$) are chosen as generalized accelerations. So,

$$\frac{\partial S}{\partial\overset{m}{\ddot{q}_j}}=\sum_{i=j}^{n_m}\left[2\,\mathrm{Tr}\left\{\frac{\partial\overset{m}{\ddot{\boldsymbol{T}}_i}}{\partial\overset{m}{\ddot{q}_j}}\,\overset{m}{\boldsymbol{B}_{1i}}\,\overset{m}{\ddot{\boldsymbol{T}}^{\mathrm{T}}_i}\right\}+\frac{\partial{}^{i}\overset{m}{\dot{\boldsymbol{\omega}}}_i^{\mathrm{T}}}{\partial\overset{m}{\ddot{q}_j}}\cdot\overset{m}{\boldsymbol{B}_{2i}}\,{}^{i}\overset{m}{\dot{\boldsymbol{\omega}}}_i\right],\quad m=1\ldots n_b;\ j=1\ldots n_m; \tag{17}$$

$$\frac{\partial S}{\partial\overset{m}{\ddot{X}_j}}=\sum_{i=1}^{n_m}2\,\mathrm{Tr}\left\{\frac{\partial\overset{m}{\ddot{\boldsymbol{T}}_i}}{\partial\overset{m}{\ddot{X}_j}}\,\overset{m}{\boldsymbol{B}_{1i}}\,\overset{m}{\ddot{\boldsymbol{T}}^{\mathrm{T}}_i}\right\},\quad m=1\ldots n_b;\ j=1\ldots 2. \tag{18}$$

Gravity is the only source of potential energy for this branched robotic system. The gravitational potential energy can be represented as

$$V=-\boldsymbol{g}^{\mathrm{T}}\cdot\sum_{m=1}^{n_b}\sum_{i=1}^{n_m}\overset{m}{\boldsymbol{T}_i}\,{}^{i}\overset{m}{\boldsymbol{c}_i} \tag{19}$$

where $\boldsymbol{g} = \left\{ g_{X_1} \; g_{X_2} \; g_{X_3} \; 0 \right\}^{\mathrm{T}}$ is the vector of the gravitational field and $\overset{m}{{}^{i}\boldsymbol{c}_i}$ is defined as

$$\left[\overset{m}{{}^{i}\boldsymbol{c}_i}\right]_{4\times 1} = \int_0^{\overset{m}{l_i}} \overset{m}{\mu_i}\, {}^{i}\boldsymbol{r}_{Q/O_i}\, d\overset{m}{\eta_i}\,. \tag{20}$$

To derive the conservative generalized forces arising from gravity, we need to differentiate potential energy with respect to generalized coordinates:

$$\frac{\partial V}{\partial \overset{m}{q_j}} = -\boldsymbol{g}^{\mathrm{T}} \cdot \sum_{i=j}^{n_m} \frac{\partial \overset{m}{\boldsymbol{T}_i}}{\partial \overset{m}{q_j}}\, {}^{i}\overset{m}{\boldsymbol{c}_i}, \quad m = 1 \ldots n_b; \;\; j = 1 \ldots n_m; \tag{21}$$

$$\frac{\partial V}{\partial \overset{m}{X_j}} = -\boldsymbol{g}^{\mathrm{T}} \cdot \sum_{i=1}^{n_m} \frac{\partial \overset{m}{\boldsymbol{T}_i}}{\partial \overset{m}{X_j}}\, {}^{i}\overset{m}{\boldsymbol{c}_i}, \quad m = 1 \ldots n_b; \;\; j = 1 \ldots 2. \tag{22}$$

The motion equations of tree-type robotic systems in the flying phase are completed by considering the remaining terms of generalized external forces. The only external force in this phase is the torque $\overset{m}{\tau_j}$ which applies to the jth joint in the mth branch. With this assumption, the dynamic motion equations in the flying phase will be completed as follows:

1. The rotational motion equation of the jth link of the mth branch in the flying phase

$$\frac{\partial S}{\partial \overset{m}{\ddot{q}_j}} + \frac{\partial V}{\partial \overset{m}{q_j}} = \overset{m}{\tau_j}, \quad m = 1 \ldots n_b; \;\; j = 1 \ldots n_m; \tag{23}$$

2. The jth translational motion equation of the first joint (O_1) in the flying phase:

$$\frac{\partial S}{\partial \overset{m}{\ddot{X}_j}} + \frac{\partial V}{\partial \overset{m}{X_j}} = 0, \quad m = 1 \ldots n_b; \;\; j = 1 \ldots 2. \tag{24}$$

For the computer simulation of the aforementioned robotic system, the inverse dynamic form of motion equations (Eqs. (23)–(24)) should be transformed into the forward dynamics form. The details are presented in the following section.

3.1 Forward dynamics

In this section, Eqs. (23) and (24) are transformed to the following forward dynamics form:

$$\boldsymbol{I}_f(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} = \Re(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}). \tag{25}$$

In Eq. (25), $\boldsymbol{I}_f(\boldsymbol{\theta})$ is the inertia matrix of this branched robotic system in the flying phase, $\ddot{\boldsymbol{\theta}}$ is the generalized acceleration vector, and $\Re(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is the right-hand side vector that includes the Coriolis and centrifugal forces.

The partial derivatives of $\overset{m}{\boldsymbol{T}_i}$, $\overset{m}{\ddot{\boldsymbol{T}}_i}$ and $\overset{m}{{}^{i}\dot{\boldsymbol{\omega}}_i}$ with respect to generalized coordinates and accelerations appear in Eqs. (17), (18), (21) and (22). To evaluate these derivatives, let us express $\boldsymbol{T}_i$ in two different forms:

$$\overset{m}{\boldsymbol{T}_i} = \overset{m}{\boldsymbol{A}_0}\,\overset{m}{\boldsymbol{E}_0}\,\overset{m}{\boldsymbol{A}_1}\,\overset{m}{\boldsymbol{E}_1} \ldots \overset{m}{\boldsymbol{A}_h}\,\overset{m}{\boldsymbol{E}_h} \ldots\ldots \overset{m}{\boldsymbol{E}_{i-1}}\,\overset{m}{\boldsymbol{A}_i} = \overset{m}{\hat{\boldsymbol{T}}_{h-1}}\,\overset{m}{\boldsymbol{A}_h}\,{}^{h}\overset{m}{\tilde{\boldsymbol{T}}_i} = \overset{m}{\boldsymbol{T}_h}\,\overset{m}{\boldsymbol{E}_h}\,{}^{h}\overset{m}{\boldsymbol{T}_i} \tag{26}$$

where $\overset{m}{\boldsymbol{A}}_0$ is the identity matrix that is inserted into transformation matrix $\overset{m}{\boldsymbol{T}}_i$ to ensure an organized arrangement. Now, the summed forms of $\overset{m}{\ddot{\boldsymbol{T}}}_i$ and ${}^i\overset{m}{\boldsymbol{\omega}}_i$ are expressed as

$$\overset{m}{\ddot{\boldsymbol{T}}}_i = \overset{m}{\ddot{\boldsymbol{T}}}_{s,i} + \overset{m}{\ddot{\boldsymbol{T}}}_{v,i}, \tag{27}$$

$${}^i\overset{m}{\boldsymbol{\omega}}_i = {}^i\overset{m}{\boldsymbol{\omega}}_{s,i} + {}^i\overset{m}{\boldsymbol{\omega}}_{v,i} \tag{28}$$

where $\overset{m}{\ddot{\boldsymbol{T}}}_{s,i}$ and ${}^i\overset{m}{\boldsymbol{\omega}}_{s,i}$ signify those terms of $\overset{m}{\ddot{\boldsymbol{T}}}_i$ and ${}^i\overset{m}{\boldsymbol{\omega}}_i$ that contain the generalized accelerations; while $\overset{m}{\ddot{\boldsymbol{T}}}_{v,i}$ and ${}^i\overset{m}{\boldsymbol{\omega}}_{v,i}$ denote those terms of $\overset{m}{\ddot{\boldsymbol{T}}}_i$ and ${}^i\overset{m}{\boldsymbol{\omega}}_i$ that do not include $\overset{m}{\ddot{q}}_j$ and $\overset{m}{\ddot{X}}_j$ as generalized accelerations. These four terms can be presented as

$$\overset{m}{\ddot{\boldsymbol{T}}}_{s,i} = \sum_{k=1}^{i} \overset{m}{\hat{\boldsymbol{T}}}_{k-1}\, \overset{m}{\boldsymbol{U}}_{1k}\, {}^k\overset{m}{\tilde{\boldsymbol{T}}}_i\, \overset{m}{\ddot{q}}_k + \sum_{k=1}^{2} \overset{m}{\boldsymbol{T}}_0\, \overset{m}{\boldsymbol{U}}_{3k}\, {}^0\overset{m}{\boldsymbol{T}}_i\, \overset{m}{\ddot{X}}_k, \tag{29}$$

$$\begin{aligned}\overset{m}{\ddot{\boldsymbol{T}}}_{v,i} = {} & \sum_{k=1}^{i} \Big(\overset{m}{\dot{\hat{\boldsymbol{T}}}}_{k-1}\, \overset{m}{\boldsymbol{U}}_{1k}\, {}^k\overset{m}{\tilde{\boldsymbol{T}}}_i + \overset{m}{\hat{\boldsymbol{T}}}_{k-1}\, \overset{m}{\boldsymbol{U}}_{2k}\, {}^k\overset{m}{\tilde{\boldsymbol{T}}}_i\, \overset{m}{\dot{q}}_k + \overset{m}{\hat{\boldsymbol{T}}}_{k-1}\, \overset{m}{\boldsymbol{U}}_{1k}\, {}^k\overset{m}{\dot{\tilde{\boldsymbol{T}}}}_i \Big)\, \overset{m}{\dot{q}}_k \\ & + \sum_{k=1}^{2} \Big(\overset{m}{\dot{\boldsymbol{T}}}_0\, \overset{m}{\boldsymbol{U}}_{3k}\, {}^0\overset{m}{\boldsymbol{T}}_i + \overset{m}{\boldsymbol{T}}_0\, \overset{m}{\boldsymbol{U}}_{3k}\, {}^0\overset{m}{\dot{\boldsymbol{T}}}_i \Big)\, \overset{m}{\dot{X}}_k,\end{aligned} \tag{30}$$

$${}^i\overset{m}{\boldsymbol{\omega}}_{s,i} = \sum_{k=1}^{i} {}^i\overset{m}{\boldsymbol{R}}_k\, {}^k\overset{m}{\boldsymbol{x}}'_{k,3}\, \overset{m}{\ddot{q}}_k, \tag{31}$$

$${}^i\overset{m}{\boldsymbol{\omega}}_{v,i} = \sum_{k=1}^{i-1} {}^i\overset{m}{\boldsymbol{R}}_k\, {}^k\overset{m}{\boldsymbol{\omega}}_k \times {}^i\overset{m}{\boldsymbol{R}}_{k+1}\, {}^{k+1}\overset{m}{\boldsymbol{x}}'_{k+1,3}\, \overset{m}{\dot{q}}_{k+1} \tag{32}$$

where ${}^k\overset{m}{\boldsymbol{x}}'_{k,3} = \{0\ 0\ 1\}^{\mathrm{T}}$. Now, the partial derivatives of $\overset{m}{\boldsymbol{T}}_i$, $\overset{m}{\ddot{\boldsymbol{T}}}_i$ and ${}^i\overset{m}{\boldsymbol{\omega}}_i$ with respect to generalized coordinates and generalized accelerations can be written as

$$\frac{\partial \overset{m}{\ddot{\boldsymbol{T}}}_i}{\partial \overset{m}{\ddot{q}}_j} = \frac{\partial \overset{m}{\boldsymbol{T}}_i}{\partial \overset{m}{q}_j} = \overset{m}{\hat{\boldsymbol{T}}}_{j-1}\, \overset{m}{\boldsymbol{U}}_{1j}\, {}^j\overset{m}{\tilde{\boldsymbol{T}}}_i, \tag{33}$$

$$\frac{\partial \overset{m}{\ddot{\boldsymbol{T}}}_i}{\partial \overset{m}{\ddot{X}}_j} = \frac{\partial \overset{m}{\boldsymbol{T}}_i}{\partial \overset{m}{X}_j} = \overset{m}{\boldsymbol{T}}_0\, \overset{m}{\boldsymbol{U}}_{3j}\, {}^0\overset{m}{\boldsymbol{T}}_i, \tag{34}$$

$$\frac{\partial {}^i\overset{m}{\boldsymbol{\omega}}_i}{\partial \overset{m}{\ddot{q}}_j} = {}^i\overset{m}{\boldsymbol{R}}_j\, {}^j\overset{m}{\boldsymbol{x}}'_{j,3}. \tag{35}$$

3.1.1 Building the inertia matrix of the mth branch

To build the inertia matrix for the *m*th branch of this tree-type robotic system in the flying phase, it is necessary to substitute Eqs. (27) and (28) and also Eqs. (33)–(35) into Eqs. (23)

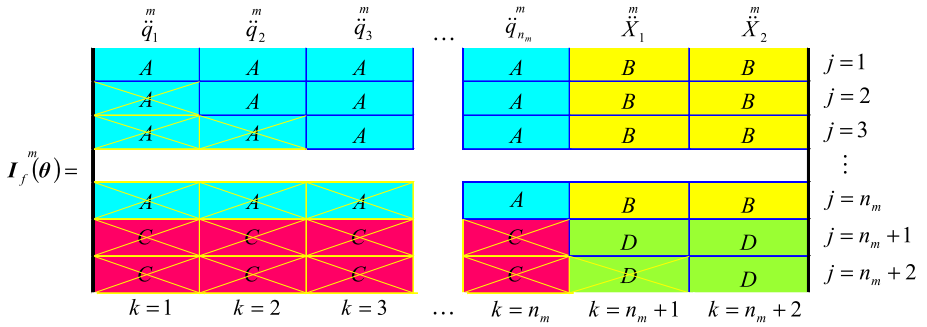

Fig. 2 Inertia matrix of the mth branch in the flying phase

and (24). Then, all the terms that contain generalized accelerations, i.e., $\ddot{q}_j^m$ and $\ddot{X}_j^m$, should be maintained on the left-hand side of the equal sign and all the remaining terms should be transferred to the right-hand side. By arranging the left-hand side terms in a matrix form, the inertia matrix for the mth branch of this system will be achieved. These procedures will be described in the following sections.

Generalized accelerations in the rotational motion equations of the mth branch: In Eq. (23), all the terms which include $\ddot{q}_k^m$ and $\ddot{X}_k^m$ as their coefficients can be grouped as

$$\left[\sum_{k=1}^{n_m} \underbrace{2\,\mathrm{Tr}\{\hat{\boldsymbol{T}}_{j-1}^m \boldsymbol{U}_{1j}^m {}^j\tilde{\boldsymbol{F}}_k^m \boldsymbol{U}_{1k}^{m\mathrm{T}} \hat{\boldsymbol{T}}_{k-1}^{m\mathrm{T}}\} + {}^j\boldsymbol{x}_{j,3}'^{\mathrm{T}} \cdot {}^j\boldsymbol{\Pi}_k^m {}^k\boldsymbol{x}_{k,3}'^m}_{A}\right] \ddot{q}_k^m$$

$$+\left[\sum_{k=n_m+1}^{n_m+2} \underbrace{2\,\mathrm{Tr}\{\hat{\boldsymbol{T}}_{j-1}^m \boldsymbol{U}_{1j}^m {}^j\boldsymbol{F}_0^m \boldsymbol{U}_{3(k-n_m)}^{m\mathrm{T}} \boldsymbol{T}_0^{m\mathrm{T}}\}}_{B}\right] \ddot{X}_{k-n_m}^m, \quad m=1\ldots n_b;\ j=1\ldots n_m, \quad \text{(I)}$$

where

$${}^j\tilde{\boldsymbol{F}}_k^m = \sum_{i=\max(k,j)}^{n_m} {}^j\tilde{\boldsymbol{T}}_i^m \boldsymbol{B}_{1i}^m {}^k\tilde{\boldsymbol{T}}_i^{m\mathrm{T}}, \tag{36}$$

$${}^j\boldsymbol{F}_0^m = \sum_{i=j}^{n_m} {}^j\tilde{\boldsymbol{T}}_i^m \boldsymbol{B}_{1i}^m {}^0\boldsymbol{T}_i^{m\mathrm{T}}, \tag{37}$$

$${}^j\boldsymbol{\Pi}_k^m = \sum_{i=\max(j,k)}^{n_m} {}^j\boldsymbol{R}_i^m \boldsymbol{B}_{2i}^m {}^i\boldsymbol{R}_k^m. \tag{38}$$

The constructive terms of Eq. (I), which are characterized by 'A' and 'B', constitute a portion of the inertia matrix of the mth branch as shown in Fig. 2.

Generalized accelerations in the translational motion equations of the mth branch: In Eq. (24), all the expressions that include $\ddot{q}_k^m$ and $\ddot{X}_k^m$ should be gathered as

$$\left[\sum_{k=1}^{n_m} \underbrace{2\,\mathrm{Tr}\{\boldsymbol{T}_0^m \boldsymbol{U}_{3(j-n_m)}^m {}^0\tilde{\boldsymbol{F}}_k^m \boldsymbol{U}_{1k}^{m\mathrm{T}} \hat{\boldsymbol{T}}_{k-1}^{m\mathrm{T}}\}}_{C}\right] \ddot{q}_k^m$$

$$+\left[\sum_{k=n_m+1}^{n_m+2}\underbrace{2\,\mathrm{Tr}\{\overset{m}{\boldsymbol{T}}_0\,\overset{m}{\boldsymbol{U}}_{3(j-n_m)}\,{}^0\hat{\boldsymbol{F}}_0\,\overset{m}{\boldsymbol{U}}^{\mathrm{T}}_{3(k-n_m)}\,\overset{m}{\boldsymbol{T}}^{\mathrm{T}}_0\}}_{D}\right]\overset{m}{\ddot{X}}_{k-n_m},$$

$$m=1\ldots n_b;\ \ j=n_m+1\ldots n_m+2, \tag{II}$$

where

$${}^0\overset{m}{\hat{\boldsymbol{F}}}_0=\sum_{i=1}^{n_m}{}^0\overset{m}{\boldsymbol{T}}_i\,\overset{m}{\boldsymbol{B}}_{1i}\,{}^0\overset{m}{\boldsymbol{T}}^{\mathrm{T}}_i, \tag{39}$$

$${}^0\overset{m}{\bar{\boldsymbol{F}}}_k=\sum_{i=k}^{n_m}{}^0\overset{m}{\boldsymbol{T}}_i\,\overset{m}{\boldsymbol{B}}_{1i}\,{}^k\overset{m}{\tilde{\boldsymbol{T}}}^{\mathrm{T}}_i. \tag{40}$$

In the inertia matrix of the mth branch, the position of Eq. (II) is indicated by 'C' and 'D'. The symmetry of this matrix can be shown by expanding its elements. So, it is not necessary to reevaluate the crossed out regions in Fig. 2.

3.1.2 Building the right-hand side vector of the motion equations of the mth branch

Here, the right-hand side vector of the motion equations of the mth branch is constructed. In Eq. (23), if all the expressions that do not contain $\overset{m}{\ddot{q}}_k$ and $\overset{m}{\ddot{X}}_k$ are moved to the right-hand side of the equal sign, we get

$$\overset{m}{R}_{q_j}=\underbrace{\overset{m}{\tau}_j+\sum_{i=j}^{n_m}\left[\boldsymbol{g}^{\mathrm{T}}\cdot\frac{\partial\overset{m}{\boldsymbol{T}}_i}{\partial\overset{m}{q}_j}\,{}^i\overset{m}{\boldsymbol{c}}_i-2\,\mathrm{Tr}\left\{\frac{\partial\overset{m}{\ddot{\boldsymbol{T}}}_i}{\partial\overset{m}{\ddot{q}}_j}\,\overset{m}{\boldsymbol{B}}_{1i}\,\overset{m}{\ddot{\boldsymbol{T}}}^{\mathrm{T}}_{v,i}\right\}-\frac{\partial{}^i\overset{m}{\dot{\boldsymbol{\omega}}}_i^{\mathrm{T}}}{\partial\overset{m}{\ddot{q}}_j}\cdot\overset{m}{\boldsymbol{B}}_{2i}\,{}^i\overset{m}{\dot{\boldsymbol{\omega}}}_{v,i}\right]}_{E};$$

$$m=1\ldots n_b,\ \ j=1\ldots n_m. \tag{41}$$

Also, in Eq. (24), all the expressions that do not have generalized accelerations are transferred to the right-hand side of the equal sign. So, we have

$$\overset{m}{R}_{X_{j-n_m}}=\underbrace{\sum_{i=1}^{n_m}\left(\boldsymbol{g}^{\mathrm{T}}\cdot\frac{\partial\overset{m}{\boldsymbol{T}}_i}{\partial\overset{m}{X}_{j-n_m}}\,{}^i\overset{m}{\boldsymbol{c}}_i-2\,\mathrm{Tr}\left\{\frac{\partial\overset{m}{\ddot{\boldsymbol{T}}}_i}{\partial\overset{m}{\ddot{X}}_{j-n_m}}\,\overset{m}{\boldsymbol{B}}_{1i}\,\overset{m}{\ddot{\boldsymbol{T}}}^{\mathrm{T}}_{v,i}\right\}\right)}_{F},$$

$$m=1\ldots n_b;\ \ j=n_m+1\ldots n_m+2. \tag{42}$$

The constructive terms of Eqs. (41) and (42), which are characterized by 'E' and 'F', form the right-hand side vector of the governing equations of the mth branch, as is demonstrated in Fig. 3.

Figures 2 and 3 illustrate the governing equations of the mth branch of the examined tree-type robotic system in the flying phase.

3.1.3 Constructing the global inertia matrix and the global right-hand side vector

In the next step, by integrating the inertia matrices and also the right-hand side vectors of each branch, the global inertia matrix and the global right-hand side vector of the whole system are obtained. Let us explain the involved procedure through an example. The branched

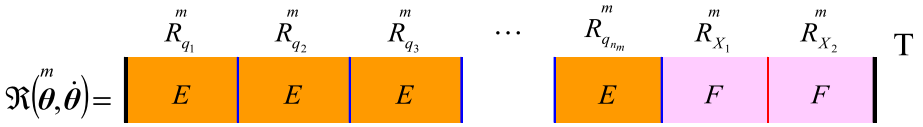

Fig. 3 Right-hand side vector of the motion equations of the mth branch

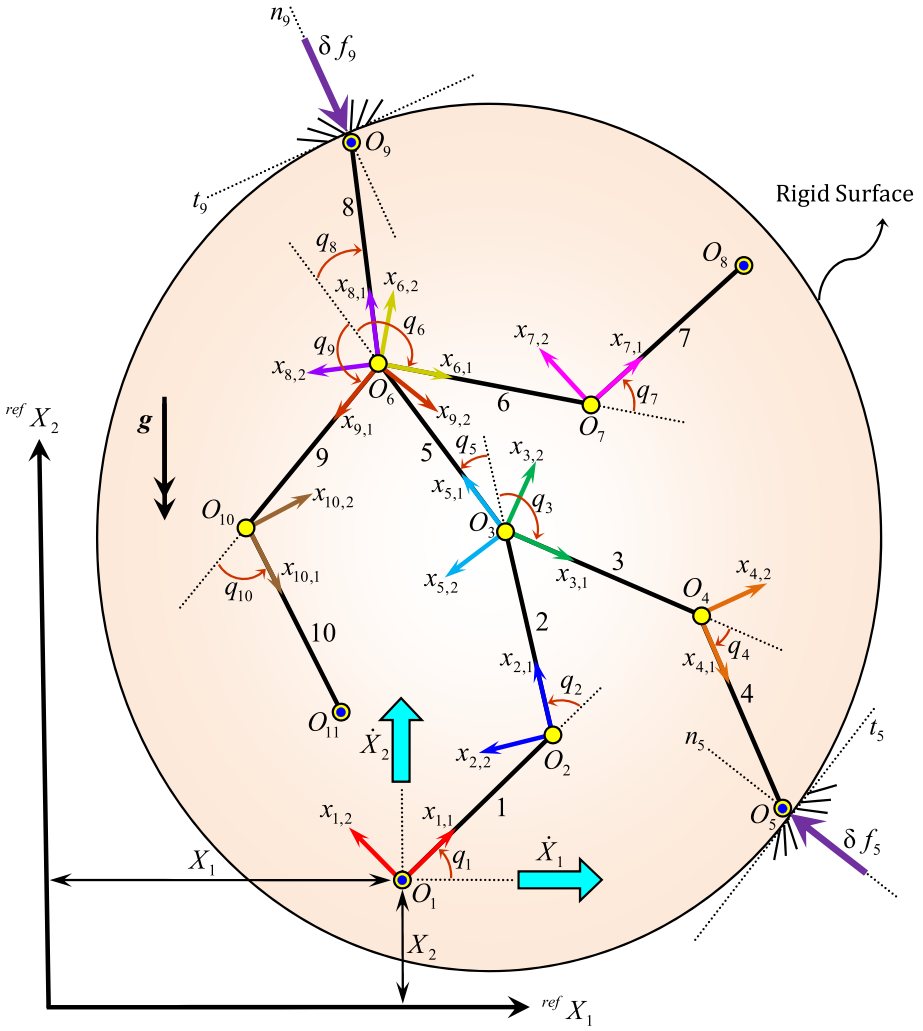

Fig. 4 A tree-type robotic system with 12 degrees of freedom confined within a curved environment

robotic system in Fig. 1, which may characterize a human body, is depicted with more details in Fig. 4. As is shown in Fig. 1, this system was divided into four branches. The first branch consists of four real links (link 1, link 2, link 3 and link 4) and has 6 degrees of freedom: q_1, q_2, q_3, q_4, X_1 and X_2. The second branch contains five links (link 1, link 2, link 5, link 6

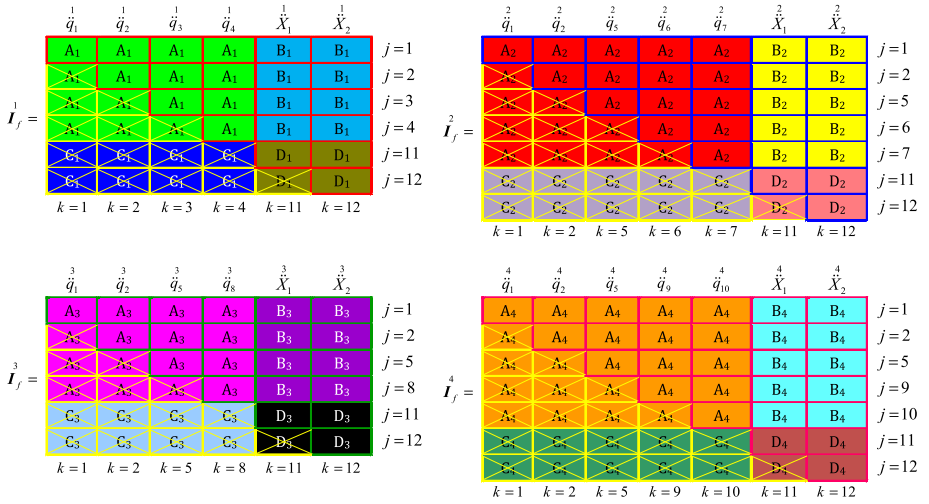

Fig. 5 Inertia matrices for each branch of the tree-type robotic system shown in Fig. 4

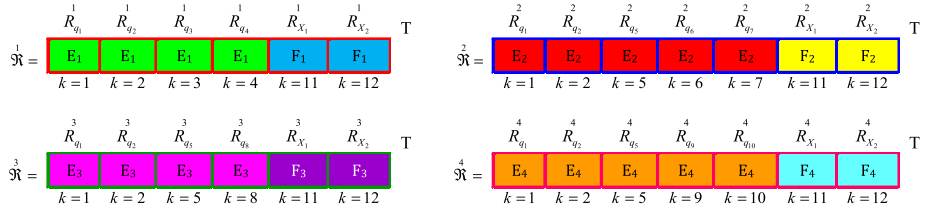

Fig. 6 Right-hand side vectors for each branch of the tree-type robotic system shown in Fig. 4

and link 7), two of which (link 1 and link 2) also appear in the first branch; so, they should be assumed massless. Moreover, branch 2 has 7 degrees of freedom: q_1, q_2, q_5, q_6, q_7, X_1 and X_2. The third branch includes four links (link 1, link 2, link 5 and link 8), with three of them (link 1, link 2 and link 5) appearing in previous branches, and therefore considered massless. This open kinematic chain also has 6 degrees of freedom: q_1, q_2, q_5, q_8, X_1 and X_2. Finally, the forth branch is composed of five links (link 1, link 2, link 5, link 9 and link 10) with three of them (link 1, link 2 and link 5) considered massless. This branch also has 7 degrees of freedom: q_1, q_2, q_5, q_9, q_{10}, X_1 and X_2. Figures 5 and 6 demonstrate the inertia matrices and the right-hand side vectors of all these branches, respectively.

The global inertia matrix of this robotic system will be obtained by assembling the inertia matrices of each branch. This matrix starts out as a 12×12 null matrix; then, each inertia matrix is cycled through and its elements are added to the previous matrix. Also, for building the global right-hand side vector of the abovementioned robotic system, the right-hand side vectors of each branch are integrated. This vector starts out as a 12×1 null vector. Then, each right-hand side vector is cycled through and its elements are added to the previous vector. Figures 7 and 8 respectively illustrate the global inertia matrix and the global right-hand side vector for the tree-type robotic system shown in Fig. 4.

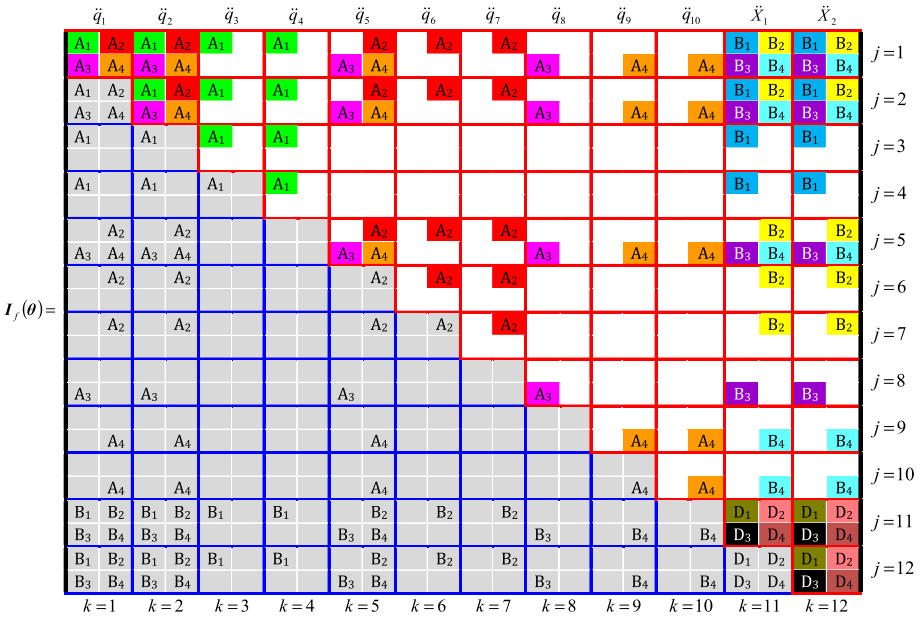

Fig. 7 Global inertia matrix for the tree-type robotic system shown in Fig. 4

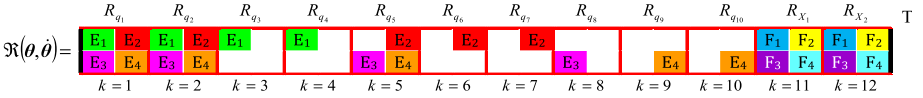

Fig. 8 Global right hand side vector for the tree-type robotic system shown in Fig. 4

4 Dynamics of the system in the impact phase

To accomplish the purpose of this section, first, the absolute velocity of each joint and end point should be evaluated. Obviously, for a branched system composed of n rigid links, there are $n+1$ joints and end points. By applying the Jacobian matrix, the velocities of these $n+1$ points can be represented as

$$\left[{}^{\text{ref}}\dot{\boldsymbol{r}}_{O_i}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\right]_{3\times 1}=\left[{}^{i}\boldsymbol{J}(\boldsymbol{\theta})\right]_{3\times(n+2)}[\dot{\boldsymbol{\theta}}]_{(n+2)\times 1},\quad i=1\ldots n+1, \tag{43}$$

where ${}^{\text{ref}}\dot{\boldsymbol{r}}_{O_i}$ is the absolute velocity of the origin of the ith body's local coordinate system expressed in the inertial reference frame. The Jacobian matrix for a tree-type manipulator composed of n rigid links is depicted in Fig. 9, in which ${}^{i}\boldsymbol{j}_{j,:}$ represents the Jacobian matrix for the jth row ($j=1\ldots 3$) and every column ($1\ldots n+2$) of the ith joint or end point ($i=1\ldots n+1$).

As shown in Fig. 10, an impact happens when any joint or end point of the aforementioned system collides with the curved walls at the angle of attack (α). So, by considering Eq. (25), the system's motion equations in the impact phase can be represented as

$$\boldsymbol{I}_f(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}=\Re(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})+{}^{j}\boldsymbol{J}^{\text{T}}(\boldsymbol{\theta})^{\text{ref}}\boldsymbol{n}_j\delta f_j(t),\quad j=1\ldots\text{Number of Contact Points}, \tag{44}$$

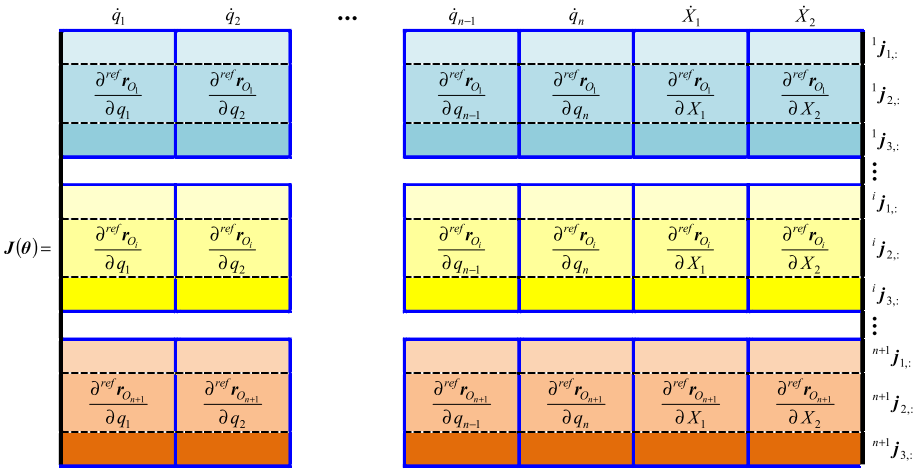

Fig. 9 Jacobian matrix for a tree-type robotic system composed of n rigid links

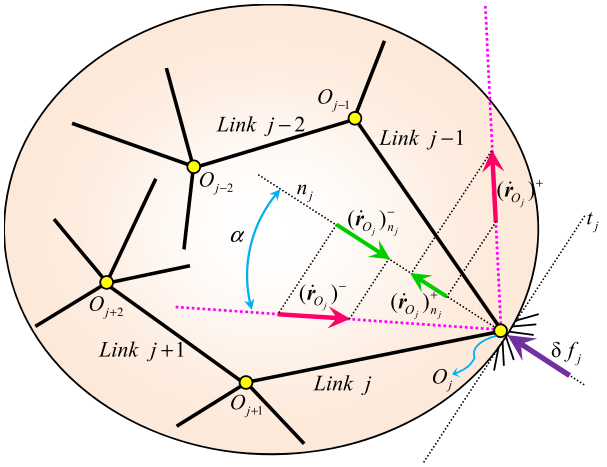

Fig. 10 A branched robotic system in the impact phase

where ${}^{\text{ref}}\boldsymbol{n}_j$ is a unit vector which is normal to the surface at the contact point, and expressed in the inertial reference frame. Also, $\delta f_j(t)$ represents the external force applied on the jth joint or end point during their collision with the curved walls. By integrating Eq. (44) over the impact time (t^-(just before impact) $\rightarrow$ t^+(just after impact)), we get

$$\boldsymbol{I}_f(\boldsymbol{\theta}^-)(\dot{\boldsymbol{\theta}}^+ - \dot{\boldsymbol{\theta}}^-) = {}^j\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{\theta}^-)^{\text{ref}}\boldsymbol{n}_j f_j \tag{45}$$

where $f_j = \int_{t^-}^{t^+} \delta f_j(t)dt$ and $\dot{\boldsymbol{\theta}}^-/\dot{\boldsymbol{\theta}}^+$ are the generalized velocities just before/after an impact. Equation (45) represents $n+2$ equations and $n+2+$ Number of Contact Points unknowns; where $n+2$ of the unknowns are related to the generalized velocities after impact ($\dot{\boldsymbol{\theta}}^+$) and the remaining unknowns are associated with the impulsive forces due to impact (f_j). So, it is essential to find as many additional equations as the number of joints and end points that collide with the curved walls during each impact. These equations can be

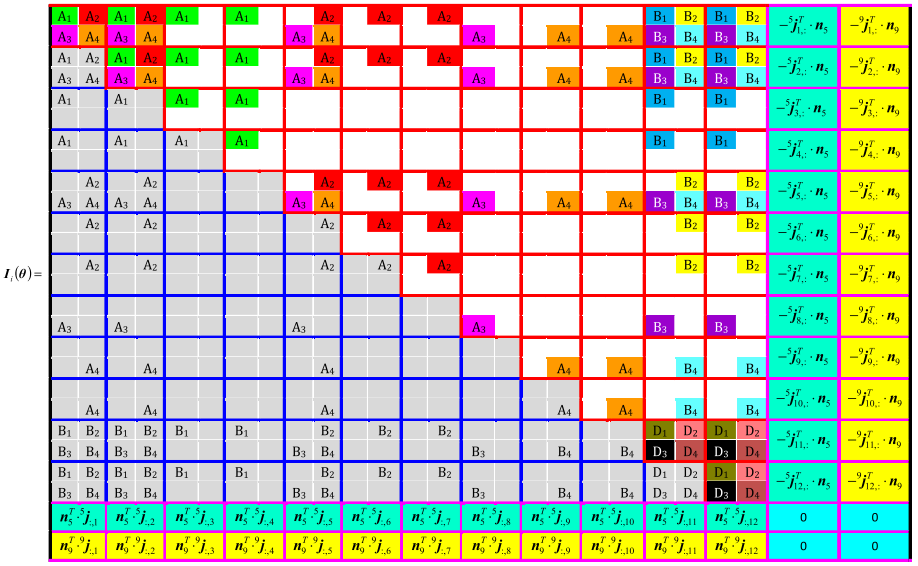

Fig. 11 A sample inertia matrix for the impact phase

obtained by relating the pre- and post-impact velocities of these colliding joints and end points, as follows:

$$\left({}^{\mathrm{ref}}\dot{\boldsymbol{r}}_{O_j}\right)^{+}_{n_j} = -e\left({}^{\mathrm{ref}}\dot{\boldsymbol{r}}_{O_j}\right)^{-}_{n_j} \quad \Rightarrow \quad {}^{\mathrm{ref}}\boldsymbol{n}_j^{\mathrm{T}\,j}\boldsymbol{J}\left(\boldsymbol{\theta}^{-}\right)\dot{\boldsymbol{\theta}}^{+} = -e\,{}^{\mathrm{ref}}\boldsymbol{n}_j^{\mathrm{T}\,j}\boldsymbol{J}\left(\boldsymbol{\theta}^{-}\right)\dot{\boldsymbol{\theta}}^{-}. \tag{46}$$

In the above equation, which is called the Newton's impact law, e denotes the coefficient of restitution. By combining Eqs. (45) and (46) and presenting them in matrix form, we get

$$\underbrace{\left[\begin{array}{c|c} \boldsymbol{I}_f(\boldsymbol{\theta}^{-}) & -{}^{j}\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{\theta}^{-})^{\mathrm{ref}}\boldsymbol{n}_j \\ \hline {}^{\mathrm{ref}}\boldsymbol{n}_j^{\mathrm{T}\,j}\boldsymbol{J}(\boldsymbol{\theta}^{-}) & 0 \end{array}\right]}_{I_i}\left\{\begin{array}{c}\dot{\boldsymbol{\theta}}^{+} \\ f_j\end{array}\right\} = \left[\begin{array}{c} \boldsymbol{I}_f(\boldsymbol{\theta}^{-}) \\ \hline -e\,{}^{\mathrm{ref}}\boldsymbol{n}_j^{\mathrm{T}\,j}\boldsymbol{J}(\boldsymbol{\theta}^{-}) \end{array}\right]\left\{\dot{\boldsymbol{\theta}}^{-}\right\}. \tag{47}$$

For example, if O_5 and O_9 in Fig. 4 simultaneously collide with the curved walls, then two rows and columns should be added to the preliminary inertia matrix derived for the flying phase in order to build the inertia matrix for the impact phase (Fig. 11). Finally, by solving the algebraic equations obtained for the impact phase (Eq. (47)), the unknown variables including the post-impact generalized velocities are determined. These variables along with the post-impact generalized coordinates become the new initial conditions for the next flying phase. It is emphasized that the configuration of the system remains unchanged during the infinitesimal duration of impact. So, the pre-impact generalized coordinates are the same as the post-impact ones, $\boldsymbol{\theta}^{-} = \boldsymbol{\theta}^{+}$.

5 Computational complexity of the proposed algorithm

Using the equations developed in previous sections, a recursive algorithm based on Gibbs–Appell formulation is proposed for systematically and efficiently deriving the motion equa-

Table 1 Number of mathematical operations required for deriving the motion equations by the Gibbs–Appell and Lagrangian methods

Dynamic formulation	Multiplication	Addition
Lagrangian formulation based on 4×4 transformation matrices [38]	$340n_m^2 n_b + 1296 n_m n_b + 504 n_b$	$\frac{463}{2} n_m^2 n_b + \frac{2111}{2} n_m n_b + 350 n_b$
Gibbs–Appell formulation based on 3×3 rotation matrices [39]	$186n_m^2 n_b + 57 n_m n_b - 162 n_b$	$148n_m^2 n_b + 145 n_m n_b - 215 n_b$
Gibbs–Appell formulation based on the combination of 3×3 rotation and 4×4 transformation matrices [present work]	$\frac{507}{2} n_m^2 n_b + \frac{1631}{2} n_m n_b + 297 n_b$	$146n_m^2 n_b + 643 n_m n_b + 165 n_b$

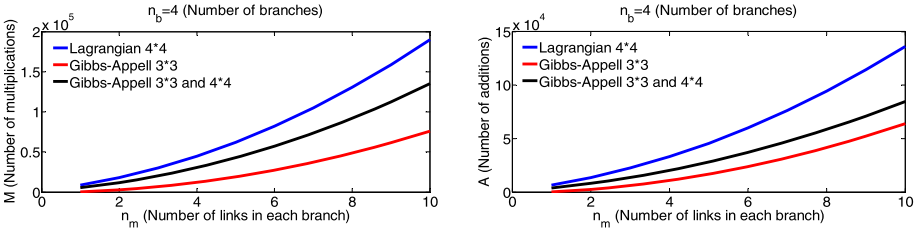

Fig. 12 Computational efficiencies of the Lagrangian and Gibbs–Appell algorithms obtained by considering the same number of branches

tions of tree-type robotic systems in both the flight and impact phases (please see the Appendix). For evaluating the efficiency of the proposed algorithm, the mathematical procedures for deriving the motion equations of tree-type systems by three different methods are presented in Table 1. These three methods include the Lagrangian formulation which uses 4×4 transformation matrices [38], the Gibbs–Appell formulation based on 3×3 rotation matrices [39], and the Gibbs–Appell formulation based on a combination of 3×3 rotation and 4×4 transformation matrices (this work). For comparison, the original algorithms proposed in [38] and [39] are extended to cover the flight and impact phases of tree-type robotic systems. The algorithm in [38] (introduced by Hollerbach) and the one in [39] (proposed by Mata et al.) were originally developed for a single open kinematic chain mechanism with a fixed manipulator base; and they do not consider the effect of system impact with the ground.

In order to compare the computational efficiency of these three algorithms, we first evaluate the effect of increasing the number of links in each branch on the number of mathematical operations necessary for deriving the motion equations of tree topology robotic systems. In this comparison, we consider the same number of branches ($n_b = 4$) for all these algorithms.

According to Fig. 12, as the number of links increases, the Gibbs–Appell model based on 3×3 rotation matrices becomes more efficient than the Lagrangian model based on 4×4 transformation matrices and the Gibbs–Appell model based on the combined use of 3×3 rotation and 4×4 transformation matrices. Now, we consider the case in which the number of links remains constant ($n_m = 4$) and the number of branches varies.

Again, as it is observed in Fig. 13, an increase in the number of branches leads to an increase in the number of mathematical operations needed for deriving the motion equations. As this figure shows, the mathematical operations of the recursive Gibbs–Appell method based on 3×3 rotation matrices are fewer than those of the other two algorithms. However,

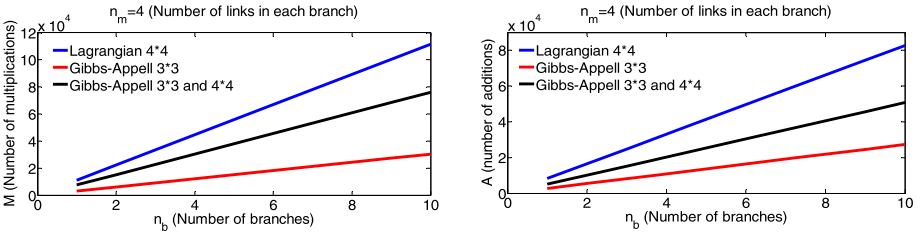

Fig. 13 Computational efficiencies of the Lagrangian and Gibbs–Appell algorithms obtained by considering the same number of links

as we have emphasized before, the algorithm proposed in this paper achieves a compromise between computational complexity and lengthy formulations. In fact, the main advantage of the dynamic algorithms based on 3×3 rotation matrices is their improved computational efficiency due to the use of fewer multiplications and additions; however, these dynamic models suffer from lengthy formulations. On the other hand, the dynamic algorithms based on 4×4 transformation matrices derive more compact motion equations than those based on 3×3 rotation matrices; but computationally, they are more complex. Indeed, this paper has achieved a compromise between computational complexity and lengthy formulations by exploiting the merits of these two algorithms.

6 Computer simulations

In this section, to validate the developed model, the tree-type robotic system shown in Fig. 4 is simulated. The following initial conditions have been chosen so as to produce the anticipated physical responses:

$$q_1|_{t=0} = \frac{\pi}{6}\ \text{rad}; \quad q_2|_{t=0} = \frac{\pi}{6}\ \text{rad}; \quad q_3|_{t=0} = -\frac{2\pi}{3}\ \text{rad}; \quad q_4|_{t=0} = \frac{\pi}{6}\ \text{rad};$$

$$q_5|_{t=0} = \frac{\pi}{6}\ \text{rad}; \quad q_6|_{t=0} = -\frac{2\pi}{3}\ \text{rad}; \quad q_7|_{t=0} = \frac{\pi}{3}\ \text{rad};$$

$$q_8|_{t=0} = 0\ \text{rad}; \quad q_9|_{t=0} = \frac{2\pi}{3}\ \text{rad}; \quad q_{10}|_{t=0} = -\frac{\pi}{3}\ \text{rad};$$

$$\dot{q}_1 = \dot{q}_2 = \dot{q}_3 = \dot{q}_4 = \dot{q}_5 = \dot{q}_6 = \dot{q}_7 = \dot{q}_8 = \dot{q}_9 = \dot{q}_{10}|_{t=0} = 0\ \frac{\text{rad}}{\text{s}};$$

$$X_1|_{t=0} = -1.366\ \text{m}; \quad \dot{X}_1|_{t=0} = 0\ \frac{\text{m}}{\text{s}}; \quad X_2|_{t=0} = -1.366\ \text{m}; \quad \dot{X}_2|_{t=0} = 0\ \frac{\text{m}}{\text{s}}.$$

The necessary parameters for the computer simulation of the abovementioned system are presented in Table 2.

By solving a set of 24 differential equations for the flying phase and also by solving the algebraic equations related to the collision of the examined system with the curved walls, the time responses of the system are obtained and illustrated in Figs. 14 through 18.

Also, the configurations of this tree-type system at different times are depicted in Fig. 18.

As mentioned before, the initial conditions of the aforementioned robotic system are intentionally adopted in such a way that a tree-type system with symmetric configuration is ultimately achieved. Since the robot's configuration and also the external impulsive forces

Table 2 Parameters needed for simulating the branched robotic system depicted in Fig. 4

Parameters	Value	Unit
Link length	$l_1 = l_2 = l_3 = l_4 = l_5 = l_6 = l_7 = l_8 = l_9 = l_{10} = 1$	m
Mass per unit length	$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \mu_7 = \mu_8 = \mu_9 = \mu_{10} = 1$	$\mathrm{kg\,m^{-1}}$
Applied torques on the joints	$\tau_1 = \tau_2 = \tau_3 = \tau_4 = \tau_5 = \tau_6 = \tau_7 = \tau_8 = \tau_9 = \tau_{10} = 0$	N m
Mass moment of inertia per unit length	$\boldsymbol{J'}_1 = \boldsymbol{J'}_2 = \boldsymbol{J'}_3 = \boldsymbol{J'}_4 = \boldsymbol{J'}_5 = \boldsymbol{J'}_6 = \boldsymbol{J'}_7 = \boldsymbol{J'}_8 = \boldsymbol{J'}_9 = \boldsymbol{J'}_{10} = \begin{bmatrix} 5.89 & 0 & 0 \\ 0 & 2.94 & 0 \\ 0 & 0 & 2.94 \end{bmatrix} \times 10^{-5}$	kg m
Acceleration of gravity	$\boldsymbol{g} = 9.81$	$\mathrm{m\,s^{-2}}$
Coefficient of restitution	$e = 1$	–

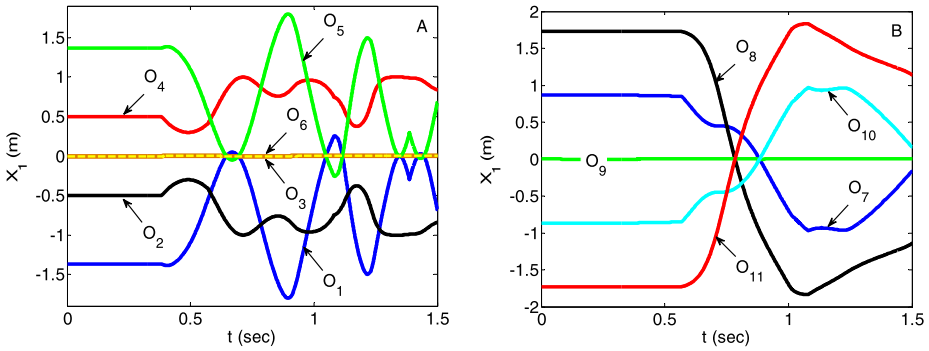

Fig. 14 Time history of the positions of joints and end points in the ${}^{\mathrm{ref}}\mathbf{X}_1$ direction

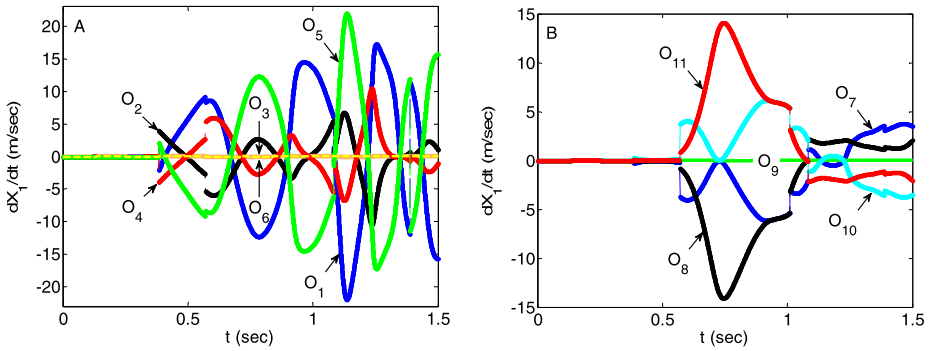

Fig. 15 Time history of the absolute velocities of joints and end points in the ${}^{\mathrm{ref}}\mathbf{X}_1$ direction

acting on its joints and end points are symmetric, the system is expected to remain symmetric during the simulation. The positions and velocities of each joint and end point in global reference frame (${}^{\mathrm{ref}}X_1\,{}^{\mathrm{ref}}X_2\,{}^{\mathrm{ref}}X_3$) are illustrated in Figs. 14 through 17. The symmetrical configuration of the simulated system can be concluded from Fig. 14, where $X_1 = 0$ m is the line of symmetry. The same conclusion could be drawn from Fig. 15, where the absolute velocity of each joint, or end point in the ${}^{\mathrm{ref}}X_1$ direction, indicates the symmetricity of the

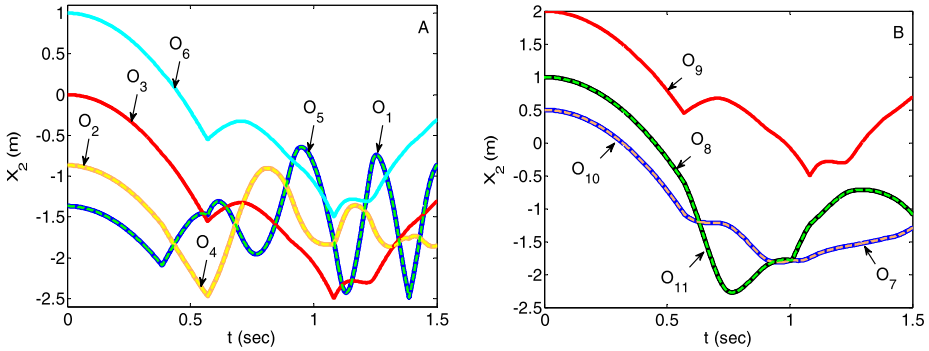

Fig. 16 Time history of the positions of joints and end points in the $^{\text{ref}}\mathbf{X}_2$ direction

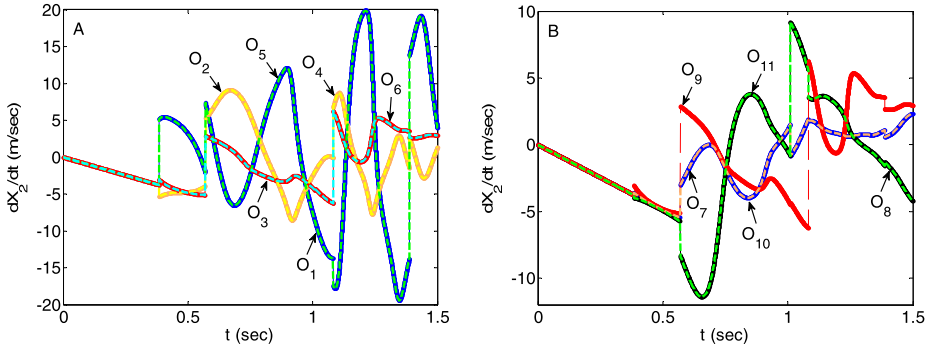

Fig. 17 Time history of the absolute velocities of joints and end points in the $^{\text{ref}}\mathbf{X}_2$ direction

system. Figure 18 displays the collisions of the first and fifth end points, O_1 and O_5, with the curved walls at $t = 0.3851$ s and $t = 1.387$ s. The other end points, O_8 and O_{11}, touch the confining walls at $t = 1.01$ s. Finally, according to this figure, two impacts occur due to the collision of the second and fourth joints, O_2 and O_4, with the surrounding surfaces at $t = 1.08$ s and one impact happens by the third joint, i.e., O_3, striking the curved walls at $t = 1.08$ s. As mentioned before, during the infinitesimal impact times, the system does not have enough time to change its configuration, i.e., $\boldsymbol{\theta}^- = \boldsymbol{\theta}^+$. This fact can be verified in Figs. 14 and 16, where the values of generalized coordinates are equal at pre- and post-impact times. However, the generalized velocities at these moments undergo a jump, as is shown in Figs. 15 and 17.

In order to study the effects of angle of attack and relative velocity on impulses, the values of these parameters are presented in Table 3.

Although, at first, one may conclude that the impulses due to the collision with confining walls increase as the relative velocities before and after impact increase, Table 3 does not confirm this conclusion. For example, in the above table, the relative velocities at the second and third impact moments (0.5695 s and 1.01 s) are almost equal (8.62 m/s and 8.64 m/s); but their impulses are completely different (31.42 N s and 4.12 N s). In fact, impulse is dependent on the relative velocity and the moment of inertia of the whole system. Since the configurations of this robotic system varies at these two impact times (Fig. 18), it is evident that the moment of inertia of the whole system has different values at these two moments.

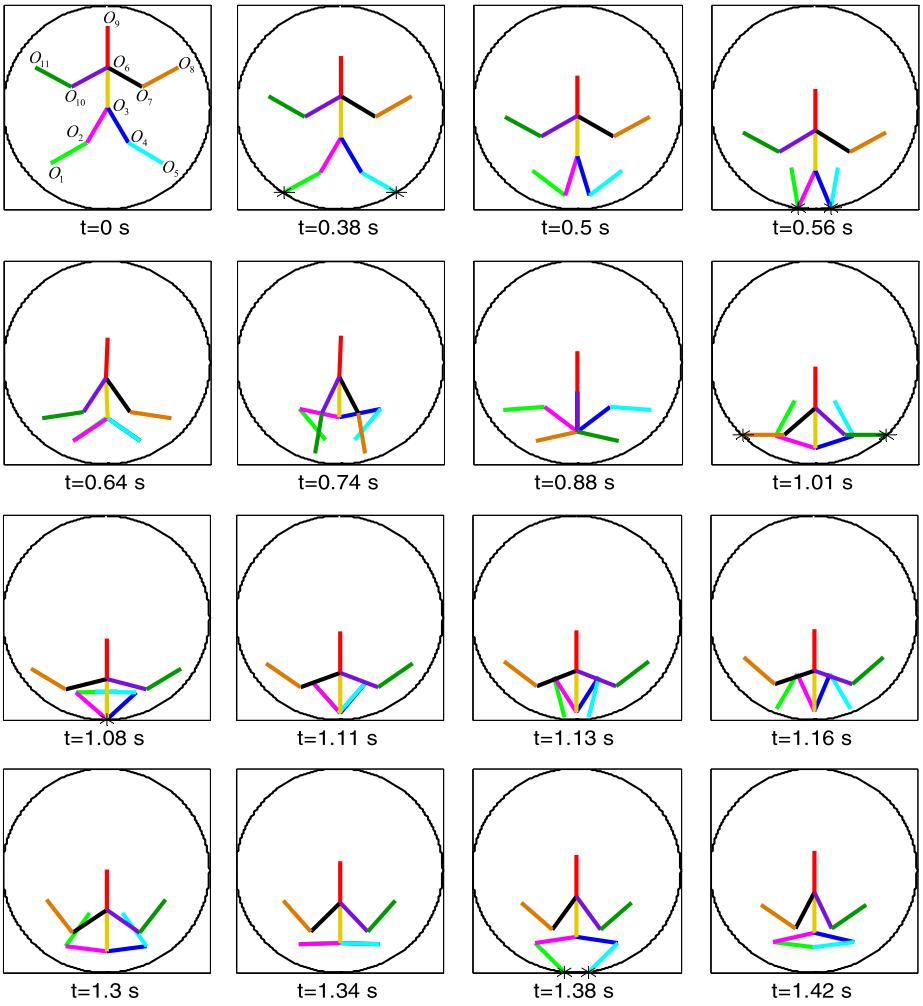

Fig. 18 Configurations of the examined branched system at different times

Table 3 Angle of attack, relative velocity and impulse values in computer simulation

t	0.3851	0.5698	1.01	1.082	1.387
α	33.12°	25.38°	36.87°	0°	34.108°
$\lvert(\dot{\boldsymbol{r}}_{O_j})^{+}_{n_j} - (\dot{\boldsymbol{r}}_{O_j})^{-}_{n_j}\rvert$	6.32	8.62	8.64	12.48	30.14
$f_j = \int_{t^-}^{t^+} \delta f_j(t)\,dt$	5.26 on O_1	31.42 on O_2	4.12 on O_8	57.46 on O_3	15.96 on O_1
	5.26 on O_5	31.42 on O_4	4.12 on O_{11}		15.96 on O_5

In order to study the accuracy of the achieved results, the time history of the mechanical energy of the whole system is investigated. Since there is no energy dissipation in the examined robotic system, the total energy of the system, including its kinetic energy (KE) and potential energy (PE), is expected to remain constant during the simulation. The energy

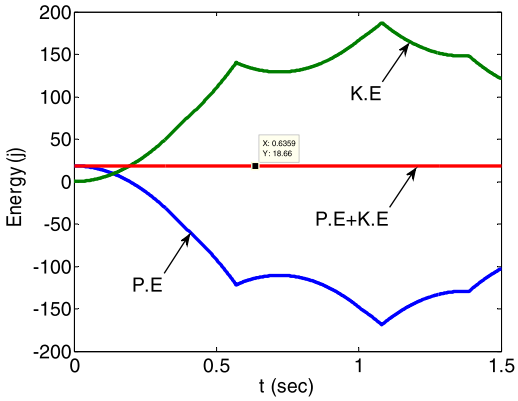

Fig. 19 Energy history of the examined system

history of the system is illustrated in Fig. 19. According to this figure, the total energy of the system (PE+KE) remains constant during the simulation at 18.66 J. This validates the stability of system responses obtained by solving the differential–algebraic motion equations. Finally, it should be noted that the stability of system responses is highly dependent on step size. So, besides employing the "ode45" solver and the EVENTS function available in the MATLAB software, a computer program based on the various Runge–Kutta methods was also used to solve these differential–algebraic equations. The time step in these methods was increased until no significant error was observed. The CPU time for deriving and solving the governing equations with a time step of $\Delta t = 10^{-5}$ s is approximately 40 hr on a PC with an Intel (R) Core (TM) i3-3220 processor running at 3.3 GHz.

7 Conclusions

In this article, the responses of a tree-type robotic manipulator have been studied in two different flying and impact phases. At first, a systematic approach has been used to divide the system into a certain number of open kinematic chains (branches). Then, a recursive algorithm based on the combination of 3×3 rotation and 4×4 transformation matrices has been employed to obtain the governing equations of the system. The main feature of dynamic algorithms based on 3×3 rotation matrices is the improvement in computational efficiency they achieve by using fewer multiplications and additions. However, such dynamic modeling suffers from lengthy formulations. On the other hand, the dynamic algorithms based on 4×4 transformation matrices derive more compact motion equations than 3×3 rotation matrices, but computationally they are highly complex. In fact, this paper has achieved a compromise between computational complexity and lengthy formulations. Finally, to demonstrate the efficiency of the proposed algorithm, a branched robotic system with 12 degrees of freedom has been simulated.

In deriving the motion equations of this complex robotic system, point contact has been assumed. So, for future works, the employed procedure in this paper can be extended to include the effects of higher-order contacts in tree-type robotic systems. Moreover, the branched systems examined here are assumed to have straight rigid links without any curvature. Hence, the proposed method in this paper can be modified, especially in the impact phase, to include the effects of impact in systems constructed of curved links.

Appendix: Proposed algorithm

Here, an algorithm based on the expressions developed in previous sections is presented. In this algorithm, all the cross-products are performed in tensor notation. The calculations are successively carried out through the following steps:

Step 1. Calculating of the compound transformation matrix:

$$\begin{aligned}
&\text{for } m = 1:1:n_b;\ i = 0:1:n_m;\ j = i:1:n_m\\
&\quad \text{if } (i = j) && {}^{i}\overset{m}{\boldsymbol{T}}_{i} = \boldsymbol{I}_{4\times 4};\\
&\quad \text{elseif } (j = i+1) && {}^{i}\overset{m}{\boldsymbol{T}}_{j} = \overset{m}{\boldsymbol{A}}_{i+1};\\
&\quad \text{else} && {}^{i}\overset{m}{\boldsymbol{T}}_{j} = {}^{i}\overset{m}{\boldsymbol{T}}_{j-1}\,\overset{m}{\boldsymbol{E}}_{j-1}\,\overset{m}{\boldsymbol{A}}_{j};
\end{aligned}$$

By knowing ${}^{i}\overset{m}{\boldsymbol{T}}_{j}$, this algorithm can evaluate ${}^{i}\overset{m}{\tilde{\boldsymbol{T}}}_{j}$:

$$\begin{aligned}
&\text{for } m = 1:1:n_b;\ i = 0:1:n_m;\ j = i:1:n_m\\
&\quad \text{if } (i = j) && {}^{i}\overset{m}{\tilde{\boldsymbol{T}}}_{i} = \boldsymbol{I}_{4\times 4};\\
&\quad \text{else} && {}^{i}\overset{m}{\tilde{\boldsymbol{T}}}_{j} = \overset{m}{\boldsymbol{E}}_{i}\,{}^{i}\overset{m}{\boldsymbol{T}}_{j};
\end{aligned}$$

$\overset{m}{\boldsymbol{T}}_{i}$ is calculated by using ${}^{i}\overset{m}{\tilde{\boldsymbol{T}}}_{j}$:

$$\begin{aligned}
&\text{for } m = 1:1:n_b\\
&\quad \overset{m}{\boldsymbol{T}}_{0} = \overset{m}{\boldsymbol{A}}_{0};\\
&\text{for } m = 1:1:n_b;\ i = 1:1:n_m\\
&\quad \overset{m}{\boldsymbol{T}}_{i} = \overset{m}{\boldsymbol{T}}_{0}\,{}^{0}\overset{m}{\tilde{\boldsymbol{T}}}_{i};
\end{aligned}$$

And finally, $\overset{m}{\hat{\boldsymbol{T}}}_{i}$ is presented as

$$\begin{aligned}
&\text{for } m = 1:1:n_b;\ i = 0:1:n_m - 1\\
&\quad \overset{m}{\hat{\boldsymbol{T}}}_{i} = \overset{m}{\boldsymbol{T}}_{i}\,\overset{m}{\boldsymbol{E}}_{i}\,.
\end{aligned}$$

Step 2. The compound rotation matrix is calculated by this algorithm as follows:

$$\begin{aligned}
&\text{for } m = 1:1:n_b;\ i = 2:1:n_m\\
&\quad {}^{i-1}\overset{m}{\boldsymbol{R}}_{i} = \overset{m}{\boldsymbol{A}}_{i}; \quad \& \quad {}^{i}\overset{m}{\boldsymbol{R}}_{i-1} = {}^{i-1}\overset{m}{\boldsymbol{R}}_{i}^{\mathrm{T}};\\
&\text{for } m = 1:1:n_b;\ j = 1:1:n_m\\
&\quad {}^{j}\overset{m}{\boldsymbol{R}}_{j} = \boldsymbol{I}_{3\times 3};
\end{aligned}$$

$$\text{for } m = 1:1:n_b;\ j = 1:1:n_m - 2;\ k = j+2:1:n_m$$
$$^{j}\overset{m}{\boldsymbol{R}}_{k} = {}^{j}\overset{m}{\boldsymbol{R}}_{k-1}\,{}^{k-1}\overset{m}{\boldsymbol{R}}_{k}; \quad \& \quad {}^{k}\overset{m}{\boldsymbol{R}}_{j} = {}^{j}\overset{m}{\boldsymbol{R}}_{k}^{\mathrm{T}}.$$

Step 3. Vectors $^{i}\overset{m}{\boldsymbol{\omega}}_{i}$ and $^{i}\overset{m}{\dot{\boldsymbol{\omega}}}_{v,i}$ can be calculated recursively as

$$\text{for } m = 1:1:n_b$$
$$^{1}\overset{m}{\boldsymbol{\omega}}_{1} = {}^{1}\overset{m}{\boldsymbol{x}}_{1,3}\,\dot{q}_1; \quad \& \quad {}^{1}\overset{m}{\dot{\boldsymbol{\omega}}}_{v,1} = \{0\ 0\ 0\}^{\mathrm{T}};$$
$$\text{for } m = 1:1:n_b;\ i = 2:1:n_m$$
$$^{i}\overset{m}{\boldsymbol{\omega}}_{i} = {}^{i}\overset{m}{\boldsymbol{R}}_{i-1}\,{}^{i-1}\overset{m}{\boldsymbol{\omega}}_{i-1} + {}^{i}\overset{m}{\boldsymbol{x}}_{i,3}\,\overset{m}{\dot{q}}_i$$
$$^{i}\overset{m}{\dot{\boldsymbol{\omega}}}_{v,i} = {}^{i}\overset{m}{\boldsymbol{R}}_{i-1}\big({}^{i-1}\overset{m}{\tilde{\boldsymbol{\omega}}}_{i-1}\,{}^{i-1}\overset{m}{\boldsymbol{R}}_{i}\,{}^{i}\overset{m}{\boldsymbol{x}}_{i,3}\,\overset{m}{\dot{q}}_i + {}^{i-1}\overset{m}{\dot{\boldsymbol{\omega}}}_{v,i-1}\big).$$

Step 4. Calculating $\overset{m}{\boldsymbol{p}}_{j}$, $\overset{m}{\boldsymbol{Q}}_{j}$, $^{i}\overset{m}{\boldsymbol{t}}_{i}$ and $^{i}\overset{m}{\boldsymbol{\chi}}_{i}$ in recursive form as follows:

$$\text{for } m = 1:1:n_b$$
$$\overset{m}{\boldsymbol{p}}_{n} = {}^{n}\overset{m}{\boldsymbol{c}}_{n};$$
$$\text{for } m = 1:1:n_b;\ j = n_m - 1:-1:1$$
$$\overset{m}{\boldsymbol{p}}_{j} = {}^{j}\overset{m}{\boldsymbol{c}}_{j} + \overset{m}{\boldsymbol{E}}_{j}\,\overset{m}{\boldsymbol{A}}_{j+1}\,\overset{m}{\boldsymbol{p}}_{j+1};$$
$$\text{for } m = 1:1:n_b$$
$$\overset{m}{\boldsymbol{Q}}_{n} = \overset{m}{\boldsymbol{B}}_{1n}\,\overset{m}{\ddot{\boldsymbol{T}}}_{v,n}^{\mathrm{T}};$$
$$\text{for } m = 1:1:n_b;\ i = n_m - 1:-1:1$$
$$\overset{m}{\boldsymbol{Q}}_{j} = \overset{m}{\boldsymbol{B}}_{1j}\,\overset{m}{\ddot{\boldsymbol{T}}}_{v,j}^{\mathrm{T}} + \overset{m}{\boldsymbol{E}}_{j}\,\overset{m}{\boldsymbol{A}}_{j+1}\,\overset{m}{\boldsymbol{Q}}_{j+1};$$
$$\text{for } m = 1:1:n_b;\ i = 1:1:n_m$$
$$^{i}\overset{m}{\boldsymbol{t}}_{i} = \overset{m}{\boldsymbol{B}}_{2i}\,{}^{i}\overset{m}{\dot{\vec{\omega}}}_{i};$$
$$\text{for } m = 1:1:n_b$$
$$^{n_m}\overset{m}{\boldsymbol{\chi}}_{n_m} = {}^{n_m}\overset{m}{\boldsymbol{t}}_{n_m}$$
$$\text{for } m = 1:1:n_b;\ j = n_m - 1:-1:1$$
$$^{j}\overset{m}{\boldsymbol{\chi}}_{j} = {}^{j}\overset{m}{\boldsymbol{t}}_{j} + {}^{j}\overset{m}{\boldsymbol{R}}_{j+1}\,{}^{j+1}\overset{m}{\boldsymbol{\chi}}_{j+1}$$

Step 5. Assembling the right-hand sides of motion equations as

$$\begin{aligned}
&\text{for } m = 1:1:n_b;\ j = 1:1:n_m \\
&\quad \overset{m}{R}_{q_j} = \overset{m}{\tau}_j + \boldsymbol{g}^{\mathrm{T}} \cdot \overset{m}{\hat{\boldsymbol{T}}}_{j-1} \overset{m}{\boldsymbol{U}}_{1j} \overset{m}{\boldsymbol{p}}_j - 2\,\mathrm{Tr}\{\overset{m}{\hat{\boldsymbol{T}}}_{j-1} \overset{m}{\boldsymbol{U}}_{1j} \overset{m}{\boldsymbol{Q}}_j\} - {}^{j}\overset{m}{\boldsymbol{x}}'^{\mathrm{T}}_{j,3} \cdot {}^{j}\overset{m}{\boldsymbol{\chi}}_j; \\
&\text{for } m = 1:1:n_b;\ j = n_m + 1:1:n_m + 2 \\
&\quad \overset{m}{R}_{X_j} = \boldsymbol{g}^{\mathrm{T}} \cdot \overset{m}{\boldsymbol{U}}_{3(j-n_m)} \overset{m}{\boldsymbol{A}}_1 \overset{m}{\boldsymbol{p}}_1 - 2\,\mathrm{Tr}\{\overset{m}{\boldsymbol{U}}_{3(j-n_m)} \overset{m}{\boldsymbol{Q}}_1\}.
\end{aligned}$$

Step 6. Preparing the recursive forms of summation variables in the inertia matrix,

- Calculating ${}^{i}\tilde{\boldsymbol{F}}_j$:

$$\begin{aligned}
&\text{for } m = 1:1:n_b;\ j = n_m:-1:1;\ i = j:-1:1 \\
&\quad \text{if } (j = i) \\
&\quad \text{if } (j = n_m) \quad {}^{n_m}\overset{m}{\tilde{\boldsymbol{F}}}_{n_m} = \overset{m}{\boldsymbol{B}}_{1n_m}; \\
&\quad \text{else} \qquad\quad {}^{i}\overset{m}{\tilde{\boldsymbol{F}}}_j = \overset{m}{\boldsymbol{B}}_{1j} + {}^{j}\overset{m}{\tilde{\boldsymbol{F}}}_{j+1}(\overset{m}{\boldsymbol{E}}_i \overset{m}{\boldsymbol{A}}_{i+1})^{\mathrm{T}}; \\
&\quad \text{else} \qquad\quad {}^{i}\overset{m}{\tilde{\boldsymbol{F}}}_j = \overset{m}{\boldsymbol{E}}_i \overset{m}{\boldsymbol{A}}_{i+1}\, {}^{i+1}\overset{m}{\tilde{\boldsymbol{F}}}_j; \qquad\qquad {}^{j}\overset{m}{\tilde{\boldsymbol{F}}}_i = {}^{i}\overset{m}{\tilde{\boldsymbol{F}}}_j^{\mathrm{T}};
\end{aligned}$$

- Calculating ${}^{j}\overset{m}{\boldsymbol{F}}_0$:

$$\begin{aligned}
&\text{for } m = 1:1:n_b;\ j = 1:1:n_m \\
&\quad {}^{j}\overset{m}{\boldsymbol{F}}_0 = {}^{j}\overset{m}{\tilde{\boldsymbol{F}}}_1 \overset{m}{\boldsymbol{A}}_1^{\mathrm{T}};
\end{aligned}$$

- Calculating ${}^{0}\overset{m}{\hat{\boldsymbol{F}}}_0$:

$$\begin{aligned}
&\text{for } m = 1:1:n_b \\
&\quad {}^{0}\overset{m}{\hat{\boldsymbol{F}}}_0 = \overset{m}{\boldsymbol{A}}_1\, {}^{1}\overset{m}{\tilde{\boldsymbol{F}}}_1 \overset{m}{\boldsymbol{A}}_1^{\mathrm{T}};
\end{aligned}$$

- Calculating ${}^{j}\overset{m}{\boldsymbol{\Pi}}_k$:

$$\begin{aligned}
&\text{for } m = 1:1:n_b;\ k = n_m:-1:1;\ j = k:-1:1 \\
&\quad \text{if } (k = j) \\
&\quad \text{if } (k = n_m) \quad {}^{n}\overset{m}{\boldsymbol{\Pi}}_n = \overset{m}{\boldsymbol{B}}_{2n_m}; \\
&\quad \text{else} \qquad\quad {}^{k}\overset{m}{\boldsymbol{\Pi}}_k = \overset{m}{\boldsymbol{B}}_{2k} + {}^{k}\overset{m}{\boldsymbol{\Pi}}_{k+1}\, {}^{k+1}\overset{m}{\boldsymbol{R}}_k; \\
&\quad \text{else} \qquad\quad {}^{j}\overset{m}{\boldsymbol{\Pi}}_k = {}^{j}\overset{m}{\boldsymbol{R}}_{j+1}\, {}^{j+1}\overset{m}{\boldsymbol{\Pi}}_k; \quad \& \quad {}^{k}\overset{m}{\boldsymbol{\Pi}}_j = {}^{j}\overset{m}{\boldsymbol{\Pi}}_k^{\mathrm{T}}.
\end{aligned}$$

Step 7. Assembling the inertia matrix for the whole system as follows:

$$\text{for } m = 1:1:n_b;\ j = 1:1:n_m;\ k = j:1:n_m$$

$$\text{if } (j = 1 \ \&\ k = 1) \quad \overset{m}{I}_{jk} = 2\,\mathrm{Tr}\big\{\overset{m}{\boldsymbol{U}}_{1j}\,{}^{j}\overset{m}{\tilde{\boldsymbol{F}}}_{k}\,\overset{m}{\boldsymbol{U}}_{1k}^{\mathrm{T}}\big\} + {}^{j}\overset{m}{\boldsymbol{x}}_{j,3}'^{\mathrm{T}}\cdot{}^{j}\overset{m}{\boldsymbol{\Pi}}_{k}\,{}^{k}\overset{m}{\boldsymbol{x}}_{k,3}';$$

$$\text{elseif } (j = 1 \ \&\ k \neq 1) \quad \overset{m}{I}_{jk} = 2\,\mathrm{Tr}\big\{\overset{m}{\boldsymbol{U}}_{1j}\,{}^{j}\overset{m}{\tilde{\boldsymbol{F}}}_{k}\,\overset{m}{\boldsymbol{U}}_{1k}^{\mathrm{T}}\,\overset{m}{\hat{\boldsymbol{T}}}_{k-1}^{\mathrm{T}}\big\} + {}^{j}\overset{m}{\boldsymbol{x}}_{j,3}'^{\mathrm{T}}\cdot{}^{j}\overset{m}{\boldsymbol{\Pi}}_{k}\,{}^{k}\overset{m}{\boldsymbol{x}}_{k,3}';\qquad \overset{m}{I}_{kj} = \overset{m}{I}_{jk};$$

$$\text{else} \quad \overset{m}{I}_{jk} = 2\,\mathrm{Tr}\big\{\overset{m}{\hat{\boldsymbol{T}}}_{j-1}\,\overset{m}{\boldsymbol{U}}_{j}\,{}^{j}\overset{m}{\tilde{\boldsymbol{F}}}_{k}\,\overset{m}{\boldsymbol{U}}_{k}^{\mathrm{T}}\,\overset{m}{\hat{\boldsymbol{T}}}_{k-1}^{\mathrm{T}}\big\} + {}^{j}\overset{m}{\boldsymbol{x}}_{j,3}'^{\mathrm{T}}\cdot{}^{j}\overset{m}{\boldsymbol{\Pi}}_{k}\,{}^{k}\overset{m}{\boldsymbol{x}}_{k,3}';\quad \overset{m}{I}_{kj} = \overset{m}{I}_{jk};$$

$$\text{for } m = 1:1:n_b;\ j = 1:1:n_m;\ k = n_m+1:1:n_m+2$$

$$\overset{m}{I}_{jk} = 2\,\mathrm{Tr}\big\{\overset{m}{\hat{\boldsymbol{T}}}_{j-1}\,\overset{m}{\boldsymbol{U}}_{1j}\,{}^{j}\overset{m}{\boldsymbol{F}}_{0}\,\overset{m}{\boldsymbol{U}}_{3(k-n_m)}^{\mathrm{T}}\,\overset{m}{\boldsymbol{T}}_{0}^{\mathrm{T}}\big\};\quad \overset{m}{I}_{kj} = \overset{m}{I}_{jk};$$

$$\text{for } m = 1:1:n_b;\ j = n_m+1:1:n_m+2;\ k = j$$

$$\overset{m}{I}_{jk} = 2\,\mathrm{Tr}\big\{\overset{m}{\boldsymbol{T}}_{0}\,\overset{m}{\boldsymbol{U}}_{3(j-n_m)}\,{}^{0}\overset{m}{\hat{\boldsymbol{F}}}_{0}\,\overset{m}{\boldsymbol{U}}_{3(k-n_m)}^{\mathrm{T}}\,\overset{m}{\boldsymbol{T}}_{0}^{\mathrm{T}}\big\};\quad \overset{m}{I}_{kj} = \overset{m}{I}_{jk}\,.$$

References

1. Muybridge, E.: Animals in Motion. Dover, New York (1957)
2. Schiehlen, W.: Computational dynamics: theory and applications of multibody systems. Eur. J. Mech. A, Solids **25**, 566–594 (2006)
3. Shah, S.V., Saha, S.K., Dutt, J.K.: Modular framework for dynamics of tree-type legged robots. Mech. Mach. Theory **49**, 234–255 (2012)
4. Vukobratovic, M., Potkonjak, V., Babkovic, K., Borovac, B.: Simulation model of general human and humanoid motion. Multibody Syst. Dyn. **17**, 71–96 (2007)
5. Bhalerao, K.D., Critchley, J., Anderson, K.: An efficient parallel dynamics algorithm for simulation of large articulated robotic systems. Mech. Mach. Theory **53**, 86–98 (2012)
6. Shah, S.V., Saha, S.K., Dutt, J.K.: Dynamics of Tree-Type Robotic Systems. Springer, Berlin (2013)
7. Nguyen, N.S., Brogliato, B.: Multiple Impacts in Dissipative Granular Chains. Springer, Berlin (2014)
8. Gilardi, G., Sharf, I.: Literature survey of contact dynamics modelling. Mech. Mach. Theory **37**, 1213–1239 (2002)
9. Khulief, Y.A.: Modeling of impacts in multibody systems: an overview. J. Comput. Nonlinear Dyn. **8**, 1–15 (2012)
10. Schiehlen, W., Seifried, R., Eberhard, P.: Elastoplastic phenomena in multibody impact dynamics. Comput. Methods Appl. Mech. Eng. **195**, 6874–6890 (2006)
11. Pfeiffer, F., Glocker, C.: Multibody Dynamics with Unilateral Contacts. Wiley, New York (1996)
12. Flores, P., Ambrósio, J.: On the contact detection for contact-impact analysis in multibody systems. Multibody Syst. Dyn. **24**, 103–122 (2010)
13. Glocker, C.: Energetic consistency conditions for standard impacts Part I: Newton-type inequality impact laws and Kane's example. Multibody Syst. Dyn. **29**, 77–117 (2013)
14. Glocker, C.: Energetic consistency conditions for standard impacts Part II: Poisson-type inequality impact laws. Multibody Syst. Dyn. **32**, 1–65 (2013)
15. Yigit, A.S., Ulsoy, A.G., Scott, R.A.: Dynamics of a radially rotating beam with impact, Part 1: theoretical and computational model. J. Vib. Acoust. **112**, 65–70 (1990)
16. Yigit, A.S., Ulsoy, A.G., Scott, R.A.: Dynamics of a radially rotating beam with impact, Part 2: experimental and simulation results. J. Vib. Acoust. **112**, 71–77 (1990)
17. Yigit, A.S., Ulsoy, A.G., Scott, R.A.: Spring–dashpot models for the dynamics of a radially rotating beam with impact. J. Sound Vib. **142**, 515–525 (1990)
18. Förg, M., Pfeiffer, F., Ulbrich, H.: Simulation of unilateral constrained systems with many bodies. Multibody Syst. Dyn. **14**, 137–154 (2005)

19. Shafei, A.M., Shafei, H.R.: A systematic method for the hybrid dynamic modeling of open kinematic chains confined in a closed environment. Multibody Syst. Dyn. **38**, 21–42 (2016)
20. Shafei, A.M., Shafei, H.R.: Dynamics behavior of flexible multiple links captured inside a closed space. J. Comput. Nonlinear Dyn. **11**, 051016 (2016)
21. Shafei, A.M., Shafei, H.R.: Oblique impact of multi-flexible-link systems. J. Vib. Control **24**, 904–923 (2018)
22. Shafei, A.M., Shafei, H.R.: Planar multibranch open-loop robotic manipulators subjected to ground collision. J. Comput. Nonlinear Dyn. **12**, 061003 (2017)
23. Mitrouchev, P.: Symbolic structural synthesis and a description method for planar kinematic chains in robotics. Eur. J. Mech. A, Solids **20**, 777–794 (2001)
24. Bae, D.S., Haug, E.J.: A recursive formulation for constrained mechanical system dynamics: Part I. Open loop system. Mech. Struct. Mach. **15**, 359–382 (1987)
25. Saha, S.K., Schiehlen, W.: Recursive kinematics and dynamics for parallel structural closed-loop multibody systems. Mech. Struct. Mach. **29**, 143–175 (2001)
26. Sohl, G.A., Bobrow, J.E.: A recursive multibody dynamics and sensitivity algorithm for branched kinematic chains. J. Dyn. Syst. Meas. Control **123**, 391–399 (2001)
27. Chenut, X., Fisette, P., Samin, J.-C.L.: Recursive formalism with a minimal dynamic parameterization for the identification and simulation of multibody systems. Application to the human body. Multibody Syst. Dyn. **8**, 117–140 (2002)
28. 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**, 45–59 (2003)
29. Lot, R., Dalio, M.: A symbolic approach for automatic generation of the equations of motion of multibody systems. Multibody Syst. Dyn. **12**, 147–172 (2004)
30. Lugris, U., Naya, M.A., Gonzalez, F., Cuadrado, J.: Performance and application criteria of two fast formulations for flexible multibody dynamics. Mech. Based Des. Struct. Mach. **35**, 381–404 (2007)
31. Gattringer, H., Bremer, H., Kastner, M.: Efficient dynamic modeling for rigid multi-body systems with contact and impact: an O(n) formulation. Acta Mech. **219**, 111–128 (2011)
32. Müller, A.: Recursive higher-order constraints for linkages with lower kinematic pairs. Mech. Mach. Theory **100**, 33–43 (2016)
33. Anderson, K.S.: Recursive derivation of explicit equations of motion for efficient dynamic/control of large multibody systems. PhD Dissertation, Stanford University (1990)
34. Jain, A.: Unified formulation of dynamics for serial rigid multibody systems. J. Guid. Control Dyn. **14**, 531–542 (1991)
35. Rosenthal, D.E., Sherman, M.A.: High performance multibody simulations via symbolic equations manipulation and Kane's method. J. Astronaut. Sci. **34**, 223–239 (1986)
36. Featherstone, R.: A divide-and-conquer articulated body algorithm for parallel $O(\log(n))$ calculation of rigid body dynamics. Part 1: basic algorithm. Int. J. Robot. Res. **18**, 867–875 (1999)
37. Poursina, M., Anderson, K.S.: An extended divide-and-conquer algorithm for a generalized class of multibody constraints. Multibody Syst. Dyn. **29**, 235–254 (2012)
38. Hollerbach, J.M.: A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. IEEE Trans. Syst. Man Cybern. **10**, 730–736 (1980)
39. Mata, V., Provenzano, S., Valero, F., Cuadrado, J.I.: Serial-robot dynamics algorithms for moderately large number of joints. Mech. Mach. Theory **37**, 739–755 (2002)
40. Korayem, M.H., Shafei, A.M.: Application of recursive Gibbs–Appell formulation in deriving the equations of motion of N-viscoelastic robotic manipulators in 3D space using Timoshenko beam theory. Acta Astronaut. **83**, 273–294 (2013)
41. Korayem, M.H., Shafei, A.M., Absalan, F., Kadkhodaei, B., Azimi, A.: Kinematic and dynamic modeling of viscoelastic robotic manipulators using Timoshenko beam theory: theory and experiment. Int. J. Adv. Manuf. Technol. **71**, 1005–1018 (2014)
42. Korayem, M.H., Shafei, A.M., Shafei, H.R.: Dynamic modeling of nonholonomic wheeled mobile manipulators with elastic joints using recursive Gibbs–Appell formulation. Sci. Iran. Trans. B, Mech. Eng. **19**, 1092–1104 (2012)
43. Korayem, M.H., Shafei, A.M.: A new approach for dynamic modeling of n-viscoelastic-link robotic manipulators mounted on a mobile base. Nonlinear Dyn. **79**, 2767–2786 (2015)
44. Korayem, M.H., Shafei, A.M., Dehkordi, S.F.: Systematic modeling of a chain of N-flexible link manipulators connected by revolute–prismatic joints using recursive Gibbs–Appell formulation. Arch. Appl. Mech. **84**, 187–206 (2014)
45. Korayem, M.H., Shafei, A.M.: Motion equation of nonholonomic wheeled mobile robotic manipulator with revolute–prismatic joints using recursive Gibbs–Appell formulation. Appl. Math. Model. **39**, 1701–1716 (2015)

46. Shafei, A.M., Shafei, H.R.: Dynamic modeling of planar closed-chain robotic manipulators in flight and impact phases. Mech. Mach. Theory **126**, 141–154 (2018)
47. Craig, J.J.: Introduction to Robotics Mechanics and Control, 2nd edn. Addison–Wesley, Reading (1986)
48. Korayem, M.H., Shafei, A.M., Doosthoseini, M., Absalan, A., Kadkhodaei, B.: Theoretical and experimental investigation of viscoelastic serial robotic manipulators with motors at the joints using Timoshenko beam theory and Gibbs–Appell formulation. J. Multi-Body Dyn. **230**, 37–51 (2016)
49. Jain, A.: Multibody graph transformations and analysis: Part I: tree topology systems. Nonlinear Dyn. **67**, 2779–2797 (2012)
50. Anderson, K.S.: An order n formulation for the motion simulation of general multi-rigid-body tree systems. Comput. Struct. **46**, 547–559 (1993)
51. Anderson, K.S., Sadowski, M.J.: An efficient method for contact/impact problems in multibody systems: tree topologies. In: Proceedings of the Fourth International Symposium on Multibody Dynamics and Vibrations, ASME Design Engineering Technical Conference 2003, DETC03, Chicago, IL, 2–6 September 2003, number DETC03/VIB-48338