International Journal of Non-Linear Mechanics 144 (2022) 104033

Contents lists available at ScienceDirect

International Journal of Non-Linear Mechanics

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

An explicit modelling method of joint-space inertia matrix for tree-chain dynamic system

Kaimeng Wang [a], Hehua Ju [a,b,c,*]

[a] College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 210000, China
[b] Key Laboratory Ministry of Industry and Information Technology, Nanjing 210000, China
[c] Laboratory of Aerospace Entry, Descent and Landing Technology, Nanjing 210000, China

ARTICLE INFO

Keywords:
Joint-space inertia matrix (JSIM)
Tree-chain dynamic system
Explicit method

ABSTRACT

The joint-space inertia matrix (JSIM) plays an important role in the dynamic simulation and control of robot mechanisms. In this paper, a novel explicit modelling method of JSIM for tree-chain dynamic system is proposed. By this method, each element of the inertia matrix has an explicit expression with respect to input parameters. Thus, the modelling process only includes the process of coordinate system establishment and parameter determination without the intermediate process of complex analysis and derivation. Moreover, the matrix elements are decoupled, which is beneficial to further study of the properties of JSIM on the one hand. On the other hand, multiple matrix elements can be computed simultaneously to reduce the computation time. Compared with the existing explicit method, this method has higher modelling and computational efficiency. Finally, a planar manipulator with three degrees of freedom (DOFs) is used to verify the proposed method in serial-chain application and show its modelling convenience. A lunar rover example is used to verify the proposed method in tree-chain application, and the modelling results are applied to the traction control in the simulation software of lunar rover.

1. Introduction

Dynamics has been an important subject in robot researches. A large number of dynamics modelling methods has been proposed in the past decades [1–5]. As the core of the dynamic model, the joint space inertial matrix (JSIM) describes the relationship between the joint axial force and acceleration. This relationship is of great importance in dynamic control and simulation of robots. As for dynamic control, JSIM were often used to decouple dynamic model, and then the controller could be realized by feedback linearization [6,7]. JSIM were also used in dynamic parameter identification because it contains all the dynamic parameters. Moreover, some researchers have used inertia matrix as an indicator of dynamic calculation accuracy and control performance [8,9].

There are two major kinds of methods to obtain the joint-space inertia matrix: recursive methods and explicit methods. Several studies have focused on the recursive JSIM calculation methods. Walker and Orin [10] first proposed three JSIM calculation methods, where the first two methods calculate JSIM by setting specific acceleration and velocity values in the Newton–Euler inverse dynamic process. The third method, well known as composite rigid body algorithm (CRBA), obtains JSIM by the inertia of composite body. And it was used in the commonly-used dynamic libraries Pinocchio [11] and RBDL [12]. Featherstone [13] represented CRBA by the six-dimensional space vector and solved the sparse problem of inertia matrix. Balafoutis [14] divided the composite linkage in CRBA into generalized linkage and augmented linkage, and improved the computational efficiency by calculating the mass and inertia tensor of the augmented linkage off-line. Then, the inertia matrix formula was expressed by the second-order Cartesian tensor to further improve the computational efficiency [15]. Lilly and Orin [16] compared three inertial matrix calculation methods, i.e., structural recursion method, inertia projection method, and improved composite rigid body method. Safeea et al. [17] used the unique attributes of each column of JSIM to minimize the number of recursive operations, and proposed the geometric inertia matrix algorithm. Saha et al. [18] simplified the Newton–Euler dynamic equation by decoupled natural orthogonal complement and obtained the recursive expression of JSIM. Besides, Shafei [19] proposed a new recursive JSIM calculation method based on a less-frequently-used recursive Gibbs–Appell formulation. Then, he improved the computational efficiency of this method through the combination of 3×3 rotation and 4×4 transformation matrices, which combines the high computational efficiency of 3×3 rotation and compact motion equations of 4×4 transformation matrices [20]. The great developments of JSIM modelling have been achieved in the mentioned recursive methods, nonetheless, they can

* Corresponding author.
E-mail address: juhehua@nuaa.edu.cn (H. Ju).

https://doi.org/10.1016/j.ijnonlinmec.2022.104033
Received 3 December 2021; Received in revised form 16 March 2022; Accepted 26 March 2022
Available online 8 April 2022

only obtain the recursive relation between two adjacent elements or columns of JSIM, but not the global explicit expression.

Compared with recursive methods, explicit methods can produce a more concise and understandable explicit expressions of JSIM systematically. The explicit expression is efficient engineering implemented and not error-prone. Furthermore, the explicit expressions can directly reflect the physical properties of the system, which is concealed by the recursive methods. In previous studies, researchers generally obtain explicit expressions of JSIM for specific system by manual derivation of Lagrangian equation. Brinker [21] applied this method in Delta Robots and Chen [22] applied this method in spatial parallel mechanism. However, this method is not suitable for computer automatic modelling, and requires a tremendous amount of work for the system with high degrees of freedom because of the partial and derivative operation involved in modelling process. Although some methods use computer symbolic automatic derivation tools such as ARM [23], MAPLE [24] and MATHEMATICA [25] to solve the partial and derivative terms in Lagrangian equations and realize computer automatic modelling, these methods are inefficient and error-prone. Afterwards, researchers attempted to obtain more efficient calculation methods of JSIM through symbolic derivation of Lagrangian equation. Uicker [26] first presented a symbolic expression of JSIM by further derivation of Lagrangian equation, which consists of 4×4 homogeneous transformation matrices and pseudo-inertia matrices. However, this method still needs the operation of taking partial derivatives of homogeneous transformation matrix to joint variables in the process of calculating JSIM. Then Li [27] further derived the Uicker's equation and eliminated the partial derivative terms to obtain an explicit JSIM modelling method, and it was applied to the JSIM calculation of spraying robot by Pan [28]. However, it is mentioned in [26] that these JSIM calculation method based on the 4×4 homogeneous transformation matrix has low computational efficiency. Some latest developments have been made to reformulate and derive the Lagrangian equation through new mathematical expressions. From [29,30] proposed an explicit JSIM calculation method based on the lie group and Boltzmann–Hamel formulation. However, these methods are only advantageous in some specific application scenarios. Besides, a widely-used method based on generalized momentum [31–34] can also obtain the explicit expression of JSIM which expressed by Jacobian matrices, nonetheless, it lacked computational efficiency due to the calculation of Jacobian matrix, and could not obtain independent expression of matrix elements. Despite the explicit calculation method of JSIM has been developed in the mentioned works, their expressions still contain some intermediate variables such as pseudo-inertia matrices, homogeneous transformation matrix and Jacobian matrices. And the computational and modelling efficiency are reduced in the process of constructing these intermediate variables.

This paper proposes a novel explicit modelling method of JSIM for tree-chain dynamic system. By this method, the explicit and decoupled expression of JSIM element can be yielded efficiently by substituting system input parameters without construction process of intermediate variables. Taking advantage of the explicit and decoupled characteristics of JSIM element, on the one hand, the symmetry property of JSIM can be proved by mathematical deduction, rather than by numerical experiments in the previous works such as those in [35,36], which can be used to improve computational efficiency by computing only the upper triangular elements of the JSIM; on the other hand, the parallel computing of JSIM elements is easily implemented. Besides, the double cross-product structure in our expression of JSIM is proved to be computational efficient in [15]. This method has higher modelling and computational efficiency than the existing explicit methods. Finally, a planar 3-DOF manipulator and a 5-DOF rocker arm of lunar rover are used to verify the modelling correctness of this method in serial-chain and tree-chain applications, respectively.

The outline of this paper is as follows. In Section 2, basic concepts used in this paper are described. Section 3 recalls the improved Lagrangian dynamic equations on which this work is based and a further derivation of these equations is conducted to fit the subsequent derivation of JSIM. The detailed derivation of JSIM is elaborated in Section 4. Then symmetry of JSIM is proved in Section 5. The implementation process and computational complexity analysis of this method are presented in Section 6. In Section 7, two applications of the propose method are given. In the last section, some remarkable conclusions are given.

2. Basic concepts and symbols

As shown in Fig. 1, the links and joints are numbered from 1 to n, the base is numbered by 0 and the inertial frame is numbered by i. The joint $\#l$ denoted as $\boldsymbol{k}_l$ connects link $\#\bar{l}$ and link $\#l$, where $\bar{l}$ represents the previous link of l. The simple joint with single DOF includes revolute joint ($\boldsymbol{R}$) and prismatic joint ($\boldsymbol{P}$), and the complex joint with multiple DOFs can be composed of simple joints. Tree-chain system can be described by kinematic chain and subtree. The kinematic chain from link $\#0$ to $\#\bar{l}$, denoted as ${}^{0}\boldsymbol{l}_{\bar{l}}$, is represented by the partial order set $(0, 1, \ldots, \bar{\bar{l}}, \bar{l}]$, and the subtree of link $\bar{l}$, denoted as ${}^{\bar{l}}\boldsymbol{L}$, is represented by the partial order set $(\bar{l}, l, \ldots, n, k, \ldots, m]$ including all descendant links (shown in the dotted box of Fig. 1). The inertial frame $\boldsymbol{F}^{[i]}$ and the joint frames $\boldsymbol{F}^{[l]}(l = 1, 2, \ldots, n)$ are established according to the rule that the initial direction of each joint frame axis is consistent with that of inertia frame and the coordinate origin is on the motion axis of joint, which is called natural coordinate system.

In this paper, vectors are expressed in lowercase letters, where the superscript denotes the starting point of the vector, the subscript denotes the end point of the vector, and the superscript left the separator "|" denotes the reference frame. For example, ${}^{i|\bar{l}}\boldsymbol{r}_l$ indicates the vector from the origin of frame $\#\bar{l}$ to the origin of frame $\#l$ with reference to $\boldsymbol{F}^{[i]}$. The cross product of the vector $\boldsymbol{r}$ can be replaced by the vector product of the screw matrix $\widetilde{\boldsymbol{r}}$, i.e., $\boldsymbol{r} \times \boldsymbol{v} = \widetilde{\boldsymbol{r}} \cdot \boldsymbol{v}$.

The structural parameters are determined according to the natural coordinate system. As shown in Fig. 2, O_l is denoted as origin of frame $\#l$, the inertia centre of the link $\#l$ as lI, and any point on the link $\#l$ as lS. The unit axis vector of the joint $\#l$ as ${}^{\bar{l}}\boldsymbol{n}_l$, ${}^{\bar{l}}\boldsymbol{l}_l$ as the structural vector from the origin of frame $\#\bar{l}$ to the origin of frame $\#l$. The angular position and linear position along ${}^{\bar{l}}\boldsymbol{n}_l$ are denoted as ϕ_l and r_l, respectively.

The forward kinematic equations expressed in natural coordinate system are derived. As for revolute joint, the rotational velocity vector for ${}^{\bar{l}}\boldsymbol{l}_l$ is denoted as

$$ {}^{i|\bar{l}}\dot{\boldsymbol{\phi}}_l = {}^{i|\bar{l}}\boldsymbol{n}_l \cdot \dot{\phi}_l \tag{1} $$

It is noted that the rotational velocity vector is additive. The rotational velocity vector for ${}^{i}\boldsymbol{l}_l$ is obtained by iterating Eq. (1),

$$ {}^{i}\dot{\boldsymbol{\phi}}_l = \sum_{k \in {}^{i}\boldsymbol{l}_l} \left({}^{i|\bar{k}}\dot{\boldsymbol{\phi}}_k \right) \tag{2} $$

The rotational acceleration vector for ${}^{i}\boldsymbol{l}_l$ can be obtained by taking the derivative of Eq. (2) with respect to time,

$$ {}^{i}\ddot{\boldsymbol{\phi}}_l = \sum_{k \in {}^{i}\boldsymbol{l}_l} \left({}^{i}\dot{\widetilde{\boldsymbol{\phi}}}_k \cdot {}^{i|\bar{k}}\dot{\boldsymbol{\phi}}_k + {}^{i|\bar{k}}\ddot{\boldsymbol{\phi}}_k \right) \tag{3} $$

As for prismatic joint, the linear position vector for ${}^{\bar{l}}\boldsymbol{l}_l$ is obtained according to

$$ {}^{i|\bar{l}}\boldsymbol{r}_l = \begin{cases} {}^{i|\bar{l}}\boldsymbol{n}_l \cdot r_l + {}^{i|\bar{l}}\boldsymbol{l}_l, & if \quad \boldsymbol{k}_l \in \boldsymbol{P} \\ {}^{i|\bar{l}}\boldsymbol{l}_l, & if \quad \boldsymbol{k}_l \in \boldsymbol{R} \end{cases} \tag{4} $$

The linear position vector for ${}^{i}\boldsymbol{l}_l$ is obtained by iterating Eq. (4)

$$ {}^{i}\boldsymbol{r}_l = \sum_{k \in {}^{i}\boldsymbol{l}_l} \left({}^{i|\bar{k}}\boldsymbol{r}_k \right) = {}^{i}\boldsymbol{r}_{\bar{l}} + {}^{i|\bar{l}}\boldsymbol{r}_l \tag{5} $$

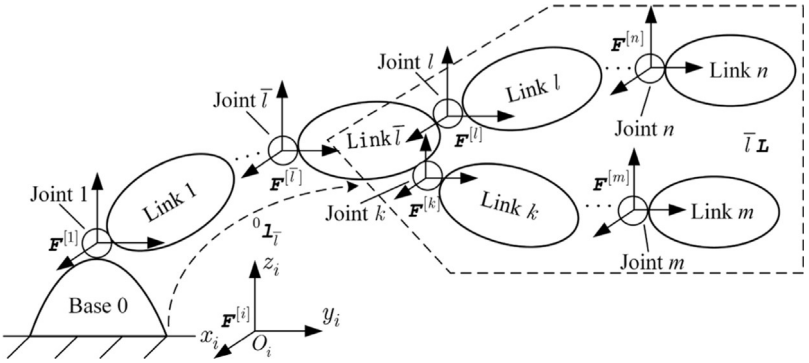


Fig. 1. Schematic diagram of tree-chain structure.

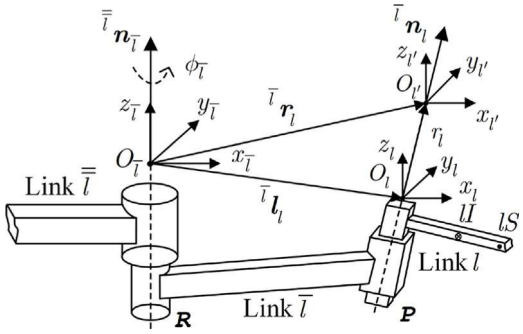


Fig. 2. Structural parameters of tree-chain system.

Similar to the derivation of Eq. (3), the translational velocity and acceleration for ${}^{i}\boldsymbol{l}_{l}$ are obtained by taking the first and second derivative of Eq. (5) with respect time, respectively,

$$ {}^{i}\dot{\boldsymbol{r}}_{l} = \sum_{k\in{}^{i}\boldsymbol{l}_{l}} \left({}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{\bar{k}} \cdot {}^{i|\bar{k}}\boldsymbol{r}_{k} + {}^{i|\bar{k}}\dot{\boldsymbol{r}}_{k} \right) \tag{6} $$

$$ {}^{i}\ddot{\boldsymbol{r}}_{l} = \sum_{k\in{}^{i}\boldsymbol{l}_{l}} \left({}^{i}\tilde{\ddot{\boldsymbol{\phi}}}_{\bar{k}} \cdot {}^{i|\bar{k}}\boldsymbol{r}_{k} + {}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{\bar{k}}^{\wedge 2} \cdot {}^{i|\bar{k}}\boldsymbol{r}_{k} + 2 \cdot {}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{\bar{k}} \cdot {}^{i|\bar{k}}\dot{\boldsymbol{r}}_{k} + {}^{i|\bar{k}}\ddot{\boldsymbol{r}}_{k} \right) \tag{7} $$

where ${}^{i}\tilde{\ddot{\boldsymbol{\phi}}}_{\bar{k}} \cdot {}^{i|\bar{k}}\boldsymbol{r}_{k}$ is coupling term of angular acceleration, ${}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{\bar{k}}^{\wedge 2} \cdot {}^{i|\bar{k}}\boldsymbol{r}_{k}$ is centripetal acceleration and ${}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{\bar{k}} \cdot {}^{i|\bar{k}}\dot{\boldsymbol{r}}_{k}$ is Coriolis acceleration. Unless otherwise stated, the formulae in the rest of this paper will use the symbols in Table A.1 of Appendix.

3. Improved Lagrangian dynamic method

Based on Lagrangian dynamic method, Ju et al. [37,38] proposed an improved Lagrangian dynamic method which could obtain the explicit inverse dynamic equations. In this paper, we will derive an explicit modelling method of JSIM based on this method. The dynamic equations by the improved Lagrangian dynamic method are recalled here for convenience.

$$ \begin{cases} \sum\limits_{k\in uL} \left(m_{k} \cdot \dfrac{\partial^{i}\dot{\boldsymbol{r}}_{kI}^{T}}{\partial \dot{r}_{u}} \cdot {}^{i}\ddot{\boldsymbol{r}}_{kI} - m_{k} \cdot \dfrac{\partial^{i}\boldsymbol{r}_{kI}^{T}}{\partial r_{k}} \cdot \boldsymbol{g} \right) = {}^{i|\bar{u}}\boldsymbol{n}_{u}^{T} \cdot \boldsymbol{f}_{u} & \text{if} \quad k_{u} \in \boldsymbol{P} \\ \sum\limits_{k\in uL} \begin{pmatrix} m_{k} \cdot \dfrac{\partial^{i}\dot{\boldsymbol{r}}_{kI}^{T}}{\partial \dot{\boldsymbol{\phi}}_{u}} \cdot {}^{i}\ddot{\boldsymbol{r}}_{kI} - m_{k} \cdot \dfrac{\partial^{i}\boldsymbol{r}_{kI}^{T}}{\partial \boldsymbol{\phi}_{u}} \cdot \boldsymbol{g} \\ + \dfrac{\partial^{i}\dot{\boldsymbol{\phi}}_{k}^{T}}{\partial \dot{\boldsymbol{\phi}}_{u}} \cdot \left({}^{i}\boldsymbol{I}_{k} \cdot {}^{i}\ddot{\boldsymbol{\phi}}_{k} + {}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{k} \cdot {}^{i}\boldsymbol{I}_{k} \cdot {}^{i}\dot{\boldsymbol{\phi}}_{k} \right) \end{pmatrix} = {}^{i|\bar{u}}\boldsymbol{n}_{u}^{T} \cdot \boldsymbol{\tau}_{u} & \text{if} \quad k_{u} \in \boldsymbol{R} \end{cases} \tag{8} $$

where m_k denotes mass of the link #k, $\boldsymbol{g}$ denotes gravity vector. $\boldsymbol{f}_u$ and $\boldsymbol{\tau}_u$ denote resultant force and torque on axis ${}^{\bar{u}}n_u$, respectively. The explicit formulae of partial are also given,

$$ \frac{\partial^{i}\boldsymbol{r}_{n}}{\partial \boldsymbol{\phi}_{k}} = \frac{\partial^{i}\dot{\boldsymbol{r}}_{n}}{\partial \dot{\boldsymbol{\phi}}_{k}} = {}^{i|\bar{k}}\tilde{\boldsymbol{n}}_{k} \cdot {}^{i|k}\boldsymbol{r}_{n}, \quad \frac{\partial^{i}\boldsymbol{r}_{n}}{\partial \boldsymbol{r}_{k}} = \frac{\partial^{i}\dot{\boldsymbol{r}}_{n}}{\partial \dot{\boldsymbol{r}}_{k}} = {}^{i|\bar{k}}\boldsymbol{n}_{k}, \quad \frac{\partial^{i}\dot{\boldsymbol{\phi}}_{n}}{\partial \dot{\boldsymbol{\phi}}_{k}} = {}^{i|\bar{k}}\boldsymbol{n}_{k} \tag{9} $$

In order to extract the joint acceleration which is implicit in Eq. (8), the Eqs. (3), (7) and (9) are substituted into Eq. (8) to obtain

$$ \begin{cases} {}^{i|\bar{u}}\boldsymbol{n}_{u}^{T} \cdot \left(\boldsymbol{M}_{\boldsymbol{P}}^{[u][*]} + \boldsymbol{h}_{\boldsymbol{P}}^{[u]} \right) = {}^{i|\bar{u}}\boldsymbol{n}_{u}^{T} \cdot \boldsymbol{f}_{u} & \text{if} \quad k_{u} \in \boldsymbol{P} \\ {}^{i|\bar{u}}\boldsymbol{n}_{u}^{T} \cdot \left(\boldsymbol{M}_{\boldsymbol{R}}^{[u][*]} + \boldsymbol{h}_{\boldsymbol{R}}^{[u]} \right) = {}^{i|\bar{u}}\boldsymbol{n}_{u}^{T} \cdot \boldsymbol{\tau}_{u} & \text{if} \quad k_{u} \in \boldsymbol{R} \end{cases} \tag{10} $$

where the $\boldsymbol{M}_{\boldsymbol{R}}^{[u][*]}$ is inertia force vector for revolute joint #u and

$$ \boldsymbol{M}_{\boldsymbol{R}}^{[u][*]} = \sum_{k\in uL} \begin{pmatrix} {}^{i}\boldsymbol{I}_{k} \cdot \sum\limits_{l\in{}^{i}\boldsymbol{l}_{k}} \left({}^{i|\bar{l}}\ddot{\boldsymbol{\phi}}_{l} + {}^{i}\tilde{\ddot{\boldsymbol{\phi}}}_{\bar{l}} \cdot {}^{i|\bar{l}}\dot{\boldsymbol{\phi}}_{l} \right) + m_{k} \cdot {}^{i|u}\tilde{\boldsymbol{r}}_{kI} \cdot \\ \sum\limits_{l\in{}^{i}\boldsymbol{l}_{kI}} \left({}^{i|\bar{l}}\ddot{\boldsymbol{r}}_{l} - {}^{i|\bar{l}}\tilde{\boldsymbol{r}}_{l} \cdot \sum\limits_{j\in{}^{i}\boldsymbol{l}_{\bar{l}}} \left({}^{i|\bar{j}}\ddot{\boldsymbol{\phi}}_{j} + {}^{i}\tilde{\ddot{\boldsymbol{\phi}}}_{\bar{j}} \cdot {}^{i|\bar{j}}\dot{\boldsymbol{\phi}}_{j} \right) \right) \end{pmatrix} \tag{11} $$

The $\boldsymbol{M}_{\boldsymbol{P}}^{[u][*]}$ is inertia force vector for prismatic joint #u and

$$ \boldsymbol{M}_{\boldsymbol{P}}^{[u][*]} = \sum_{k\in uL} \left(m_{k} \cdot \sum_{l\in{}^{i}\boldsymbol{l}_{kI}} \left({}^{i|\bar{l}}\ddot{\boldsymbol{r}}_{l} - {}^{i|\bar{l}}\tilde{\boldsymbol{r}}_{l} \cdot \sum_{j\in{}^{i}\boldsymbol{l}_{\bar{l}}} \left({}^{i|\bar{j}}\ddot{\boldsymbol{\phi}}_{j} + {}^{i}\tilde{\ddot{\boldsymbol{\phi}}}_{\bar{j}} \cdot {}^{i|\bar{j}}\dot{\boldsymbol{\phi}}_{j} \right) \right) \right) \tag{12} $$

The $\boldsymbol{h}_{\boldsymbol{P}}^{[u]}$ is bias force vector for prismatic joint #u and

$$ \boldsymbol{h}_{\boldsymbol{P}}^{[u]} = \sum_{k\in uL} \left(m_{k} \cdot \sum_{l\in{}^{i}\boldsymbol{l}_{kI}} \left({}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{\bar{l}}^{\wedge 2} \cdot {}^{i|\bar{l}}\boldsymbol{r}_{l} + 2 \cdot {}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{\bar{l}} \cdot {}^{i|\bar{l}}\dot{\boldsymbol{r}}_{l} \right) - m_{k} \cdot \boldsymbol{g} \right) \tag{13} $$

The $\boldsymbol{h}_{\boldsymbol{R}}^{[u]}$ is bias force vector for revolute joint #u and

$$ \boldsymbol{h}_{\boldsymbol{R}}^{[u]} = \sum_{k\in uL} \begin{pmatrix} m_{k} \cdot {}^{i|u}\tilde{\boldsymbol{r}}_{kI} \cdot \sum\limits_{l\in{}^{i}\boldsymbol{l}_{kI}} \left({}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{\bar{l}}^{\wedge 2} \cdot {}^{i|\bar{l}}\boldsymbol{r}_{l} + 2 \cdot {}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{\bar{l}} \cdot {}^{i|\bar{l}}\dot{\boldsymbol{r}}_{l} \right) \\ + {}^{i}\tilde{\dot{\boldsymbol{\phi}}}_{k} \cdot {}^{i}\boldsymbol{I}_{k} \cdot {}^{i}\dot{\boldsymbol{\phi}}_{k} - m_{k} \cdot {}^{i|u}\tilde{\boldsymbol{r}}_{kI} \cdot \boldsymbol{g} \end{pmatrix} \tag{14} $$

It can be seen from Eqs. (11)–(14) that only $\boldsymbol{M}_{\boldsymbol{R}}^{[u][*]}$ and $\boldsymbol{M}_{\boldsymbol{P}}^{[u][*]}$ contain the acceleration terms. Therefore, only these two terms need to be analysed and derived in the next section.

4. Derivation of joint-space inertia matrix

The process of obtaining JSIM is also the process of expressing all the joint accelerations explicitly in the improved Lagrangian equations. According to Eqs. (11) and (12), the $\boldsymbol{M}_{\boldsymbol{R}}^{[u][*]}$ and $\boldsymbol{M}_{\boldsymbol{P}}^{[u][*]}$ contain five different terms that include the joint acceleration:

$$ \sum_{k\in uL} \left({}^{i}\boldsymbol{I}_{k} \cdot \sum_{l\in{}^{i}\boldsymbol{l}_{k}} \left({}^{i|\bar{l}}\ddot{\boldsymbol{\phi}}_{l} \right) \right) \tag{15} $$

$$ \sum_{k\in uL} \left(m_{k} \cdot {}^{i|u}\tilde{\boldsymbol{r}}_{lI} \cdot \sum_{l\in{}^{i}\boldsymbol{l}_{kI}} \left(-{}^{i|\bar{l}}\tilde{\boldsymbol{r}}_{l} \cdot \sum_{j\in{}^{i}\boldsymbol{l}_{\bar{l}}} \left({}^{i|\bar{j}}\ddot{\boldsymbol{\phi}}_{j} \right) \right) \right) \tag{16} $$

$$ \sum_{k\in uL} \left(m_{k} \cdot {}^{i|u}\tilde{\boldsymbol{r}}_{kI} \cdot \sum_{l\in{}^{i}\boldsymbol{l}_{k}} \left({}^{i|\bar{l}}\ddot{\boldsymbol{r}}_{l} \right) \right) \tag{17} $$

$$\sum_{k\in uL}\left(m_k\cdot\sum_{l\in{}^iI_{kI}}\left(-{}^{i|\bar{l}}\widetilde{\boldsymbol{r}}_l\cdot\sum_{j\in{}^iI_{\bar{l}}}\left({}^{i|\bar{j}}\ddot{\boldsymbol{\phi}}_j\right)\right)\right) \tag{18}$$

$$\sum_{k\in uL}\left(m_k\cdot\sum_{l\in{}^iI_{kI}}\left({}^{i|\bar{l}}\ddot{\boldsymbol{r}}_l\right)\right) \tag{19}$$

To express the joint acceleration explicitly, Eqs. (15)–(19) are derived and simplified respectively. We take Eq. (16), the most complex of them, as an example to give the derivation process. Eq. (16) contains three layers of iteration structure, which is too complicated to expand directly. Thus, we need to first derive the equivalent form of the internal two-layer iterative structure,

$$\begin{aligned}&\sum_{l\in{}^iI_{kI}}\left(-{}^{i|\bar{l}}\widetilde{\boldsymbol{r}}_l\cdot\sum_{j\in{}^iI_{\bar{l}}}\left({}^{i|\bar{j}}\ddot{\boldsymbol{\phi}}_j\right)\right)\\&\quad=-{}^{i|1}\widetilde{\boldsymbol{r}}_2\cdot({}^i\ddot{\boldsymbol{\phi}}_1)-{}^{i|2}\widetilde{\boldsymbol{r}}_3\cdot({}^i\ddot{\boldsymbol{\phi}}_1+{}^{i|1}\ddot{\boldsymbol{\phi}}_2)-\cdots\\&\qquad-{}^{i|k}\widetilde{\boldsymbol{r}}_{kI}\cdot({}^i\ddot{\boldsymbol{\phi}}_1+{}^{i|1}\ddot{\boldsymbol{\phi}}_2+\cdots+{}^{i|\bar{k}}\ddot{\boldsymbol{\phi}}_k)\\&\quad=\sum_{l\in{}^iI_k}\left(-{}^{i|l}\widetilde{\boldsymbol{r}}_{kI}\,{}^{i|\bar{l}}\ddot{\boldsymbol{\phi}}_l\right)\end{aligned} \tag{20}$$

Then Eq. (16) can be re-expressed with Eq. (20),

$$\begin{aligned}&\sum_{k\in uL}\left(m_k\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{kI}\cdot\sum_{l\in{}^iI_{kI}}\left(-{}^{i|\bar{l}}\widetilde{\boldsymbol{r}}_l\cdot\sum_{j\in{}^iI_{\bar{l}}}\left({}^{i|\bar{j}}\ddot{\boldsymbol{\phi}}_j\right)\right)\right)\\&=\sum_{k\in uL}\left(m_k\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{kI}\cdot\sum_{l\in{}^iI_k}\left(-{}^{i|l}\widetilde{\boldsymbol{r}}_{kI}\,{}^{i|\bar{l}}\ddot{\boldsymbol{\phi}}_l\right)\right)\end{aligned} \tag{21}$$

Eq. (21) can be expanded as

$$\begin{aligned}&\sum_{k\in uL}\left(m_k\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{kI}\cdot\sum_{l\in{}^iI_k}\left(-{}^{i|l}\widetilde{\boldsymbol{r}}_{kI}\,{}^{i|\bar{l}}\ddot{\boldsymbol{\phi}}_l\right)\right)=\\&\quad\mathrm{m}_u\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{uI}\cdot\left(-{}^{i|1}\widetilde{\boldsymbol{r}}_{uI}\,{}^i\ddot{\boldsymbol{\phi}}_1-\cdots-{}^{i|u}\widetilde{\boldsymbol{r}}_{uI}\,{}^{i|\bar{u}}\ddot{\boldsymbol{\phi}}_u\right)+\cdots\\&\qquad+\mathrm{m}_c\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{cI}\cdot\left(-{}^{i|1}\widetilde{\boldsymbol{r}}_{cI}\,{}^i\ddot{\boldsymbol{\phi}}_1-\cdots-{}^{i|u}\widetilde{\boldsymbol{r}}_{cI}\,{}^{i|\bar{u}}\ddot{\boldsymbol{\phi}}_u-\cdots-{}^{i|c}\widetilde{\boldsymbol{r}}_{cI}\,{}^{i|\bar{c}}\ddot{\boldsymbol{\phi}}_c\right)+\cdots\\&\qquad+\mathrm{m}_n\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{nI}\cdot\left(-{}^{i|1}\widetilde{\boldsymbol{r}}_{nI}\,{}^i\ddot{\boldsymbol{\phi}}_1-\cdots-{}^{i|u}\widetilde{\boldsymbol{r}}_{nI}\,{}^{i|\bar{u}}\ddot{\boldsymbol{\phi}}_u-\cdots-{}^{i|c}\widetilde{\boldsymbol{r}}_{nI}\,{}^{i|\bar{c}}\ddot{\boldsymbol{\phi}}_c-\cdots\right.\\&\qquad\left.-{}^{i|n}\widetilde{\boldsymbol{r}}_{nI}\,{}^{i|\bar{n}}\ddot{\boldsymbol{\phi}}_n\right)\end{aligned} \tag{22}$$

where $c\in uL$ and $n\in cL$. By combining the coefficients of the same joint acceleration, the equivalent form of Eq. (16) can be expressed as

$$\begin{aligned}&\sum_{k\in uL}\left(m_k\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{kI}\cdot\sum_{l\in{}^iI_{kI}}\left(-{}^{i|\bar{l}}\widetilde{\boldsymbol{r}}_l\cdot\sum_{j\in{}^iI_{\bar{l}}}\left({}^{i|\bar{j}}\ddot{\boldsymbol{\phi}}_j\right)\right)\right)\\&\quad=\sum_{k\in{}^iI_{\bar{u}}}\left(\sum_{l\in uL}\left(-m_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{\phi}}_k\right)\\&\qquad+\sum_{k\in uL}\left(\sum_{l\in kL}\left(-m_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{\phi}}_k\right)\end{aligned} \tag{23}$$

The derivation process of other equations is similar, so we will only give the results here. The equivalent form of Eq. (15) is

$$\begin{aligned}&\sum_{k\in uL}\left({}^iI_k\cdot\sum_{l\in{}^iI_k}\left({}^{i|\bar{l}}\ddot{\boldsymbol{\phi}}_l\right)\right)\\&=\sum_{k\in{}^iI_{\bar{u}}}\left(\sum_{l\in uL}\left({}^iI_l\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{\phi}}_k\right)+\sum_{k\in uL}\left(\sum_{l\in kL}\left({}^iI_l\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{\phi}}_k\right)\end{aligned} \tag{24}$$

The equivalent form of Eq. (17) is

$$\begin{aligned}&\sum_{k\in uL}\left(m_k\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{kI}\cdot\sum_{l\in{}^iI_k}\left({}^{i|\bar{l}}\ddot{\boldsymbol{r}}_l\right)\right)\\&=\sum_{k\in{}^iI_{\bar{u}}}\left(\sum_{l\in uL}\left(m_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{r}}_k\right)+\sum_{k\in uL}\left(\sum_{l\in kL}\left(m_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{r}}_k\right)\end{aligned} \tag{25}$$

The equivalent form of Eq. (18) is

$$\begin{aligned}&\sum_{k\in uL}\left(m_k\cdot\sum_{l\in{}^iI_{kI}}\left(-{}^{i|\bar{l}}\widetilde{\boldsymbol{r}}_l\cdot\sum_{j\in{}^iI_{\bar{l}}}\left({}^{i|\bar{j}}\ddot{\boldsymbol{\phi}}_j\right)\right)\right)\\&=\sum_{k\in{}^iI_{\bar{u}}}\left(\sum_{l\in uL}\left(-m_l\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{\phi}}_k\right)+\sum_{k\in uL}\left(\sum_{l\in kL}\left(-m_l\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{\phi}}_k\right)\end{aligned} \tag{26}$$

And the equivalent form of Eq. (19) is

$$\begin{aligned}&\sum_{k\in uL}\left(m_k\cdot\sum_{l\in{}^iI_{kI}}\left({}^{i|\bar{l}}\ddot{\boldsymbol{r}}_l\right)\right)\\&=\sum_{k\in{}^iI_{\bar{u}}}\left(\sum_{l\in uL}\left(m_l\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{r}}_k\right)+\sum_{k\in uL}\left(\sum_{l\in kL}\left(m_l\right)\cdot{}^{i|\bar{k}}\ddot{\boldsymbol{r}}_k\right)\end{aligned} \tag{27}$$

Then, Eqs. (24)–(27) are substituted into Eqs. (11) and (12),

$$\begin{aligned}{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\boldsymbol{M}_{\boldsymbol{R}}^{[u][*]}&=\sum_{k\in{}^iI_{\bar{u}}}\begin{pmatrix}{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in uL}\left({}^iI_l-\mathrm{m}_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k\cdot\ddot{\boldsymbol{\phi}}_k\\+{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in uL}\left(\mathrm{m}_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k\cdot\ddot{\boldsymbol{r}}_k\end{pmatrix}\\&\quad+\sum_{k\in uL}\begin{pmatrix}{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in kL}\left({}^iI_l-\mathrm{m}_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k\cdot\ddot{\boldsymbol{\phi}}_k\\+{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in kL}\left(\mathrm{m}_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k\cdot\ddot{\boldsymbol{r}}_k\end{pmatrix}\end{aligned} \tag{28}$$

$$\begin{aligned}{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\boldsymbol{M}_{\boldsymbol{P}}^{[u][*]}&=\sum_{k\in{}^iI_{\bar{u}}}\begin{pmatrix}{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in uL}\left(-m_l\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k\cdot\ddot{\boldsymbol{\phi}}_k\\+{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in uL}\left(m_l\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k\cdot\ddot{\boldsymbol{r}}_k\end{pmatrix}\\&\quad+\sum_{k\in uL}\begin{pmatrix}{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in kL}\left(-m_l\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k\cdot\ddot{\boldsymbol{\phi}}_k\\+{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in kL}\left(m_l\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k\cdot\ddot{\boldsymbol{r}}_k\end{pmatrix}\end{aligned} \tag{29}$$

From Eqs. (28) and (29), the explicit expressions of the inertia matrix elements can be obtained according to the joint types and relation between $\boldsymbol{k}_u$ and $\boldsymbol{k}_k$. There are eight different expressions for the element $\boldsymbol{M}^{[u][k]}$ in the u-th row and kth column of the JSIM:

CASE 1: $\boldsymbol{k}_u\in\boldsymbol{R}$, $\boldsymbol{k}_k\in\boldsymbol{R}$ and $k\in{}^iI_{\bar{u}}$,

$$\boldsymbol{M}^{[u][k]}={}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in uL}\left({}^iI_l-m_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k \tag{30}$$

CASE 2: $\boldsymbol{k}_u\in\boldsymbol{R}$, $\boldsymbol{k}_k\in\boldsymbol{P}$ and $k\in{}^iI_{\bar{u}}$,

$$\boldsymbol{M}^{[u][k]}={}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in uL}\left(m_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k \tag{31}$$

CASE 3: $\boldsymbol{k}_u\in\boldsymbol{R}$, $\boldsymbol{k}_k\in\boldsymbol{R}$ and $k\in uL$,

$$\boldsymbol{M}^{[u][k]}={}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in kL}\left({}^iI_l-m_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k \tag{32}$$

CASE 4: $\boldsymbol{k}_u\in\boldsymbol{R}$, $\boldsymbol{k}_k\in\boldsymbol{P}$ and $k\in uL$,

$$\boldsymbol{M}^{[u][k]}={}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in kL}\left(m_l\cdot{}^{i|u}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k \tag{33}$$

CASE 5: $\boldsymbol{k}_u\in\boldsymbol{P}$, $\boldsymbol{k}_k\in\boldsymbol{R}$ and $k\in{}^iI_{\bar{u}}$,

$$\boldsymbol{M}^{[u][k]}=-{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in uL}\left(m_l\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k \tag{34}$$

CASE 6: $\boldsymbol{k}_u\in\boldsymbol{P}$, $\boldsymbol{k}_k\in\boldsymbol{P}$ and $k\in{}^iI_{\bar{u}}$,

$$\boldsymbol{M}^{[u][k]}={}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in uL}\left(m_l\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k \tag{35}$$

CASE 7: $\boldsymbol{k}_u\in\boldsymbol{P}$, $\boldsymbol{k}_k\in\boldsymbol{R}$ and $k\in uL$,

$$\boldsymbol{M}^{[u][k]}=-{}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in kL}\left(m_l\cdot{}^{i|k}\widetilde{\boldsymbol{r}}_{lI}\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k \tag{36}$$

CASE 8: $\boldsymbol{k}_u\in\boldsymbol{P}$, $\boldsymbol{k}_k\in\boldsymbol{P}$ and $k\in uL$,

$$\boldsymbol{M}^{[u][k]}={}^{i|\bar{u}}\boldsymbol{n}_u^T\cdot\sum_{l\in kL}\left(m_l\right)\cdot{}^{i|\bar{k}}\boldsymbol{n}_k \tag{37}$$

So far, the explicit expressions of the JSIM are derived in all cases.

5. Symmetry proof of JSIM

The symmetry of JSIM has been researched in Ref. [13], however, it is only a rule found in numerical experiments without theoretical proof. In this study, we will use the explicit expression of JSIM derived in Section 4 to prove the symmetry.

Given any inertia matrix element $\boldsymbol{M}^{[u][k]}(u \neq k)$in JSIM, the proof can be divided into four cases according to the types of $\boldsymbol{k}_u$ and $\boldsymbol{k}_k$:

(1) If $\boldsymbol{k}_u \in \boldsymbol{R}$, $\boldsymbol{k}_k \in \boldsymbol{R}$ and $k \in {}^{i}\boldsymbol{l}_{\bar{u}}$, according to Eq. (30),

$$\boldsymbol{M}^{[u][k]} = {}^{i|\bar{u}}\boldsymbol{n}_u^T \cdot \sum_{l \in uL} \left({}^{i}\boldsymbol{I}_l - m_l \cdot {}^{i|u}\widetilde{\boldsymbol{r}}_{lI} \cdot {}^{i|k}\widetilde{\boldsymbol{r}}_{lI} \right) \cdot {}^{i|\bar{k}}\boldsymbol{n}_k$$

Then its symmetrical element $\boldsymbol{M}^{[k][u]}$ can be obtained according to Eq. (32),

$$\boldsymbol{M}^{[k][u]} = {}^{i|\bar{k}}\boldsymbol{n}_k^T \cdot \sum_{l \in uL} \left({}^{i}\boldsymbol{I}_l - m_l \cdot {}^{i|k}\widetilde{\boldsymbol{r}}_{lI} \cdot {}^{i|u}\widetilde{\boldsymbol{r}}_{lI} \right) \cdot {}^{i|\bar{u}}\boldsymbol{n}_u$$

Because $\boldsymbol{M}^{[k][u]} \in \mathbb{R}$, then

$$\begin{aligned}
\boldsymbol{M}^{[k][u]} &= \left(\boldsymbol{M}^{[k][u]} \right)^{\mathrm{T}} \\
&= \left({}^{i|\bar{k}}\boldsymbol{n}_k^T \cdot \sum_{l \in uL} \left({}^{i}\boldsymbol{I}_l - m_l \cdot {}^{i|k}\widetilde{\boldsymbol{r}}_{lI} \cdot {}^{i|u}\widetilde{\boldsymbol{r}}_{lI} \right) \cdot {}^{i|\bar{u}}\boldsymbol{n}_u \right)^{\mathrm{T}} \\
&= {}^{i|\bar{u}}\boldsymbol{n}_u^T \cdot \sum_{l \in uL} \left(\left({}^{i}\boldsymbol{I}_l \right)^{\mathrm{T}} - m_l \cdot \left({}^{i|u}\widetilde{\boldsymbol{r}}_{lI} \right)^{T} \cdot \left({}^{i|k}\widetilde{\boldsymbol{r}}_{lI} \right)^{\mathrm{T}} \right) \cdot {}^{i|\bar{k}}\boldsymbol{n}_k \\
&= {}^{i|\bar{u}}\boldsymbol{n}_u^T \cdot \sum_{l \in uL} \left({}^{i}\boldsymbol{I}_l - m_l \cdot {}^{i|u}\widetilde{\boldsymbol{r}}_{lI} \cdot {}^{i|k}\widetilde{\boldsymbol{r}}_{lI} \right) \cdot {}^{i|\bar{k}}\boldsymbol{n}_k = \boldsymbol{M}^{[u][k]}
\end{aligned}$$

(2) If $\boldsymbol{k}_u \in \boldsymbol{R}$, $\boldsymbol{k}_k \in \boldsymbol{P}$ and $k \in {}^{i}\boldsymbol{l}_{\bar{u}}$, according to Eq. (31),

$$\boldsymbol{M}^{[u][k]} = {}^{i|\bar{u}}\boldsymbol{n}_u^T \cdot \sum_{l \in uL} \left(m_l \cdot {}^{i|u}\widetilde{\boldsymbol{r}}_{lI} \right) \cdot {}^{i|\bar{k}}\boldsymbol{n}_k$$

Then its symmetrical element $\boldsymbol{M}^{[k][u]}$ can be obtained according to Eq. (36),

$$\boldsymbol{M}^{[k][u]} = -{}^{i|\bar{k}}\boldsymbol{n}_k^T \cdot \sum_{l \in uL} \left(m_l \cdot {}^{i|u}\widetilde{\boldsymbol{r}}_{lI} \right) \cdot {}^{i|\bar{u}}\boldsymbol{n}_u$$

$\boldsymbol{M}^{[u][k]} = \left(\boldsymbol{M}^{[k][u]} \right)^{\mathrm{T}} = \boldsymbol{M}^{[k][u]}$ can be proved similarly.

(3) If $\boldsymbol{k}_u \in \boldsymbol{P}$, $\boldsymbol{k}_k \in \boldsymbol{P}$ and $k \in {}^{i}\boldsymbol{l}_{\bar{u}}$, according to Eq. (35),

$$\boldsymbol{M}^{[u][k]} = {}^{i|\bar{u}}\boldsymbol{n}_u^T \cdot \sum_{l \in uL} \left(m_l \right) \cdot {}^{i|\bar{k}}\boldsymbol{n}_k$$

Then its symmetrical element $\boldsymbol{M}^{[k][u]}$ can be obtained according to Eq. (37),

$$\boldsymbol{M}^{[k][u]} = {}^{i|\bar{k}}\boldsymbol{n}_k^T \cdot \sum_{l \in uL} \left(m_l \right) \cdot {}^{i|\bar{u}}\boldsymbol{n}_u$$

$\boldsymbol{M}^{[u][k]} = \left(\boldsymbol{M}^{[k][u]} \right)^{\mathrm{T}} = \boldsymbol{M}^{[k][u]}$ can be proved similarly.

(4) If $\boldsymbol{k}_u \in \boldsymbol{P}$, $\boldsymbol{k}_k \in \boldsymbol{R}$ and $k \in {}^{i}\boldsymbol{l}_{\bar{u}}$, according to Eq. (34),

$$\boldsymbol{M}^{[u][k]} = -{}^{i|\bar{u}}\boldsymbol{n}_u^T \cdot \sum_{l \in uL} \left(m_l \cdot {}^{i|k}\widetilde{\boldsymbol{r}}_{lI} \right) \cdot {}^{i|\bar{k}}\boldsymbol{n}_k$$

Then its symmetrical element $\boldsymbol{M}^{[k][u]}$ can be obtained according to Eq. (33),

$$\boldsymbol{M}^{[k][u]} = {}^{i|\bar{k}}\boldsymbol{n}_k^T \cdot \sum_{l \in uL} \left(m_l \cdot {}^{i|k}\widetilde{\boldsymbol{r}}_{lI} \right) \cdot {}^{i|\bar{u}}\boldsymbol{n}_u$$

$\boldsymbol{M}^{[u][k]} = \left(\boldsymbol{M}^{[k][u]} \right)^{\mathrm{T}} = \boldsymbol{M}^{[k][u]}$ can be proved similarly.

So far, the proof of the symmetry of the JSIM in all cases has been completed.

6. Implementation and computational complexity analysis of JSIM

In this section, we illustrate the implementation process by pseudocode and analyse the computation complexity of each step. The flowchart of implementation process of JSIM for n DOFs tree-chain dynamic system is shown in Fig. 1 of Appendix. Compared with traditional modelling, the explicit method proposed in this paper has no intermediate construction process of intermediate variables. Thus, it is convenient for the realization of computer automatic modelling. Besides, topology and joint types are explicitly expressed in the modelling process, which makes it possible to change topologies and joint types in real time.

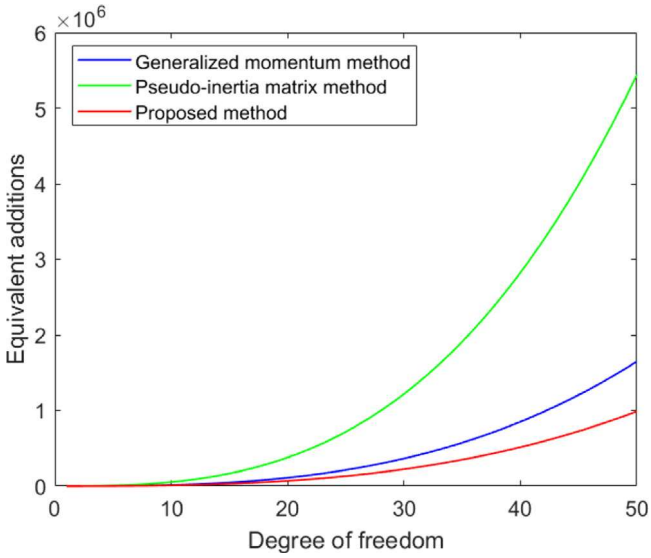


Fig. 3. Comparison of the equivalent additions for three explicit methods.

When all joints are revolute, the computational complexity is the highest. Thus, only the case where all joints are revolute is considered in the analysis of computational complexity. The computational complexity of the JSIM is analysed by the number of additions and multiplications. Generally, the multiplications of 3×3 matrix can be categorized in the following classes:

(1) Multiplication of two ordinary matrices needs 27 scalar multiplications and 18 scalar additions;

(2) Multiplication of two skew-symmetric matrices need 12 scalar multiplications and 3 scalar additions;

(3) Multiplication of skew-symmetric matrix and vector need 6 scalar multiplications and 3 scalar additions;

(4) Multiplication of ordinary matrix and vector need 9 scalar multiplications and 6 scalar additions.

Based on these four types of matrix operations, the detailed comparisons of computational process and complexity of three explicit JSIM methods are listed in Table 1. Since the process of establishing coordinate system and determining parameters of the two methods are the same, only the computational complexity of calculating inertia matrix is given here. Using the symmetry of JSIM to calculate only the lower triangle and diagonal elements, the proposed method requires $(3.5n^3+16n^2+12.5n)$ multiplications and $(3.5n^3+10n^2+6.5n)$ additions. Then equivalent additions are introduced in order to simplify comparisons, where equivalent additions = 1.1 * multiplications + additions. In Fig. 3, the equivalent additions of the three methods with the degree of freedom ranging from 1 to 50 are compared. The computational complexity of the proposed method is lower than the generalized momentum method and pseudo-inertia matrix method, and the gap becomes more obvious with the increase of the degree of freedom. Furthermore, benefited from the independence of the matrix elements, it is possible to perform parallel computing on the matrix elements to reduce computational time furtherly.

Table 1
Comparison of computational process and complexity of explicit JSIM methods.

Dynamic method	Generalized momentum method [31]		
Steps		Computational Complexity	
		Multiplications	Additions
1	Construct the Jacobian matrix J_P^l, J_O^l	$3n^2+3n$	$3n^2+3n$
2	Matrix calculations: $m_l \cdot J_P^{l\,T} \cdot J_P^l$	$4n^2$	$2n^2$
3	Matrix calculations: $J_O^{l\,T} \cdot I_l \cdot J_O^l$	$3n^2+9n$	$2n^2+6n$
4	$m_l \cdot J_P^{l\,T} \cdot J_P^l + J_O^{l\,T} \cdot I_l \cdot J_O^l$	0	n^2
5	$\boldsymbol{M} = \sum_{l=1}^{n} \left(m_l \cdot J_P^{l\,T} \cdot J_P^l + J_O^{l\,T} \cdot I_l \cdot J_O^l \right)$	$7n^3+12n^2+3n$	$5n^3+10n^2+2n$
Dynamic method	Pseudo-inertia matrix method [28]		
1	Construct the Pseudo-inertia matrix $\boldsymbol{I}_j$	24n	12n
2	Basic elements: Trace$(U_{j,k} \cdot I_j \cdot U_{j,u}{}^T)$	128	99
3	Calculate the basic element $(n^3+3n^2+2n)/6$ times	$21n^3+63n^2+42n$	$16.5n^3+49.5n^2+33n$
4	Addition of the basic elements	0	$1.5n^3-1.5n$
Total		$21n^3+63n^2+66n$	$18n^3+49.5n^2+43.5n$
Dynamic method	The proposed method in this paper		
1	Basic elements: ${}^i\boldsymbol{I}_l - m_l \cdot {}^{i|k}\widetilde{\boldsymbol{r}}_{lI} \cdot {}^{i|u}\widetilde{\boldsymbol{r}}_{lI}$	21	12
2	Calculate the basic element $(n^3+3n^2+2n)/6$ times	$3.5n^3+10.5n^2+7n$	$2n^3+6n^2+4n$
3	Calculate ${}^{i|\bar{u}}\boldsymbol{n}_u^T \cdot () \cdot {}^{i|\bar{k}}\boldsymbol{n}_k$ $(n^2+n)/2$ times	$5.5n^2+5.5n$	$4n^2+4n$
4	Addition of the basic elements	0	$1.5n^3-1.5n$
Total		$3.5n^3+16n^2+12.5n$	$3.5n^3+10n^2+6.5n$

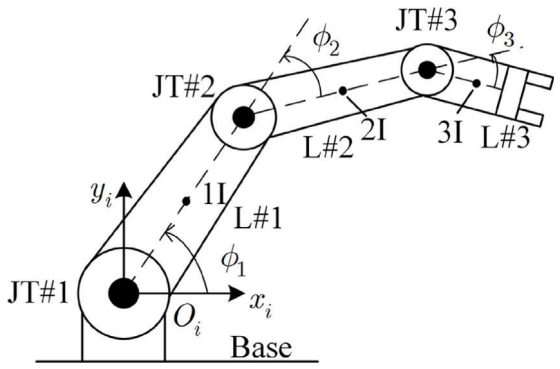


Fig. 4. Planar three DOF manipulator (JT — joint, L — link)

7. Application of JSIM modelling method

The correctness of the proposed method is verified through a planar 3-DOF manipulator in [39]. Then this method is further applied in a lunar rover example.

7.1. Example of 3-DOF serial-chain planar manipulator

As shown in Fig. 4, the manipulator moves on the x-y plane, the gravity is along the negative y direction, and the rotation axis is along the vertical direction of the x-y plane (denoted as the z direction). As for link $\#k(k = 1, 2, 3)$, the distance from the origin of the joint frame $\boldsymbol{F}^{[k]}$ to the centre of mass is denoted as l_{kI}, the length as l_k, the mass as m_k, the axis vector as ${}^{\bar{k}}\boldsymbol{n}_k = [0, 0, 1]^{\mathrm{T}}$ and the moment of inertia as $\boldsymbol{I}_k$. The angular position of joint $\#k$ is denoted as ϕ_k, $\cos\phi_1$ is denoted as C_1 and $\cos(\phi_1 + \phi_2)$ is denoted as C_{12}.

The JSIM is obtained by substituting the kinematic parameters into the explicit expression of JSIM directly. Because all joints are revolute, the upper triangular and diagonal elements of the JSIM can be obtained by Eq. (32)

$$\boldsymbol{M}^{[1][1]} = {}^i\boldsymbol{n}_1^T \cdot \begin{pmatrix} {}^i\boldsymbol{I}_1 - m_1 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{1I}^{\wedge 2} + {}^i\boldsymbol{I}_2 - m_2 \cdot \\ {}^{i|1}\widetilde{\boldsymbol{r}}_{2I}^{\wedge 2} + {}^i\boldsymbol{I}_3 - m_3 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{3I}^{\wedge 2} \end{pmatrix} \cdot {}^i\boldsymbol{n}_1$$
$$= I_1 + I_2 + I_3 + m_1 l_{1I}^2 + m_2 \left(l_1^2 + l_1 l_2 \mathrm{C}_1 + l_{2I}^2 \right) + m_3 \left(l_1^2 + l_2^2 + l_{3I}^2 + 2 l_1 l_2 \mathrm{C}_1 + 2 l_1 l_{3I} \mathrm{C}_3^1 + 2 l_2 l_{3I} \mathrm{C}_2 \right)$$
$$\boldsymbol{M}^{[1][2]} = {}^i\boldsymbol{n}_1^T \cdot \begin{pmatrix} {}^i\boldsymbol{I}_2 - m_2 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{2I} \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{2I} \\ + {}^i\boldsymbol{I}_3 - m_3 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{3I} \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{3I} \end{pmatrix} \cdot {}^{i|1}\boldsymbol{n}_2$$
$$= I_2 + I_3 + m_2 \left(l_{2I}^2 + l_1 l_{2I} \mathrm{C}_1 \right) + m_3 \left(l_2^2 + l_{3I}^2 + l_1 l_2 \mathrm{C}_1 + l_1 l_{3I} \mathrm{C}_{12} + 2 l_2 l_{3I} \mathrm{C}_2 \right)$$
$$\boldsymbol{M}^{[1][3]} = {}^i\boldsymbol{n}_1^T \cdot \left({}^i\boldsymbol{I}_3 - m_3 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{3I} \cdot {}^{i|3}\widetilde{\boldsymbol{r}}_{3I} \right) \cdot {}^{i|2}\boldsymbol{n}_3$$
$$= I_3 + m_3 \left(l_{3I}^2 + l_1 l_{3I} \mathrm{C}_{12} + l_2 l_{3I} \mathrm{C}_2 \right)$$
$$\boldsymbol{M}^{[2][2]} = {}^{i|1}\boldsymbol{n}_2^T \cdot \begin{pmatrix} {}^i\boldsymbol{I}_2 - m_2 \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{2I}^{\wedge 2} \\ + {}^i\boldsymbol{I}_3 - m_3 \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{3I}^{\wedge 2} \end{pmatrix} \cdot {}^{i|1}\boldsymbol{n}_2$$
$$= I_2 + I_3 + m_2 l_{2I}^2 + m_3 \left(l_2^2 + l_{3I}^2 + 2 l_2 l_{3I} \mathrm{C}_2 \right)$$
$$\boldsymbol{M}^{[2][3]} = {}^{i|1}\boldsymbol{n}_2^T \cdot \left({}^i\boldsymbol{I}_3 - m_3 \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{3I} \cdot {}^{i|3}\widetilde{\boldsymbol{r}}_{3I} \right) \cdot {}^{i|2}\boldsymbol{n}_3$$
$$= I_3 + m_3 \left(l_{3I}^2 + l_2 l_{3I} \mathrm{C}_2 \right)$$
$$\boldsymbol{M}^{[3][3]} = {}^{i|2}\boldsymbol{n}_3^T \cdot \left({}^i\boldsymbol{I}_3 - m_3 \cdot {}^{i|3}\widetilde{\boldsymbol{r}}_{3I}^{\wedge 2} \right) \cdot {}^{i|2}\boldsymbol{n}_3$$
$$= I_3 + m_3 l_{3I}^2$$

The lower triangle element can be obtained by Eq. (30)

$$\boldsymbol{M}^{[2][1]} = {}^{i|1}\boldsymbol{n}_2^T \cdot \begin{pmatrix} {}^i\boldsymbol{I}_2 - m_2 \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{2I} \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{2I} \\ + {}^i\boldsymbol{I}_3 - m_3 \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{3I} \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{3I} \end{pmatrix} \cdot {}^i\boldsymbol{n}_1$$
$$= I_2 + I_3 + m_2 \left(l_{2I}^2 + l_1 l_{2I} \mathrm{C}_2^1 \right) + m_3 \left(l_2^2 + l_{3I}^2 + l_1 l_2 \mathrm{C}_2^1 + l_1 l_{3I} \mathrm{C}_3^1 + 2 l_2 l_{3I} \mathrm{C}_3^2 \right)$$
$$\boldsymbol{M}^{[3][1]} = {}^{i|2}\boldsymbol{n}_3^T \cdot \left({}^i\boldsymbol{I}_3 - m_3 \cdot {}^{i|3}\widetilde{\boldsymbol{r}}_{3I} \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{3I} \right) \cdot {}^i\boldsymbol{n}_1$$
$$= I_3 + m_3 \left(l_{3I}^2 + l_1 l_{3I} \mathrm{C}_3^1 + l_2 l_{3I} \mathrm{C}_3^2 \right)$$
$$\boldsymbol{M}^{[3][2]} = {}^{i|2}\boldsymbol{n}_3^T \cdot \left({}^i\boldsymbol{I}_3 - m_3 \cdot {}^{i|3}\widetilde{\boldsymbol{r}}_{3I} \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{3I} \right) \cdot {}^{i|1}\boldsymbol{n}_2$$
$$= I_3 + m_3 \left(l_{3I}^2 + l_2 l_{3I} \mathrm{C}_3^2 \right)$$

In order to show the completeness of the algorithm, all the elements of the JSIM are calculated. In fact, only the lower triangle and diagonal elements need to be calculated. By comparison with the computation results in [29], the correctness of the proposed method is proved. In order to make meaningful comparisons of modelling complexity, some major processes in the dynamic modelling method already mentioned

Table 2
Comparison of modelling complexity.

Main process of Modelling	Recursive method	Explicit method		
	CRBA and its improved methods	Generalized momentum method	Pseudo-inertia matrix method	Proposed method
Parameter substitution	Need	Need	Need	Need
Establishment of Pseudo-inertia matrix	Need	No Need	Need	No Need
Establishment of Jacobian matrix	No Need	Need	No Need	No Need
Recursive process	Need	No Need	No Need	No Need

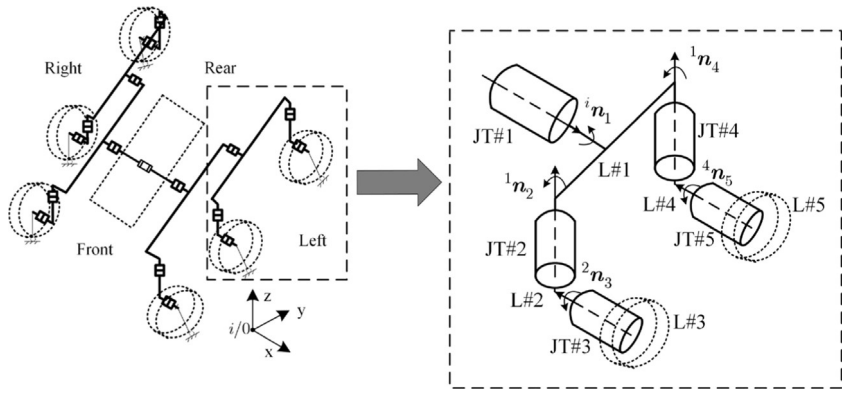
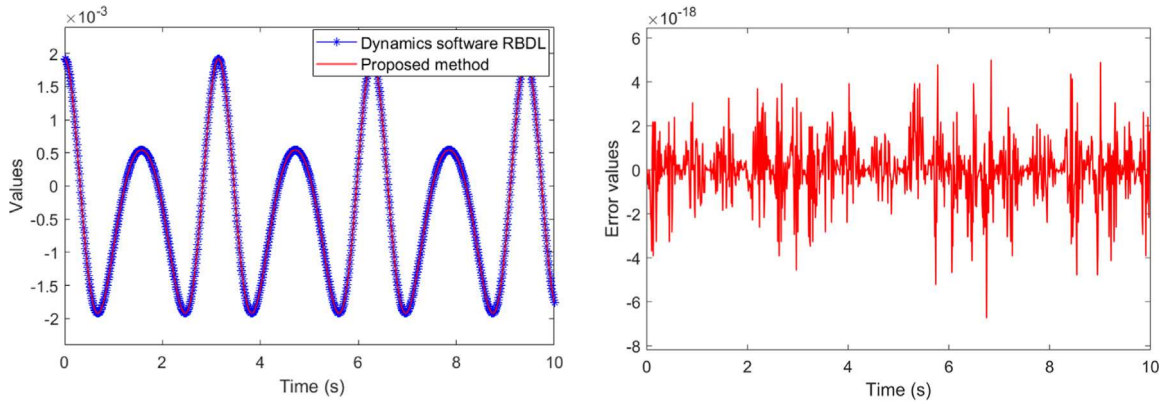


Fig. 5. Left rear rocker arm of the lunar rover.



(a)Values of the $\boldsymbol{M}^{[2][1]}$ obtained by the commercial dynamics software RBDL and proposed methods

(b)Calculation error of the $\boldsymbol{M}^{[2][1]}$ between the generalized momentum and proposed methods

Fig. 6. Comparison of the JSIM calculation results between commercial dynamics software RBDL and our proposed method.

in this paper are provided in Table 2. It can be seen from the comparison results that the proposed method has fewer modelling steps, thus, it has higher modelling efficiency.

7.2. Example of the lunar rover

The freedom degree of the whole lunar rover is so large that the modelling results of JSIM takes up too much paper, therefore, we select the left rear rocker arm of the lunar rover for verifying the correctness of the proposed method in tree-chain applications. Finally, we give the simulation results for the traction control of the whole lunar rover in rugged terrain.

The mechanical diagram of the left rear rocker arm is shown in Fig. 5. The rocker arm has two branches at link #1, where the axis #2 and #4 are along the direction of z-axis and the axis #3 and #5 are along the direction of x-axis. The detailed kinematic and dynamic parameters are shown in Table 3.

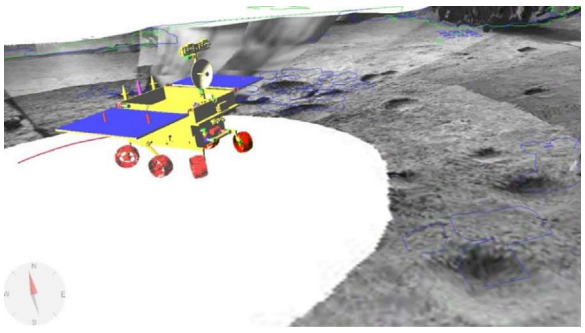


Fig. 7. Circular motion of Lunar rover on rugged terrain.

The equations of diagonal and upper triangular elements are obtained according to Eq. (32), where $\boldsymbol{M}^{[2][4]}$, $\boldsymbol{M}^{[2][5]}$, $\boldsymbol{M}^{[3][4]}$, $\boldsymbol{M}^{[3][5]}$ and

Table 3
Parameters of tree-chain manipulator.

Number l	${}^{\tilde{l}}\boldsymbol{n}_l$	${}^{\tilde{l}}\boldsymbol{r}_l$(m)	${}^{l}\boldsymbol{r}_{lI}$(m)	$\boldsymbol{I}_l$ (kg m^2)	m_l(kg)
1	[1,0,0] $^{\mathrm{T}}$	[0,0,0] $^{\mathrm{T}}$	[0.05,0,0] $^{\mathrm{T}}$	[0 0 0;0 0.0833 0;0 0 0.0833]	1
2	[0,0,1] $^{\mathrm{T}}$	[0.1,−0.05,0] $^{\mathrm{T}}$	[0,0,−0.04] $^{\mathrm{T}}$	[0.0427 0 0;0 0. 0427 0;0 0 0]	0.8
3	[1,0,0] $^{\mathrm{T}}$	[0.01,0,−0.08] $^{\mathrm{T}}$	[0.03,0,0] $^{\mathrm{T}}$	[0 0 0;0 0. 018 0;0 0 0. 018]	0.6
4	[0,0,1] $^{\mathrm{T}}$	[0.1,0.05,0] $^{\mathrm{T}}$	[0,0,−0.04] $^{\mathrm{T}}$	[0. 0427 0 0;0 0.0427 0;0 0 0]	0.8
5	[1,0,0] $^{\mathrm{T}}$	[0.01,0,−0.08] $^{\mathrm{T}}$	[0.03,0,0] $^{\mathrm{T}}$	[0 0 0;0 0. 018 0;0 0 0. 018]	0.6

Table A.1
Symbols description.

Symbols	Description	Symbols	Description
$\boldsymbol{F}^{[l]}$	Reference frame of joint #l	$\boldsymbol{M}_{\boldsymbol{R}}^{[u][*]}$	Inertial force vector of revolute joint #u
$\boldsymbol{F}^{[i]}$	Inertial frame	$\boldsymbol{M}_{\boldsymbol{P}}^{[u][*]}$	Inertial force vector of prismatic joint #u
${}^{i}\boldsymbol{l}_l$	Kinematic chain from link #i to #l	$\boldsymbol{M}^{[u][k]}$	uth row and kth column element of JSIM
$u\boldsymbol{L}$	Subtree of link u	$\boldsymbol{h}^{[u]}$	uth element of bias force vector
${}^{\tilde{l}}\boldsymbol{R}_l$	Rotation matrix from $\boldsymbol{F}^{[l]}$ to $\boldsymbol{F}^{[\tilde{l}]}$	$\boldsymbol{h}_{\boldsymbol{R}}^{[u]}$	Non-inertial force vector of revolute joint #u
ϕ_l	Angular position along ${}^{\tilde{l}}\boldsymbol{n}_l$	$\boldsymbol{h}_{\boldsymbol{P}}^{[u]}$	Non-inertial force vector of prismatic joint #u
$\dot{\phi}_l(\ddot{\phi}_l)$	Rotational velocity (acceleration) of joint #l	${}^{i}\boldsymbol{I}_k$	Inertia tensor with respect to the inertia centre of link #k expressed in Frame #i
${}^{\tilde{l}}\dot{\boldsymbol{\phi}}_l({}^{\tilde{l}}\ddot{\boldsymbol{\phi}}_l)$	Vector form of $\dot{\phi}_l(\ddot{\phi}_l)$	m_k	Mass of link #k
r_l	Linear position along ${}^{\tilde{l}}\boldsymbol{n}_l$	$\boldsymbol{g}$	Gravity vector
$\dot{r}_l(\ddot{r}_l)$	Translational velocity (acceleration) from origin of $\boldsymbol{F}^{[\tilde{l}]}$ to origin of $\boldsymbol{F}^{[\tilde{l}]}$	$\boldsymbol{k}_l$	Joint #l
${}^{\tilde{l}}\boldsymbol{r}_l$	Linear position vector from origin of $\boldsymbol{F}^{[\tilde{l}]}$ to origin of $\boldsymbol{F}^{[l]}$	$\boldsymbol{R}$	Revolute joint
${}^{\tilde{l}}\boldsymbol{l}_l$	Structural vector from origin of $\boldsymbol{F}^{[\tilde{l}]}$ to origin of $\boldsymbol{F}^{[l]}$	$\boldsymbol{P}$	Prismatic joint
${}^{\tilde{l}}\dot{\boldsymbol{r}}_l({}^{\tilde{l}}\ddot{\boldsymbol{r}}_l)$	Translational velocity (acceleration) vector from origin of $\boldsymbol{F}^{[\tilde{l}]}$ to origin of $\boldsymbol{F}^{[l]}$	$\boldsymbol{f}_u$	Resultant force on axis ${}^{\tilde{u}}\boldsymbol{n}_u$ except gravity
${}^{\tilde{l}}\boldsymbol{n}_l$	Unit axis vector of joint #l	$\boldsymbol{\tau}_u$	Resultant torque on axis ${}^{\tilde{u}}\boldsymbol{n}_u$ except gravity torque
$l\boldsymbol{S}$	Point on the link #l	$\widetilde{\square}$	Cross-product operator
$l\boldsymbol{I}$	Mass centre of the link #l	${}^{\square}|_{\square}$	Projection operator
$\hat{\square}$	Exponent operator	$\dot{\square}$	Derivative operator

their symmetry elements are zero because of the branched structure.

$$\boldsymbol{M}^{[1][1]} = {}^{i}\boldsymbol{n}_1^T \cdot \begin{pmatrix} {}^{i}\boldsymbol{I}_1 - m_1 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{1I}^{\wedge 2} + {}^{i}\boldsymbol{I}_2 - m_2 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{2I}^{\wedge 2} + \\ {}^{i}\boldsymbol{I}_3 - m_3 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{3I}^{\wedge 2} + {}^{i}\boldsymbol{I}_4 - m_4 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{4I}^{\wedge 2} \\ + {}^{i}\boldsymbol{I}_5 - m_5 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{5I}^{\wedge 2} \end{pmatrix} \cdot {}^{i}\boldsymbol{n}_1$$

$$\boldsymbol{M}^{[1][2]} = {}^{i}\boldsymbol{n}_1^T \cdot \begin{pmatrix} {}^{i}\boldsymbol{I}_2 - m_2 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{2I} \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{2I} \\ + {}^{i}\boldsymbol{I}_3 - m_3 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{3I} \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{3I} \end{pmatrix} \cdot {}^{i|1}\boldsymbol{n}_2$$

$$\boldsymbol{M}^{[1][3]} = {}^{i}\boldsymbol{n}_1^T \cdot \left({}^{i}\boldsymbol{I}_3 - m_3 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{3I} \cdot {}^{i|3}\widetilde{\boldsymbol{r}}_{3I} \right) \cdot {}^{i|2}\boldsymbol{n}_3$$

$$\boldsymbol{M}^{[1][4]} = {}^{i}\boldsymbol{n}_1^T \cdot \begin{pmatrix} {}^{i}\boldsymbol{I}_4 - m_4 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{4I} \cdot {}^{i|4}\widetilde{\boldsymbol{r}}_{4I} \\ + {}^{i}\boldsymbol{I}_5 - m_5 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{5I} \cdot {}^{i|4}\widetilde{\boldsymbol{r}}_{5I} \end{pmatrix} \cdot {}^{i|1}\boldsymbol{n}_4$$

$$\boldsymbol{M}^{[1][5]} = {}^{i}\boldsymbol{n}_1^T \cdot \left({}^{i}\boldsymbol{I}_5 - m_5 \cdot {}^{i|1}\widetilde{\boldsymbol{r}}_{5I} \cdot {}^{i|5}\widetilde{\boldsymbol{r}}_{5I} \right) \cdot {}^{i|4}\boldsymbol{n}_5$$

$$\boldsymbol{M}^{[2][2]} = {}^{i|1}\boldsymbol{n}_2^T \cdot \left({}^{i}\boldsymbol{I}_2 - m_2 \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{2I}^{\wedge 2} + {}^{i}\boldsymbol{I}_3 - m_3 \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{3I}^{\wedge 2} \right) \cdot {}^{i|1}\boldsymbol{n}_2$$

$$\boldsymbol{M}^{[2][3]} = {}^{i|1}\boldsymbol{n}_2^T \cdot \left({}^{i}\boldsymbol{I}_3 - m_3 \cdot {}^{i|2}\widetilde{\boldsymbol{r}}_{3I} \cdot {}^{i|3}\widetilde{\boldsymbol{r}}_{3I} \right) \cdot {}^{i|2}\boldsymbol{n}_3$$

$$\boldsymbol{M}^{[3][3]} = {}^{i|2}\boldsymbol{n}_3^T \cdot \left({}^{i}\boldsymbol{I}_3 - m_3 \cdot {}^{i|3}\widetilde{\boldsymbol{r}}_{3I}^{\wedge 2} \right) \cdot {}^{i|2}\boldsymbol{n}_3$$

$$\boldsymbol{M}^{[4][4]} = {}^{i|1}\boldsymbol{n}_4^T \cdot \left({}^{i}\boldsymbol{I}_4 - m_4 \cdot {}^{i|4}\widetilde{\boldsymbol{r}}_{4I}^{\wedge 2} \right) \cdot {}^{i|1}\boldsymbol{n}_4$$

$$\boldsymbol{M}^{[4][5]} = {}^{i|1}\boldsymbol{n}_4^T \cdot \left({}^{i}\boldsymbol{I}_5 - m_5 \cdot {}^{i|4}\widetilde{\boldsymbol{r}}_{5I} \cdot {}^{i|5}\widetilde{\boldsymbol{r}}_{5I} \right) \cdot {}^{i|4}\boldsymbol{n}_5$$

$$\boldsymbol{M}^{[5][5]} = {}^{i|4}\boldsymbol{n}_5^T \cdot \left({}^{i}\boldsymbol{I}_5 - m_5 \cdot {}^{i|5}\widetilde{\boldsymbol{r}}_{5I}^{\wedge 2} \right) \cdot {}^{i|4}\boldsymbol{n}_5$$

Then the proposed explicit method and generalized momentum method are simulated, where the simulation step is 0.01s and the simulation time is 10 s. Assuming the joint angles vary by 5sin(t), the values of inertial matrix elements $\boldsymbol{M}^{[2][1]}$ are recorded in Fig. 6(a). The error of calculation results between these two methods are in 1e-18 level as shown in Fig. 6(b), which verifies correctness of the proposed method in tree-chain dynamic system.

Finally, the modelling results are used in traction control of the lunar rover. The simulation software of lunar rover is developed by c++ and open inventor library. To perform the circling around motion, we assume that the rover starts at coordinate (5 m,5 m), the front direction joint angle is 5°, the middle direction angle is 0°, and the rear direction angle is −5°, as shown in Fig. 7. The simulation results of the rover are shown in Figs. 8 and 9, where Fig. 8 shows that the rover's direction joints can be controlled to the desired angle, and Fig. 9 shows the torque fluctuation caused by wheel contact with the rough surface. Due to the rocker arm, the impact torques of the rear and middle direction joints are smaller than those of the front direction joint.

8. Conclusion

This paper proposes a novel explicit modelling method of the JSIM for tree-chain dynamic systems. The explicit expressions of JSIM elements are directly given in this method. As a result, the modelling of JSIM can be implemented with only system input parameters substitution process without construction of intermediate variables. Moreover, the elements of the matrix are independent of each other. Using this property, we prove the symmetry of the inertia matrix mathematically. Furthermore, it is advantageous to the implementation of parallel computing to reduce computational complexity. By comparing the computational complexity and modelling steps with the existing explicit methods, it is proved that our method has higher modelling and computational efficiency. To validate the correctness of the proposed method in the application of serial-chain systems, a numerical example of a 3-DOF serial manipulator is given. In addition, a simulation of the lunar rover's left rear rocker arm is used to validate the proposed method in the application of tree-chain systems. Finally, this method is

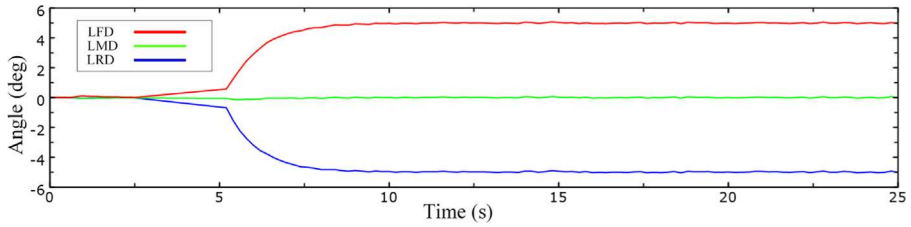


Fig. 8. The angle of left front direction (LFD), left middle direction (LMD) and left rear direction (LRD) joint.

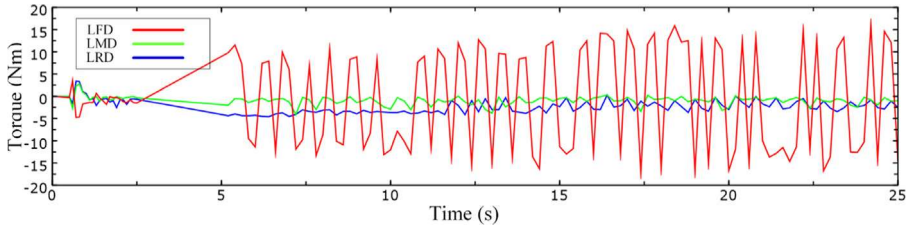


Fig. 9. Driving torques of LFD, LMD and LMD joints.

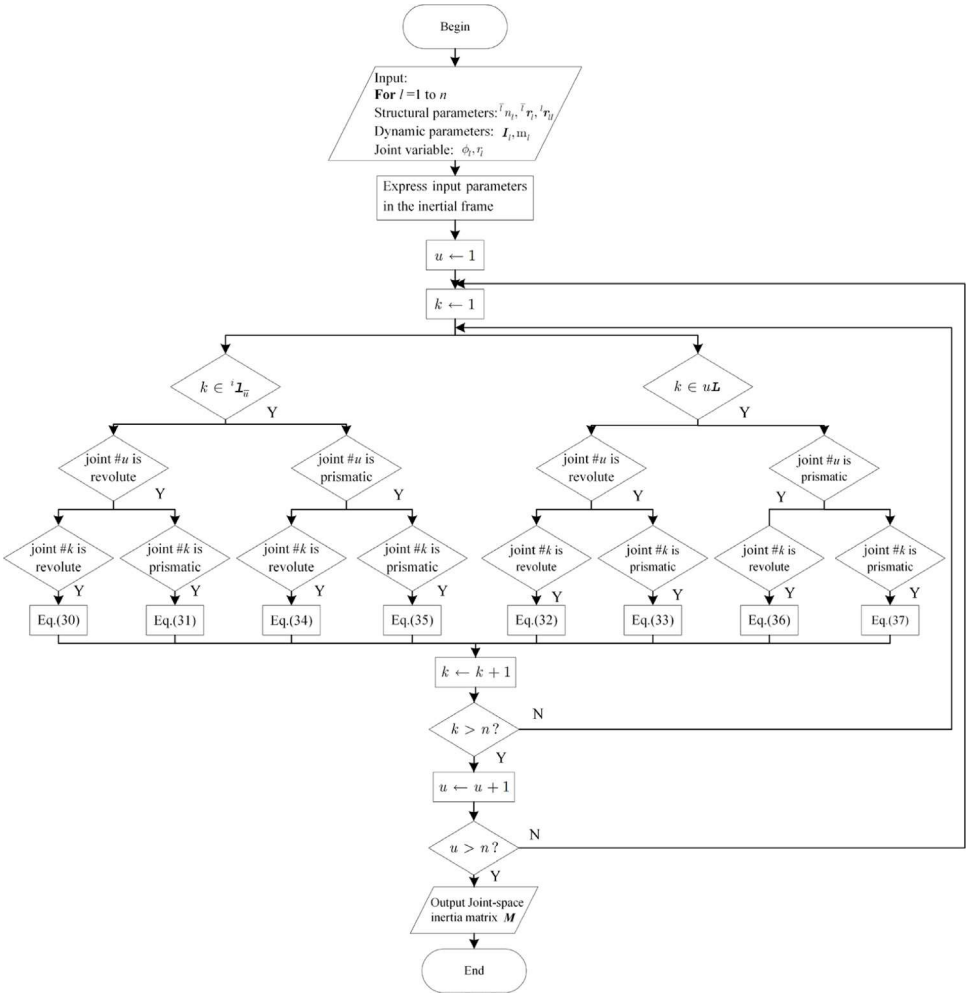


Fig. A.1. The flowchart of implementation process of JSIM.

applied to the lunar rover's traction control. This method will be combined with a parallel algorithm in the future to improve computational efficiency.

CRediT authorship contribution statement

Kaimeng Wang: Conceptualization, Software, Investigation, Formal analysis, Writing – original draft. **Hehua Ju:** Methodology, Data curation, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was funded by the National Natural Science Foundation of China, grant number 61673010. The author sincerely acknowledges the support.

Appendix

The symbols used in this paper is described in Table A.1.
See Fig. A.1.

References

[1] A. Mohan, S.K. Saha, A recursive, numerically stable, and efficient simulation algorithm for serial robots, Multibody Syst. Dyn. 17 (4) (2007) 291–319.
[2] J. Liu, R. Liu, Dynamics modelling and simulation of an industrial manipulator subject to constraint, Mech. Mech. Eng. (2016) 3–13.
[3] A. Agarwal, S.V. Shah, S. Bandyopadhyay, S.K. Saha, Dynamics of serial kinematic chains with large number of degrees-of-freedom, Multibody Syst. Dyn. 32 (2014) 273–298.
[4] M.H. Korayem, A.M. Shafei, Motion equation of nonholonomic wheeled mobile robotic manipulator with revolute–prismatic joints using recursive Gibbs–Appell formulation, Appl. Math. Model. 84 (2014) 187–206.
[5] T.T. Do, V.H. Vu, Z.H. Liu, Symbolic differentiation algorithm for inverse dynamics of serial robots with flexible joints, J. Mech. Robot. 13 (6) (2021) 501–509.
[6] Fei Yn, Wu Qh, Tracking control of robot manipulators via output feedback linearization, Front. Mech. Eng. China 1 (2006) 329–335.
[7] F. Piltan, M.H. Yarmahmoudi, M. Mirzaie, et al., Design novel fuzzy robust feedback linearization control with application to robot manipulator, Int. J. Intell. Syst. Appl. 5 (5) (2013) 1–10.
[8] M. Fonseca, B.V. Adorno, P. Fraisse, Task-space admittance controller with adaptive inertia matrix conditioning, J. Intell. Robot. Syst. 101 (41) (2021).
[9] R. Featherstone, An empirical study of the joint space inertia matrix, Int. J. Robot. Res. 23 (9) (2004) 859–871.
[10] M.W. Walker, D.E. Orin, Efficient dynamics computer simulation of robotic mechanisms, Trans. ASME J. Dyn. Syst. 104 (3) (1982) 205–211.
[11] J. Carpentier, G. Saurel, G. Buondonno, The Pinocchio C++ library—A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives, in: IEEE Int. Symposium. Syst. Integr, 2019.
[12] Felis L. Martin, RBDL: an efficient rigid-body dynamics library using recursive algorithms, Auton. Robots 41 (2) (2017) 495–511.
[13] R. Featherstone, Rigid Body Dynamics Algorithms, Springer, New York, 2008.
[14] C.A. Balafoutis, R.V. Patel, Efficient computation of manipulator inertia matrices and the direct dynamics problem, IEEE Trans. Syst. Man. Cybern. 19 (5) (1989) 1313–1321.
[15] C.A. Balafoutis, R.V. Patel, Efficient modelling and computation of manipulator dynamics using orthogonal cartesian tensors, IEEE J. Robot. Autom. 4 (6) (1988) 665–676.
[16] K.W. Lilly, D.E. Orin, Alternate formulations for the manipulator inertia matrix, Int. J. Robot. Res. 10 (1) (1991) 64–74.
[17] M. Safeea, R. Bearee, P. Neto, Reducing the computational complexity of mass-matrix calculation for high DOF robots, in: IEEE/RSJ Int. Conf. Intell. Rob. Syst, 2018.
[18] S.K. Saha, A decomposition of the manipulator inertia matrix, IEEE Trans. Robot. Autom. 13 (2) (1997) 301–304.
[19] A.M. Shafei, H.R. Shafei, A systematic method for the hybrid dynamic modelling of open kinematic chains confined in a closed environment, Multibody Syst. Dyn. 38 (1) (2016) 21–42.
[20] A.M. Shafei, H.R. Shafei, Dynamic modelling of tree-type robotic systems by combining 3 × 3 rotation and 4 × 4 transformation matrices, Multibody Syst. Dyn. 44 (2018) 367–395, http://dx.doi.org/10.1007/s11044-018-09642-4.
[21] J. Brinker, B. Corves, Lagrangian based dynamic analyses of delta robots with serial-parallel architecture, in: Symposium on Robot Design, Dynamics and Control, Springer, 2016.
[22] X. Chen, C. Sun, Dynamic modelling of spatial parallel mechanism with multi-spherical joint clearances, Int. J. Adv. Robot. Syst. 16 (5) (2019).
[23] C.P. Neuman, J.J. Murray, Symbolically efficient formulations for computational robot dynamics, J. Robot. Syst. 4 (6) (1987) 743–769.
[24] R. Boughdiri, H. Nasser, H. Bezine, et al., Lagrange dynamic modelling of a multi-fingered robot hand in free motion considering the coupling dynamics, Adv. Mater. Res. 165 (2012) 1659–1663.
[25] M.P. Cheng, C.C. Tsai, Dynamic modelling and tracking control of a non-holonomic wheeled mobile manipulator with two robotic arms, in: 42nd IEEE Conference on Decision and Control, 2003, pp. 2932–2937.
[26] J.M. Hollerbach, A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity, IEEE Trans. Syst. Man. Cybern. (1980) 730–736.
[27] C.J. Li, A new lagrangian formulation of dynamics for robot manipulators, J. Dyn. Syst. Meas. Control 111 (4) (1989) 545.
[28] J.F. Pan, J.R. Wan, et al., Dynamics modelling of spraying robot using Lagrangian method with co-simulation analysis, in: IEEE International Conference on Electrical Engineering and Mechatronics Technology, 2021.
[29] P.J. From, An explicit formulation of singularity-free dynamic equations of mechanical systems in lagrangian form—part one: single rigid bodies, Model. Identif. Control 33 (2) (2012) 45–60.
[30] P.J. From, An explicit formulation of singularity-free dynamic equations of mechanical systems in lagrangian form—part two: multibody systems, Model. Identif. Control 33 (2) (2012) 61–68.
[31] B. Siciliano, L. Sciavicco, Robotics: Modelling, Planning and Control, Springer, New York, 2010.
[32] B. Siciliano, O. Khatib, Springer Handbook of Robotics, Springer, New York, 2016.
[33] G. Garofalo, C. Ott, A. Albu-Schäffer, On the closed form computation of the dynamic matrices and their differentiations, in: IEEE/RSJ Inter. Conf. Intell. Rob. Syst, 2013.
[34] M.A. Renaud, Near Minimum Iterative Analytical Procedure for Obtaining a Robot Manipulator Dynamic Model, Springer Berlin Heidelberg, 1986.
[35] V.D. Tourassis, C.P. Neuman, Properties and structure of dynamic robot models for control engineering applications, Mech. Mach. Theory 20 (1) (1985) 27–40.
[36] V.D. Tourassis, C.P. Neuman, The inertial characteristics of dynamic robot models, Mech. Mach. Theory 20 (1) (1985) 41–52.
[37] H.H. Ju, Axis-invariants based tree-chain robot dynamic modelling and calculation method. China Patent, CN201810933332.3, 2020.
[38] H.H. Ju, Axis-invariants based MAS dynamic modelling method, China Patent, CN201810933374.7, 2020.
[39] Z. Hussain, N.Z. Azlan, KANE's method for dynamic modelling, in: IEEE Inter. Conf. Autom. Control. Intell. Syst, 2016.