Robotica: (2020) volume 38, pp. 2138–2150. © Cambridge University Press 2020
doi:10.1017/S0263574719001899

Comparison of Kane's and Lagrange's Methods in Analysis of Constrained Dynamical Systems

Amin Talaeizadeh, Mahmoodreza Forootan, Mehdi Zabihi and Hossein Nejat Pishkenari*

Department of Mechanical Engineering, Sharif University of Technology, Tehran, Iran.
E-mails: amtalaiezadeh@gmail.com, mahmood.foroutan@yahoo.com, mehdizabihi@yahoo.com

(Accepted December 21, 2019. First published online: February 10, 2020)

SUMMARY
Dynamic modeling is a fundamental step in analyzing the movement of any mechanical system. Methods for dynamical modeling of constrained systems have been widely developed to improve the accuracy and minimize computational cost during simulations. The necessity to satisfy constraint equations as well as the equations of motion makes it more critical to use numerical techniques that are successful in decreasing the number of computational operations and numerical errors for complex dynamical systems. In this study, performance of a variant of Kane's method compared to six different techniques based on the Lagrange's equations is shown. To evaluate the performance of the mentioned methods, snake-like robot dynamics is considered and different aspects such as the number of the most time-consuming computational operations, constraint error, energy error, and CPU time assigned to each method are compared. The simulation results demonstrate the superiority of the variant of Kane's method concerning the other ones.

KEYWORDS: Kane method; Lagrange's method; Snake robot; Constrained dynamical systems; Derivation of motion equations.

1. Introduction

Derivation of the motion equations plays an essential role in analyzing mechanical systems. Many analytical methods such as Newton–Euler equations,[1] Lagrange's method,[2] Kane's method,[3–5] Gibbs–Appell's approach[6,7] Maggi's method,[8] and Udwadia–Kalaba approach[9] have been developed in this regard. Different numerical methods employed in solving motion equations result in different aspects such as computational errors and time consumption. These aspects would be more evident when the system of interest is constrained via nonholonomic constraints.

Many approaches, including the well-known Lagrange's method, were employed for deriving motion equations for constrained multibody systems. Lagrange's method has been used to derive motion equations of nonholonomic constrained systems by considering the constraints of the system as Lagrange's multipliers (constraint reaction forces).[10] A large number of research studies were conducted to improve the Lagrange's method such as a representation of a matrix form of the Lagrangian equations for constrained systems[11] and introducing a novel approach to derive the motion equations of nonholonomic constrained systems by the elimination of the constraint reaction forces.[12]

However, this method has a few drawbacks, including constraint violation and computational error accumulation. These are the result of applying constraint equations in acceleration form and a large number of differential equations in the solving process. Methods used for elimination include

* Corresponding author. E-mail: nejat@sharif.edu

augmenting corrective terms within the dynamical equation formulations,[13] correcting state variables after every integration step,[14,15] and solving constraint equations in position and velocity level.[15] Moreover, to deal with the problem of error accumulation, an explicit form of motion equations was presented for the constrained systems.[16]

Kane's method is a well-known new approach that can partially resolve the Lagrange's method's problems. In this method, as many variables, called generalized speeds, as the system's degrees of freedom (DOF) are used instead of generalized velocities, and the number of differential equations decreases (compared to the number of generalized coordinates) in the constrained systems. Kane's method also uses constraint equations at the velocity level to eliminate constraints violation.[3,4] This method has been further developed during the last decade, and a new matrix form of Kane's equations was presented that leads to a significant reduction in computational efforts.[17] Moreover, Kane's method was also applied in the study of impulsive constraints,[4] and it is extended to the non-minimal nonholonomic form in ref. [18]. Furthermore, to derive linearized motion equations, an approach is developed based on Kane's method.[19]

Although in Kane's method constraints are expressed at the velocity level, some researches were conducted on applying constraints at the acceleration level. A new variation of Kane's method was proposed, which employs the acceleration form of the constraint equations.[20,21] This issue causes constraints violation in the numerical solution, which can be eliminated by using constraint stabilization terms.[22] The extensions of Kane's method were applied to derive an explicit form of motion equations for a constrained system by using nonholonomic partial accelerations.[23] Applications of Kane's method in modeling dynamic systems are found in refs. [17, 24, 25].

We chose snake-like robots as an appropriate case for this study. A survey on modeling and motion equations of the snake-like robots is published in ref. [26]. Moreover, the equations of motion of a snake-like robot are obtained with Gibbs–Appell's method in ref. [27]. In this work, Lagrange's and Kane's methods are applied to derive the motion equations of a passive wheel snake-like robot with N modules.

In this robot, each module contains a homogenous link connected to neighboring modules by revolute joints and a wheel placed at the link's center of mass (CoM). The orientation of each link with respect to its wheel's axis is fixed and considered arbitrary (except $\pm\pi/2$ for i-th link ($i \geq 2$) where the equations will be singular). Lagrange's equations are solved by six methods (in refs. [1, 10, 28]) named as integrated multiplier (IM), augmented (Aug), elimination (El), Greenwood (GW), embedding (Em), and modified Lagrange (ML) methods. Finally, simulation results for snake-like robots with 2- to 100-link provided enough data to make a comparison between these methods concerning required CPU time and computation errors. This paper is organized as follows: all methods and their numerical solution procedures are described in Section 2, the snake robot considered in this work is introduced in Section 3, and the methods are compared via simulation results in Section 4.

2. Lagrange's Method

In this study, Lagrange's method is applied to derive motion equations of constrained systems. Considering a system with m DOF and r constraints, its generalized coordinates and velocities are presented as follows:

$$\mathbf{q} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}^T, \quad \dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \cdots & \dot{q}_n \end{bmatrix}^T, \qquad n = m + r. \tag{1}$$

r nonholonomic constraint equations can be presented in the following matrix form:

$$\mathbf{c}_{r\times 1} = \begin{bmatrix} c_1 & c_2 & \cdots & c_r \end{bmatrix}^T = \mathbf{a}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \mathbf{b}_{r\times 1} = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^T_{\ r\times 1}. \tag{2}$$

The matrix $\mathbf{a}$ and vector $\mathbf{b}$ depend on the type of constraints. The time derivative of constraint equations in matrix form is

$$\mathbf{a}_{r\times n}\ddot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{a}}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{b}}_{r\times 1} = 0. \tag{3}$$

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

The Lagrange equations for constrained systems can be written as follows:

$$\begin{gathered}\ell = T - U,\\ \frac{d}{dt}\left(\frac{\partial \ell}{\partial \dot{q}_i}\right) - \frac{\partial \ell}{\partial q_i} = \mathbf{Q} + \sum_{k=1}^{r} \lambda_k a_{ik}, \quad i = 1, \cdots, n,\end{gathered} \tag{4}$$

where T is kinetic energy, U is potential energy, ℓ is Lagrangian, $\mathbf{Q}$ is generalized forces, t is time, and λ_k is Lagrange's multiplier.

Equation (4) could be written in the following matrix form:

$$\mathbf{M}_{n\times n}\ddot{\mathbf{q}}_{n\times 1} = \underbrace{(\mathbf{Q}_{n\times 1} - \mathbf{B}_{n\times 1})}_{\mathbf{F}_{n\times 1}} + \mathbf{a}^T_{n\times r}\boldsymbol{\lambda}_{r\times 1}, \qquad \boldsymbol{\lambda} = \begin{bmatrix}\lambda_1\\ \vdots \\ \lambda_r\end{bmatrix}, \tag{5}$$

where $\mathbf{M}_{n\times n}$ is the mass matrix and $\mathbf{B}_{n\times 1}$ is the bias vector.

Six methods were developed for solving these equations. In the following, these methods are briefly described.

3. IM Method

In the IM method, the states of the system and the state time derivatives are considered as follows:

$$\mathbf{z}_{(2n+r)\times 1} = \left[\mathbf{q}_{n\times 1}\ \dot{\mathbf{q}}_{n\times 1}\ \boldsymbol{\mu}_{r\times 1}\right]^T, \quad \dot{\mathbf{z}}_{(2n+r)\times 1} = \left[\dot{\mathbf{q}}_{n\times 1}\ \ddot{\mathbf{q}}_{n\times 1}\ \dot{\boldsymbol{\mu}}_{r\times 1}\right]^T, \tag{6}$$

where $\dot{\boldsymbol{\mu}}_{r\times 1} = \boldsymbol{\lambda}_{r\times 1}$.

By considering (3) and (5), governing equations derived by the IM method are as follows:[10]

$$\begin{gathered}\underbrace{\begin{bmatrix}\mathbf{I}_{n\times n} & [0]_{n\times n} & [0]_{n\times r}\\ [0]_{n\times n} & \mathbf{M}_{n\times n} & -\mathbf{a}^T{}_{n\times r}\\ [0]_{r\times n} & -\mathbf{a}_{r\times n} & [0]_{r\times r}\end{bmatrix}}_{\mathbf{M}_{IM(2n+r)\times(2n+r)}} \times \begin{bmatrix}\dot{\mathbf{q}}_{n\times 1}\\ \ddot{\mathbf{q}}_{n\times 1}\\ \dot{\boldsymbol{\mu}}_{r\times 1}\end{bmatrix} = \underbrace{\begin{bmatrix}\dot{\mathbf{q}}_{n\times 1}\\ \mathbf{F}_{n\times 1}\\ \dot{\mathbf{a}}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{b}}_{r\times 1}\end{bmatrix}}_{\mathbf{F}_{IM(2n+r)\times 1}},\\ \dot{\mathbf{z}}_{(2n+r)\times 1} = \mathbf{M}^{-1}_{IM(2n+r)\times(2n+r)}\mathbf{F}_{IM(2n+r)\times 1}.\end{gathered} \tag{7}$$

In this method, $2n + r$ equations should be solved. In other words, the most time-consuming part of the numerical solution is inverse operation on $\mathbf{M}$. So the dimension of $\mathbf{M}$ is a useful criterion for comparing the different methods from time-consumption aspect.

For solving these equations, each initial condition for $\boldsymbol{\mu}$ is admissible because only the derivative form of this vector appears in the equations. For simplicity, zeros are used for initial conditions of $\boldsymbol{\mu}$.

4. Au Method

In the Au method, the states of the system and the state time derivatives are defined as follows:

$$\mathbf{z}_{(n+r)\times 1} = \begin{bmatrix}\mathbf{q}_{n\times 1}\\ \boldsymbol{\mu}_{r\times 1}\end{bmatrix}, \quad \dot{\mathbf{z}}_{(n+r)\times 1} = \begin{bmatrix}\dot{\mathbf{q}}_{n\times 1}\\ \dot{\boldsymbol{\mu}}_{r\times 1}.\end{bmatrix} \tag{8}$$

By considering Eqs. (3) and (5), governing equations derived by Au method are as follows:[10]

$$\begin{gathered}\underbrace{\begin{bmatrix}\mathbf{M}_{n\times n} & -\mathbf{a}^T{}_{n\times r}\\ -\mathbf{a}_{r\times n} & [0]_{r\times r}\end{bmatrix}}_{\mathbf{M}_{Au((n+r)\times(n+r))}} \times \begin{bmatrix}\ddot{\mathbf{q}}_{n\times 1}\\ \dot{\boldsymbol{\mu}}_{r\times 1}\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{F}_{n\times 1}\\ \dot{\mathbf{a}}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{b}}_{r\times 1}\end{bmatrix}}_{\mathbf{F}_{Au((n+r)\times 1)}},\\ \begin{bmatrix}\ddot{\mathbf{q}}_{n\times 1}\\ \dot{\boldsymbol{\mu}}_{r\times 1}\end{bmatrix} = \mathbf{M}^{-1}_{Au((n+r)\times(n+r))}\mathbf{F}_{Au((n+r)\times 1)}.\end{gathered} \tag{9}$$

In this method, based on the dimension of $\mathbf{M}$, $n + r$ equations should be solved.

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

5. El Method

In the El method, the states of the system are selected similar to the Au method. By dividing Eq. (5) into two sets of equations and calculating Lagrange's multiplier by using the first set, as Eq. (10):[10]

$$\begin{aligned}
&\mathbf{M}_{n\times n}\ddot{\mathbf{q}}_{n\times 1} = \mathbf{F}_{n\times 1} + \mathbf{a}^{T}{}_{n\times r}\boldsymbol{\lambda}_{r\times 1},\\
&\begin{bmatrix} \mathbf{M}'_{r\times n} \\ \mathbf{M}''_{(n-r)\times n} \end{bmatrix} \ddot{\mathbf{q}}_{n\times 1} = \begin{bmatrix} \mathbf{F}'_{r\times 1} \\ \mathbf{F}''_{(n-r)\times 1} \end{bmatrix} + \begin{bmatrix} \mathbf{a}'^{T}{}_{r\times r} \\ \mathbf{a}''^{T}{}_{(n-r)\times r} \end{bmatrix} \boldsymbol{\lambda}_{r\times 1} \Rightarrow \\
&\begin{cases} \mathbf{M}'_{r\times n}\ddot{\mathbf{q}}_{n\times 1} = \mathbf{F}'_{r\times 1} + \mathbf{a}'^{T}{}_{r\times r}\boldsymbol{\lambda}_{r\times 1} \Rightarrow \boldsymbol{\lambda}_{r\times 1} = \mathbf{a}'^{-T}{}_{r\times r}\left(\mathbf{M}'_{r\times n}\ddot{\mathbf{q}}_{n\times 1} - \mathbf{F}'_{r\times 1}\right) \\ \mathbf{M}''_{(n-r)\times n}\ddot{\mathbf{q}}_{n\times 1} = \mathbf{F}''_{(n-r)\times 1} + \mathbf{a}''^{T}{}_{(n-r)\times r}\boldsymbol{\lambda}_{r\times 1} \end{cases}.
\end{aligned} \tag{10}$$

Lagrange's multiplier could be placed in the second set of equations and motion equations could be solved by adding Eq. (3) to this set:

$$\begin{aligned}
&\left.\begin{array}{l} \left(\mathbf{M}''_{(n-r)\times n} - \mathbf{a}''^{T}{}_{(n-r)\times r}\mathbf{a}'^{-T}{}_{r\times r}\mathbf{M}'_{r\times n}\right)\ddot{\mathbf{q}}_{n\times 1} = \mathbf{F}''_{(n-r)\times 1} - \mathbf{a}''^{T}{}_{(n-r)\times r}\mathbf{a}'^{-T}{}_{r\times r}\mathbf{F}'_{r\times 1} \\ \mathbf{a}_{r\times n}\ddot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{a}}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{b}}_{r\times 1} = 0 \end{array}\right\} \Rightarrow \\
&\underbrace{\begin{bmatrix} \left(\mathbf{M}''_{(n-r)\times n} - \mathbf{a}''^{T}{}_{(n-r)\times r}\mathbf{a}'^{-T}{}_{r\times r}\mathbf{M}'_{r\times n}\right)_{(n-r)\times n} \\ \mathbf{a}_{r\times n} \end{bmatrix}}_{\mathbf{M}_{El(n\times n)}} \ddot{\mathbf{q}}_{n\times 1} = \\
&\underbrace{\left[\left(\mathbf{F}''_{(n-r)\times 1} - \mathbf{a}''^{T}{}_{(n-r)\times r}\mathbf{a}'^{-T}{}_{r\times r}\mathbf{F}'_{r\times 1}\right)_{(n-r)\times 1} - \dot{\mathbf{a}}_{r\times n}\dot{\mathbf{q}}_{n\times 1} - \dot{\mathbf{b}}_{r\times 1}\right]}_{\mathbf{F}_{El(n\times 1)}} \ddot{\mathbf{q}}_{n\times 1} = \mathbf{M}^{-1}_{El(n\times n)}\mathbf{F}_{El(n\times 1)}.
\end{aligned} \tag{11}$$

In this method, based on the dimension of **M**, n equations should be solved.

6. GW Method

In the GW method, the system's states are similar to the previous method. By deriving $\ddot{\mathbf{q}}_{n\times 1}$ from Eq. (5) and placing in Eq. (3), Lagrange's multipliers can be calculated:[28]

$$\begin{aligned}
&\left.\begin{array}{l} \ddot{\mathbf{q}}_{n\times 1} = \mathbf{M}^{-1}{}_{n\times n}\left(\mathbf{F}_{n\times 1} + \mathbf{a}^{T}{}_{n\times r}\boldsymbol{\lambda}_{r\times 1}\right) \\ \mathbf{a}_{r\times n}\ddot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{a}}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{b}}_{r\times 1} = 0 \end{array}\right\} \Rightarrow \\
&\mathbf{a}_{r\times n}\left(\mathbf{M}^{-1}{}_{n\times n}\left(\mathbf{F}_{n\times 1} + \mathbf{a}^{T}{}_{n\times r}\boldsymbol{\lambda}_{r\times 1}\right)\right) + \dot{\mathbf{a}}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{b}}_{r\times 1} = 0, \\
&\boldsymbol{\lambda}_{r\times 1} = -\left(\mathbf{a}_{r\times n}\mathbf{M}^{-1}{}_{n\times n}\mathbf{a}^{T}{}_{n\times r}\right)^{-1}\left(\mathbf{a}_{r\times n}\mathbf{M}^{-1}{}_{n\times n}\mathbf{F}_{n\times 1} + \dot{\mathbf{a}}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{b}}_{r\times 1}\right).
\end{aligned} \tag{12}$$

By placing Lagrange's multipliers in Eq. (5), the motion equations could be solved by:

$$\begin{aligned}
&\underbrace{\mathbf{M}_{n\times n}}_{\mathbf{M}_{GW(n\times n)}}\ddot{\mathbf{q}}_{n\times 1} = \underbrace{\mathbf{F}_{n\times 1} - \mathbf{a}^{T}{}_{n\times r}\left(\mathbf{a}_{r\times n}\mathbf{M}^{-1}{}_{n\times n}\mathbf{a}^{T}{}_{n\times r}\right)^{-1}\left(\mathbf{a}_{r\times n}\mathbf{M}^{-1}{}_{n\times n}\mathbf{F}_{n\times 1} + \dot{\mathbf{a}}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \dot{\mathbf{b}}_{r\times 1}\right)}_{\mathbf{F}_{GW(n\times n)}}, \\
&\ddot{\mathbf{q}}_{n\times 1} = \mathbf{M}^{-1}_{GW(n\times n)}\mathbf{F}_{GW(n\times n)}.
\end{aligned} \tag{13}$$

In this method, based on the dimension of **M**, n equations should be solved.

7. Em Method

In the Em method, $n-r$ independent generalized velocities, as same as number of system DOF, could be found and after solving them, the other generalized velocities could be calculated from them in the constraint equations. In this method, generalized coordinates are divided into two groups and the states of the system and the state time derivatives are introduced as follows:

$$\mathbf{z}_{(2n-r)\times 1} = \begin{bmatrix} \mathbf{q}_{1(r\times 1)} \\ \mathbf{q}_{2((n-r)\times 1)} \\ \dot{\mathbf{q}}_{2((n-r)\times 1)} \end{bmatrix}, \quad \dot{\mathbf{z}}_{(2n-r)\times 1} = \begin{bmatrix} \dot{\mathbf{q}}_{1(r\times 1)} \\ \dot{\mathbf{q}}_{2((n-r)\times 1)} \\ \ddot{\mathbf{q}}_{2((n-r)\times 1)} \end{bmatrix}. \tag{14}$$

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

Equation (2) and $\dot{\mathbf{q}}_{n\times 1}$ could be written as follows:[10]

$$\begin{bmatrix} \mathbf{a}_{1(r\times r)} & \mathbf{a}_{2(r\times(n-r))} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_{1(r\times 1)} \\ \dot{\mathbf{q}}_{2((n-r)\times 1)} \end{bmatrix} + \mathbf{b}_{r\times 1} = 0 \Rightarrow \dot{\mathbf{q}}_{1(r\times 1)} = -\mathbf{a}^{-1}_{1(r\times r)} \left(\mathbf{a}_{2(r\times(n-r))}\dot{\mathbf{q}}_{2((n-r)\times 1)} + \mathbf{b}_{r\times 1}\right),$$
$$\dot{\mathbf{q}}_{n\times 1} = \begin{bmatrix} -\mathbf{a}^{-1}_{1(r\times r)}\mathbf{a}_{2(r\times(n-r))} \\ \mathbf{I}_{(n-r)\times(n-r)} \end{bmatrix} \dot{\mathbf{q}}_{2((n-r)\times 1)} + \begin{bmatrix} -\mathbf{a}^{-1}_{1(r\times r)}\mathbf{b}_{r\times 1} \\ [0]_{(n-r)\times(n-r)} \end{bmatrix} = \mathbf{w}_{n\times(n-r)}\dot{\mathbf{q}}_{2((n-r)\times 1)} + \mathbf{x}_{n\times 1}. \tag{15}$$

Therefore, $\ddot{\mathbf{q}}_{n\times 1}$ could be written as:

$$\ddot{\mathbf{q}}_{n\times 1} = \mathbf{w}_{n\times(n-r)}\ddot{\mathbf{q}}_{2((n-r)\times 1)} + \dot{\mathbf{w}}_{n\times(n-r)}\dot{\mathbf{q}}_{2((n-r)\times 1)} + \dot{\mathbf{x}}_{n\times 1}, \tag{16}$$

By placing $\ddot{\mathbf{q}}_{n\times 1}$ in Eq. (5):

$$\mathbf{M}_{n\times n}\left(\mathbf{w}_{n\times(n-r)}\ddot{\mathbf{q}}_{2((n-r)\times 1)} + \dot{\mathbf{w}}_{n\times(n-r)}\dot{\mathbf{q}}_{2((n-r)\times 1)} + \dot{\mathbf{x}}_{n\times 1}\right) = \mathbf{F}_{n\times 1} + \mathbf{a}^{T}{}_{n\times r}\boldsymbol{\lambda}_{r\times 1}. \tag{17}$$

Moreover, by using the property of $[\mathbf{w}]^T_{(n-r)\times n}[\mathbf{a}]^T_{n\times r} = [0]_{(n-r)\times r}$ (proved in Appendix A), and multiplying Eq. (17) by $\mathbf{w}^T$, the governing motion equations could be solved as follows:

$$\begin{aligned} &\underbrace{\left(\mathbf{w}^T_{(n-r)\times n}\mathbf{M}_{n\times n}\mathbf{w}_{n\times(n-r)}\right)}_{\mathbf{M}_{Em((n-r)\times(n-r))}}\ddot{\mathbf{q}}_{2((n-r)\times 1)} \\ &= \underbrace{\mathbf{w}^T_{(n-r)\times n}\left(\mathbf{F}_{n\times 1} - \mathbf{M}_{n\times n}\dot{\mathbf{w}}_{n\times(n-r)}\dot{\mathbf{q}}_{2((n-r)\times 1)} - \mathbf{M}_{n\times n}\dot{\mathbf{x}}_{n\times 1}\right)}_{\mathbf{F}_{Em((n-r)\times 1)}} + \underbrace{\mathbf{w}^T_{(n-r)\times n}\mathbf{a}^{T}{}_{n\times r}}_{[0]_{(n-r)\times r}}\boldsymbol{\lambda}_{r\times 1} \\ &\ddot{\mathbf{q}}_{2((n-r)\times 1)} = \mathbf{M}^{-1}_{Em((n-r)\times(n-r))}\mathbf{F}_{Em((n-r)\times 1)}. \end{aligned} \tag{18}$$

In this method, based on the dimension of $\mathbf{M}$, $n-r$ equations should be solved.

8. ML Method

In the ML method, generalized speeds ($\mathbf{u}_{(n-r)\times 1}$) are defined as the number of systems' DOF. The states of the system and the state time derivatives are considered as follows:

$$\mathbf{z}_{(2n-r)\times 1} = \begin{bmatrix} \mathbf{q}_{n\times 1} \\ \mathbf{u}_{(n-r)\times 1} \end{bmatrix}, \quad \dot{\mathbf{z}}_{(2n-r)\times 1} = \begin{bmatrix} \dot{\mathbf{q}}_{n\times 1} \\ \dot{\mathbf{u}}_{(n-r)\times 1} \end{bmatrix}. \tag{19}$$

Generalized velocities with respect to generalized speeds and its time derivation are shown in the following equation:[1]

$$\begin{aligned} \dot{\mathbf{q}}_{n\times 1} &= \mathbf{w}'_{n\times(n-r)}\mathbf{u}_{(n-r)\times 1} + \mathbf{x}'_{n\times 1}, \\ \ddot{\mathbf{q}}_{n\times 1} &= \dot{\mathbf{w}}'_{n\times(n-r)}\mathbf{u}_{(n-r)\times 1} + \mathbf{w}'_{n\times(n-r)}\dot{\mathbf{u}}_{(n-r)\times 1} + \dot{\mathbf{x}}'_{n\times 1}, \end{aligned} \tag{20}$$

By placing $\ddot{\mathbf{q}}_{n\times 1}$ in Eq. (5), using the property of $[\mathbf{w}']^T_{(n-r)\times n}[\mathbf{a}]^T_{n\times r} = [0]_{(n-r)\times r}$ (Appendix B) and multiplying it by $\mathbf{w}'^T$, the governing motion equations are:

$$\begin{aligned} &\underbrace{\left(\mathbf{w}'^T_{(n-r)\times n}\mathbf{M}_{n\times n}\mathbf{w}'_{n\times(n-r)}\right)}_{\mathbf{M}_{ML((n-r)\times(n-r))}}\dot{\mathbf{u}}_{((n-r)\times 1)} \\ &= \underbrace{\mathbf{w}'^T_{(n-r)\times n}\left(\mathbf{F}_{n\times 1} - \mathbf{M}_{n\times n}\dot{\mathbf{w}}'_{n\times(n-r)}\mathbf{u}_{((n-r)\times 1)} - \mathbf{M}_{n\times n}\dot{\mathbf{x}}'_{n\times 1}\right)}_{\mathbf{F}_{ML((n-r)\times 1)}} + \underbrace{\mathbf{w}'^T_{(n-r)\times n}\mathbf{a}^{T}{}_{n\times r}}_{[0]_{(n-r)\times r}}\boldsymbol{\lambda}_{r\times 1} \\ &\dot{\mathbf{u}}_{((n-r)\times 1)} = \mathbf{M}^{-1}_{ML((n-r)\times(n-r))}\mathbf{F}_{ML((n-r)\times 1)}, \end{aligned} \tag{21}$$

Similar to Em method based on the dimension of $\mathbf{M}$, $n-r$ equations should be solved in this method.

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

Table I. Some characteristics of the seven methods.

	Lagrange's method						Kane's method
	IM	AU	EL	GW	EM	ML	
Number of equations	$2n+r$	$n+r$	n	n	$n-r$	$n-r$	$n-r$
Constraint equations used at	Acceleration level				Velocity level		
Lagrange's multipliers derived directly	✓	✓	✗	✗	✗	✗	✗

9. A Variant of Kane's Method

A new approach based on Kane's method proposed in ref. [17] is applied in this work. The states of the system are identical to the ML method, so Eq. (20) is also valid in this method. The matrix form of governing motion equations could be written as follows:[17,24,25]

$$\mathbf{M}_{\mathrm{Kane}(n-r)\times(n-r)}\dot{\mathbf{u}}_{(n-r)\times 1}+\mathbf{N}_{\mathrm{Kane}(n-r)\times n}\dot{\mathbf{q}}_{n\times 1}+\mathbf{G}_{\mathrm{Kane}(n-r)\times 1}=\mathbf{F}_{\mathrm{Kane}(n-r)\times 1}, \tag{22}$$

In which:

$$\mathbf{M}_{\mathrm{Kane}((n-r)\times(n-r))}(i,j)=\sum_{k=1}^{N_b} m_k\left(\frac{\partial \mathbf{V}_k{}^T}{\partial u_i}\frac{\partial \mathbf{V}_k}{\partial u_j}\right)+\frac{\partial \boldsymbol{\omega}_k{}^T}{\partial u_i}\mathbf{I}_k\frac{\partial \boldsymbol{\omega}_k}{\partial u_j},\quad 1\leqslant i,j\leqslant(n-r),$$

$$\mathbf{N}_{\mathrm{Kane}((n-r)\times n)}(i,j)=\sum_{k=1}^{N_b}\left[m_k\left(\frac{\partial \mathbf{V}_k{}^T}{\partial q_j}\frac{\partial \mathbf{V}_k}{\partial u_i}\right)+\frac{\partial \boldsymbol{\omega}_k{}^T}{\partial q_j}\mathbf{I}_k\frac{\partial \boldsymbol{\omega}_k}{\partial u_i}\right],\quad 1\leqslant i\leqslant(n-r)\ ,\quad 1\leqslant j\leqslant n,$$

$$\mathbf{G}_{\mathrm{Kane}((n-r)\times 1)}(i)=\sum_{k=1}^{N_b}\left[m_k\frac{\partial \mathbf{V}_k{}^T}{\partial t}\frac{\partial \mathbf{V}_k}{\partial u_i}+\frac{\partial \boldsymbol{\omega}_k{}^T}{\partial t}\mathbf{I}_k\frac{\partial \boldsymbol{\omega}_k}{\partial u_i}+\boldsymbol{\omega}_k{}^T\mathbf{I}_k\left[\hat{\boldsymbol{\omega}}_k\right]^T\frac{\partial \boldsymbol{\omega}_k}{\partial u_i}\right]^T, \tag{23}$$

$$\left[\hat{\boldsymbol{\omega}}_k\right]=\begin{bmatrix}0 & -\omega_{k_z} & \omega_{k_y}\\ \omega_{k_z} & 0 & -\omega_{k_x}\\ -\omega_{k_y} & \omega_{k_x} & 0\end{bmatrix},$$

$$\mathbf{F}_{\mathrm{Kane}(n-r)\times 1}=\mathbf{w}'^{T}_{(n-r)\times n}\mathbf{Q}_{n\times 1},$$

where N_b is the number of the body in the system, $\mathbf{V}_k$ and ω_k are linear and angular velocities of k-th body in the system, and m_k and $\mathbf{I}_k$ are mass and moment of inertia of the body. For a detailed derivation of Eq. (23), take a look at Appendix C. The reference frame used in ref. [17] is the inertial reference frame, and all of the partial differentiations are conducted based on the inertial reference frame.

If the generalized speeds introduced in the variant of Kane's method and ML method are identical to each other, the motion equations derived by these methods are entirely equivalent.

In Table I, these seven methods are compared with different computation aspects.

10. Snake-like Robot

The dynamical model of N-link snake-like robot, which is studied in this research, is shown in Fig. 1. The purpose is to compare Lagrange's and Kane's methods in different computational aspects. The CoM of each link is assumed to be in the middle of the link. The angle between the axis of the i-th link and the plane of the i-th wheel is $\alpha_i(1\leqslant i\leqslant n)$ which is constant, and $\theta_i(1\leqslant i\leqslant n)$ is the orientation of i-th link with respect to the x-axis. In Fig. 1, x and y are positions of the first link's tip along x and y directions, respectively. The bold-short lines in the Fig. 1 are simplified symbolic representation of the wheels and their corresponding plane. e_i is a vector along the i-th wheel axle. There is no side slip at the wheels; therefore, the CoM's velocity of each link is perpendicular to the wheel's axis. In other words, each link has a nonholonomic constraint. As a result, N-link snake-like robot has two DOF. m_i, L_i, and I_i are the mass, length, and the moment of inertia of i-th link, respectively. A torsion spring placed at each joint between i-th link and $i+1$-th link is denoted by $k_i(1\leqslant i\leqslant n-1)$.

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

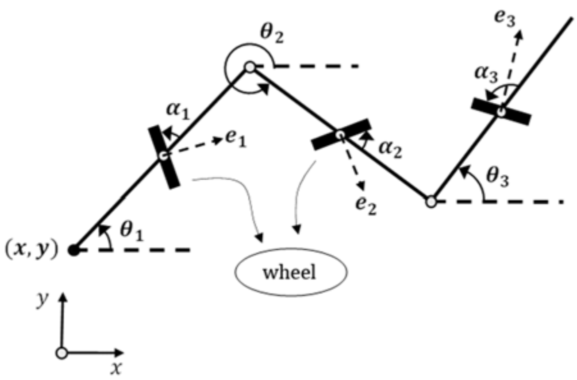

Fig. 1. Schematic of a snake-like robot.

Generalized coordinates and velocities in the snake-like model can be chosen as follows:

$$\begin{aligned}\mathbf{q} &= \left[q_1\ q_2\ q_3\ \cdots\ q_{i+2}\ \cdots\ q_{N+2}\right]^T = \left[x\ y\ \theta_1\ \cdots\ \theta_i\ \cdots\ \theta_N\right]^T,\\ \dot{\mathbf{q}} &= \left[\dot{q}_1\ \dot{q}_2\ \dot{q}_3\ \cdots\ \dot{q}_{i+2}\ \cdots\ \dot{q}_{N+2}\right]^T = \left[\dot{x}\ \dot{y}\ \dot{\theta}_1\ \cdots\ \dot{\theta}_i\ \cdots\ \dot{\theta}_N\right]^T.\end{aligned} \tag{24}$$

The kinetic energy of the snake-like robot is

$$\begin{aligned}&T = \sum_{i=1}^{N}\left(\tfrac{1}{2}m_i\mathbf{v}_i.\mathbf{v}_i + \tfrac{1}{2}\boldsymbol{\omega}_i\mathbf{I_i}\boldsymbol{\omega}_i\right),\\ &\boldsymbol{\omega}_i = \left[0\ 0\ \dot{\theta}_i\right]^T,\\ &\mathbf{v_1} = \left[\dot{x}\ \dot{y}\ 0\right]^T + \boldsymbol{\omega}_1 \times \tfrac{L_1}{2}\left[\cos\theta_1\ \sin\theta_1\ 0\right]^T,\\ &\mathbf{v_i} = \mathbf{v_{i-1}} + \boldsymbol{\omega}_{i-1} \times \tfrac{L_{i-1}}{2}\left[\cos\theta_{i-1}\ \sin\theta_{i-1}\ 0\right]^T + \boldsymbol{\omega}_i \times \tfrac{L_i}{2}\left[\cos\theta_i\ \sin\theta_i\ 0\right]^T, \qquad 2 < i < N,\end{aligned} \tag{25}$$

where $\mathbf{v_i}$ is the velocity and $\omega_\mathbf{i}$ is the angular velocity of i-th link.

The potential energy stored in the torsion springs could be considered as the potential energy of the robot:

$$U = \frac{1}{2}\sum_{i=1}^{N-1} k_i(\theta_{i+1} - \theta_i)^2. \tag{26}$$

Free position of each torsion spring is where its two neighboring links are in line.

The only external force applied to the system is gravitational forces acting on the links. As there is not any displacement in the vertical direction (the movement is 2D in the horizontal plane), gravitational forces will not contribute to the generalized active force. The generalized forces of Lagrange's method can be obtained from potential energy as follows:

$$\mathbf{Q} = -\frac{\partial U}{\partial\,\{\mathbf{q}\}}. \tag{27}$$

Since each link's velocity is always perpendicular to its link's wheel axle, nonholonomic constraint equations are calculated as follows:

$$\begin{aligned}&c_i = \mathbf{v_i}.\,\mathbf{e_i} = 0\,, \quad \mathbf{e_i} = \left[\cos\left(\pi/2 - \alpha_i - \theta_i\right)\ -\sin\left(\pi/2 - \alpha_i - \theta_i\right)\ 0\right]^T \quad 1 \leqslant i \leqslant N\\ &\mathbf{c} = \left[c_1\ \ldots\ c_i\ \cdots\ c_n\right] = \mathbf{a}_{N\times(N+2)}\dot{\mathbf{q}}_{(N+2)\times 1} = [0]_{(N+2)\times 1},\end{aligned} \tag{28}$$

where e_i is a vector along the i-th wheel axle since c_i is time independent, b_i is equal to zero.

In the ML and the variant of Kane's method, generalized speeds defined as follows, in which the values in the vector in Eq. (29) are scalar.

$$\mathbf{u}_{2\times 1} = \begin{bmatrix} v_1 \\ \dot{\theta}_1 \end{bmatrix}. \tag{29}$$

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

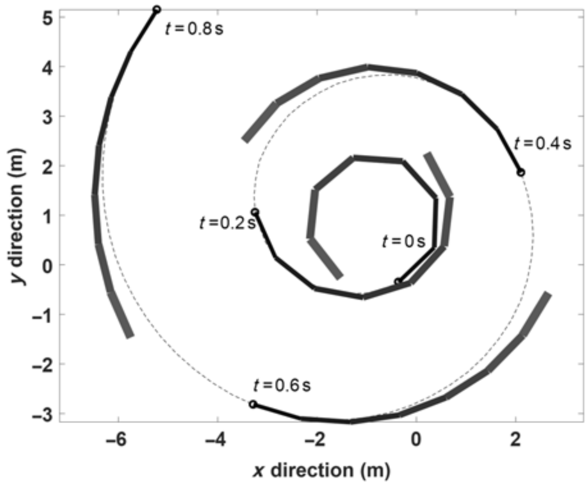

Fig. 2. The position of seven-link snake-like robot at five different instances.

Here, v_1 is magnitude of CoM of the first link velocity which is parallel to the plane of the wheel, $v_1 = \mathbf{v}_1.[\cos(\alpha_1 + \theta_1)\ \sin(\alpha_1 + \theta_1)\ 0]^T$. For a detailed formulation of the generalized force of the variant of Kane's method, please see Appendix D.

11. Simulation Results

In this chapter, we compare the results of different methods that were introduced in the previous section in these aspects:

- CPU time.
- Energy error
- Constraints error.

As a case study, seven-link snake-like robots are studied. Figure 2 depicts the snapshots of a seven-link snake-like robot motion. The system parameters and initial conditions are set as follows:

$$M = [1;\ 1;\ 1;\ 1;\ 1;\ 1;\ 1]\ \text{kg}$$

$$L = [1;\ 1;\ 1;\ 1;\ 1;\ 1;\ 1]\ \text{m}$$

$$I = 0.0833 \times [1;\ 1;\ 1;\ 1;\ 1;\ 1;\ 1]\ \text{kg.m}^2$$

$$K = 5000 \times [1;\ 1;\ 1;\ 1;\ 1;\ 1;\ 1]\ \text{kg.m}$$

$$\alpha_i(t = 0) = [0.0175;\ 0.0349;\ 0.0524;\ 0.0698;\ 0.0873;\ 0.1047;\ 0.1222]\ \ \text{rad}$$

$$\theta_i(t = 0) = \ [0.7679;\ 1.536;\ 2.304;\ 3.072;\ 3.840;\ 4.608;\ 5.376]\ \ \text{rad}$$

$$\dot{\theta}_i(t = 0)(i = 1, ..., 7) = 0$$

$$x(t = 0) = 0,\ y(t = 0) = 0\ \text{m}$$

$$\dot{x}(t = 0) = 0,\ \dot{y}(t = 0) = 0\ \text{m/s}$$

The position of the robot is plotted in five different instances in Fig. 2. The instances are t = 0, 0.2, 0.4, 0.6, and 0.8 s. Robot links are plotted with solid lines. For better illustration, the line thickness for different links differs. The head of the robot is represented by O sign, which is the same x, y position in Fig. 1. The position of the robots head (O sign) is plotted by the dashed line in all times for better understanding of robot movement. The result depicted in this figure is obtained using Kane's method.

In numerical solution, constraints errors defined in (30) are illustrated in Fig. 3.

$$\text{constraints\_error} = \|c\|_2 = \left(\sum_{i=1}^{n} {c_i}^2\right)^{1/2}. \tag{30}$$

As expected, Fig. 3 shows that Kane's, Em, and ML methods that apply constraints at velocity level have fewer constraints error than other methods that apply constraints at acceleration level.

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

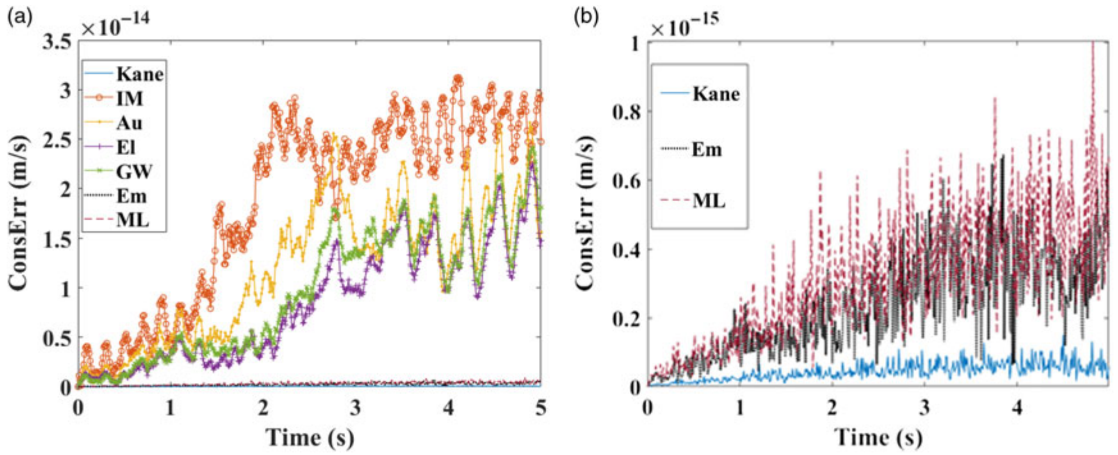

Fig. 3. Constraints error in seven-link snake-like robot in different methods. In the figure (b), for better comparison only the constraint errors of three methods ML, Em, and Kane are depicted.

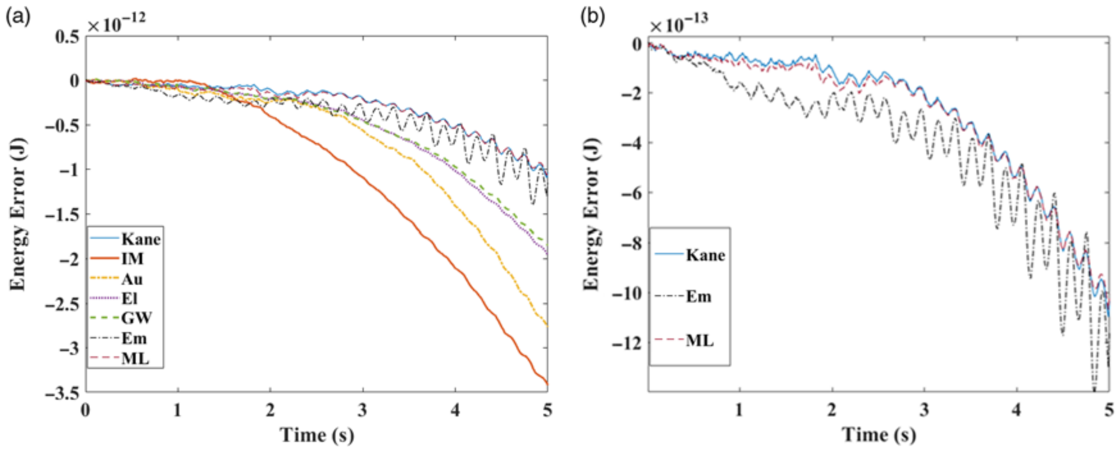

Fig. 4. (a) Energy error in seven-link snake-like robot in different methods, (b) the enlargement of the energy error of three ML, Em, and the variant of Kane's methods.

Also, as it is illustrated in Fig. 3(b), constraint error of the variant of Kane's method is much less than the two others.

In the numerical solution, energy errors introduced in (31) are illustrated in Fig. 4.

$$\text{Energy\_error} = T + U - E_0, \tag{31}$$

In which E_0 is initial mechanical energy of the robot.

Fig. 4 shows that the variant of Kane's, Em, and ML methods have $n - r$ motion equations, as mentioned in Table I, which are less than those of other methods, have less energy error in numerical solutions.

Moreover, another simulation is presented for 2- to 100-link snake-like robots. The Euclidean norm of constraints error and energy error in simulation for these robots is illustrated in Figs. 5 and 6, respectively. If we consider error at different times as a vector, we have

$$\text{Error} = \left[e(0)\ e(dt)\ e(2dt)\ \cdots\ e(t_{\text{end}})\right], \tag{32}$$

Then error norm would be like Eq. (33).

$$\text{Error\_norm} = \sqrt{\sum_{i=1} e(t(i))^2}. \tag{33}$$

The results confirm high precision of Kane's, Em, and ML methods in computational aspects.

Figure 7 illustrates the CPU time consumption of each method in the simulation of 2- to 100-link snake-like robot.

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

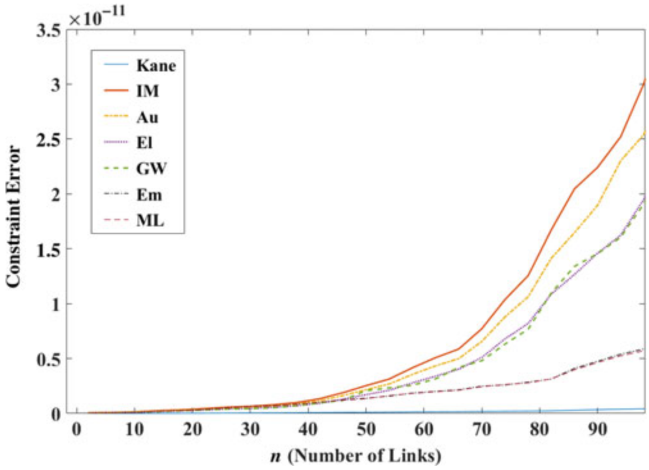

Fig. 5. The Euclidean norm of constraints error in simulation for 2-link to 100-link snake-like robots.

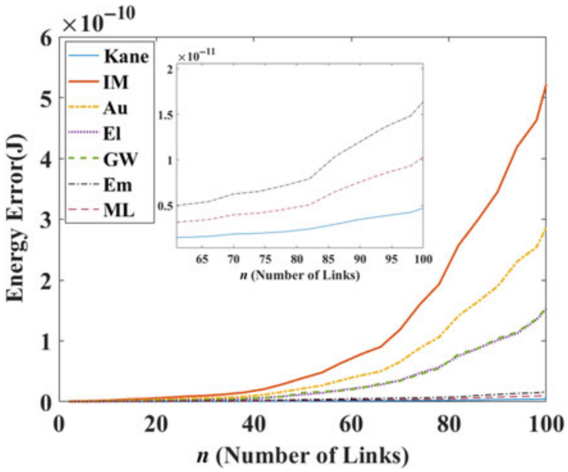

Fig. 6. The Euclidean norm of energy error in simulations for 2-link to 100-link snake-like robots.

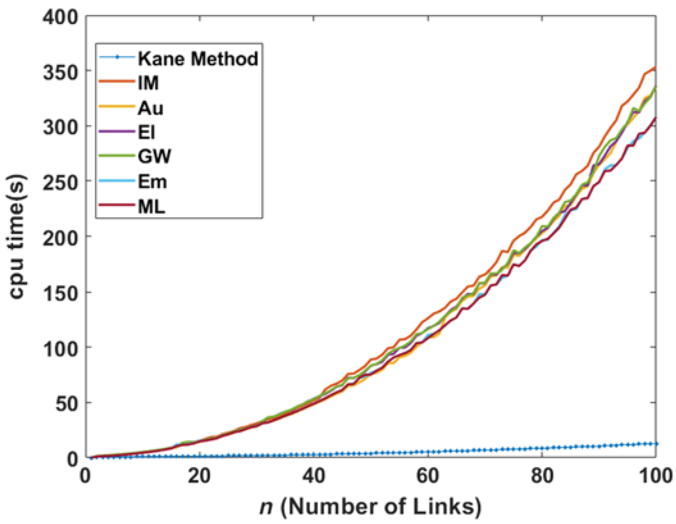

Fig. 7. The comparison between seven methods' CPU time consumption in the simulation of 2-link to 100-link snake-like robot.

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

As shown in Fig. 7, CPU time consumption in the variant of Kane's method is significantly less than the other methods, because the variant of Kane's method uses a recursive method, which some differential phrases can be derived from the others and it does not need to calculate them again. It means that the variant of Kane's method is more appropriate for robots with a high number of links.

12. Conclusion

In this study, a comparative study on the numerical performance of a variant of Kane's method and six different numerical techniques based on the Lagrange's equations was presented. The resulting constraint and energy errors and the CPU time needed for the numerical simulation were considered as the criteria for this comparison. The conducted simulations on a snake-like robot illustrate that the CPU time in Lagrange-based methods is about 10 times higher than the variant of Kane's method. Since the constraints were applied in the velocity form in the variant of Kane's, Em, and ML methods, the resulting constraints error is less than that of the other methods. Moreover, the energy error of these three methods is less than that of the other methods. These results indicate that the variant of Kane's method is more appropriate with respect to the other methods for this application, especially in the sense of required CPU time for computations.

References

1. H. Baruh, *Analytical Dynamics* (WCB/McGraw-Hill, Boston, 1999).
2. G. R. Fowles and G. L. Cassiday, *Analytical Mechanics* (Thomson Brooks/Cole, Belmont, CA, 2005).
3. T. R. Kane, "Dynamics of nonholonomic systems," *J. Appl. Mech.* **28**(4), 574–578 (1961).
4. T. R. Kane and D. A. Levinson, *Dynamics: Theory and Applications* (McGraw Hill, New York, 1985).
5. C. M. Roithmayr and D. H. Hodges, *Dynamics: Theory and Application of Kane's Method* (Cambridge University Press, Cambridge, UK, 2016).
6. J. Willard Gibbs, "On the fundamental formulae of dynamics," *American J. Math.* **2**(1), 49–64 (1879).
7. P. Appell, "Sur les mouvements de roulment; equations du mouvement analougues a celles de lagrange," *Comptes Rendus* **129**, 317–320 (1899).
8. J. G. Papastavridis, "Maggi's equations of motion and the determination of constraint reactions," *J. Guidance Control Dyn.* **13**(2), 213–220 (1990).
9. F. E. Udwadia and R. E. Kalaba, "A new perspective on constrained motion," *Proc. R. Soc. London Ser. A Math. Phys. Sci.* **439**(1906), 407–410 (1992).
10. J. Ginsberg, *Engineering Dynamics*, vol. 10 (Cambridge University Press, Cambridge, UK, 2008).
11. N. V. Khang, "Kronecker product and a new matrix form of Lagrangian equations with multipliers for constrained multibody systems," *Mech. Res. Commun.* **38**(4), 294–299 (2011).
12. E. Jarzkebowska, K. Augustynek and A. Urbaś, "Computational Reference Dynamical Model of a Multibody System with First Order Constraints," *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2017) pp. V006T10A016–V006T10A016.
13. F. Marques, A. P. Souto and P. Flores, "On the constraints violation in forward dynamics of multibody systems," *Multi. Syst. Dyn.* **39**(4), 385–419 (2017).
14. W. Blajer, "A geometric unification of constrained system dynamics," *Multi. Syst. Dyn.* **1**(1), 3–21 (1997).
15. W. Blajer, "Elimination of constraint violation and accuracy aspects in numerical simulation of multibody systems," *Multi. Syst. Dyn.* **7**(3), 265–284 (2002).
16. D. J. Braun and M. Goldfarb, "Eliminating constraint drift in the numerical simulation of constrained dynamical systems," *Comput. Methods Appl. Mech. Eng.* **198**(37–40), 3151–3160 (2009).
17. H. Nejat Pishkenari, S. A. Yousefsani, A. L. Gaskarimahalle and S. B. G. Oskouei, "A fresh insight into Kane's equations of motion," *Robotica* **35**(3), 498–510 (2017).
18. A. H. Bajodah, D. H. Hodges and Y.-H. Chen, "Nonminimal generalized Kane's impulse-momentum relations," *J. Guidance Control Dyn.* **27**(6), 1088–1092 (2004).
19. D. L. Peterson, G. Gede and M. Hubbard, "Symbolic linearization of equations of motion of constrained multibody systems," *Multi. Syst. Dyn.* **33**(2), 143–161 (2015).
20. A. H. Bajodah, D. H. Hodges and Y.-H. Chen, "New form of Kane's equations of motion for constrained systems," *J. Guidance Control Dyn.* **26**(1), 79–88 (2003).
21. C. M. Roithmayr, A. H. Bajodah, D. H. Hodges and Y.-H. Chen, "Corrigendum: New form of Kane's equations of motion for constrained systems," *J. Guidance Control Dyn* **30**(1), 286–288 (2007).
22. A. H. Bajodah, D. H. Hodges and Y.-H. Chen, "Nonminimal Kane's equations of motion for multibody dynamical systems subject to nonlinear nonholonomic constraints," *Multi. Syst. Dyn.* **14**(2), 155–187 (2005).
23. C. M. Roithmayr and D. H. Hodges, "Forces associated with non-linear non-holonomic constraint equations," *Int. J. Non-Linear Mech.* **45**(4), 357–369 (2010).
24. H. Tavakoli Nia, H. Nejat Pishkenari and A. Meghdari, "A recursive approach for the analysis of snake robots using Kane's equations," *Robotica* **24**(2), 251–256 (2006).

25. W. Raman-Nair and M. White, "A model for deployment of a freefall lifeboat from a moving ramp into waves," *Multi. Syst. Dyn.* **29**(3), 327–342 (2013).
26. A. A. Transeth, K. Y. Pettersen and P. Liljebäck, "A survey on snake robot modeling and locomotion," *Robotica* **27**(7), 999–1015 (2009).
27. G. Vossoughi, H. Pendar, Z. Heidari and S. Mohammadi, "Assisted passive snake-like robots: Conception and dynamic modeling using Gibbs–Appell method," *Robotica* **26**(3), 267–276 (2008).
28. D. T. Greenwood, *Advanced Dynamics* (Cambridge University Press, Cambridge, UK, 2006).

A. Appendix

$$\begin{aligned}
&[\mathbf{w}]^T_{(n-r)\times n}[\mathbf{a}]^T_{n\times r} = \left([\mathbf{a}]_{n\times r}[\mathbf{w}]_{(n-r)\times n}\right)^T \\
&[\mathbf{a}]_{n\times r}\,[\mathbf{w}]_{(n-r)\times n} = \left[[\mathbf{a}_1]_{r\times r}\ [\mathbf{a}_2]_{r\times(n-r)}\right]\begin{bmatrix}-[\mathbf{a}_1]^{-1}{}_{r\times r}[\mathbf{a}_2]_{r\times(n-r)}\\ [I]_{(n-r)\times(n-r)}\end{bmatrix} = \\
&-[\mathbf{a}_1]_{r\times r}[\mathbf{a}_1]^{-1}{}_{r\times r}[\mathbf{a}_2]_{r\times(n-r)} + [\mathbf{a}_2]_{r\times(n-r)} = [0]_{r\times(n-r)} \Rightarrow [\mathbf{w}]^T_{(n-r)\times n}[\mathbf{a}]^T_{n\times r} = [0]_{(n-r)\times r}
\end{aligned} \quad . \tag{A1}$$

B. Appendix

Assume that there are $n-r$ independent generalized speeds $\mathbf{u}_{(n-r)\times 1}$. We can write the generalized velocities versus these generalized speeds as follows:

$$\dot{\mathbf{q}}_{n\times 1} = \mathbf{w}'_{n\times(n-r)}\mathbf{u}_{(n-r)\times 1} + \mathbf{x}'_{n\times 1}. \tag{B1}$$

Using constraint equation, we have

$$\begin{aligned}
&\mathbf{a}_{r\times n}\dot{\mathbf{q}}_{n\times 1} + \mathbf{b}_{r\times 1} = [0]_{r\times 1} \ \Rightarrow \mathbf{a}_{r\times n}(\mathbf{w}'_{n\times(n-r)}\mathbf{u}_{(n-r)\times 1} + \mathbf{x}'_{n\times 1}) + \mathbf{b}_{r\times 1} = [0]_{r\times 1} \\
&\Rightarrow \mathbf{a}_{r\times n}\mathbf{w}'_{n\times(n-r)}\mathbf{u}_{(n-r)\times 1} + \mathbf{a}_{r\times n}\mathbf{x}'_{n\times 1} + \mathbf{b}_{r\times 1} = [0]_{r\times 1}
\end{aligned} \quad . \tag{B2}$$

Now since u_i s are independent, from the last expression we conclude that:

$$\mathbf{a}_{r\times n}\mathbf{w}'_{n\times(n-r)} = [0]_{r\times(n-r)} \quad , \quad \mathbf{a}_{r\times n}\mathbf{x}'_{n\times 1} + \mathbf{b}_{r\times 1} = [0]_{r\times 1}. \tag{B3}$$

Hence, we have $(\mathbf{a}\mathbf{w}')^T = \mathbf{w}'^T\mathbf{a}^T = [0]_{(n-r)\times r}$.

C. Appendix

$$\boldsymbol{F}_k = \sum \boldsymbol{f}_i \cdot \frac{\partial v_i}{\partial u_k} = \sum \boldsymbol{f}_i \cdot \frac{\partial \boldsymbol{r}_i}{\partial q'_k}, \tag{C1}$$

where q'_k indicates the generalized quasi-coordinate associated with u_k. On the other hand, we have

$$\dot{\mathbf{q}}_{n\times 1} = \mathbf{w}'_{n\times(n-r)}\mathbf{u}_{(n-r)\times 1} + \mathbf{x}'_{n\times 1} \quad \Rightarrow \delta\mathbf{q} = \mathbf{w}'\delta\mathbf{q}', \tag{C2}$$

where δ stands for variation. On the other hand, we have

$$\frac{\partial \boldsymbol{r}_i}{\partial q'_k} = \sum_{l=1}^{n} \frac{\partial \boldsymbol{r}_i}{\partial q_l}\frac{\partial q_l}{\partial q'_k} = \sum_{l=1}^{n} \frac{\partial \boldsymbol{r}_i}{\partial q_l} w'_{lk}, \tag{C3}$$

$$\begin{aligned}
\boldsymbol{F} &= \sum \boldsymbol{f}_i \cdot \tfrac{\partial v_i}{\partial \{\boldsymbol{u}\}} = \sum \boldsymbol{f}_i \cdot \tfrac{\partial \boldsymbol{r}_i}{\partial \{\boldsymbol{q}'\}} = \sum \left(\tfrac{\partial \boldsymbol{r}_i}{\partial \{\boldsymbol{q}'\}}\right)^T \boldsymbol{f}_i = \sum \left(\tfrac{\partial \boldsymbol{r}_i}{\partial \{\boldsymbol{q}\}}\mathbf{w}'\right)^T \boldsymbol{f}_i \\
&= \sum w'^T \left(\tfrac{\partial r_i}{\partial \{q\}}\right)^T f_i = w'^T \sum f_i \cdot \tfrac{\partial r_i}{\partial \{q\}} = w'^T Q.
\end{aligned} \quad . \tag{C4}$$

Here, **Q** is the generalized force vector in Lagrange's method, but **F** is the generalized force vector in Kane's method. For more details, please see ref. [1].

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press

D. Appendix

$$F(i) = \sum_{k=1}^{n-1} T_k \left(\frac{\partial \omega_{k+1}}{\partial u_i} - \frac{\partial \omega_k}{\partial u_i} \right) \ 1 \leqslant i \leqslant 2 \,, \tag{D1}$$

$$T_i = k_i(q_i - q_{i+1}), \tag{D2}$$

T_i is the torque of torsional springs.

https://doi.org/10.1017/S0263574719001899 Published online by Cambridge University Press