Dynamics Computation of Link Mechanisms Employing COG Jacobian

Takashi Sonoda and Kazuo Ishii

Department of Brain Science and Engineering
Kyushu Institute of Technology
2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan
sonoda-takashi@edu.brain.kyutech.ac.jp
ishii@brain.kyutech.ac.jp

Daigoro Isobe

Department of Engineering Mechanics and Energy
University of Tsukuba
1-1-1 Tennodai, Tsukuba-shi, Ibaraki 305-8573, Japan
isobe@kz.tsukuba.ac.jp

***Abstract* - We show the dynamics computation employing Jacobian that relates the center of gravity (COG) to joints of link mechanisms, in this paper. COG Jacobian is used for the behavior planning and the control of humanoids. And, it usually expresses the relationship between the joints and COG of a robot's whole body. However, in this scheme, it's calculated regarding each link and not the robot's whole-body. Moreover, we can obtain the torques by relationship between COG Jacobian and the applied forces to COG by using principle of virtual work. The loaded forces to COG can be obtained by employing Newton's and Euler's equations of motion. By the scheme, we can calculate the inverse dynamics regardless of open- and closed-link mechanisms. In addition, the forward dynamics can be calculated by employing COG Jacobian.**

Index Terms -Dynamics, Link Mechanism, COG Jacobian.

I. INTRODUCTION

Machines and robots are modeled into link mechanisms of rigid-body when we compute their dynamics analyses. The link mechanisms simplify structures of components and are employed as the kinematic model. Various methods based on Newton-Euler equation of motion are developed as dynamics of link mechanisms. In serial mechanisms, Newton-Euler formulation that is efficient and is able to be solved at computation cost proportioned to DOF of a mechanism is proposed. Also, the laws of an inertial force occurring at the center of gravity (COG) of the link that is moving and balancing with external forces are used in this method. Moreover, issues of dynamics are application to not only open- and closed-link mechanisms, but mechanisms containing redundant actuators, collisions and contacts. Therefore unified dynamics computation schemes treating their issues are hoped. A method of numerical analysis solving simultaneous equations of constraints corresponding to each joint is used a lot of present versatile dynamics simulators [1]. However, this method needs very heavy computation cost and long computation time at the analyses of the complex structural mechanisms. Therefore, at closed-link mechanisms, as an efficient method such as Newton-Euler formulation are also required because their mechanisms are usually complicated. The solution of equation of motion employing Lagrange multipliers has been known as inverse dynamics for closed-link mechanisms, but it is not efficient. After considering closed-link mechanisms to be temporal virtual open-link mechanisms of tree structure, there is a method of solving inverse dynamics using Jacobian matrix changing from generalized forces of open-link mechanisms to the ones of closed-link mechanisms [2][3]. However, though this method needs to cut some arbitrary joints, to choose the cutting joints and the generalized coordinates is problems in complex mechanisms. Nakamura and Yamane show the method that is employed as 6 DOF virtual links to all joints of each link as developed method than the above method and it is applied to mechanisms with discontinuous changes of constraints [4]. According to this method, these above problems are solved, but it also needs to compute as virtual open-link mechanisms. The computation of the change processes is not necessary at the unified scheme. Isobe suggests a method applying Finite Element Method to inverse dynamics [5][6]. It is able to represent equations of inverse dynamics as equations of matrices by coordinate transformation on dimensions of forces. Therefore, it can be applied to mechanisms with changes of constraints. The links are divided to some elements and some nodal points, and forces are applied to the nodal points without distinguishing static forces and kinetic forces. As others, there is the method that uses the motor algebra, and it enables to derive the equations of motion of open- and closed-link mechanisms [7]. However, we have to solve simultaneous equations of forces regarding passive joints at closed-link mechanisms.

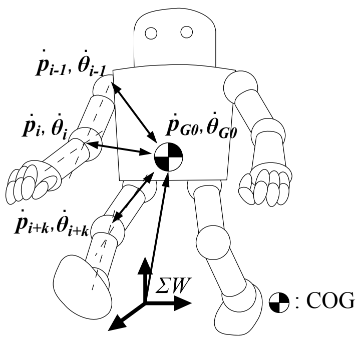


Fig. 1 Humanoid model. COG Jacobian is usually calculated to COG of the whole-body of the humanoid. And, it shows the relationship between the joints and COG of the humanoid.

In this paper, we show a computational method that is employing Jacobian connecting COG of links with active joints. This Jacobian usually connects COG of the whole-body with active joints, and it is used for motion planning and

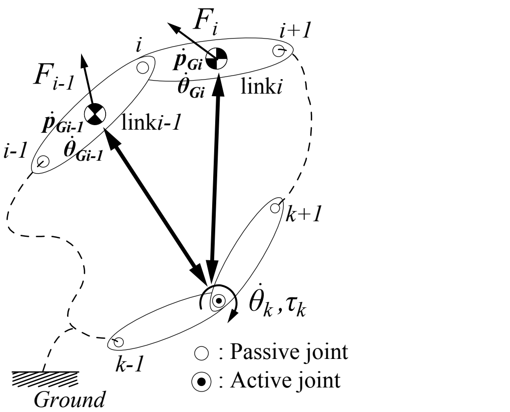


Fig. 2 Multi-rigid-body model. And, the forces and the moments regarding the inertia by the motion are applied to all rigid-bodies.

controls of humanoids [8]-[11]. This method is deriving COG Jacobian matrices that connect COG of each link with active joints, and calculating the inverse dynamics by using the principle of virtual work. In this method, the equation of motion is derived efficiently because COG Jacobian matrices can be derived in the derivation process of velocities or accelerations. Moreover, it has the feature that is able to represent equations of inverse dynamics as equations of matrices. As the examples, we show the applications to an open-link mechanism, a close-link mechanism, and a mechanism with redundant actuators.

II. Dynamics Employeng COG Jacobian

A. COG Jacobian

This section presents the differences between the general COG Jacobian and the COG Jacobian of the proposed method.

The orientation of COG of robot is denoted by $\boldsymbol{q}_{G0} = [\boldsymbol{p}_{G0}^T, \boldsymbol{\theta}_{G0}^T]^T$, and the orientation vector regarding to all joints is denoted by $\boldsymbol{q} = [\boldsymbol{p}_1^T, \cdots, \boldsymbol{p}_n^T, \boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_n^T]^T$, where $\boldsymbol{p}_{G0}$ and $\boldsymbol{\theta}_{G0}$ are the dimensions of COG of the robot, $\boldsymbol{p}_i$ and $\boldsymbol{\theta}_i$ are the dimensions of joint i, n is number of the joints such as in shown Fig.1. The COG Jacobian matrix of whole-body is described as

$$\boldsymbol{G}_0(\boldsymbol{q}) = \frac{\partial \boldsymbol{q}_{G0}(\boldsymbol{q})}{\partial \boldsymbol{q}}. \tag{1}$$

The COG Jacobian matrix is only one in this case. However, the COG Jacobian matrices are derived in relation to each link in the suggested method by us as in shown Fig.2.

At first, the orientations of the mechanisms with holonomic constraint are decided by the active joints only. The function $\boldsymbol{g}_i(\boldsymbol{\theta})$ is given such that the orientations $\boldsymbol{q}_i = [\boldsymbol{p}_{Gi}^T, \boldsymbol{\theta}_{Gi}^T]^T$ of the link i ($i = 1, \cdots, N$) are expressed as

$$\boldsymbol{q}_i = \boldsymbol{g}_i(\boldsymbol{\theta}_m), \tag{2}$$

where $\boldsymbol{p}_{Gi}$ and $\boldsymbol{\theta}_{Gi}$ are the dimensions of the COG of the link i, $\boldsymbol{\theta}_m \in R^M$ is the vector expressing the displacements of the active joints, N is number of the links and M is number of the active joints. Thus, the link velocity $\dot{\boldsymbol{q}}_i$ at the COG of link i is written in the following form:

$$\dot{\boldsymbol{q}}_i = \boldsymbol{G}_i(\boldsymbol{\theta}_m)\dot{\boldsymbol{\theta}}_m, \tag{3}$$

where the following matrix is the COG Jacobian matrix of the link i:

$$\boldsymbol{G}_i(\boldsymbol{\theta}_m) = \frac{\partial \boldsymbol{g}_i}{\partial \boldsymbol{\theta}_m}, \tag{4}$$

Where

$$\boldsymbol{G}_i = [\boldsymbol{G}_{i1}, \cdots, \boldsymbol{G}_{iM}], \tag{5}$$

$$\boldsymbol{G}_{ik} = \frac{\partial \boldsymbol{g}_i}{\partial \theta_{mk}}, \ (k = 1, \cdots, M). \tag{6}$$

Similarly, the link acceleration   at the COG of link i is written in the following form:

$$\ddot{\boldsymbol{q}}_i = \boldsymbol{G}_i(\boldsymbol{\theta}_m)\ddot{\boldsymbol{\theta}}_m + \dot{\boldsymbol{G}}_i(\boldsymbol{\theta}_m)\dot{\boldsymbol{\theta}}_m, \tag{7}$$

where $\dot{\boldsymbol{G}}_i = [\dot{\boldsymbol{G}}_{i1}, \cdots, \dot{\boldsymbol{G}}_{iM}]$ is the time-derivative of $\boldsymbol{G}_i(\boldsymbol{\theta})$.

Equation (7) can be combined with equations of all links as following equation:

$$\begin{bmatrix} \ddot{\boldsymbol{q}}_1 \\ \vdots \\ \ddot{\boldsymbol{q}}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_1 \\ \vdots \\ \boldsymbol{G}_N \end{bmatrix} \ddot{\boldsymbol{\theta}}_m + \begin{bmatrix} \dot{\boldsymbol{G}}_1 \\ \vdots \\ \dot{\boldsymbol{G}}_N \end{bmatrix} \dot{\boldsymbol{\theta}}_m \tag{8}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{G}\ddot{\boldsymbol{\theta}}_m + \dot{\boldsymbol{G}}\dot{\boldsymbol{\theta}}_m, \tag{9}$$

where $\boldsymbol{G} = [\boldsymbol{G}_1^T, \cdots, \boldsymbol{G}_N^T]^T \in R^{N \times M}$ is the COG Jacobian matrix regarding to the all links and $\dot{\boldsymbol{G}} = [\dot{\boldsymbol{G}}_1^T, \cdots, \dot{\boldsymbol{G}}_N^T]^T \in R^{N \times M}$ is same.

B. Inverse Dynamics

The external force vector $\boldsymbol{F}_i$ around the COG of the link i is expressed as the following equation:

$$\boldsymbol{F}_i = \boldsymbol{M}_i \ddot{\boldsymbol{q}}_i + \boldsymbol{b}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i), \tag{10}$$

where $\boldsymbol{M}_i$ is the mass matrix of the link i, $\boldsymbol{b}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)$ is the term regarding gravitational forces etc. And,

$$\boldsymbol{F}_i = \begin{bmatrix} \boldsymbol{f}_{Gi} \\ \boldsymbol{n}_{Gi} \end{bmatrix}, \tag{11}$$

where $\boldsymbol{f}_{Gi}$ and $\boldsymbol{n}_{Gi}$ are the forces of the translation and the rotation around the COG of the link i. Equation (9) can be combined with all links as

$$\begin{bmatrix} \boldsymbol{F}_1 \\ \vdots \\ \boldsymbol{F}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_1 & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{M}_N \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_1 \\ \vdots \\ \ddot{\boldsymbol{q}}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_N \end{bmatrix} \tag{12}$$

$$\boldsymbol{F} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{b}\,. \tag{13}$$

Next, we compute $\boldsymbol{\tau}_{mi} = [\tau_{1i}, \cdots, \tau_{Mi}]^T$ which is the torque vector regarding to the link i. We obtain $\boldsymbol{\tau}_{mi}$ by the product of the transposed matrix of $\boldsymbol{G}_i$ and $\boldsymbol{F}_i$ like we do in static dynamics. Thus, $\boldsymbol{\tau}_{mi}$ is computed by

$$\boldsymbol{\tau}_{mi} = \boldsymbol{G}_i^T \boldsymbol{F}_i\,. \tag{14}$$

And we obtain the torques of all active joints by using

$$\boldsymbol{\tau}_m = \boldsymbol{G}^T \boldsymbol{F}\,, \tag{15}$$

where $\boldsymbol{\tau}_m = [\boldsymbol{\tau}_1^T, \cdots, \boldsymbol{\tau}_M^T]^T$.

Using (9) and (13), (15) is written as

$$\begin{aligned} \boldsymbol{\tau}_m &= \boldsymbol{G}^T \boldsymbol{M}\boldsymbol{G}\ddot{\boldsymbol{\theta}}_m + \boldsymbol{G}^T \boldsymbol{M}\dot{\boldsymbol{G}}\dot{\boldsymbol{\theta}}_m + \boldsymbol{G}^T \boldsymbol{b} \\ &= \boldsymbol{H}(\boldsymbol{\theta}_m)\ddot{\boldsymbol{\theta}}_m + \boldsymbol{B}(\boldsymbol{\theta}_m, \dot{\boldsymbol{\theta}}_m), \end{aligned} \tag{16}$$

where

$$\boldsymbol{H}(\boldsymbol{\theta}_m) = \boldsymbol{G}^T \boldsymbol{M}\boldsymbol{G}\,, \tag{17}$$

$$\boldsymbol{B}(\boldsymbol{\theta}_m, \dot{\boldsymbol{\theta}}_m) = \boldsymbol{G}^T (\boldsymbol{M}\dot{\boldsymbol{G}}\dot{\boldsymbol{\theta}}_m + \boldsymbol{b})\,. \tag{18}$$

Equation (16) is the form of the generalized equation of motion. Thus, we find the following relationships: $\boldsymbol{H}(\boldsymbol{\theta})$ is the inertia matrix, and $\boldsymbol{B}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is the term regarding centrifugal, colliolis and gravitational forces.

Computations for inverse dynamics employing COG Jacobian consist of following three steps:

1. compute $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ using (2), (3) and (7),
2. compute $\boldsymbol{H}(\boldsymbol{\theta})$ and $\boldsymbol{B}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ using obtained $\boldsymbol{G}$ and $\dot{\boldsymbol{G}}$ in step 1,
3. compute $\boldsymbol{\tau}_m$ using $\boldsymbol{H}(\boldsymbol{\theta})$ and $\boldsymbol{B}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$.

Moreover, in the case that the end-effecters of a mechanism is applied the external forces which is denoted by $\boldsymbol{f}_e$, (16) is written the following form by using the Jacobian matrix $\boldsymbol{J}$ that connects the active joints and the end-effecters:

$$\boldsymbol{\tau}_m = \boldsymbol{H}(\boldsymbol{\theta}_m)\ddot{\boldsymbol{\theta}}_m + \boldsymbol{B}(\boldsymbol{\theta}_m, \dot{\boldsymbol{\theta}}_m) + \boldsymbol{J}^T \boldsymbol{f}_e\,. \tag{19}$$

C. Forward Dynamics

We show the primitive method of forward dynamics employing COG Jacobian.

At first, derive the equation of motion using (16). Next, solve the derived equation for $\ddot{\boldsymbol{\theta}}_m$. Then, we obtain the solutions by

$$\ddot{\boldsymbol{\theta}}_m = \boldsymbol{H}(\boldsymbol{\theta}_m)^{-1}\{\boldsymbol{\tau}_m - \boldsymbol{B}(\boldsymbol{\theta}_m, \dot{\boldsymbol{\theta}}_m)\}\,. \tag{20}$$

However, computation of the inverse matrix of $\boldsymbol{H}(\boldsymbol{\theta})$ needs the large computation cost, when a mechanism has many active joints. Therefore, we suggest using more efficient methods such as the ones shown in [12] or [13].

D. Features of COG Jacobian Matrices

COG Jacobian matrices have the feature that the elements of the matrices are decided by a structure of mechanisms. It is shown by the following examples.

Fig. 3 is examples of mechanisms. (a) in Fig.3 shows a primitive open-link mechanism with N links. If we already obtain $\boldsymbol{G}$, then the torques of the active joints is expressed as

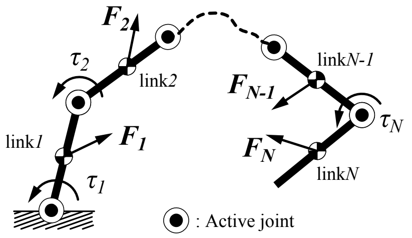
$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_{11}^T & \boldsymbol{G}_{21}^T & \cdots & \boldsymbol{G}_{N1}^T \\ 0 & \boldsymbol{G}_{22}^T & \cdots & \boldsymbol{G}_{N2}^T \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{G}_{NN}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \vdots \\ \boldsymbol{F}_N \end{bmatrix}. \tag{21}$$

The COG Jacobian matrix $\boldsymbol{G}$ is expressed as the form that is upper triangular matrix because the torques have correlation: i.e. the torques of the active joints are affected by the torques of the active joints connected towards the end link than itself.

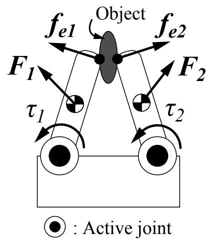
Next, (b) in Fig. 3 shows a robot hand picking an object. If we already obtained $\boldsymbol{G}$, then the torques of the active joints is expressed as

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_{11}^T & 0 & \boldsymbol{J}_1^T & 0 \\ 0 & \boldsymbol{G}_{22}^T & 0 & \boldsymbol{J}_2^T \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{f}_{e1} \\ \boldsymbol{f}_{e2} \end{bmatrix}, \tag{22}$$

where $\boldsymbol{J}_i$ and $\boldsymbol{f}_{ei}$ are the Jacobian matrices and the reaction forces. If $\boldsymbol{f}_{ei}$ are known, we can consider as that the left and the right finger are the independent mechanisms. Therefore, because the left and the right finger have no correlation, the elements except diagonal ones of $\boldsymbol{G}$ are zero.



(a) Robot arm model of serial mechanisms.



(b) Robot hand model.

Fig. 3 Examples for showing the features of COG Jacobian matrices on two types of mechanisms. (a) is a serial link mechanism. (b) is a robot hand model picking a object.

III. Numerical Examples

At first, the proposed scheme employing COG Jacobian matrices is applied to an open-link mechanism with 2 links (see Fig.4(a)). Next, we show the application to a closed-link mechanisms without redundant actuators (see Fig.4(b)). Finally, we apply to a mechanism with redundant actuators (see Fig.4(c)).

A. Open-Link Mechanism

The active joints of the mechanism in the shown Fig. 4(a) are denoted by $\boldsymbol{\theta}_m = [\theta_1, \theta_2]^T$. The number N and M are equal to two together. The vector $\boldsymbol{q}_i$ that expresses the orientation of link i is calculated by the following equations:

$$\boldsymbol{q}_1 = \boldsymbol{g}_1(\boldsymbol{\theta}_m) = R_z^{\theta_1}\begin{bmatrix} l_{G1} \\ 0 \\ \theta_1 \end{bmatrix}, \tag{23}$$

$$\boldsymbol{q}_2 = \boldsymbol{g}_2(\boldsymbol{\theta}_m) = R_z^{\theta_1}\begin{bmatrix} l_1 \\ 0 \\ \theta_1 \end{bmatrix} + R_z^{\theta_1+\theta_2}\begin{bmatrix} l_{G2} \\ 0 \\ \theta_2 \end{bmatrix}, \tag{24}$$

where l_i is the length of the link i, l_{Gi} is the length from the end to the COG of the link i, and

$$R_z^{\theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{25}$$

Next, we compute the velocities and the accelerations at the COG of the link i.

The velocities are expressed as

$$\dot{\boldsymbol{q}}_1 = \boldsymbol{G}_1(\boldsymbol{\theta}_m)\dot{\boldsymbol{\theta}}_m = \begin{bmatrix} -l_{G1}\sin\theta_1 & 0 \\ l_{G1}\cos\theta_1 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \tag{26}$$

$$\dot{\boldsymbol{q}}_2 = \boldsymbol{G}_2(\boldsymbol{\theta}_m)\dot{\boldsymbol{\theta}}_m = \begin{bmatrix} -l_{G2}\sin(\theta_1+\theta_2) - l_1\sin\theta_1 & -l_{G2}\sin(\theta_1+\theta_2) \\ l_{G2}\cos(\theta_1+\theta_2) + l_1\cos\theta_1 & l_{G2}\cos(\theta_1+\theta_2) \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{27}$$

and the accelerations are expressed similarly as

$$\ddot{\boldsymbol{q}}_1 = \boldsymbol{G}_1(\boldsymbol{\theta}_m)\ddot{\boldsymbol{\theta}}_m + \dot{\boldsymbol{G}}_1(\boldsymbol{\theta}_m)\dot{\boldsymbol{\theta}}_m, \tag{28}$$

$$\ddot{\boldsymbol{q}}_2 = \boldsymbol{G}_2(\boldsymbol{\theta}_m)\ddot{\boldsymbol{\theta}}_m + \dot{\boldsymbol{G}}_2(\boldsymbol{\theta}_m)\dot{\boldsymbol{\theta}}_m, \tag{29}$$

where

$$\dot{\boldsymbol{G}}_1(\boldsymbol{\theta}_m) = \begin{bmatrix} -l_{G1}\dot{\theta}_1\cos\theta_1 & 0 \\ -l_{G1}\dot{\theta}_1\sin\theta_1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{30}$$

$$\dot{\boldsymbol{G}}_2(\boldsymbol{\theta}_m) = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \\ 0 & 0 \end{bmatrix}, \tag{31}$$

$$K_{11} = -l_{G2}(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1+\theta_2) - \dot{\theta}_1 l_1\cos\theta_1, \tag{32}$$

$$K_{12} = -l_{G2}(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1+\theta_2), \tag{33}$$

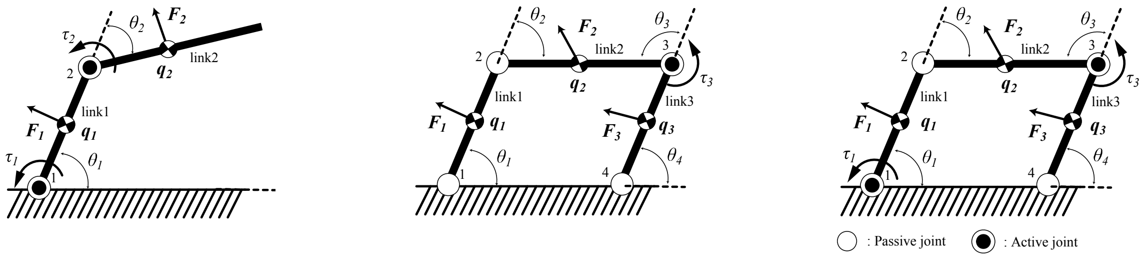
$$K_{21} = -l_{G2}(\dot{\theta}_1 + \dot{\theta}_2)\sin(\theta_1+\theta_2) - \dot{\theta}_1 l_1\sin\theta_1, \tag{34}$$

and

$$K_{22} = -l_{G2}(\dot{\theta}_1 + \dot{\theta}_2)\sin(\theta_1+\theta_2). \tag{35}$$

Substituting (28) and (29) for (10), we obtain the forces as

$$\begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_1 & 0 \\ 0 & \boldsymbol{M}_2 \end{bmatrix}\begin{bmatrix} \ddot{\boldsymbol{q}}_1 \\ \ddot{\boldsymbol{q}}_2 \end{bmatrix}. \tag{36}$$



(a) Model of serial link mechanisms with 2 links.

(b) Model of closed link mechanisms without redundant actuators.

(c) Model of closed link mechanisms with a redundant actuator.

Fig. 4 Three types of link mechanisms for the dynamic analysis examples employing the COG Jacobian matrices.

Finally, the torques of the active joints is obtained by

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_1 \\ \boldsymbol{G}_2 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{bmatrix}. \tag{37}$$

Figure 5 shows the target motion of the mechanism (length of each link: 0.10 (m); weight: 1.54 x 10^{-2} (kg); moment of inertia: 1.29 x 10^{-5} (kgm^2); COG at midpoint). Then, we obtain the torque curves of the active joints using (37) (see Fig. 6).

B. Closed-Link Mechanism without Redundant Actuators

The active joint of the mechanism are denoted by $\theta_m = \theta_3$. And, N = 3, and M = 1 of the mechanism. The passive joint angle θ_1 and θ_4 can be expressed by θ_3 only: i.e. they are the functions of θ_3 as

$$\theta_1 = \theta_1(\theta_3) \text{ and } \theta_4 = \theta_4(\theta_3). \tag{38}$$

Therefore, the vector $\boldsymbol{q}$ is expressed by the following equations:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \boldsymbol{q}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_1(\theta_3) \\ \boldsymbol{g}_2(\theta_3) \\ \boldsymbol{g}_3(\theta_3) \end{bmatrix}. \tag{39}$$

Next, differentiate (39) with respect to time, then $\dot{\boldsymbol{q}}$ is expressed as

$$\dot{\boldsymbol{q}} = \boldsymbol{G}(\theta_m)\dot{\theta}_m = \begin{bmatrix} \boldsymbol{G}_1 \\ \boldsymbol{G}_2 \\ \boldsymbol{G}_3 \end{bmatrix} \dot{\theta}_3, \tag{40}$$

$$\boldsymbol{G}_i = \frac{\partial \boldsymbol{g}_i}{\partial \theta_3}. \tag{41}$$

Now we have $\boldsymbol{G}$, and $\boldsymbol{F}$ is obtained by using (13). Since the torque $\boldsymbol{\tau}_3$ of the active joint θ_3 is computed by

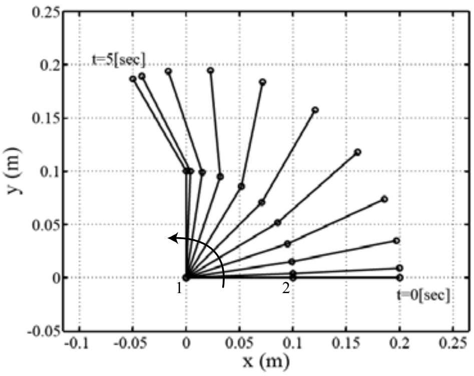


Fig. 5 Target motion

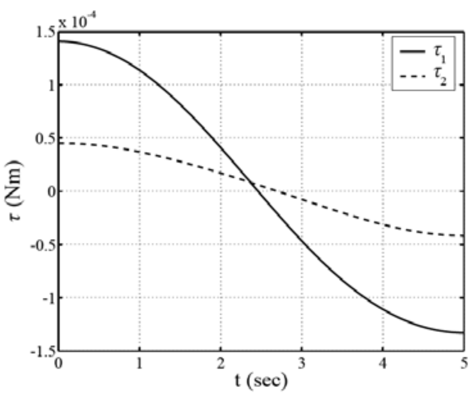


Fig. 6 Torque curves

$$[\tau_3] = \begin{bmatrix} \boldsymbol{G}_1 \\ \boldsymbol{G}_2 \\ \boldsymbol{G}_3 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{F}_3 \end{bmatrix}. \tag{42}$$

C. Closed-Link Mechanism with Redundant Actuators

The active joint of the mechanism are denoted by $\boldsymbol{\theta}_m = [\theta_1, \theta_3]^T$. And, N = 3, and M = 2 of the mechanism. The passive joint angle θ_4 is the functions of θ_1 and θ_3 as

$$\theta_4 = \theta_4(\theta_1, \theta_3). \tag{43}$$

Therefore, $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are expressed as follows:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \boldsymbol{q}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_1(\theta_1, \theta_3) \\ \boldsymbol{g}_2(\theta_1, \theta_3) \\ \boldsymbol{g}_3(\theta_1, \theta_3) \end{bmatrix}, \tag{44}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{G}(\theta_m)\dot{\theta}_m = \begin{bmatrix} \boldsymbol{G}_1 \\ \boldsymbol{G}_2 \\ \boldsymbol{G}_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_3 \end{bmatrix}, \tag{45}$$

where

$$\boldsymbol{G}_i = \left[\frac{\partial \boldsymbol{g}_i}{\partial \theta_1}, \frac{\partial \boldsymbol{g}_i}{\partial \theta_3} \right]. \tag{46}$$

And the torque and are computed by

$$\begin{bmatrix} \tau_1 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_1 \\ \boldsymbol{G}_2 \\ \boldsymbol{G}_3 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{F}_3 \end{bmatrix}. \tag{47}$$

In this paper, we don't discuss about the optimization of the actuational redundancy. However, it is enabled by using Jacobian matrices that are obtained from the constraint equations of joints (hints shown in Ref. [2]-[4]).

Figure 7 shows the target motion for the closed-link mechanisms of the subsection *B* and *C* (length of each link: 0.20 (m); weight: 0.628 (kg); moment of inertia: 2.11 x 10^{-3} (kgm^2); COG at midpoint). We obtain the torque curves of the active joints using (42) and (47) (see Fig. 8).

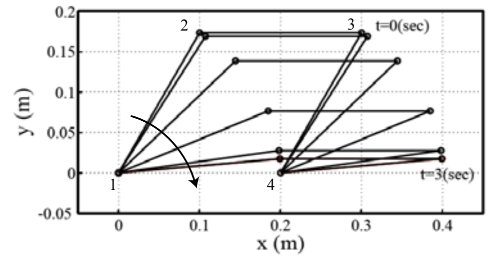


Fig. 7 Target motion

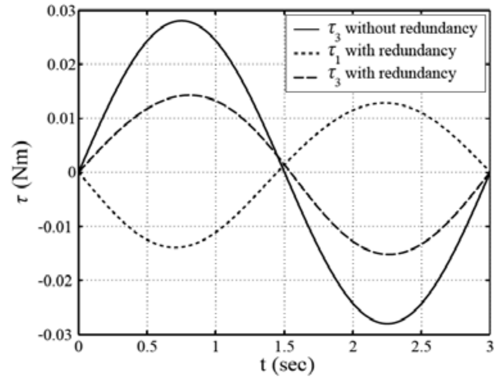


Fig. 8 Torque curves

IV. Conclusion

We propose the unified scheme that is able to be applied to various link mechanisms for dynamics and show the numerical examples of the three mechanisms of difference type together in this paper. The proposed method employs COG Jacobian matrices for dynamics analyses. We can derive efficiently the equation of motion by using it.

COG Jacobian matrices are able to be obtained in a deriving process of velocities or accelerations. They has feature which COG matrices are affected by structures of a mechanism. Moreover, an obtained equation of motion is expressed as the matrix form. Consequently, those features show that dynamics computation employing COG Jacobian is effective as a scheme that expresses the dynamics of link mechanisms.

References

[1] E.J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical System*. Boston: Allyn and Bacon, 1989.

[2] Y. Nakamura: "Dynamics Computation of Parallel Mechanisms," Journal of the Robotics Society of Japan, Vol.10, No.6, pp.709-714, 1992.

[3] T. Yoshikawa, Y. Yokokohji and N. Watanabe: "Dynamics Analysis of Closed Link Mechanisms with Redundant Actuators," Journal of the Robotics Society of Japan, Vol.11, No.6, pp.913-917, 1993.

[4] Y. Nakamura, K. Yamane and F. Nagashima: "Dynamics Computation of Structure-Varying Kinematic Chains and Its Application to Human Figure," Journal of the Robotics Society of Japan, Vol.16, No.8, pp.1152-1159, 1998.

[5] D. Isobe: "A Unified Numerical Scheme for Calculating Inverse Dynamics of Open/Closed Link Mechanisms," Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics Society IECON'01, pp.341-344, 2001.

[6] D. Isobe, A. Yagi and S. Sato: "General-Purpose Expression of Structural Connectivity in the Parallel Solution Scheme and Its Application," JSME International Journal Series C, Vol.49, No.3, pp.789-798, 2006.

[7] K. Sugimoto: "Deviation of Equation of Motion of Closed Loop Mechanisms," Journal of the Robotics Society of Japan, Vol.15, No.3, pp.460-467, 1997.

[8] T. Sugihara and Y. Nakamura: "Whole-body Cooperative Reaction Force Manipulation on Legged Robots with COG Jacobian involving Implicit Representation of Unactuated Coordinates," Journal of the Robotics Society of Japan, Vol.24, No.2, pp.222-231, 2007.

[9] R. Boulic, R. Mas and D. Thalmann: "Inverse Kinetics for Center of Mass Position Control and Posture Optimizaion," Proceedings of European Workshop on Combined Real and Synthetic Image Processing for Broadcast and video Production, 1994.

[10]P. Baerlocher, *Inverse kinematics techniques of the interactive posture control of articulated figures.* PhD thesis, Lausanne, 2001.

[11]Y. Tamiya, M. Inaba and H. Inoue: "Realtime Balance Compensation for Dynamics Motion of Full-Body Humanoid Standing on One Leg," Journal of the Robotics Society of Japan, Vol.17, No.2, pp.268-274, 1999.

[12]K. Yamane and Y. Nakamura: "Efficient Parallel Dynamics Computation of Human Figure," Proceedings of the 2002 IEEE International Conference on Robotics & Automation, pp.530-537, 2002.

[13]R. Featherstone: "A Divide-and-Conquer Articulated-Body Algorithm for Parallel O(log(n)) Calculation of Rigid-Body Dynamics," International Journal of Robotics Research, Vol.18, No.9, pp.867-892, 1999.