Unified Dynamics Calculation employing COG Jacobian for Link Mechanisms

Takashi Sonoda¹*, Kazuo Ishii¹ and Daigoro Isobe²

¹Kyushu Institute of Technology and ²University of Tukuba

1-2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan and

2 1-1-1 Tennodai, Tsukuba-shi, Ibaraki 305-8573, Japan

*e-mail: sonoda-takashi@edu.brain.kyutech.ac.jp

Abstract—We show the unified dynamics computation employing Jacobian that relates the center of gravity (COG Jacobian) to joints of link mechanisms. COG Jacobian is used for the behavior planning and the control of humanoids. Also, it usually expresses the relationship between the joints and COG of a robot’s whole-body. However, this scheme that is suggested by us calculated regarding each link and not the robot’s whole-body. Moreover, we can obtain the torques of the active joints in mechanisms by relating COG Jacobian to the applied forces to COG and by using principle of virtual work. The COG Jacobian is obtained via the process calculating the accelerations at COG of links and the equation of motion expressed matrix forms by using COG Jacobian. Therefore, we can compute the equation of motion of a mechanism effectively, and the forward dynamics which can be calculated by the obtained equation. We can calculate the inverse dynamics regardless of open- and closed-link mechanisms.

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 to 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. And, 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 not only open- and closed-link mechanisms, but applications to mechanisms containing redundant actuators, collisions and contacts are also expected. Therefore unified dynamics computation schemes treating their issues are hoped. A Method of equations of motion solving simultaneous equations of constraints corresponding to each joint is used in 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. Then, after considering closed-link mechanisms to be virtual open-link mechanisms of tree structure temporarily and solving as a problem of open-link mechanisms, 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]. This method needs to cut some arbitrary joints. Therefore, in complex mechanisms, to choose the cutting joints and the generalized coordinates is problems. 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 apply to mechanisms with discontinuous changes of constraints [4], and this method is the solution to above problems. However, this method 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]. But, we have to solve simultaneous equations of forces regarding passive joints at closed-link mechanisms.
II. DYNAMICS EMPLOYING 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 \( \mathbf{q}_{rg} = [\mathbf{p}_{rg}^T, \mathbf{q}_{rg}^T]^T \), and the orientation vector regarding to all joints is denoted by \( \mathbf{q} = [\mathbf{p}_i^T, \cdots, \mathbf{p}_n^T, \theta_1^i, \cdots, \theta_n^i]^T \), where \( \mathbf{p}_{rg} \) and \( \theta^i_2 \) are the dimensions of COG of the robot, \( \mathbf{p}_i \) and \( \theta_i \) are the dimensions of joint \( i \), \( n \) is number of the joints such as shown in Fig.1. The COG Jacobian matrix of whole-body is described as

\[
\mathbf{G}_0(\mathbf{q}) = \frac{\partial \mathbf{q}_{rg}}{\partial \mathbf{q}}.
\]

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 shown in Fig.2.

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

\[
\mathbf{q}_i = \mathbf{g}_i(\theta_m),
\]

where \( \mathbf{p}_{gi} \) and \( \theta_{gi} \) are the dimensions of the COG of the link \( i \), \( \theta_m \in \mathbb{R}^m \) 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 \( \dot{\mathbf{q}}_i \) at the COG of link \( i \) is written in the following form:

\[
\dot{\mathbf{q}}_i = \mathbf{G}_i(\theta_m)\dot{\theta}_m,
\]

where the following matrix is the COG Jacobian matrix of the link \( i \):

\[
\mathbf{G}_i(\theta_m) = \frac{\partial \mathbf{q}_i}{\partial \theta_m},
\]

Where

\[
\mathbf{G}_i = [\mathbf{G}_{i1}, \cdots, \mathbf{G}_{im}],
\]

\[
\mathbf{G}_{ik} = \frac{\partial \mathbf{q}_i}{\partial \theta_{ik}}, \quad (k=1, \cdots, M).
\]

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

\[
\ddot{\mathbf{q}}_i = \mathbf{G}_i(\theta_m)\ddot{\theta}_m + \mathbf{G}_i(\theta_m)\dot{\theta}_m,
\]

where \( \mathbf{G}_i = [\mathbf{G}_{i1}, \cdots, \mathbf{G}_{im}] \) is the time-derivative of \( \mathbf{G}_i(\theta) \).

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

\[
\begin{bmatrix}
\mathbf{\ddot{q}}_1 \\
\vdots \\
\mathbf{\ddot{q}}_N
\end{bmatrix} =
\begin{bmatrix}
\mathbf{G}_1 \\
\vdots \\
\mathbf{G}_N
\end{bmatrix}
\begin{bmatrix}
\mathbf{\ddot{\theta}}_m \\
\dot{\mathbf{\theta}}_m
\end{bmatrix}
\]

\[
\mathbf{\ddot{q}} = \mathbf{G}\mathbf{\ddot{\theta}}_m + \mathbf{G}\dot{\mathbf{\theta}}_m,
\]

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

B. Inverse Dynamics

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

\[
\mathbf{F}_i = \mathbf{M}_i \ddot{\mathbf{q}}_i + \mathbf{b}(\mathbf{q}_i, \dot{\mathbf{q}}_i),
\]

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

\[
\mathbf{F}_i = \begin{bmatrix}
\mathbf{f}_{gi} \\
\mathbf{n}_{gi}
\end{bmatrix},
\]
where \( f_{gi} \) and \( n_{gi} \) 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:

\[
\begin{bmatrix}
F_1 \\
\vdots \\
F_N
\end{bmatrix} = \begin{bmatrix}
M_1 & 0 & \tilde{q}_1 & b_1 \\
\vdots & \ddots & \vdots & \vdots \\
0 & \cdots & M_N & \tilde{q}_N \\
b_1 & \cdots & b_N
\end{bmatrix} \begin{bmatrix}
\dot{q}_1 \\
\vdots \\
\dot{q}_N
\end{bmatrix} + \begin{bmatrix}
b_1 \\
\vdots \\
b_N
\end{bmatrix}
\]

(12)

\[ F = M\dot{q} + b. \]

(13)

Next, we compute \( \tau_{\text{act}} = [\tau_{1i}, \cdots, \tau_{Mi}]^T \) which is the torque vector regarding to the link \( i \). We obtain \( \tau_{\text{act}} \) by the product of the transposed matrix of \( G_i \) and \( F_i \) like we do in static dynamics. Thus, \( \tau_{\text{act}} \) is computed by

\[ \tau_{\text{act}} = G_i^T F_i. \]

(14)

And we obtain the torques of all active joints by using

\[ \tau = G^T F, \]

(15)

where \( \tau = [\tau_1, \cdots, \tau_M]^T \).

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

\[ \tau = G^T M \dot{\theta} + G^T M \ddot{\theta} + G^T b \]

(16)

\[ = H(\theta) \ddot{\theta} + B(\theta, \dot{\theta}), \]

where

\[ H(\theta) = G^T M G, \]

(17)

\[ B(\theta, \dot{\theta}) = G^T (M \ddot{\theta} + b). \]

(18)

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

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

1. compute \( \dot{q}, \ddot{q} \) and \( \dddot{q} \) using (2), (3) and (7),
2. compute \( H(\theta) \) and \( B(\theta, \dot{\theta}) \) using obtained \( G \) and \( \dot{G} \) in step 1,
3. compute \( \tau_{\text{act}} \) using \( H(\theta) \) and \( B(\theta, \dot{\theta}) \).

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

\[ \tau = H(\theta) \ddot{\theta} + B(\theta, \dot{\theta}) + J^T f_e. \]

(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 \( \ddot{\theta}. \) Then, we obtain the solutions by

\[ \ddot{\theta} = H(\theta)^{-1} [\tau_{\text{act}} - B(\theta, \dot{\theta})]. \]

(20)

However, computation of the inverse matrix of \( H(\theta) \) 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].

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 \( G \), then the torques of the active joints is expressed as

\[
\begin{bmatrix}
\tau_1 \\
\tau_2 \\
\vdots \\
\tau_N
\end{bmatrix} = \begin{bmatrix}
G_{11} & G_{12} & \cdots & G_{1N} \\
0 & G_{22} & \cdots & G_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & G_{NN}
\end{bmatrix} \begin{bmatrix}
F_1 \\
F_2 \\
\vdots \\
F_N
\end{bmatrix}.
\]

(21)

The COG Jacobian matrix \( G \) 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 \( \mathbf{G} \), then the torques of the active joints is expressed as

\[
\begin{bmatrix}
\tau_1 \\
\tau_2 
\end{bmatrix} = \begin{bmatrix}
\mathbf{G}_1^T & \mathbf{0} \\
\mathbf{0} & \mathbf{G}_2^T 
\end{bmatrix}
\begin{bmatrix}
\mathbf{J}_1^T \\
\mathbf{J}_2^T 
\end{bmatrix}
\begin{bmatrix}
\mathbf{F}_1 \\
\mathbf{F}_2 
\end{bmatrix},
\]

(22)

where \( \mathbf{J}_i \) and \( \mathbf{f}_i \) are the Jacobian matrices and the reaction forces. If \( \mathbf{f}_i \) 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 \( \mathbf{G} \) are zero.

III. 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 \( \theta_\alpha = [\theta_1, \theta_2]^T \). The number \( N \) and \( M \) are equal to two together. The vector \( \mathbf{q}_i \) that expresses the orientation of link \( i \) is calculated by the following equations:

\[
\mathbf{q}_1 = \mathbf{g}_1(\theta_\alpha) = R_1 \begin{bmatrix}
\theta_1 \\
\theta_2 
\end{bmatrix},
\]

(23)

\[
\mathbf{q}_2 = \mathbf{g}_2(\theta_\alpha) = R_1 \begin{bmatrix}
\theta_1 \\
\theta_2 
\end{bmatrix} + R_2 \begin{bmatrix}
\theta_1 \\
\theta_2 
\end{bmatrix},
\]

(24)

where \( l_i \) is the length of the link \( i \), \( l_{i0} \) is the length from the end to the COG of the link \( i \), and

\[
R_0 = \begin{bmatrix}
\cos \theta_i & -\sin \theta_i & 0 \\
\sin \theta_i & \cos \theta_i & 0 \\
0 & 0 & 1 
\end{bmatrix}.
\]

(25)

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

The velocities are expressed as

\[
\dot{\mathbf{q}}_i = \dot{\mathbf{g}}_i(\theta_\alpha) \dot{\theta}_\alpha = \begin{bmatrix}
-l_{i1} \sin \theta_1 \\
l_{i1} \cos \theta_1 \\
0
\end{bmatrix},
\]

(26)

\[
\dot{\mathbf{q}}_2 = \begin{bmatrix}
-l_{i2} \sin(\theta_1 + \theta_2) - l_1 \sin \theta_1 - l_{i2} \sin(\theta_1 + \theta_2) \\
l_{i2} \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \end{bmatrix},
\]

(27)

and the accelerations are expressed similarly as

\[
\ddot{\mathbf{q}}_1 = \mathbf{G}_1(\theta_\alpha) \ddot{\theta}_\alpha = \begin{bmatrix}
-l_{i1} \dot{\theta}_1 \cos \theta_1 \\
l_{i1} \dot{\theta}_1 \sin \theta_1 \\
0
\end{bmatrix},
\]

(28)

\[
\ddot{\mathbf{q}}_2 = \mathbf{G}_2(\theta_\alpha) \ddot{\theta}_\alpha = \begin{bmatrix}
K_{11} & K_{12} \\
K_{21} & K_{22} 
\end{bmatrix}
\]

(29)

where

\[
\mathbf{G}_1(\theta_\alpha) = K_1 \begin{bmatrix}
\dot{\theta}_1 & \dot{\theta}_2 
\end{bmatrix},
\]

(30)

\[
\mathbf{G}_2(\theta_\alpha) = K_2 \begin{bmatrix}
\dot{\theta}_1 & \dot{\theta}_2 
\end{bmatrix},
\]

(31)

\[
K_1 = -l_{i2} (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2) - \dot{\theta}_1 l_1 \cos \theta_1,
\]

(32)

\[
K_2 = -l_{i2} (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2) - \dot{\theta}_1 l_1 \sin \theta_1,
\]

(33)

\[
K_{22} = -l_{i2} (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2). \]

(34)

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

\[
\begin{bmatrix}
\mathbf{F}_1 \\
\mathbf{F}_2 
\end{bmatrix} = \begin{bmatrix}
M_1 & 0 \\
0 & M_2 
\end{bmatrix} \dot{\mathbf{q}}_i,
\]

(36)

Finally, the torques of the active joints is obtained by

\[
\begin{bmatrix}
\tau_1 \\
\tau_2 
\end{bmatrix} = \begin{bmatrix}
\mathbf{G}_1^T & \mathbf{0} \\
\mathbf{0} & \mathbf{G}_2^T 
\end{bmatrix}
\begin{bmatrix}
\mathbf{F}_1 \\
\mathbf{F}_2 
\end{bmatrix}.
\]

(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 \( \theta_\alpha = \theta_1 \). And, \( N = 3 \), and \( M = 1 \) of the mechanism. The passive joint angle \( \theta_1 \) and \( \theta_2 \) can be expressed by \( \theta_3 \) only: i.e. they are the
functions of $\theta_3$ as

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

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

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} g_1(\theta_3) \\ g_2(\theta_3) \\ g_3(\theta_3) \end{bmatrix}. \quad (39)$$

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

$$\dot{\mathbf{q}} = \mathbf{G}(\theta_m)\dot{\theta}_m = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \quad (40)$$

where

$$\mathbf{G}_i = \frac{\partial g_i}{\partial \theta_3}. \quad (46)$$

And the torque and are computed by

$$\mathbf{r}_3 = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} G_1^T F_1 \\ G_2^T F_2 \\ G_3^T F_3 \end{bmatrix}. \quad (47)$$

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.

C. Closed-Link Mechanism with Redundant Actuators

The active joint of the mechanism are denoted by $\theta_m = [\theta_1, \theta_3]^T$. And, $N = 3$, and $M = 2$ of the mechanism. The passive joint angle $\theta_2$ is the functions of $\theta_1$ and $\theta_3$ as

$$\theta_2 = \theta_2(\theta_1, \theta_3). \quad (43)$$

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

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} g_1(\theta_1, \theta_3) \\ g_2(\theta_1, \theta_3) \\ g_3(\theta_1, \theta_3) \end{bmatrix}, \quad (44)$$
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: $2.11 \times 10^{-3}$ (kgm$^2$); COG at midpoint). We obtain the torque curves of the active joints using (42) and (47) (see Fig. 8).

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 have feature which COG matrices are affected by structures of a mechanism. Moreover, an obtained equation of motion is expressed as the matrix form. Those features show that dynamics computation employing COG Jacobian is effective as a scheme that expresses the dynamics of link mechanisms.

REFERENCES