Proceedings of the 1999 IEEE/RSJ
International Conference on
Intelligent Robots and Systems

Dynamics Computation of Closed Kinematic Chains for Motion Synthesis of Human Figures

Katsu YAMANE and Yoshihiko NAKAMURA
(E-mail: [katz, nakamura]@ynl.t.u-tokyo.ac.jp)

Dept. of Mechano-Informatics, Univ. of Tokyo
7-3-1 Hongo Bunkyo-ku Tokyo, 113-8656 JAPAN

Abstract

This paper discusses the dynamics computation of closed kinematic chains, especially those found in motions of human figures. A number of efficient dynamics computation algorithms have been established in robotics for open kinematic chains and particular types of closed kinematic chains, such as parallel five-bar link mechanisms and the Stewart platform. The dynamics computation of closed kinematic chains, however, is still challenging and among open research issues. In this paper, we describe the mobility of closed kinematic chains by the minimal set of independent variables, which we call the generalized coordinates of a closed kinematic chain. We then develop a systematic procedure to find them out, and establish the computational algorithms for the inverse and forward dynamics of any closed kinematic chains. The numerical examples show the effectiveness of the algorithms in particular for computing high-degrees-of-freedom human/animal motions.

Key Words: Closed kinematic chains, Dynamics computation, Generalized coordinates, Human motion.

1 Introduction

Many algorithms have been developed in robotics for inverse and forward dynamics computation of closed kinematic chains [1]-[5], some of them achieving order-n complexity. Their main fields of application are control and simulation of industrial robots, which usually have relatively low degrees of freedom and less closed loops, and are designed to have simple structures such as planar, parallel or symmetric ones. Since industrial robots are artifacts designed by ourselves, we know best their degrees of freedom and which variables represent the mobility of the structure, except for the singular points.

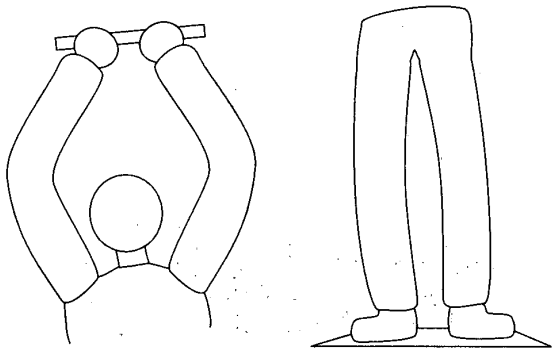

Figure 1: Closed kinematic chains in human motion

When human or animal bodies move, one finds many complicated structures including closed kinematic chains as illustrated in Fig.1. Holding a bar by the both hands, or standing with the both feet on the ground, for example, generates a closed kinematic chain. Dynamics computation of such systems is coming into need for simulation, control, and motion generation in the fields of computer graphics animations and humanoid research. Among many motions of human figures, locomotion is frequently taken as a research subject and some methods for its simulation have been proposed[6]-[8]. Although they provide efficient methods for computing the dynamics of walking human figures, more complicated interactions such as those with other human figures' bodies, for example, are not considered.

The dynamics computation algorithms currently used in general purpose motion analysis softwares [1] can handle any mechanisms and simulate their motions. However, they tend to require enormous amount of computation because of the large number

0-7803-5184-3/99/$10.00 © 1999 IEEE

of coordinates they use. Several efficient order-n algorthims are also proposed[4, 9], however, the number of closed loops are assumed to be small in order to make use of the sparsity of inertia matrix. Motions of human figures highly interacting with the environment or each others, therefore, will increase the computational load because of the number of unknowns in the equation.

In robotics, the dynamics computation algorithms have been developed taking account of their efficiency and adopting the minimal number of coordinates. The algorithms were extended from open kinematic chains to closed ones, where the closed chain is transformed into equivalent tree structure by virtually cutting some joints in closed loops. Most of them use the Lagrange multipliers to compute the constraint force and moment at the cut joints[2]. Alternative approach was proposed by Nakamura and Ghodoussi [10], where the Jacobian matrix of unactuated joints with respect to actuated ones is used instead of the Lagrange multipliers. A similar method is used in [11] to derive a linear form of parallel mechanisms. This approach uses the minimal number of coordinates and computationally efficient. Another advantage of the approach is that the values obtained during the computation have clear physical meanings. However, the computation of the Jacobian matrix was shown only for simple closed kinematic chains such as parallel five-bar link structures. Therefore, the systematic computation of the Jacobian matrix of unactuated joints with respect to actuated ones for the general closed kinematic chains remains an open research issue.

In this paper, we first introduce generalized coordinate of a closed kinematic chain, which are defined as the independent variables that represent the mobility of the kinematic chain. As for a designed manipulator, we know in advance its degrees of freedom and the variables that represent the motion. For closed kinematic chains found in motions of human figures, on the other hand, since we cannot predict their structures, the generalized coordinates or the degrees of freedom are not defined or computed in advance. We develop a general algorithm that systematically select the generalized coordinates and compute the degree of freedom. The developed algorithm is used with the previously proposed efficient computation algorithm of Nakamura et al.[10] and applied to compute the inverse and forward dynamics of general closed kinematic chains. The usability of the algorithm is verified by an example of dynamics simulation, followed by the conclusions.

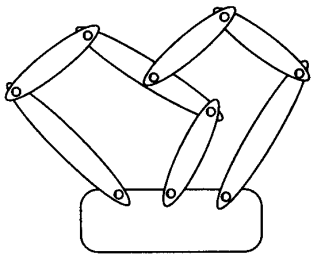

Figure 2: Closed kinematic chain

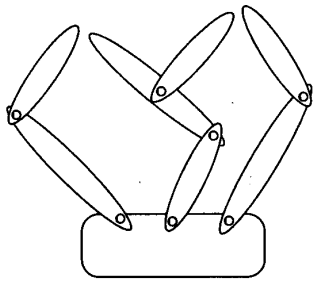

Figure 3: Tree-structure open kinematic chain

2 Dynamics Computation of General Closed Kinematic Chains

2.1 Generalized Coordinates of Closed Kinematic Chains

Consider a closed kinematic chain in Fig.2. Let N_J be the total number of joints in the chain, $\boldsymbol{\theta}_J \in \boldsymbol{R}^{N_J}$ the whole joint angles, N_A the number of actuated joints, $\boldsymbol{\theta}_A \in \boldsymbol{R}^{N_A}$ the actuated joints and $\boldsymbol{\tau}_A \in \boldsymbol{R}^{N_A}$ the actuator torques. In this section, we assume that the mechanism has rotational or translational joints of single-degree-of-freedom for simplicity sake. Introducing multi-degrees-of-freedom joints requires no essential modification to the algorithm, as discussed in [12].

Suppose that the closed chain is virtually cut at some joints and forms a tree-structure open kinematic chain in Fig.3. Let N_O be the number of joints in the tree-structure chain, $\boldsymbol{\theta}_O \in \boldsymbol{R}^{N_O}$ the joint angles and $\boldsymbol{\tau}_O \in \boldsymbol{R}^{N_O}$ the joint torques. We assume at this moment that all joints in the tree structure, including those unactuated in the original closed chain, are actuated to control the virtual tree structure.

Suppose that the tree structure makes the same motion as the original closed chain without force or mo-

ment interaction at the virtually cut joints. The joint torques $\boldsymbol{\tau}_O$ required to generate the motion is computed by recursive inverse dynamics algorithms for open kinematic chains [13, 14, 15]. Note that nonzero values may be obtained for the elements of $\boldsymbol{\theta}_O$ corresponding to the unactuated joints in the original closed kinematic chain.

Let the original closed kinematic chain have N_F degrees of freedom, $\boldsymbol{\theta}_G \in \boldsymbol{R}^{N_F}$ be the generalized coordinates that describe the mobility of the closed kinematic chain, and $\boldsymbol{\tau}_G$ be the generalized force. We can form $\boldsymbol{\theta}_G$ by selecting appropriate N_F joints from $\boldsymbol{\theta}_J$, for instance. Since the generalized coordinates determine the motion of the whole mechanism, $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_O$ can be written as follows:

$$\boldsymbol{\theta}_O = \boldsymbol{\theta}_O(\boldsymbol{\theta}_G) \tag{1}$$

$$\boldsymbol{\theta}_A = \boldsymbol{\theta}_A(\boldsymbol{\theta}_G) \tag{2}$$

From Eq.(1), the d'Alembert's principle, and the principle of virtual work, the joint torques of the tree structure $\boldsymbol{\tau}_O$ and the generalized forces $\boldsymbol{\tau}_G$ satisfy the following equation[10]:

$$\boldsymbol{\tau}_G^T \delta\boldsymbol{\theta}_G = \boldsymbol{\tau}_O^T \delta\boldsymbol{\theta}_O = \boldsymbol{\tau}_O^T \boldsymbol{W} \delta\boldsymbol{\theta}_G \tag{3}$$

where

$$\boldsymbol{W} \triangleq \frac{\partial \boldsymbol{\theta}_O}{\partial \boldsymbol{\theta}_G} \in \boldsymbol{R}^{N_O \times N_F} \tag{4}$$

$\delta\boldsymbol{\theta}_O$ and $\delta\boldsymbol{\theta}_G$ are the virtual displacements of $\boldsymbol{\theta}_O$ and $\boldsymbol{\theta}_G$, respectively. Similarly, Eq.(2) and the principle of virtual work yield

$$\boldsymbol{\tau}_G^T \delta\boldsymbol{\theta}_G = \boldsymbol{\tau}_A^T \delta\boldsymbol{\theta}_A = \boldsymbol{\tau}_A^T \boldsymbol{S} \delta\boldsymbol{\theta}_G \tag{5}$$

where

$$\boldsymbol{S} \triangleq \frac{\partial \boldsymbol{\theta}_A}{\partial \boldsymbol{\theta}_G} \in \boldsymbol{R}^{N_A \times N_F} \tag{6}$$

$\delta\boldsymbol{\theta}_A$ is the virtual displacement of $\boldsymbol{\theta}_A$. Since Eqs.(3) and (5) hold for any $\delta\boldsymbol{\theta}_G$, we have the following equations:

$$\boldsymbol{\tau}_G = \boldsymbol{W}^T \boldsymbol{\tau}_O \tag{7}$$

$$\boldsymbol{\tau}_G = \boldsymbol{S}^T \boldsymbol{\tau}_A \tag{8}$$

We can compute the actuator torque $\boldsymbol{\tau}_A$ from those of the tree structure $\boldsymbol{\tau}_O$ through the generalized force $\boldsymbol{\tau}_G$.

Nakamura et al.[10] did not use the generalized coordinates explicitly assuming that $\delta\boldsymbol{\theta}_G$ is taken as a subspace of $\delta\boldsymbol{\theta}_A$. As explained above, introducing the generalized coordinates eliminates unnecessary assumptions and restrictions on virtual cut joints, and on the placement of actuated joints.

The inverse dynamics computation of general closed kinematic chains consists of the following four steps:

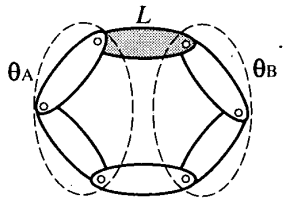

Figure 4: Closed loop

(1) Compute $\boldsymbol{W}$ and $\boldsymbol{S}$

(2) Compute $\boldsymbol{\tau}_O$ by inverse dynamics computation for the tree structure

(3) Compute $\boldsymbol{\tau}_G$ by Eq.(7)

(4) Compute $\boldsymbol{\tau}_A$ by solving the linear equation (8)

If the mechanism does not have actuation redundancy, namely, if the number of actuators equals to the degree of freedom, $\boldsymbol{S}$ becomes a square matrix. Thus, $\boldsymbol{\tau}_A$ is computed by

$$\boldsymbol{\tau}_A = \boldsymbol{S}^{-T} \boldsymbol{W}^T \boldsymbol{\tau}_O \tag{9}$$

Otherwise $\boldsymbol{\tau}_A$ is not determined uniquely, and some optimization method should be applied. Refer to [16] for methods of optimizing actuation redundancy.

2.2 Computation of W and S

For many practical planar closed kinematic chains, $\boldsymbol{W}$ and $\boldsymbol{S}$ become constant and can be formed from visual inspection. It is also known that they are computed relatively easily for some special closed kinematic chains such as parallel mechanisms. In this subsection we provide a general method for computing the two matrices.

Consider a loop illustrated in Fig.4. The linear and angular velocities of the shadowed link L is computed from $\dot{\boldsymbol{\theta}}_A$ as well as $\dot{\boldsymbol{\theta}}_B$ by multiplying the Jacobian matrices $\boldsymbol{J}_A$ and $\boldsymbol{J}_B$ of the position and orientation of link L with respect to $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$, respectively. The closed loop imposes the constraint that the velocity of link L computed from $\boldsymbol{\theta}_A$ should be equal to that from $\boldsymbol{\theta}_B$, namely,

$$\begin{pmatrix} \boldsymbol{J}_A & -\boldsymbol{J}_B \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\theta}}_A \\ \dot{\boldsymbol{\theta}}_B \end{pmatrix} = \boldsymbol{O} \tag{10}$$

Extending the discussion to the whole mechanism, the constraint due to the ith closed loop is written in the form

$$\boldsymbol{J}_{Li} \dot{\boldsymbol{\theta}}_J = \boldsymbol{O} \tag{11}$$

where $\boldsymbol{J}_{Li}$ is a 6 by N_J matrix. The columns of $\boldsymbol{J}_{Li}$ consist of those of the Jacobian matrices of link L with respect to the joint angles, which can be calculated in the same way as serial kinematic chains[17].

Let N_L be the number of independent closed loop in the structure. Then we get N_L constraint matrices $\boldsymbol{J}_{Li}(i = 1, 2, \ldots, N_L)$ which forms the matrix $\boldsymbol{J}_C \in \boldsymbol{R}^{6N_L \times N_J}$ as

$$\boldsymbol{J}_C \triangleq \begin{pmatrix} \boldsymbol{J}_{L1} \\ \boldsymbol{J}_{L2} \\ \vdots \\ \boldsymbol{J}_{LN_L} \end{pmatrix} \tag{12}$$

Although $\boldsymbol{J}_C$ represents all the kinematic constraints in the mechanism, not all the rows in $\boldsymbol{J}_C$ are independent, namely, $\boldsymbol{J}_C$ is not always full rank. We extract linearly independent rows from $\boldsymbol{J}_C$ and form $\boldsymbol{J}_{Cm} \in \boldsymbol{R}^{m \times N_J}$ where m is the rank of $\boldsymbol{J}_C$. From Eq.(11), $\boldsymbol{J}_{Cm}$ satisfies:

$$\boldsymbol{J}_{Cm}\dot{\boldsymbol{\theta}}_J = \boldsymbol{O} \tag{13}$$

Equation (13) represents the m independent constraints of the closed loops. Since we have N_J joints under m constraints, the remaining degrees of freedom (mobility) of the whole mechanism N_F becomes

$$N_F = N_J - m \tag{14}$$

Now we form $\boldsymbol{J}_S$ by extracting m independent columns from $\boldsymbol{J}_{Cm}$, and $\boldsymbol{J}_G$ by gathering the remaining columns. Also divide $\boldsymbol{\theta}_J$ into $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_G$ according to the division of $\boldsymbol{J}_{Cm}$. From Eq.(13), $\boldsymbol{J}_S$, $\boldsymbol{J}_G$, $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_G$ satisfy the equation

$$\boldsymbol{J}_{Cm}\dot{\boldsymbol{\theta}}_J = \begin{pmatrix} \boldsymbol{J}_S & \boldsymbol{J}_G \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\theta}}_S \\ \dot{\boldsymbol{\theta}}_G \end{pmatrix} = \boldsymbol{O} \tag{15}$$

Equivalently,

$$\boldsymbol{J}_S\dot{\boldsymbol{\theta}}_S = -\boldsymbol{J}_G\dot{\boldsymbol{\theta}}_G \tag{16}$$

Since $\boldsymbol{J}_S$ is always invertible, $\dot{\boldsymbol{\theta}}_S$ is uniquely determined by

$$\dot{\boldsymbol{\theta}}_S = \boldsymbol{H}\dot{\boldsymbol{\theta}}_G \tag{17}$$

$$\boldsymbol{H} \triangleq \frac{\partial \boldsymbol{\theta}_S}{\partial \boldsymbol{\theta}_G} = -\boldsymbol{J}_S^{-1}\boldsymbol{J}_G \tag{18}$$

Equation (17) implies that we can choose $\boldsymbol{\theta}_G$ as the generalized coordinates. It is worth pointing out that the generalized coordinate is automatically selected through the process of forming $\boldsymbol{J}_S$.

The Jacobian matrices $\boldsymbol{W}$ and $\boldsymbol{S}$ are formed from $\boldsymbol{H}$ immediately as follows:

- $\boldsymbol{W}$: If the ith joint of $\boldsymbol{\theta}_O$ is not a member of the generalized coordinates and corresponds with the jth one of $\boldsymbol{\theta}_S$, then include the jth row of $\boldsymbol{H}$ as the ith row of $\boldsymbol{W}$. If it is a member of the generalized coordinates and corresponds with the jth joint of $\boldsymbol{\theta}_G$, then include a unit vector with jth element being 1 and others 0 as the ith row of $\boldsymbol{W}$. This procedure is shown in Fig.5.
- $\boldsymbol{S}$: If the ith joint of $\boldsymbol{\theta}_A$ is not a member of the generalized coordinates and corresponds with the jth one of $\boldsymbol{\theta}_S$, then include the jth row of $\boldsymbol{H}$ as the ith row of $\boldsymbol{S}$. If it is a member of the generalized coordinates and corresponds with the jth joint of $\boldsymbol{\theta}_G$, then include a unit vector with jth element being 1 and others 0 as the ith row of $\boldsymbol{S}$. This procedure is shown in Fig.6

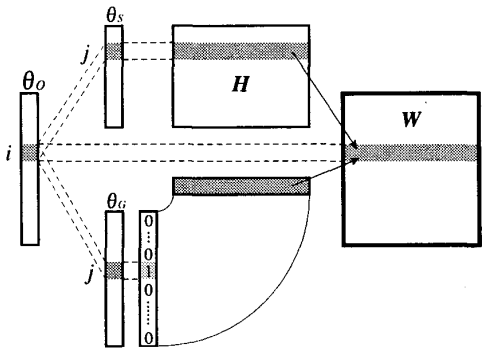

Figure 5: Forming **W** from **H**

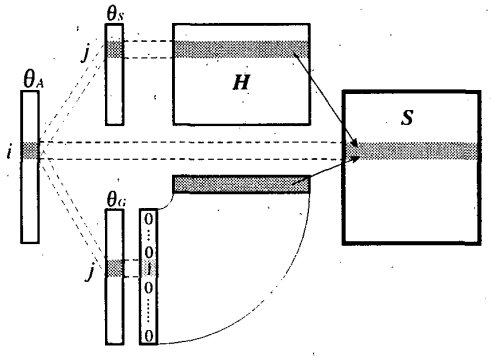

Figure 6: Forming **S** from **H**

2.3 Relationship of Accelerations

Differentiating Eq.(17) by time yields

$$\ddot{\boldsymbol{\theta}}_S = \boldsymbol{H}\ddot{\boldsymbol{\theta}}_G + \dot{\boldsymbol{H}}\dot{\boldsymbol{\theta}}_G \tag{19}$$

which calculates the acceleration of dependent joints $\ddot{\boldsymbol{\theta}}_S$ from generalized acceleration $\ddot{\boldsymbol{\theta}}_G$. This computation is required in forward dynamics computation. In this subsection, computation of the second term of the right-hand side of Eq.(19) is presented.

From Eq.(18) we have

$$\dot{\boldsymbol{H}}\dot{\boldsymbol{\theta}}_G = -\left\{\frac{d}{dt}(\boldsymbol{J}_S^{-1})\boldsymbol{J}_G + \boldsymbol{J}_S^{-1}\dot{\boldsymbol{J}}_G\right\}\dot{\boldsymbol{\theta}}_G \tag{20}$$

On the other hand, $\boldsymbol{J}_S^{-1}\boldsymbol{J}_S = \boldsymbol{E}$ yields

$$\frac{d}{dt}(\boldsymbol{J}_S^{-1})\boldsymbol{J}_S + \boldsymbol{J}_S^{-1}\dot{\boldsymbol{J}}_S = \boldsymbol{O} \tag{21}$$

Using Eqs.(16) and (21), Eq.(20) becomes

$$\begin{aligned}\dot{\boldsymbol{H}}\dot{\boldsymbol{\theta}}_G &= -\boldsymbol{J}_S^{-1}(\dot{\boldsymbol{J}}_S\dot{\boldsymbol{\theta}}_S + \dot{\boldsymbol{J}}_G\dot{\boldsymbol{\theta}}_G) \\ &= -\boldsymbol{J}_S^{-1}\dot{\boldsymbol{J}}_{Cm}\dot{\boldsymbol{\theta}}_J\end{aligned} \tag{22}$$

$\dot{\boldsymbol{J}}_{Cm}\dot{\boldsymbol{\theta}}_J$ is formed by extracting the elements of $\dot{\boldsymbol{J}}_C\dot{\boldsymbol{\theta}}_J$ corresponding to $\boldsymbol{J}_{Cm}$, where $\dot{\boldsymbol{J}}_C\dot{\boldsymbol{\theta}}_J$ is computed in the same algorithm as one for serial manipulators [15].

3 Inverse and Forward Dynamics of General Closed Kinematic Chains

3.1 Inverse Dynamics

The inverse dynamics of general closed kinematic chains is computed by the following steps. First, compute the Jacobian matrices $\boldsymbol{W}$ and $\boldsymbol{S}$ for the given configuration by the procedure explained in section 2.2. Next, compute the joint torques required to the virtual tree structure to generate the desired joint accelerations by applying inverse dynamics algorithms for open kinematic chains, and transform them into the generalized forces by Eq.(7). Finally, compute the actuator torques of the closed kinematic chain by Eq.(9) or by solving Eq.(8).

3.2 Forward Dynamics

Although several forward dynamics algorithms are known for open kinematic chains [1, 3, 5, 13], it is difficult to apply them to closed chains due to the complexity of their structure. The unit vector approach[13], however, can be extended to closed chains easily.

The equation of motion of closed kinematic chains is written in the same form as open chains as

$$\boldsymbol{\tau}_G = \boldsymbol{A}(\boldsymbol{\theta}_G)\ddot{\boldsymbol{\theta}}_G + \boldsymbol{b}(\boldsymbol{\theta}_G, \dot{\boldsymbol{\theta}}_G) \tag{23}$$

where $\boldsymbol{\tau}_G \in \boldsymbol{R}^{N_F}$ is the generalized force, $\boldsymbol{A} \in \boldsymbol{R}^{N_F \times N_F}$ is the inertia matrix and $\boldsymbol{b} \in \boldsymbol{R}^{N_F}$ includes the sum of centrifugal, Colliolis and gravity forces. In open kinematic chains, the joint angles are usually used as the generalized coordinate and thus the joint torques are the generalized force. Therefore, the accelerations of all joints are computed directly by Eq.(23). In closed kinematic chains, on the other hand, the joint torque vector and the generalized force may differ even in their dimensions. Additional computations of transformation of the joint torques into the generalized force and the generalized acceleration into the joint acceleration are required.

The forward dynamics algorithm based on the inverse dynamics algorithm explained in the previous subsection and the unit vector approach is summarized as follows:

(1) Transform the input joint torques $\boldsymbol{\tau}_A$ into the generalized force $\boldsymbol{\tau}_G$ by Eq.(8).

(2) Compute the inverse dynamics for the zero generalized acceleration and let the resultant generalized force be $\boldsymbol{b}$. Using Eq.(19), the accelerations of dependent joints $\ddot{\boldsymbol{\theta}}_S$ are given by $\dot{\boldsymbol{H}}\dot{\boldsymbol{\theta}}_G$, whose computation method is shown in section 2.3.

(3) Execute the following computation for $i = 1, 2, \ldots, N_F$:

  (a) Compute the inverse dynamics with $\ddot{\boldsymbol{\theta}}_G = \boldsymbol{e}_i$, where $\boldsymbol{e}_i \in \boldsymbol{R}^{N_F}$ is a unit vector whose ith element is 1 and others 0. The accelerations of dependent joints are computed by substituting $\boldsymbol{e}_i$ for $\ddot{\boldsymbol{\theta}}_G$ in Eq.(19).

  (b) Let the computed generalized force be $\boldsymbol{f}_i$ and calculate $\boldsymbol{a}_i$ by $\boldsymbol{a}_i = \boldsymbol{f}_i - \boldsymbol{b}$.

  (c) Include $\boldsymbol{a}_i$ as the i th column of $\boldsymbol{A}$.

(4) Using $\boldsymbol{\tau}_G, \boldsymbol{b}$ and $\boldsymbol{A}$, compute the generalized acceleration by

$$\ddot{\boldsymbol{\theta}}_G = \boldsymbol{A}^{-1}(\boldsymbol{\tau}_G - \boldsymbol{b}) \tag{24}$$

(5) Compute $\ddot{\boldsymbol{\theta}}_S$ by Eq.(19), where $\dot{\boldsymbol{H}}\dot{\boldsymbol{\theta}}_G$ is already computed in step 2, to get the accelerations of all joints.

This algorithm requires $O(Nn)$ computations, where N, n are number of links and degrees of freedom of a closed kinematic chain, respectively. More efficient $O(N)$ algorithms[4, 9], making use of sparsity of matrices, may be more efficient than ours for simple closed kinematic chains. But they fail to achieve that efficiency for extremely compliated closed kinematic chains. The complexity of our method, on the other hand, is always $O(Nn)$ for kinematic chains with any number of closed loops. Moreover, as the number of closed loops increases, the degrees of freedom n becomes smaller compared to the number of links N. In motion synthesis and generation, which is an interesting issue in controlling humanoid robots or human figures in computer graphics, we find another advantage of this approach since, through the forward dynamics computation, we can obtain the inertia matrix with respect to joint angles that allows us to inspect the physical consistency of a motion given as joint angle data or keyframes.

4 Example of Dynamics Simulation

Figure 8 shows an example of dynamics simulation of a closed kinematic chain, where a monkey holds a swing. The algorithm is implemented using Microsoft Visual C++, and runs on a PC with Pentium Pro 200MHz processor and an OpenGL graphic board. The human figure has 16 degrees of freedom (4 for each arm and leg) as illustrated in Fig.7, and the swing consists of six spherical joints, thus we have 34 degrees of freedom and two closed loops in total. Connections between the hands and the swing are represented by two virtual links, which are used to describe closed loops in our link connectivity description[12]. Therefore it is quite easy to cut both or one of the connections during the simulation by eliminating the virtual links, to make the character fall down. We applied zero torques except for the case when we need to restrict the joint angles within their limits. The sampling time for the forward dynamics is approximately 25msec.

5 Conclusions

The results obtained in this research is summarized by the following four items:

(1) We introduced the concept of the generalized coordinates of a closed kinematic chain, which allows a general formulation of the inverse dynamics computation of closed kinematic chains.

(2) We developed a systematic algorithm for computing the degrees of freedom and selecting the generalized coordinates of closed kinematic chains and compute the jacobian matrices for the inverse dynamics computation.

(3) Inverse and forward dynamics algorithms of general closed kinematic chains are established.

(4) The developed algorithms were implemented. An example of motion simulation of human figures verified their feasibility and compuational efficiency.

Acknowledgments

We would like to thank Mr. Fumio Nagashima, Fujitsu Laboratories Ltd. for offering the programming environment, Firstsight. This research was supported by the Advanced Software Enrichment Project of the Information-technology Promotion Agency (IPA), Japan, and Humanoid Robotics Project, MITI, Japan.

References

[1] E.J. Haug. *Computer Aided Kinematics and Dynamics of Mechanical Systems.* Allyn and Bacon Series in Engineering, 1989.

[2] J.F. Kleinfinger and W. Khalil. "Dynamic Modeling of Closed-Loop Robots". In *Proceedings of the 16th International Symposium on Industrial Robots*, pp.401–412, 1986.

[3] Farid M.L. Amirouche. *Computational Methods in Multibody Dynamics.* Prentice – Hall, 1992.

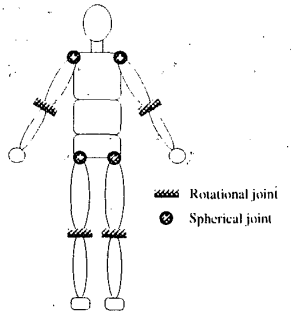

Figure 7: 16 d.o.f. human figure in the simulation

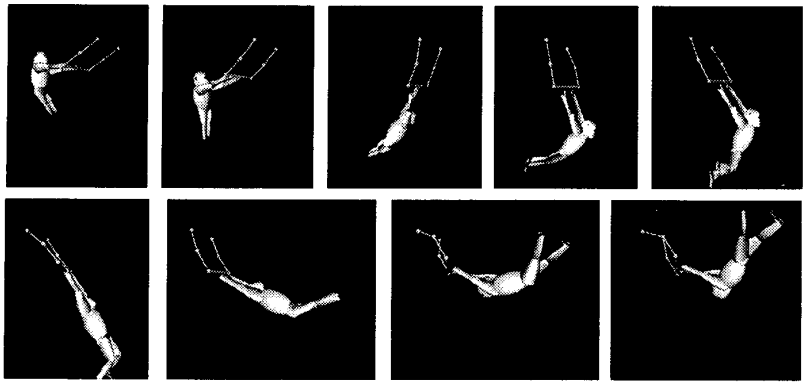

Figure 8: Example of dynamics simulation of a closed kinematic chain

[4] C. Lubich, U. Nowak, U. Pöhle, and C. Engstler. *MEXX - Numerical Software for the Integration of Constrained Mechanical Multibody Systems.* ftp://na.uni-tuebingen.de/pub/engstler/mexx/SC-92-12.ps.gz, 1992.

[5] R. Featherstone. *Robot Dynamics Algorithm.* Kluwer, 1987.

[6] B. Perrin, C. Chevallereau, and C. Verdier. "Calculation of the Direct Dynamics Model of Walking Robots: Comparison Between Two Methods". In *Proceedings of IEEE International Conference on Robotics and Autmation*, vol.2, pp.1088–1093, 1997.

[7] O. Bruneau and F.B. Ouezdou. "Dynamic Walk Simulation of Various Bipeds via Ankle Trajectory". In *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol.1, pp.58–63, 1998.

[8] D. Surla and M. Rackovic. "Closed-Form Mathematical Model of the Dynamics of Anthropomorphic Locomotion Robotic Mechanism". In *2nd ECPD International Conference on Advanced Robotics, Intelligent Automation and Active Systems*, pp.327–331, 1996.

[9] David Baraff. "Linear-Time Dynamics Using Lagrange Multipliers". In *Computer Graphics Proceedings, Annual Conference Series*, pp.137–146, 1996.

[10] Y. Nakamura and M. Ghodoussi. "Dynamics Computation of Closed-Link Robot Mechanisms with Nonredundant and Redundant Actuators". *IEEE Transactions on Robotics and Automation*, vol.5, no.3, pp.294–302, 1989.

[11] A. Codourey and E. Burdet. "A Body-Oriented Method for Finding a Linear Form of the Dynamic Equation of Fully Parallel Robots". In *Proceedings of IEEE International Conference on Robotics and Automation*, 1997.

[12] K. Yamane and Y. Nakamura. "Dynamics Computation of Structure-Varying Kinematic Chains for Motion Synthesis of Humanoid". In *Proceedings of IEEE International Conference on Robotics and Automation*, pp.714–721, 1999.

[13] M.W. Walker and D.E. Orin. "Efficient Dynamic Computer Simulation of Robot Manipulators". *ASME Journal on Dynamic Systems, Measurment and Control*, vol.104, pp.205–211, 1982.

[14] J.Y.S. Luh, M.W. Walker, and R.P.C. Paul. "On-line Computational Scheme for Mechanical Manipulators". *ASME Journal on Dynamic Systems, Measurment and Control*, vol.104, pp.69–76, 1980.

[15] Y. Nakamura, Y. Yokokohji, H. Hanafusa, and T. Yoshikawa. "Unified Recursive Formulation of Kinematics and Dynamics for Robot Manipulators". In *Proceedings of Japan-USA Symposium on Flexible Automation*, pp.53–60, 1986.

[16] Y. Nakamura and T. Ropponen. "Actuation Redundancy of a Closed Link Manipulator". In *Proceedings of 1990 American Control Conference*, pp.2294–2299, 1990.

[17] J.J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley, 1986.