Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Feed-Forward Control of Link Mechanisms under Various Boundary Conditions by Using a Parallel Solution Scheme

Daigoro Isobe
Inst. of Engineering Mechanics and Systems
University of Tsukuba
Tsukuba-shi, Ibaraki 305-8573, Japan
isobe@kz.tsukuba.ac.jp

Youichi Chikugo
Canon Inc.
Oota-ku, Tokyo 146-8501, Japan

Daisaku Imaizumi
NEC Co.
Minato-ku, Tokyo 108-8001, Japan

Shunsuke Sato and Atsushi Yagi
Graduate School
University of Tsukuba
Tsukuba-shi, Ibaraki 305-8573, Japan

Abstract – **In this paper, we describe a parallel solution scheme for inverse dynamics, and its application to feed-forward control of link mechanisms under various boundary conditions. The conditions include such cases as open- and closed-loops, and even one that continuously changes its form from an open- to a closed-loop. The dynamic equations conducted by generally used schemes such as the Newton-Euler method or the Lagrangian method, include interdependent variables between the constituting links which make it highly complicated to derive inverse dynamics of the closed-loop link mechanisms, or of the continuously transforming ones. The proposed scheme is developed by using the Finite Element Method (FEM), and evaluates the entire system as a continuum. The system is subdivided into finite elements, and the nodal forces are evaluated by equations of motion in a matrix form. The joint torque in the system is then calculated by converting the obtained nodal forces. Therefore, information from the entire system can be handled in parallel, which makes it seamless in application to open/closed-loop or continuously transforming mechanisms. The control results of link mechanisms under various boundary conditions reveal the possibility of using the proposed solution scheme for feed-forward control, independent of the system configuration of link mechanisms.**

I. INTRODUCTION

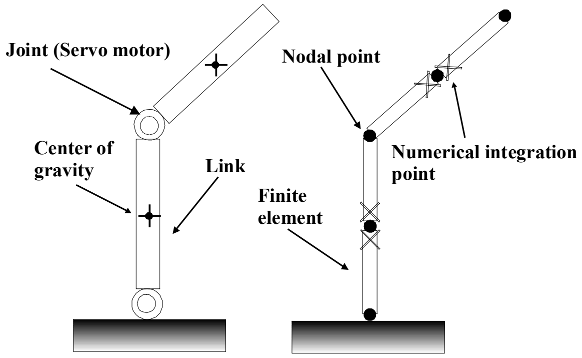
Dynamic equations conducted by generally used schemes such as the Newton-Euler method or the Lagrangian method, include interdependent variables between the constituting links which make it highly complicated to derive inverse dynamics of closed-loop link mechanisms, or of continuously transforming ones. Generally, robotic tasks include motions that generate open and closed loops alternately, and the dynamic equations of the system (or the numerical algorithm) require instant revision during the motion. Therefore, a unified solution scheme for calculating the inverse dynamics is strongly desired, particularly for massive, quick robots controlled by force.

Isobe and Nakagawa proposed the application of the Finite Element Method (FEM), a widely used computational tool for analyzing, for example, structures and fluids, to a control system of connected piezoelectric actuators, and achieved good control not only of the actuator itself but also of the entire system [1]. After finding that the FEM can be used as a control scheme of a continuum, Isobe et al. implemented the FEM in a two-dimensional solution scheme of inverse dynamics for open- [2] and closed-loop link mechanisms [3]. Taking advantage of the characteristic of the FEM, i.e., the capability of expressing the behavior of each discrete element as well as that of the entire continuous system, local information such as nodal forces and displacements can be calculated in parallel. The nodal forces are calculated incrementally in a matrix form, which does not require any revision of the outside frame, and the variables inside can be revised by simply changing the input data in the case of a physical change in the hardware system. The obtained nodal forces are then used to calculate the joint torque in the link systems.

In this paper, we describe a three-dimensional version of the parallel solution scheme for calculating inverse dynamics of link mechanisms. Link mechanisms are modeled using linear Timoshenko beam elements with a single integration point. Nodal forces for obtaining target trajectories are calculated using the FEM, and the joint torque of each link is calculated based on a matrix-formed conversion equation between nodal forces and the joint torque. Some numerical tests are carried out for several types of link mechanisms in order to verify the validity and flexibility of the scheme. The proposed scheme is also implemented in a control system to evaluate the performance under actual control with dynamics compensation, and some control experiments are carried out using a two-dimensional, nongear link mechanism, which can change its boundary condition in several ways.

II. PARALLELL SOLUTION SCHEME FOR N-LINK MECHANISM

A link mechanism constituted of a joint and a rigid link member, is modeled by using two Timoshenko beam elements with nodal points expressing the center of gravity and motors. Fig. 1 shows the general concept of the modeling. The total mass of two elements is concentrated at the nodal



(a)Link mechanism (b)Finite elements

Fig. 1 Modeling of link mechanism by finite elements

point expressing the center of gravity. The mass of a motor is lumped at the corresponding nodal points.

Fig. 2 shows the nodal forces (based on global coordinates) acting on the i-th link (i=1~n) in a three-dimensional open-loop n-link mechanism. The joint torque τ_{ix} required around the x-elemental axis on the i-th link, for example, is determined by adding an i+1-th joint torque $\tau_{(i+1)x}$ to the sum of inertia moments acting on this link, and is expressed by nodal forces F_{iy} and $F_{i\phi x}$ based on elemental (or link) coordinates as follows:

$$\tau_{ix} = l_{iC}F_{iy} + l_i(\sum_{j=i+1}^{n} F_j)_y + F_{i\phi x} + \tau_{(i+1)x}, \tag{1}$$

where l_{iC} is the length between the former joint and the center of gravity and l_i is the link length. By considering other components around the y- and z-axes, and arranging them into global coordinates (X, Y, Z) in a matrix form, the joint torque vector is expressed as

$$\{\tau^n\} = [L^n][T^n]\{P^n\}, \tag{2}$$

where $\{P^n\}$ is a vector related to nodal force, defined as

$$\{P^n\} = \left\{ \begin{array}{c} P_1 \\ P_2 \\ \cdot \\ \cdot \\ P_n \end{array} \right\}, \quad where \quad \{P_i\} = \left\{ \begin{array}{c} F_{iX} \\ F_{iY} \\ F_{iZ} \\ \sum_{j=i+1}^{n} F_{jX} \\ \sum_{j=i+1}^{n} F_{jY} \\ \sum_{j=i+1}^{n} F_{jZ} \\ F_{i\phi X} \\ F_{i\phi Y} \\ F_{i\phi Z} \end{array} \right\}. \tag{3}$$

The transformation matrix $[T^n]$ is expressed as

$$[T^n] = [h^n][T^n_{GE}], \tag{4}$$

where $[h^n]$ is a correction matrix between x-y and z-x coordinate systems, which simply inverts their signs in the y-axis direction. $[T^n_{GE}]$ is a transformation matrix between global and elemental coordinates which is expressed as

$$\left[\; T^n_{GE} \; \right] = \left[\begin{array}{ccccccc} T_1 & & & & & & \\ & T_2 & & & 0 & & \\ & & T_3 & & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & 0 & & & & \cdot & \\ & & & & & & T_n \end{array} \right], \tag{5}$$

where

$$\left[\; T_i \; \right] = \left[\begin{array}{ccc} A_i & 0 & 0 \\ 0 & A_i & 0 \\ 0 & 0 & A_i \end{array} \right], \tag{6a}$$

and

$$\left[\; A_i \; \right] = \left[\begin{array}{ccc} cos\phi_{iXx} & cos\phi_{iYx} & cos\phi_{iZx} \\ cos\phi_{iXy} & cos\phi_{iYy} & cos\phi_{iZy} \\ cos\phi_{iXz} & cos\phi_{iYz} & cos\phi_{iZz} \end{array} \right], \tag{6b}$$

where ϕ_{iXx}, for example, represents the rotational angle between X-global and x-elemental coordinates. $[L^n]$ is a matrix related to member length and is expressed as

$$[L^n] = [T^n_\Lambda][\Lambda^n], \tag{7}$$

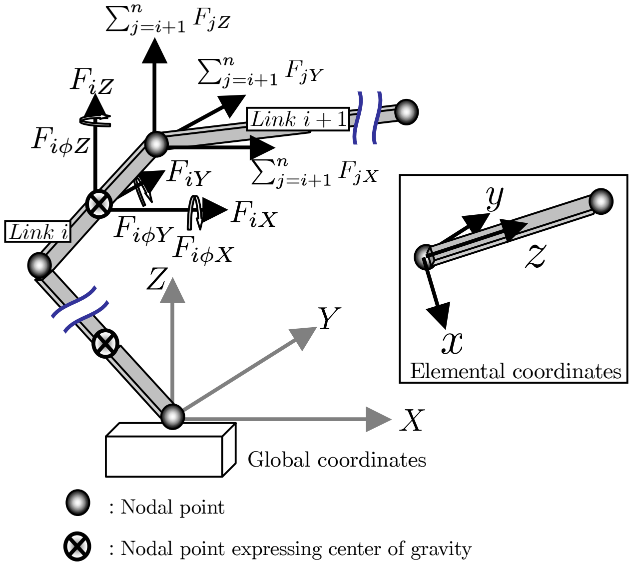


Fig. 2 Nodal forces acting on i -th link in an open-loop n-link mechanism

where [$T^n{}_\Lambda$] is a transformation matrix between each elemental coordinate, and is expressed as

$$\left[\ T^n_\Lambda\ \right] = \begin{bmatrix} T_{11} & T_{12} & T_{13} & \cdot & \cdot & \cdot & T_{1n} \\ & T_{22} & T_{23} & \cdot & \cdot & \cdot & T_{2n} \\ & & T_{33} & \cdot & \cdot & \cdot & T_{3n} \\ & & & \cdot & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot \\ & 0 & & & & \cdot & \cdot \\ & & & & & & T_{nn} \end{bmatrix}. \tag{8}$$

[T_{ij}] (i,j=1~n) is expressed using matrix [A_i] shown above:

$$[T_{ij}] = [A_i][A_j]^T. \tag{9}$$

[Λ^n] is expressed as

$$\left[\ \Lambda^n\ \right] = \begin{bmatrix} \Lambda_1 & & & & & & \\ & \Lambda_2 & & & 0 & & \\ & & \Lambda_3 & & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & 0 & & & & \cdot & \\ & & & & & & \Lambda_n \end{bmatrix}, \tag{10}$$

where

$$\left[\ \Lambda_i\ \right] = \begin{bmatrix} 0 & l_{iC} & 0 & 0 & l_i & 0 & 1 & 0 & 0 \\ l_{iC} & 0 & 0 & l_i & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

Information on the i+1~n link is summed by multiplying the [L^n] matrix by vector [T^n]{P^n}, which is the nodal force vector transformed into elemental coordinates. In cases of closed-loop link mechanisms, the above matrix is divided into multiple parts, as shown below, to fix the configuration of passive joints as well as the torque allocation undertaken by active joints.

$$[L^n] = \begin{bmatrix} L^a & 0 \\ 0 & L^b \end{bmatrix} \tag{12}$$

The suffixes a and b are the numbers of links (a+b=n) when the mechanism is divided into two parts. This is the only process that is different between the algorithms of open- and closed-loop link mechanisms, which of course, can be automatically alternated in the program. A vector related to incremental nodal forces acting on the i-th link is defined using the nodal numbers k (=2i):

$$\{\Delta p_k\} = \left\{ \begin{array}{c} \Delta f_{kX} \\ \Delta f_{kY} \\ \Delta f_{kZ} \\ \sum_{h=k+1}^{2n+1} \Delta f_{hX} \\ \sum_{h=k+1}^{2n+1} \Delta f_{hY} \\ \sum_{h=k+1}^{2n+1} \Delta f_{hZ} \\ \Delta f_{k\phi X} \\ \Delta f_{k\phi Y} \\ \Delta f_{k\phi Z} \end{array} \right\}. \tag{13}$$

Thus, the vector related to the nodal force acting on the i-th link at $t+\Delta t$ is successively calculated using the above vector as follows:

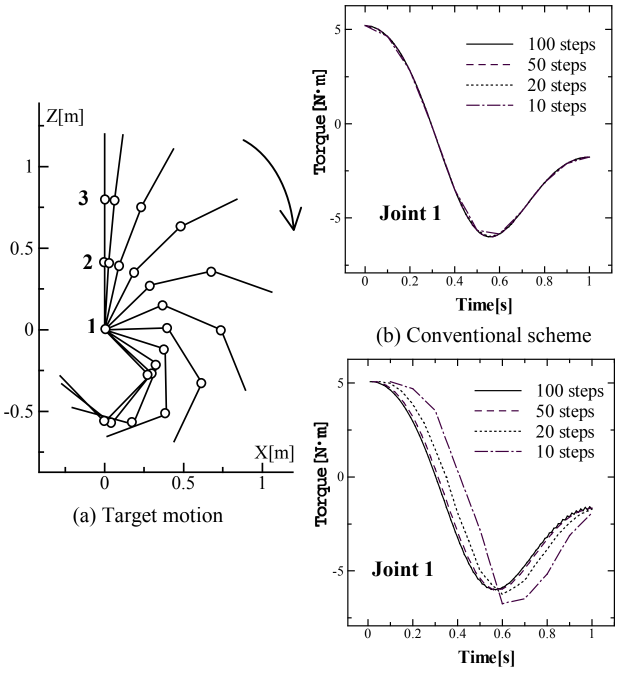
$$\{P_i\}_{t+\Delta t} = \{P_i\}_t + \{\Delta p_k\}. \tag{14}$$

The successive values of the n-link joint torque are then obtained by substituting (14) into (2). Newmark's β method (β=1/4) is used as the time integration scheme to solve the incremental kinematic equation [4].

III. NUMERICAL EXAMPLES

The parallel solution scheme is applied to the joint torque calculation of an in-plane open-loop mechanism as an example. In order to confirm the accuracy of the scheme, we compared the calculated torque curves with those obtained by the Newton-Euler method. Fig. 3(a) shows the target trajectory for a 1.0 s motion given in vertical plane for a three-link mechanism (length of each link: 40 cm; weight: 215 g; center of gravity at midpoint). Figs. 3(b) and 3(c) show the torque curves obtained by using the conventional and the proposed schemes. As shown in Fig. 3(b), the accuracy of the torque values does not depend upon the number of incremental steps when we use the dynamic equations that supply exact solutions. In contrast, the proposed scheme depends upon the number of incremental steps (see Fig. 3(c)), since the torque values are calculated approximately by summing the incremental information of each step. However, the results agree well if a sufficient number of steps are chosen. The influence of incremental steps is discussed later in this paper.

Although it needs special attention on the number of incremental steps, we can obtain torque curves for a closed-loop link mechanism (see Fig. 4(a)) or even those for a continuously transforming mechanism (see Fig. 4(b)) without revising any part of the numerical algorithm in the parallel solution scheme. This is one of the biggest merits of using the scheme.



(a) Target motion

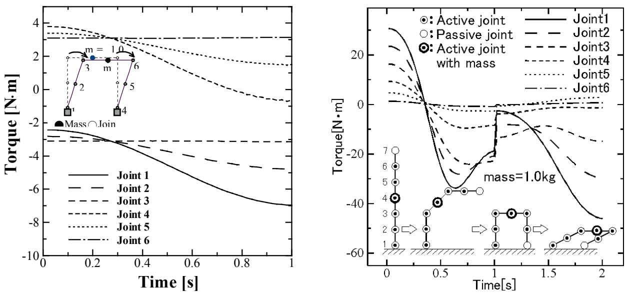
(b) Conventional scheme

(c) Proposed scheme

Fig. 3 Accuracy of torque curves against number of steps

IV. CONTROL EXPERIMENTS

As mentioned before, the scheme yields an approximate solution by summing the incremental information of each step. Therefore, this slight difference in the calculation procedures must be investigated to enable application to a control system. The proposed parallel solution scheme is implemented into a control system as shown in Fig. 5, to investigate its performance when applied in actual control. The time possession of each process in the control procedure is shown in Fig. 6. There are two main calculation processes during the control. One is the calculation of inverse dynamics, which may differ, of course, between the conventional and the proposed schemes. The other is the calculation of control and output function, which is a process common to both schemes. The calculation time of inverse dynamics for each step is about three times longer than that when using the conventional dynamic equations. However, the process time is sufficiently short compared to the time of the entire process as shown in Fig. 6, when, for example, the sampling time is selected to be 10 ms.



(a) Closed-loop

(b) Transforming type

Fig. 4 Torque curves for various types of link mechanisms [3]

A link mechanism with no gear shaft attached to the motors is used in the experiments to maximize the effect of the dynamics. Input torque τ is calculated using

$$\tau = J\dot{q}_d + Dq_d + f_c + \tau_{I.D.} + \tau_{feedback}, \qquad (15)$$

where J is the inertial moment of motors, D is the viscosity coefficient of motors, f_c is the dynamic friction force, $\tau_{I.D.}$ is the input torque calculated by FEM, and q_d and $\dot{q}_d$ are the target angle and angular velocity, respectively. The viscosity coefficient of motors D and the dynamic friction force f_c are

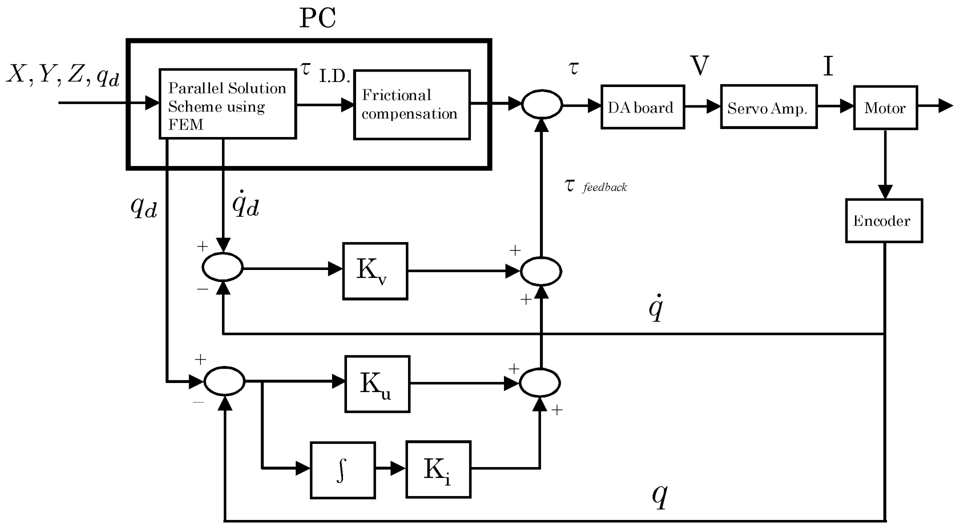


Fig. 5 Outline of the control system with parallel solution scheme

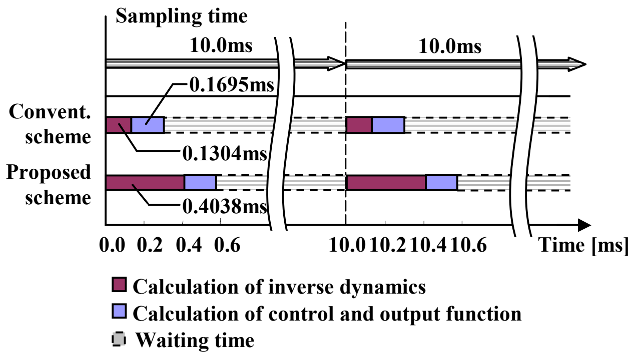


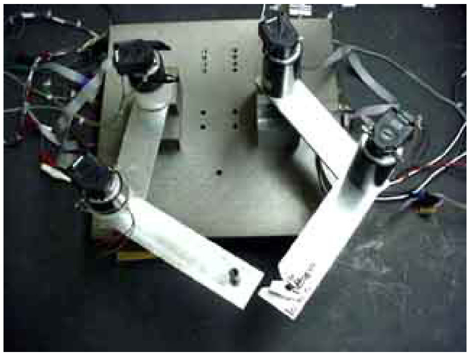
Fig. 6 Time possession of each process

identified beforehand by simple experiments. $\tau_{feedback}$ is the PID feedback torque which is obtained using

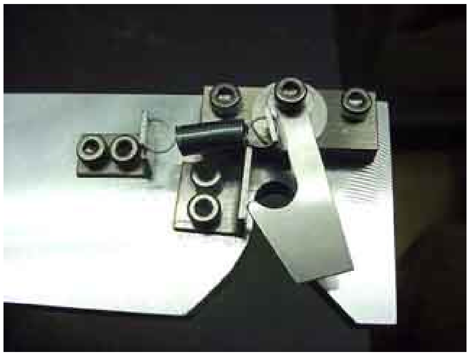
$$\tau_{feedback} = K_u(q_d - q) + K_i \Sigma(q_d - q) + K_v(\dot{q}_d - \dot{q}), \quad (16)$$

where q and $\dot{q}$ are the actual angle and angular velocity acquired from the attached encoders, respectively. K_u, K_i and K_v are the feedback gain for the angle, the integrated value and the angular velocity, respectively.

A link mechanism with a constraint device, as shown in Fig. 7, is used in the control experiment. PID feedback with feed-forward control using calculated inverse dynamics is applied. A target motion is given as shown in Fig. 8. Two open-loop link mechanisms are operated independently until 3.0 s, when both arms are then connected by the constraint device and operate as one closed-loop link mechanism from that time on. Fig. 9 shows the joint torque curves for the continuous transformation, calculated by the proposed scheme during the control. Surplus forces are intentionally generated in the manipulator on the right-hand side during 2.3~2.6 s, to enable a smooth connection. Therefore, the joint torque values of Joint 3 and Joint 4 slightly increase during that time period.

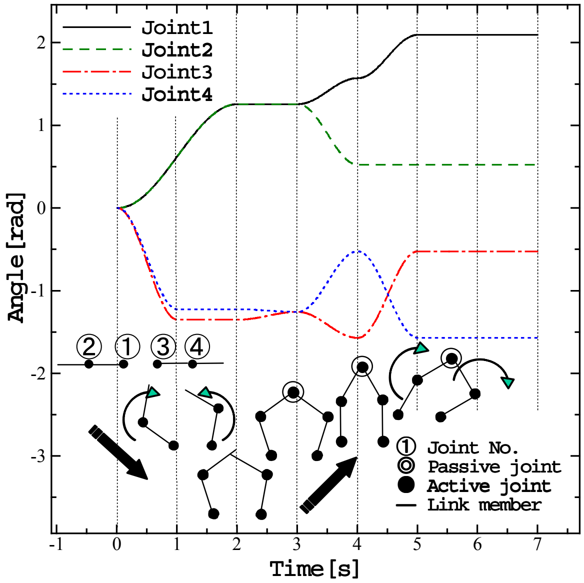


(a) General outline

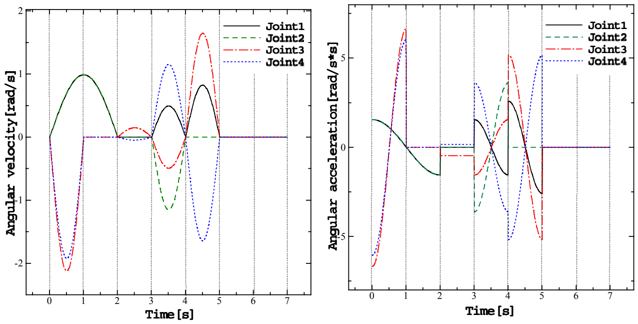


(b) Constraint device

Fig. 7 Link mechanism used in experiments



(a) Angle



(b) Angular velocity

(c) Angular acceleration

Fig. 8 Target motion

Fig. 10 shows the control results for each joint. Except for a short time period during the connection stage, where surplus forces are acting in order for the constraint device to work smoothly, it is evident that the control gives good convergence against the target motion. Although the proposed scheme requires incremental and successive calculations in the algorithm, the result of the control experiment clearly shows that the performance of a control

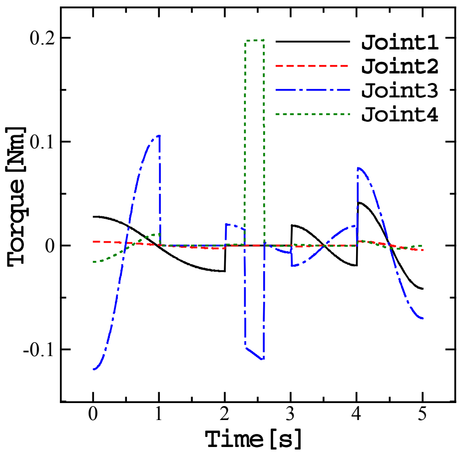


Fig. 9 Joint torque curves

system with this scheme presents no problem in actual use.

V. CONCLUDING REMARKS

The proposed solution scheme derives nodal forces in parallel and converts them to the joint torque, which can conveniently be applied to many types of link mechanisms under various boundary conditions. No revision of the basic numerical algorithm is required during the transformation process of the mechanisms. This function cannot be realized by using the conventional schemes based upon the generally used dynamic equations. It may achieve stability and smoothness in continuous motions of robotic architecture. The control results of link mechanisms under various boundary conditions reveal the possibility of using the proposed solution scheme for feed-forward control, independent of the system configuration of link mechanisms. Application of the scheme to flexible manipulators is scheduled.

VI. ACKNOWLEDGEMENT

This research is partially supported by the Grant-in-Aid for Encouragement of Young Scientists (No.12750176) of Japan Society for the Promotion of Science (JSPS).

VII. REFERENCES

[1] D. Isobe and H. Nakagawa, A Parallel Control System for Continuous Architecture Using Finite Element Method, *J. Intel. Mat. Syst. and Structures, 9*(12), 1998, pp.1038-1045.

[2] D.Isobe, H.Takeuchi and T.Ueda, A Finite-Element Approach to Control Link Mechanisms: Its Concept and Basic Simulation, *Proc. of the Int. Conf. on Comp. Eng. Science ICES 2000*, Los Angeles, 2000, pp.1648-1653.

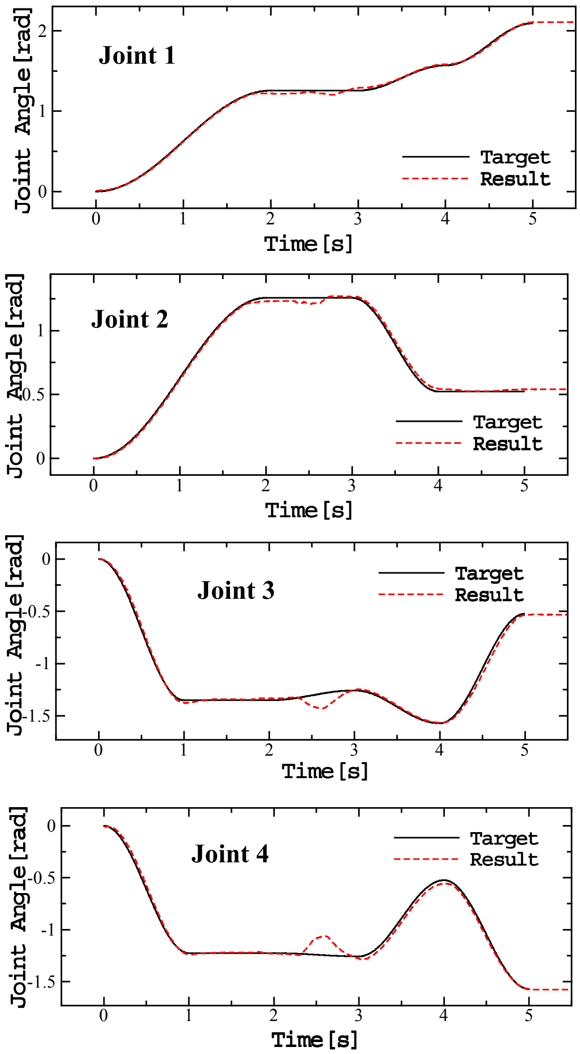


Fig. 10 Control results

[3]D. Isobe, A Unified Numerical Scheme for Calculating Inverse Dynamics of Open/Closed Link Mechanisms, *Proc. 27th Annual Conf. IEEE Industrial Electronics Society IECON'01,* Denver, 2001, pp.341-344.

[4]D. Isobe, Deriving Inverse Dynamics for Link Mechanisms by Using Finite Element Method, *Journal of the Robotics Society of Japan*, *20*(6), 2002, pp.647-653, in Japanese.