Pergamon

*Robotics & Computer-Integrated Manufacturing*, Vol. 13, No. 3, pp. 203–215, 1997
© 1997 Elsevier Science Ltd. All rights reserved
Printed in Great Britain
0736–5845/97 $17.00+0.00

PII:S0736–5845(97)00003–3

● *Paper*

# THE PATH PRECOMPENSATION METHOD FOR FLEXIBLE ARM ROBOT

JIH-HUA CHIN and SHIN-TYI LIN

Department of Mechanical Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China

**This paper constructed a closed loop path precompensation method for a flexible arm robot. A torque computation method taking care of the elastic arm deformation was first proposed and discussed. A concept of partial deformation compensation was subsequently proposed to improve the torque profiles and the trajectory fidelity. The advantage of this concept was first shown by examples of planar trajectory. After the construction of the closed-loop path precompensation method for a flexible arm, the torque method and partial deformation compensation were incorporated to track the spatial trajectory. Numerical simulations were given to show the usefulness of the proposed concept and method. © 1997 Elsevier Science Ltd.**

**Keywords—tracking; flexible arm; precompensation**

## 1. INTRODUCTION

Robots are used to perform tasks such as spray painting, arc welding, laser cutting and deburring. The dynamics of a rigid robot is more predictable, but the accuracy depends on a rigid and sometimes massive design, which makes it slow and heavy. In contrast, a lightweight flexible arm robot may be faster and less expensive, but it is more likely to deform elastically, which reduces the end point accuracy. In order to track the trajectory with precision, the effects of elastic deformation will be taken into consideration for a flexible arm.

The dynamic modeling of the flexible arm has been investigated by many authors. The two most popular methods for flexible arm modeling are (1) the Lagrangian assumed modes method, and (2) the Lagrangian finite-element method. For example, Book[1] presented an efficient formulation based on the Lagrangian equation. Usoro *et al.* [2] presented a finite-element/Lagrangian method. A similar approach was used by Sunada.[3] King *et al.*[4] presented a fast and accurate simulation algorithm. Many authors presented different methods to control the flexible arm. Cannon and Schmitz[5] carried out mathematical modeling and the initial experiment to explain the end-point feedback control. Book[6] derived the linear dynamic model for a flexible arm. Book *et al.*[7] also introduced three control schemes based on feedback from the state variables. Ower[8] used a Lagrangian dynamic method to model the two-link flexible arm, and designed a multivariable control system based on classical methods.

In the above papers, different position control methods for the flexible arm are employed in stepwise moving from a certain point to a desired point, and the features of the trajectory do not enter into consideration. Asada *et al.*[9] proposed a special moving coordinate system and an efficient algorithm for computing the actuator torque. Examples of tracking trajectory were given. Their algorithm leads to undesired vibration due to its full scale deflection compensation. However, Asada's open-loop torque algorithm (Ref. 9) can be invaluable if incorporated in a closed-loop environment.

In a novel type of trajectory tracking control[10–13] the actual position is considered during the generation of trajectory command values, and it has been shown the trajectory precision is improved. Among these works Hasegawa and Mizutani[12] derived an approximate trajectory relation to offer a trajectory control technique. Sakaue and Sugimoto[13] proposed a real-time control algorithm for a straight line path, which compensated both the actual errors of position and orientation. Chin and Tsai[10] presented a path tracking algorithm for robotic continuous path machining. Chin and Lin[11] further proposed a cross-coupling precompensation method for continuous path tracking. Their papers[10,11] proposed control with the path precompensation method for tracking trajectory, but limited to use for the rigid robot or machine tool.

The purpose of this paper is to propose a path precompensation method for a flexible arm. A torque compensation method comparable to Asada's torque algorithm [9] was proposed and incorporated within

*Acknowledgements*—This work was supported by the National Science Council of the Republic of China under Grant Number NSC82-0422-E-009-330.

the proposed closed loop. A concept of partial compensation for elastic deformation was subsequently proposed, and finally the closed-loop path precompensation method for a flexible arm was established. Planar and spatial trajectories were given as examples to show the improvement obtained.

## 2. PATH PRECOMPENSATION METHOD FOR A PLANAR FLEXIBLE ARM SYSTEM

### 2.1. *Modeling for the flexible arm*

Figure 1 shows a flexible arm system with $n$ flexible links operating in the $(X,Y)$ plane. The approach to flexible arm modeling is based on the Lagrangian assumed mode method.[1] The dynamic model of a flexible arm is given (Ref. 9) as follows:

(a) The joint equation:

$$\sum_{j=1}^{n}(H_{ij}\ddot{\theta}_j + A_{ij}\dot{\theta}_j + G_{ij}) = f_{\theta_i}, (i = 1,2,...,n). \quad (1)$$

(b) The deflection equation:

$$\sum_{j=1}^{n}\sum_{r=1}^{m_i}\left\{\prod_{ijpr}(\ddot{q}_{jr}\cos\theta_{ij} + 2\dot{\theta}_j\dot{q}_{jr}\sin\theta_{ij}) + [k_{ipr}\delta_{ij} - \prod_{ijpr}(\dot{\theta}_j^2\cos\theta_{ij} - \ddot{\theta}_j\sin\theta_{ij})]q_{jr}\right\}$$
$$- \sum_{j=1}^{n} T_{ijp}(\ddot{\theta}_j\cos\theta_{ij} - \dot{\theta}_{ij}\sin\ \theta_{ij})$$
$$= f_{qip}(p = 1,2,...,m_i; i = 1,2,...,n). \quad (2)$$

In order to derive the generalized forces, we consider the virtual work done by the torques shown in Fig. 2:

$$\delta W_i = \tau_{i,i-1}\delta\theta_i + \tau_{i,i+1}\delta\beta_i.$$

The generalized forces are found to be

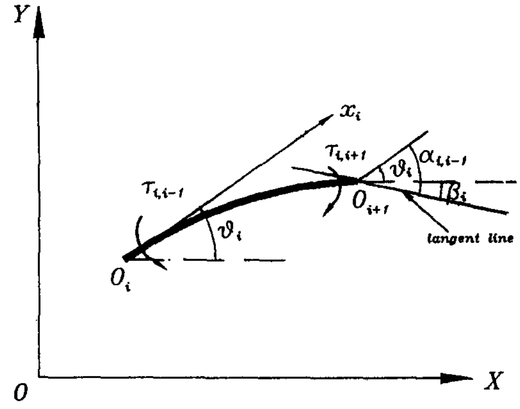

Fig. 2. Joint torque acting on link $i$.

$$f_{\theta i} = \tau_{i,i-1} - \tau_{i+1,i}, (i = 1,2,...,n-1)$$

$$f_{\theta n} = \tau_{n,n-1}$$

$$f_{qip} = \phi_{ip}'(L_i)\tau_{i+1,i}.$$

Asada *et al.*[9] showed that the equations are simplified significantly when represented in the virtual link coordinate system (VLCS) as shown in Fig. 3.

The relationship between the two joint displacements $\theta_i$ and $\hat{\theta}_i$ is

$$\theta_i = \hat{\theta}_i + \alpha_{i,i-1}.$$

Using the coordinate transformation, the dynamic equations can be rewritten as:

$$\sum_{r=1}^{m_i}[\hat{m}_{ipr}\hat{\ddot{q}}_{ir} + (\hat{k}_{ipr} - \hat{m}_{ipr}\hat{\dot{\theta}}_i^2)\hat{q}_{ir}]$$
$$- \sum_{j=1}^{i}\hat{T}_{ijp}(\hat{\ddot{\theta}}_j\cos\hat{\theta}_{ij} + \hat{\dot{\theta}}_j^2\sin\hat{\theta}_{ij})$$
$$= \hat{f}_{qip}(p = 1,2,...,m_i; i = 1,2,...,n) \quad (3)$$

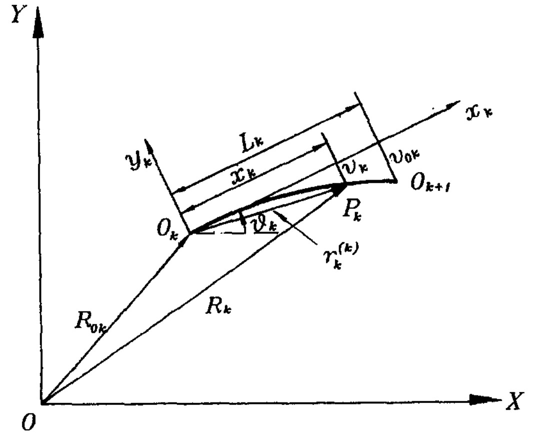

Fig. 1. Deformation of link $k$ represented in tangent coordinate.

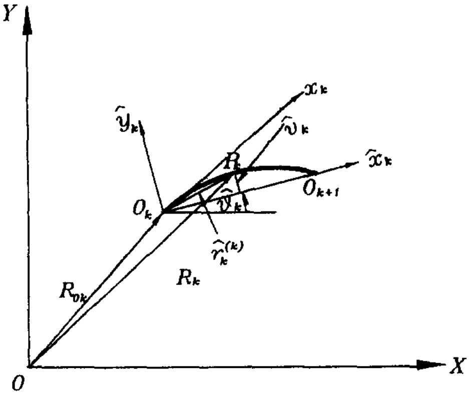

Fig. 3. Deformation of link $k$ represented in VLCS.

$$\sum_{j=1}^{n}(\hat{H}_{ij}\ddot{\hat{\theta}}_j + \hat{A}_{ij}\dot{\hat{\theta}}_j^2 + \hat{B}_{ij}\dot{\hat{\theta}}_j + \hat{G}_{ij}) = \hat{f}_{\theta i}(i = 1, 2, ..., n) \tag{4}$$

where

$$\hat{f}_{qip} = -\hat{\phi}_{ip}{}'(0)\tau_{i,i-1} + \hat{\phi}_{ip}{}'(L_i)\tau_{i+1,i}$$
$$(p = 1, 2, ..., m_i; i = 1, 2, ..., n-1)$$

$$\hat{f}_{qnp} = -\hat{\phi}_{np}{}'(0)\tau_{n,n-1}, (p = 1, 2, ...m_n)$$

$$\hat{f}_{\theta i} = \tau_{i,i-1} - \tau_{i+1,i}, (i = 1, 2, ..., n-1)$$

$$\hat{f}_{\theta n} = \tau_{n,n-1}$$

$$\hat{H}_{ij} = \sum_{p=1}^{m_i}\sum_{r=1}^{m_j}\hat{m}_{ipr}\hat{q}_{ip}\hat{q}_{jr}\delta_{ij}$$
$$+ \sum_{p=1}^{m_j}\hat{T}_{ijp}\hat{q}_{ip} \sin\hat{\theta}_{ij}$$

jir

$$- \sum_{r=1}^{m_j}\hat{T}_{\cancel{ij}}\hat{q}_{jr} \sin\hat{\theta}_{ij} + X_{ij}\cos\hat{\theta}_{ij}$$

$$\hat{A}_{ij} = \sum_{p=1}^{m_j}\hat{T}_{ijp}\hat{q}_{ip}\cos\hat{\theta}_{ij} - \sum_{r=1}^{m_j}\hat{T}_{jir}\hat{q}_{jr}\cos\hat{\theta}_{ij} + X_{ij}\sin\hat{\theta}_{ij}$$

$$\hat{B}_{ij} = 2[\sum_{p=1}^{m_i}\sum_{r=1}^{m_j}\hat{m}_{ipr}\hat{q}_{ip}\dot{\hat{q}}_{jr}\delta_{ij} - \sum_{r=1}^{m_j}\hat{T}_{jir}\dot{\hat{q}}_{jr}\sin\hat{\theta}_{ij}]$$

$$\hat{G}_{ij} = -\sum_{r=1}^{m_j}\hat{T}_{jir}\ddot{\hat{q}}_{jr}\cos\hat{\theta}_{ij}$$

$$\hat{\Pi}_{ijpr} = \hat{m}_{ipr}\delta_{ij}, \hat{T}_{ijpr} = \begin{cases} \hat{e}_{ip}L_j, & i > j \\ \hat{s}_{ip}, & i = j \\ 0, & i < j \end{cases}$$

$$\hat{s}_{ip} = \int_0^{l_i}\rho_i A_i \hat{x}_i \hat{\phi}_{ip} \mathrm{d}\hat{x}_i, \hat{e}_{ip} = \int_0^{l_i}\rho_i A_i \hat{\phi}_{ip}\mathrm{d}\hat{x}_i$$

$$\hat{m}_{ipr} = \hat{m}_{ip}\delta_{pr}, \hat{h}_{ipr} = \hat{k}_{ip}\delta_{pr}$$

$$\hat{m}_{ip} = \int_0^{l_i}\rho_i A_i(\hat{\phi}_{ip})^2 \, \mathrm{d}\hat{x}_i, \hat{k}_{ip} = \int_0^{l_i} E_i I_i\left(\frac{\mathrm{d}^2\hat{\phi}_{ip}}{\mathrm{d}\hat{x}_i^2}\right)^2 \mathrm{d}\hat{x}_i$$

Equations (3) and (4) are the dynamic equations for a flexible arm in the VLCS. These equations were first used by Asada *et al.*[9]

### 2.2. *Path precompensation method for rigid arm*

The concept of path precompensation for a robot was systematically proposed by Chin and Tsai.[10] Some similar early works could also be sporadically identified.[12,13]

The concept of the path precompensation method of any function path is shown in Fig. 4. The trajectory is represented by a spatial curve: $\mathbf{r}(u) = \mathbf{P} = x(u)\mathbf{i} + y(u)\mathbf{j} + z(u)\mathbf{k}$, $u \in [a, b]$.

The planned velocity is

$$V_t = V_b\mathbf{t}$$

where $\mathbf{t}$ is the unit tangent vector on the curve

$$\mathbf{t} = \frac{\mathrm{d}\mathbf{r}/\mathrm{d}u}{|\mathrm{d}\mathbf{r}/\mathrm{d}u|} = \frac{\mathbf{r}'}{|\mathbf{r}'|}.$$

The position error $E_r$ is defined as the shortest distance from the actual position $P_i(x_i,y_i,z_i)$ of the robot arm tip to a point $P(x,y,z)$ on the curve.

$$E_r(u) = P - P_i$$
$$= (x(u) - x_i)\mathbf{i} + (y(u) - y_i)\mathbf{j} + (z(u) - z_i)\mathbf{k}$$

If $E_r$ is minimum the differentiation of error $E_r$ must be zero, thus

$$(P - P_i)^T \bullet P' = 0.$$

$E_r$ can be obtained by solving the above equation. The control law for the path precompensation can be set as follows:

$$V = V_b\mathbf{t} + k_v E_r \tag{5}$$

where $k_v$ is a gain which depends on, for example, the sampling time of the controller, etc.

In order to reduce the steady-state error, an integral component can be added to the control law:

$$V = V_b\mathbf{t} + k_v E_r + k_i\int_0^t E_r \, \mathrm{d}t \tag{6}$$

where $k_i$ is the integral gain.

### 2.3. *Path precompensation method for flexible arm*

The flexible arm differs from the rigid arm by its

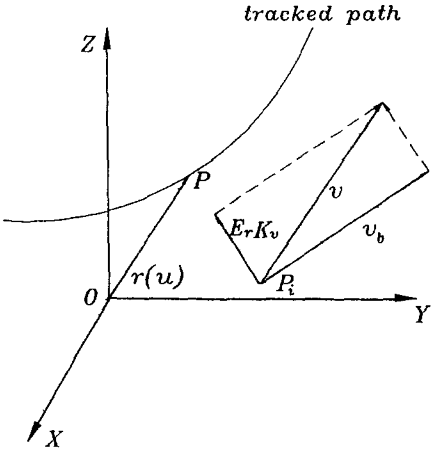

**Fig. 4.** Path precompensation method.

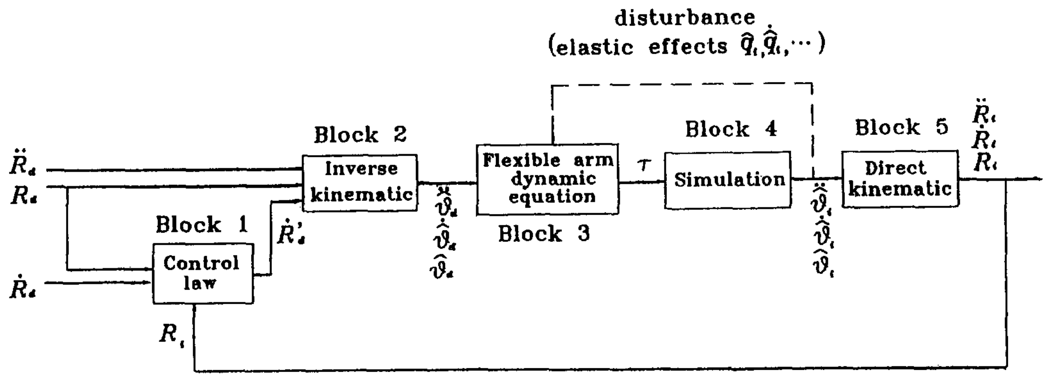

Fig. 5. Block diagram of path precompensation method for a flexible arm.

elastic effects. This can be clearly seen from the following rearranged dynamic equations.

$$\begin{bmatrix} M_{qq} & M_{\theta q} \\ M_{\theta q} & M_{\theta\theta} \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} G_q \\ G_\theta \end{bmatrix} = \begin{bmatrix} F_q \\ F_\theta \end{bmatrix} \tag{7}$$

where $q = [\hat{q}_{11}, \hat{q}_{12}, ..., \hat{g}_{1m}, ..., \hat{q}_{nm_n}]^T$ = coordinates for elastic behavior, $\theta = \left[\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_n\right]^T$ = coordinates for joints.

Many control ways for the flexible arm moving from a certain point to a desired point have been given.[7,8,14] We propose to guide the flexible arm tracking a desired trajectory in this paper. The computation for command value with vibration control is a high computational load for the control. Baruh and Tadikenda[15] discussed the advantage and disadvantage of different ways of treating the elastic effects. Since we are interested in the trajectory precision, so we concentrate on $\theta$ and treat the elastic deformation as a known disturbance, in this way the above equation can be rewritten as

$$\begin{aligned}(M_{\theta\theta} - M_{\theta q}M_{qq}^{-1}M_{\theta q})\ddot{\theta} + (G_\theta - M_{\theta q}M_{qq}^{-1}G_q) \\ = F_\theta - M_{\theta q}M_{qq}^{-1}F_q.\end{aligned} \tag{8}$$

Figure 5 shows the block diagram of the closed-loop path precompensation method proposed for the system of a flexible arm. In Fig. 5, block 1 executes the compensation according to Eqs (5) or (6). The driving torques are obtained in block 3. The computation algorithm for the driving torques taking care of the effects of the elastic deformation will be described in the following section.

### 2.4. *Open-loop torque computation method*

The flexible arm can be driven in an open-loop manner as shown in Fig. 6. Block 1 solves inverse kinematics to get the joint angles of the associated virtual links. Block 4 transforms the joint angles to the Cartesian space. Block 2 executes the computation algorithm for the actuator torques. The first step in block 2 is to plan the joint position, velocity and acceleration of the virtual link. Then the deformation of the actual link and the driving torques are computed. There are three approaches to calculating the required open-loop torques $\tau$ in Fig. 6 depending on how the effect of the elastic motion is considered.

2.4.1. *Method 1.* Elastic effects are neglected and the links are taken to be rigid. By planning $\hat{\theta}_i$ and setting $\hat{q}_{ip}$ equal to zero in Eq. (7), we can derive the following equation:

$$M_{\theta\theta}\ddot{\theta} + G_\theta = F_\theta$$

From the above equation, the approximate torques $\tau_i'$ can be obtained.

2.4.2. *Method 2.* Asada[9] proposed an algorithm for computing the required torques as follows. The

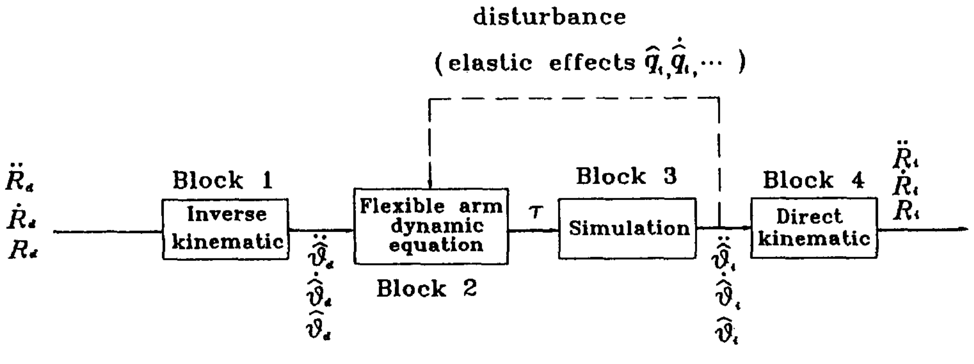
disturbance
(elastic effects $\hat{q}_t, \dot{\hat{q}}_t, \cdots$)
Block 1
Inverse kinematic
Flexible arm dynamic equation
Block 2
Block 3
Simulation
Block 4
Direct kinematic
$\tau$
$\ddot{R}_d$
$\dot{R}_d$
$R_d$
$\ddot{\hat{\vartheta}}_d$
$\dot{\hat{\vartheta}}_d$
$\hat{\vartheta}_d$
$\ddot{\hat{\vartheta}}_t$
$\dot{\hat{\vartheta}}_t$
$\hat{\vartheta}_t$
$\ddot{R}_t$
$\dot{R}_t$
$R_t$

Fig. 6. Block diagram of an open-loop control for a flexible arm.

elastic effects are considered in the computation for the driving torques to a certain extent. The steps for computation algorithm are as follows.

(1) The approximate actuator torque $\tau_i'$ obtained by Method 1 is applied to Eq. (7) to yield the generalized force $F_q$ for the deflection system. By rearranging Eq. (7) into the following form:

$$M_{qq}\ddot{q} + G_q = F_q - M_{\theta q}\ddot{\theta}(p = 1,2,3; i = 1,2)$$

the deflection acceleration $\hat{\ddot{q}}$ can be obtained.

(2) Substituting $\hat{\theta}_i$ and $\hat{q}_{ip}$ into Eq. (7), the driving torques for the flexible arm are found.

2.4.3. *Method 3.* Since $\hat{\ddot{q}}$ obtained by Asada *et al.*[9] are approximate quantities, we first proposed a way to obtain more accurate $\hat{\ddot{q}}$.

(1) Substituting $\hat{\theta}_i, \hat{q}, \hat{\dot{q}}$ into Eq. (8) to get the approximate torques $\tau'$.

(2) Substituting the approximate torques $\tau'$ into Eq. (8) to yield the deflection acceleration, and applying $\hat{\theta}$ and $\hat{q}$ into Eq. (7) to yield the actuator torques.

However, investigation shows that torques computed from Asada's method and the proposed Method 3 lead to arm oscillations as side effects. To suppress the undesired oscillation, we propose a concept of partial deformation compensation, that is, we modify the deflection acceleration $\hat{\ddot{q}}$ to $k_a\hat{\ddot{q}}$ and $k_f\hat{q}$, respectively, where $k_a$, $k_f$ are factors smaller than 1.

## 2.5. *Implementation on two-link flexible arm*

In this section, the path precompensation schemes are implemented on a two-link planar flexible arm, and the torque computation methods are studied.

2.5.1. *Dynamic equations for two-link flexible arm.* The dynamic equations for a two-link flexible arm can be established from Eqs (3) and (4) as follows:

$$\hat{m}_{ip}\hat{\ddot{q}}_{ip}+\hat{c}_{ip}\hat{\dot{q}}_{ip} + \hat{k}\hat{q}_{ip} - \sum_{j=1}^{i} \hat{T}_{ijr}\hat{\ddot{\theta}}_j \cos \hat{\theta}_{ij} \\ = \hat{f}_{ip}(p = 1,2,3; i = 1,2) \quad (10)$$

$$\sum_{j=1}^{2} X_{ij}\hat{\ddot{\theta}}_j \cos \hat{\theta}_{ij} - \sum_{j=1}^{2}\sum_{r=1}^{m_j} \hat{T}_{jir}\hat{\ddot{q}}_{jr} \cos \hat{\theta}_{ij} = \hat{f}_{\theta i}(i = 1,2) \quad (11)$$

where

$$\hat{f}_{ip} = -p\pi\tau_1/L_1 + (-1)^{p+1}p\pi\tau_2/L_1 \quad (12)$$

$$\hat{f}_{ip} = -p\pi\tau_2/L_2 \quad (13)$$

$$\hat{f}_{\theta_1} = \tau_1 - \tau_2 \quad (14)$$

$$\hat{f}_{\theta_2} = \tau_2 \quad (15)$$

$$\hat{m}_{ip} = \rho_i A_i L_i/2$$

$$\hat{k}_{ip} = E_i I_i(p\pi)^4/L_i^3/2$$

$$\hat{c}_{ip} = 2\mu_i\sqrt{\hat{m}_{ip}\hat{k}_{ip}}$$

$$\hat{T}_{ijp} = \begin{cases} 0, i < j \\ (-1)^{p+1}\rho_i A_i L_i^2/p\pi, i = j \\ 1 - (-1)^p \rho_i A_i L_i L_j/p\pi, i > j \end{cases}$$

$$X_{11} = (\rho_1 A_1 L_1/3 + \rho_2 A_2 L_2 + m_{L1} + m_{L2})L_1^2$$

$$X_{22} = (\rho_2 A_2 L_2/3 + m_{L2})L_2^2$$

$$X_{12} = X_{21} = (\rho_1 A_1 L_1/2 + \rho_2 A_2 L_2 + m_{L1} + m_{L2})L_1 L_2.$$

The two-link flexible arm is configured as follows. The flexible arm consists of rectangular steel beams 1 m in length. The thickness of link 1 and link 2 is 3.1 mm and 2.4 mm, respectively. The width is 20 mm for both links. The mass density $\rho$ is $7.8\times10^{-6}$ kg/mm$^3$, and Young's module $E$ is $2.1\times10^4$ kg/mm$^2$. The damping coefficients $\mu_1$ and $\mu_2$ are both 0.3.

2.5.2. *Open-loop torque computation.* In this section, open-loop torque computation methods are implemented for a two-link flexible arm. Figure 7 shows a desired trajectory to be tracked by the three open-loop torque computation methods. The flexible arm is reqduired to move from start point $P_1$(750,750) to end point $P_2$(800,800). Figure 8a and b show profiles of the actuator torques computed by Asada's method for various $k_a$, and Fig. 9 shows the tracking error for the respective torques. Figure 10a and b show the profiles error for the respective torques. Figure 10a and b show profiles of the actuator

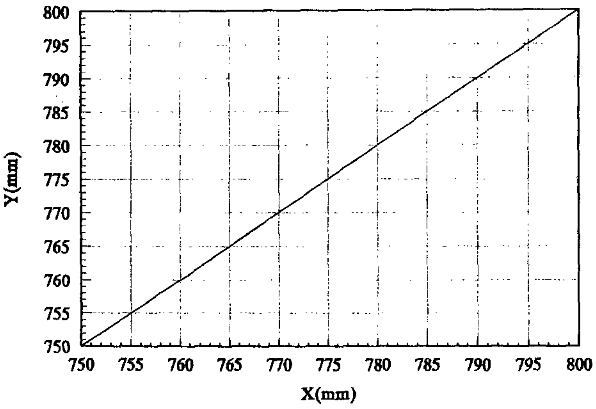

Fig. 7. Desired trajectory.

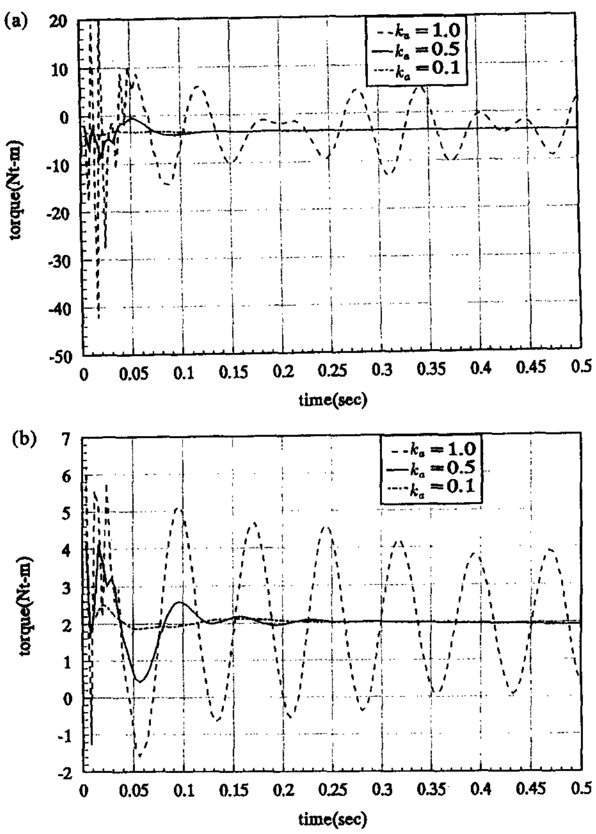

**Fig. 8.** (a) Computed torque $\tau_1$ for joint 1 by Asada's method. (b) Computed torque $\tau_2$ for joint 2 by Asada's method.

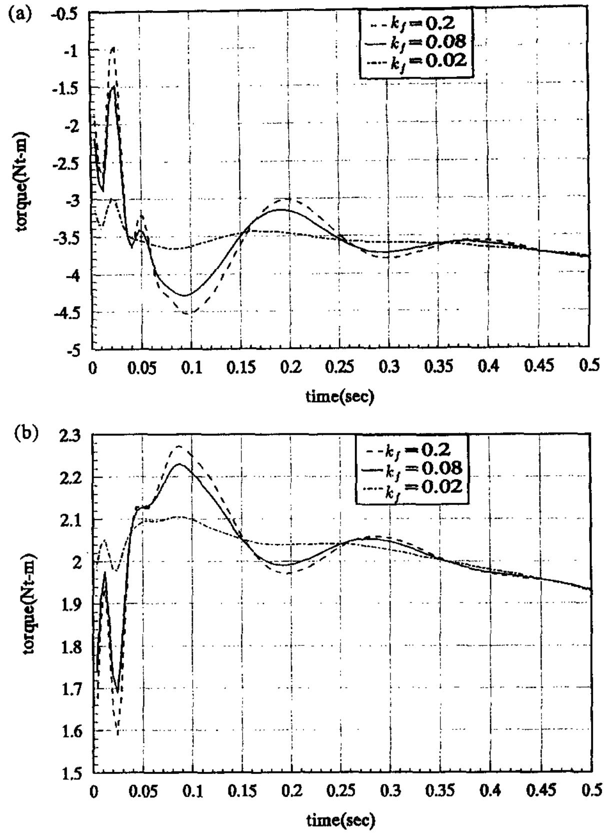

**Fig. 10.** (a) Computed torque $\tau_1$ for joint 1 by proposed method. (b) Computed torque $\tau_2$ for joint 2 by proposed method.

torques computed by the proposed method for various $k_f$, and Fig. 11 shows the tracking error for the respective torques.

Note that when $k_a$ and $k_f$ are 0, the computed torque by Asada's and the proposed method are the same as the torque computation by Method 1. It can be seen from Figs 8 and 10 that the torque profiles become more oscillatory for bigger $k_a$ and $k_f$. This means that the torque varies more versatilely to cope with the arm flexure. While $k_a$ is 1, the actuator torque is very large at the beginning of motion, and the elastic motion grows rapidly. Figures 8–11 indicate that the proposed partial consideration $k_a$ and $k_f$ helps to smooth the arm behavior without decreasing the trajectory fidelity.

Further investigation reveals that the proposed torque computation method generate $\hat{\ddot{q}}$ which might enlarge the torques $\tau_i$. While this can be avoided by using smaller $k_f$, the trajectory error might increase owing to too small $k_f$, because small $k_f$ means the arm is treated more like a rigid arm. The dilemma is that, in pursuing minimal elastic oscillation, the position error gradually grows. Before an optimum strategy could be found, we would say the proposed method is comparable to, not better than, Asada's method, and for the sake of comparison, Asada's method will be used in the following works. The CPU time of a PC with 486 CPU is 1.4 msec for Method 1, 2.1 msec for Method 2, 2.9 msec for Method 3, and sampling time is 4 msec.

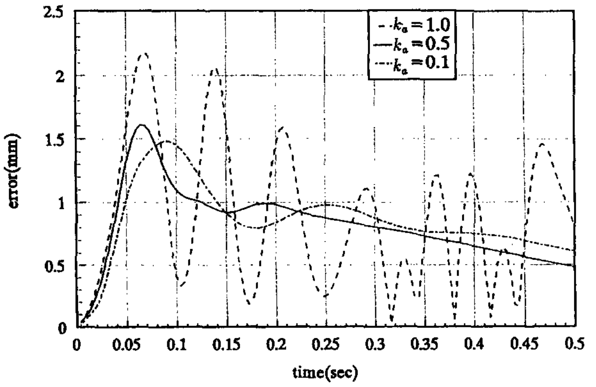

**Fig. 9.** Trajectory error for Asada's method with $\mu = 0.3$.

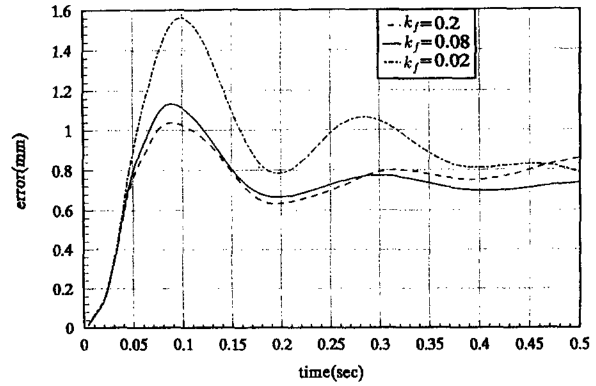

**Fig. 11.** Trajectory error for proposed method $\mu = 0.3$.

2.5.3. *Closed-loop path precompensation method for a flexible arm.* In this section, the flexible arm is guided by the path precompensation method to track three types of path.

2.5.4. *Tracking a circular path.* The flexible arm moves along a circle centered at (500,500) with radius 750 mm; the starting point is $P_1(1250,500)$.

Figure 12a shows the error history for tracking a circular path with the closed-loop precompensation method with different $k_v$. In the figure shown, bigger $k_v$ results in smaller errors. Figure 12b shows the comparisons of error history from different methods. It is seen that Asada's torque method ($k_a = 1$) is entangled with elastic oscillation, while the proposed partial deformation compensation ($k_a = 0.5$) diminishes the elastic oscillation. However, Asada's method, either $k_a = 1$ or $k_a = 0.5$, is subjected to divergence due to its open-loop nature. Satisfactory results are obtained by the proposed closed-loop path precompensation method.

2.5.5. *Tracking a straight path.* The flexible arm moves along a straight line as shown in Fig. 7 from initial point $P_1(750,750)$ to end point $P_2(830,830)$.

Figure 13 shows the error history for tracking a straight path with precompensation and open-loop torque computation. Again Asada's open-loop torque computation method leads to divergence, while the closed-loop path precompensation method with $k_a = 0.5$ yields satisfactory results.

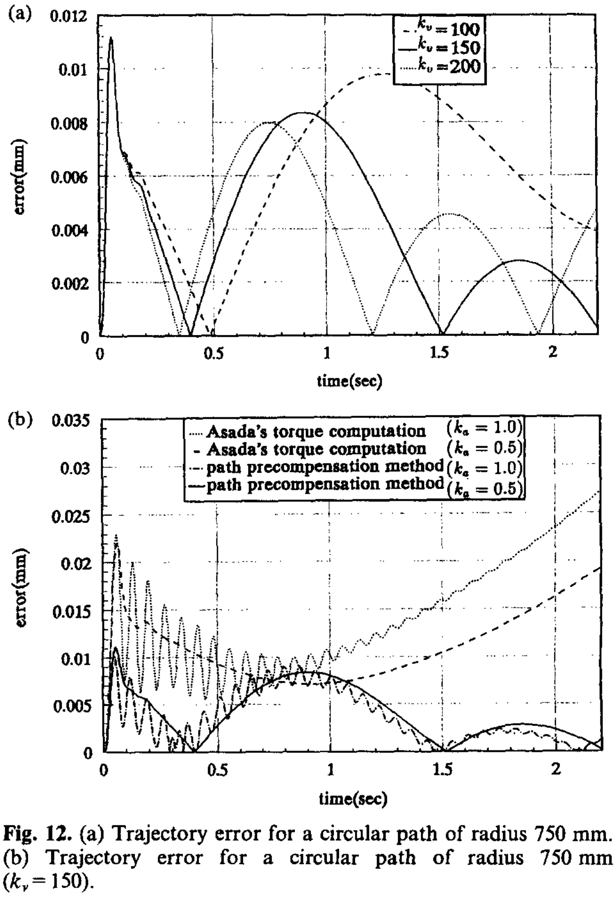

**Fig. 12.** (a) Trajectory error for a circular path of radius 750 mm. (b) Trajectory error for a circular path of radius 750 mm ($k_v = 150$).

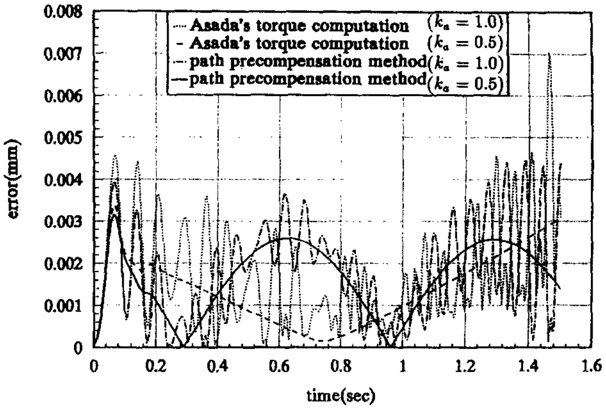

**Fig. 13.** Trajectory error for a straight path ($k_v = 150$).

Note that $k_a = 1$ excites drastic elastic vibrations which may jeopardize the stability of the robot. This fact speaks for the significance of the proposed $k_a$ and $k_f$.

2.5.6. *Tracking a parabolic path.* The flexible arm moves along a parabolic path as shown in Fig. 14 which is represented by

$$y = \frac{-1}{400}(x - 1080)^2 + 828.$$

The flexible arm starts at point $P_1(1160,812)$.

Figure 15 shows the error history for tracking the parabolic path with closed-loop path precompensation method and open-loop torque computation. Again the closed-loop precompensation method with $k_a = 0.5$ brings the best results.

From Figs 12, 13 and 15, we see that the proposed partial deformation compensation ($k_a = 0.5$) improves the results in all cases. But satisfactory results are obtained only by the closed-loop precompensation method.

Note that the path precompensation method controls the end point, not the oscillations. The elastic effects are treated as a known disturbance. Thus, the stability of the flexible arm system depends

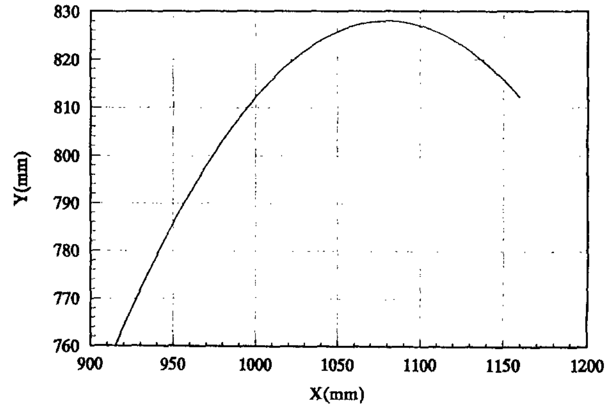

**Fig. 14.** Desired parabolic path.

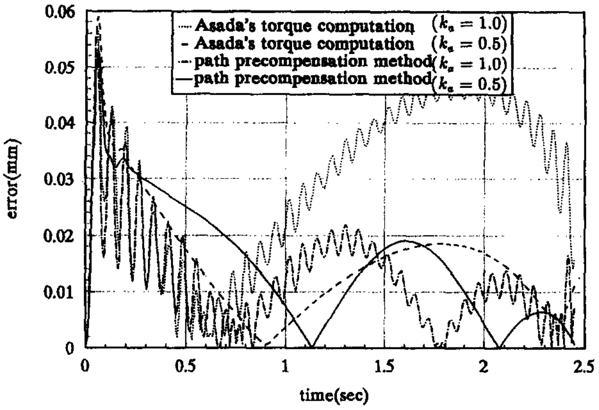

Fig. 15. Trajectory error for a parabolic path ($k_v$/it> =150).

on the disturbance (the effects of the elastic deformation) and the accuracy of torque computation, that is, if the disturbance diverges or torque computation leads to gross inaccuracy, the system may be unstable. So it is necessary to smooth the vibration by the proposed idea, for example, use $k_a = 0.5$. CPU time for the closed-loop path precompensation method is 3.0 msec.

## 3. PATH PRECOMPENSATION METHOD FOR SPATIAL CURVE

Since the purpose of this study is to propose the path precompensation method for a flexible robot arm in order that it can track a spatial trajectory, so a three-link flexible robot arm will be given in this section.

### 3.1. *Dynamic modeling for a three-link flexible arm*

In order to stay compatible with the planar model in Section 2, a three-link flexible arm shown in Fig. 16 is considered. The flexible arm consists of two flexible links constrained in the horizontal $(X,Y)$ plane and one rigid link projecting from the plane. The equations for the first two flexible links are derived in Section 2.3. Note that the two planar flexible links have no flexibility in the $Z$-direction. The addition of the third rigid arm will not deflect the two flexible links out of their plane.

The total kinetic energy for the flexible arm is

$$T = T_1 + T_2 + T_3. \tag{16}$$

The total potential energy for the flexible arm is

$$U = U_1 + U_2 + U_3. \tag{17}$$

The Lagrangian function for a three-link flexible arm can be written as:

$$\begin{aligned} L &= L_1 + L_2 + L_3 \\ &= (T_1 - U_1) + (T_2 - U_2) + (T_3 - U_3) \\ &= L_1(\theta_1, \dot\theta_1, q_{1i} \dot q_{1i}) + L_2(\theta_1, \dot\theta_1, \theta_2, \dot\theta_2, q_{1i}, \dot q_{1i}, q_{2i}, \dot q_{2i}) \\ &\quad + L_3(\theta_1, \theta_2, \theta_3, \dot\theta_1, \dot\theta_2, \dot\theta_3). \end{aligned} \tag{18}$$

By applying the Lagrangian equation, the dynamic model of the flexible arm becomes:

(1) The joint $(\theta_1,\theta_2,\theta_3)$ equation

for $\theta_1$:

$$\frac{\mathrm{d}[\partial(L_1 + L_2 + L_3)/\partial\dot\theta_1]}{\mathrm{d}t} - \frac{\partial(L_1 + L_2 + L_3)}{\partial\theta_1} = f_{\theta_1} \tag{19}$$

for $\theta_2$:

$$\frac{\mathrm{d}[\partial(L_2 + L_3)/\partial\dot\theta_2]}{\mathrm{d}t} - \frac{\partial(L_2 + L_3)}{\mathrm{d}\theta_2} = f_{\theta_2} \tag{20}$$

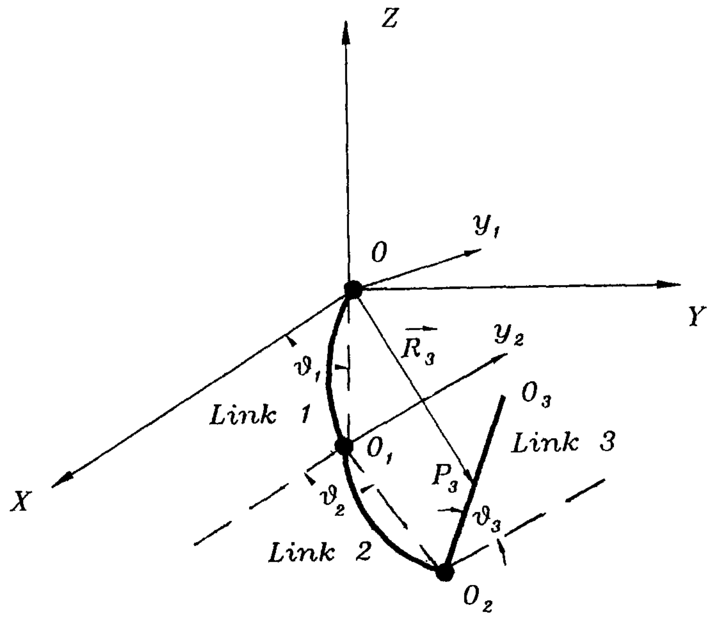

Fig. 16. Position of link 3 represented in VLCS.

for $\theta_3$:

$$\frac{\mathrm{d}[\partial(L_3)/\partial\dot{\theta}_3]}{\mathrm{d}t}-\frac{\partial(L_3)}{\partial\theta_1}=f_{\theta_3} \tag{21}$$

(2) The deflection $(q_{1i},q_{2i})$ equation

for $q_{1i}$:

$$\frac{\mathrm{d}[\partial(L_1+L_2)/\partial\dot{q}_{1i}]}{\mathrm{d}t}-\frac{\partial(L_1+L_2)}{\partial q_{1i}}=f_{1i},\quad i=1,2,\ldots,m_i \tag{22}$$

for $q_{2i}$:

$$\frac{\mathrm{d}[\partial(L_1+L_2)/\partial\dot{q}_{2i}]}{\mathrm{d}t}-\frac{\partial(L_1+L_2)}{\partial q_{2i}}=f_{2i}, i=1,2,\ldots,m_i. \tag{23}$$

In Section 2.5.1, we have derived the dynamic equations for the two-link flexible arm. So for construction of Lagrangian's equations, we need

$$\frac{\mathrm{d}(\partial L_3/\partial\dot{\theta}_1)}{\mathrm{d}t},\ \frac{\partial L_3}{\partial\theta_1},\ \frac{\mathrm{d}(\partial L_3/\partial\dot{\theta}_2)}{\mathrm{d}t},\ \frac{\partial L_3}{\partial\theta_2},\ \frac{\mathrm{d}(\partial L_3/\partial\dot{\theta}_3)}{\mathrm{d}t},\ \text{and}\ \frac{\partial L_3}{\partial\theta_3}$$

where

$$\begin{aligned}\frac{\mathrm{d}(\partial L_3/\partial\dot{\theta}_1)}{\mathrm{d}t}=\frac{m}{2}\{&2l_3l_1^2\ddot{\theta}_1+l_1l_3^2\ddot{\theta}_2\cos\theta_3\cos(\beta-\theta_1)\\&-l_1l_3^2\dot{\theta}_2\dot{\theta}_3\sin\theta_3\cos(\beta-\theta_1)\\&-l_1l_3^2\dot{\theta}_2(\dot{\theta}_2-\dot{\theta}_1)\cos\theta_3\sin(\beta-\theta_1)\\&+l_1l_3^2\dot{\theta}_3^2\cos\theta_3\sin(\beta-\theta_1)\\&-l_1l_3^2\dot{\theta}_3(\dot{\theta}_1-\dot{\theta}_2)\sin\theta_3\cos(\beta-\theta_1)\\&+2l_1l_2l_3\ddot{\theta}_2\cos(\theta_1-\theta_2)\\&-2l_1l_2l_3\dot{\theta}_2(\dot{\theta}_1-\dot{\theta}_2)\sin(\theta_1-\theta_2)\}\end{aligned} \tag{24}$$

$$\begin{aligned}\frac{\partial L_3}{\partial\theta_1}=\frac{m}{2}\{&l_1\dot{\theta}_1\dot{\theta}_2l_3^2\cos\theta_3\sin(\beta-\theta_1)\\&-l_1\dot{\theta}_1\dot{\theta}_3l_3^2\sin\theta_3\cos(\beta-\theta_1)\\&-2l_1l_2\dot{\theta}_1\dot{\theta}_2l_3\sin(\theta_2-\theta_1)\}\end{aligned} \tag{25}$$

$$\begin{aligned}\frac{\mathrm{d}(\partial L_3/\partial\dot{\theta}_2)}{\mathrm{d}t}=\frac{m}{2}\Big\{&\frac{2}{3}l_2^3\ddot{\theta}_2\cos^2\theta_3-\frac{3}{4}l_2^3\dot{\theta}_2\dot{\theta}_3\cos\theta_3\sin\theta\\&+2l_3l_2^3\ddot{\theta}_2+l_1l_3^2\ddot{\theta}_1\cos\theta_3\cos(\beta-\theta_1)\\&-l_1l_3^2\dot{\theta}_1\dot{\theta}_1\sin\theta_3\cos(\beta-\theta_1)\\&+l_1l_3^2\dot{\theta}_1(\dot{\theta}_2-\dot{\theta}_1)\cos\theta_3\sin(\beta-\theta_1)\\&+l_2l_3^2\ddot{\theta}_3\sin\theta_3+l_2l_3^2\dot{\theta}_3\dot{\theta}_2\cos\theta_3\\&+2l_1l_2l_3\ddot{\theta}_1\cos(\theta_1-\theta_2)\\&-2l_1l_2l_3\dot{\theta}_1(\dot{\theta}_1-\dot{\theta}_2)\sin(\theta_1-\theta_2)\Big\}\end{aligned} \tag{26}$$

$$\begin{aligned}\frac{\partial L_3}{\partial\theta_2}=\frac{m}{2}\{&-l_1\dot{\theta}_1\dot{\theta}_2l_3^2\cos\theta_3\sin(\beta-\theta_1)\\&-l_1\dot{\theta}_1\dot{\theta}_3l_3^2\sin\theta_3\cos(\beta-\theta_1)\\&-2l_1l_2\dot{\theta}_1\dot{\theta}_2\sin(\theta_2-\theta_1)\}\end{aligned} \tag{27}$$

$$\begin{aligned}\frac{\mathrm{d}(\partial L_3/\partial\dot{\theta}_3)}{\mathrm{d}t}=\frac{m}{2}\Big\{&\frac{2}{3}l_3^3\ddot{\theta}_3-l_1l_3^2\ddot{\theta}_1\sin\theta_3\sin(\beta-\theta_1)\\&-l_1l_3^2\dot{\theta}_1\dot{\theta}_3\cos\theta_3\sin(\beta-\theta_1)\\&+l_1l_3^2\dot{\theta}_1(\dot{\theta}_2-\dot{\theta}_1)\sin\theta_3\sin(\beta-\theta_1)\Big\}\\&+l_2l_3^2\ddot{\theta}_2\sin\theta_3\\&+l_2l_3^2\dot{\theta}_3\dot{\theta}_2\cos\theta_3\end{aligned} \tag{28}$$

$$\begin{aligned}\frac{\partial L_3}{\partial\theta_3}=\frac{m}{2}\Big\{&-\frac{2}{3}l_3^3\dot{\theta}_2^2\cos\theta_3\sin\theta_3\\&-l_1\dot{\theta}_1\dot{\theta}_2l_3^2\sin\theta_3\cos(\beta-\theta_1)\\&-l_1\dot{\theta}_1\dot{\theta}_3l_3^2\cos\theta_3\sin(\beta-\theta_1).\\&+l_2l_3^2\dot{\theta}_2\dot{\theta}_3\cos\theta_3\Big\}\\&-\Big\{\frac{1}{2}mgl_3^2\cos\theta_3\Big\}\end{aligned} \tag{29}$$

The three driving torques $\tau_1$, $\tau_2$ and $\tau_3$ are exerted in the joints $O$, $O_1$ and $O_2$.

We assumed that the longitudinal deformation and torsional modes are negligible.

The generalized forces are found from the total virtual work to be

for joint:

$$f_{\theta_1}=\tau_1-\tau_2$$

$$f_{\theta_2}=\tau_2$$

$$f_{\theta_3}=\tau_3$$

for deformation:

$$f_{1i}=\phi_{1p}{}'(0)\tau_1-\phi_{1p}{}'l_1\tau_2,\ i=1,2,\ldots,m_1$$

$$f_{2i} = \phi_{2p}{}'(0)\tau_2, i = 1, 2, ..., m_2.$$

### 3.2. *Planning for spatial curve and path precompensation method*

Since the cubic spline is the most common trajectory applied by a robot, this will be proposed for a flexible arm under the path precompensation method in this section. A cubic spline segment between two points $P_k$ and $P_{k+1}$ as shown in Fig. 17 is given as:

$$R_k(u) = \sum_{i=1}^{4} A_{ik}u^{i-1} 0 \leq u \leq 1, 2 \leq k \leq n-1 \quad (30)$$

where $u$ is the parameter along the curve, $n$ is the number of the data point, $A_{ik}$ are the coefficients of the polynomial, and $R_k(u)$ is any position on the curve between point $P_k$ and $P_{k+1}$.

The position of the curve segment can be obtained as:

$$R_k(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ P_k{}' \\ P_{k+1}{}' \end{bmatrix} \quad (31)$$

where $P_k$,$P_{k+1}$ is the position vector of data point,

$$P_k{}', \ P_{k+1}{}'$$

are the slopes of the data points.

For the condition of position, slope and curvature continuity at the connected joints, Eq. (31) yields the following relationships:

$$P_{k-1}{}' + 4P_k{}' + P_{k+1}{}' = 3(P_{k+1} - P_{k-1}), 2 \leq k \leq n-1. \quad (32)$$

By applying Eq. (32) for all data points, we get the matrix equation:

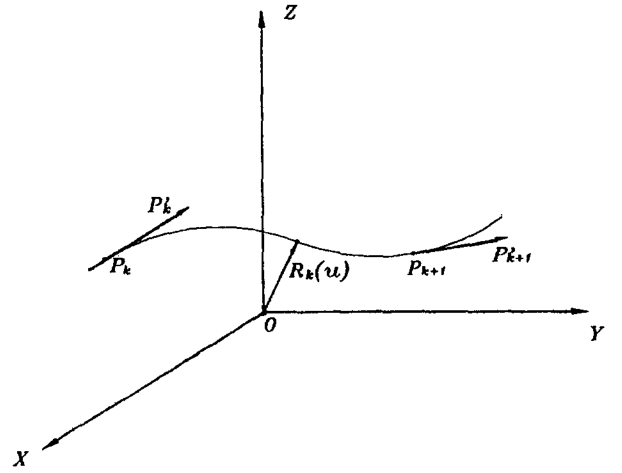

Fig. 17. Cubic spline segment.

$$\begin{bmatrix} 1 & 4 & 1 & 0 & \cdots \\ 0 & 1 & 4 & 1 & \cdots \\ & & \ddots & & \\ \cdots & 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} P_1{}' \\ P_2{}' \\ \\ P_n{}' \end{bmatrix} = \begin{bmatrix} 3(P_3 - P_1) \\ 3(P_4 - P_2) \\ \\ 3(P_n - P_{n-2}) \end{bmatrix}. \quad (33)$$

Therefore, together with boundary conditions the slope value can be solved to obtain any position vector on the cubic spline curve. Any position vector of the segment on the spatial cubic spline is represented as

$$\mathbf{r}(u) = x(u)\mathbf{i} + y(u)\mathbf{j} + z(u)\mathbf{k}, u \in [0, 1] \quad (34)$$

where

$$x(u) = \sum_{i=1}^{4} A_{ikx}u^{i-1},$$

$$y(u) = \sum_{i=1}^{4} A_{iky}u^{i-1},$$

$$z(u) = \sum_{i=1}^{4} A_{ikz}u^{i-1}.$$

The position error is of the following form:

$$\mathbf{E}_r = (x(u) - x_i)\mathbf{i} + (y(u) - y_j)\mathbf{j} + (z(u) - z_k)\mathbf{k}.$$

The tangential velocity $V_b$ in Eq. (5) or Eq. (6) is required constant by conventional CNC machining, which is usually performing along a straight line or a circular arc. Since robots usually track spatial curves of higher order, the non-constant variation of curvature along the curve will enter into consideration.

So we make the following consideration concerning the velocity $\dot{R}$ and acceleration

$$\ddot{R}$$

for the tracking trajectory. If $u$ is a function of time $t$, the velocity

$$\dot{R}$$

and acceleration

$$\ddot{R}$$

can be represented as

$$\dot{R} = \frac{\partial R}{\partial u} \bullet \frac{\partial u}{\partial t} = R'\dot{u}$$

$$\ddot{R} = R''(\dot{u})^2 + R'\ddot{u}.$$

Since

$$\dot{u}$$

is usually a constant value $c$,

$$\ddot{u}$$

is zero. The velocity and acceleration become

**Table 1** The data points used in forming a cubic spline

| Point | Position coordinate (m) |
|---|---|
| 1 | (0.830,0.031,0.371) |
| 2 | (0.800,0.042,0.430) |
| 3 | (0.700,0.049,0.471) |
| 4 | (0.500,0.032,0.375) |
| 5 | (0.400,0.017,0.293) |
| 6 | (0.300,0.000,0.200) |

$$\dot{R} = R'\dot{u} = cR' \tag{35}$$

$$\ddot{R} = R''(\dot{u})^2 = c^2R'. \tag{36}$$

So the duration of tracking is determined by the constant $c$.

The path precompensation method for cubic spline requires the $u$ value that corresponds to the minimum error $E_r$. That is, the roots for

$$E_r' = 0$$

will be found, and the command velocity is compensated as

$$\mathbf{V} = \dot{\mathbf{R}} + k_v\mathbf{E}_r = c\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}u} + k_v\mathbf{E}_r. \tag{37}$$

The difference between Eq. (37) and Eq. (5) is that $V_b t$ remains constant but

$$c\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}u}$$

varies from point to point. Since

$$c(=\dot{u})$$

is the explicit term used in handling the parametric form curve, Eq. (37) offers more implementation convenience and a way to see the correlation between the machining velocity and the curve condition. The block diagram for tracking spatial trajectory using the path precompensation method is the same as that in Fig. 5.

**Table 2** The planned trajectory by cubic spline method for initial $R'=(-30,10,60)$

| Data points (m) | Equations of cubic spline for each segment |
|---|---|
| 0.830,0.031,0.371 | $x(u)=0.830-0.030u+0.017u^2-0.017u^3$ |
| | $y(u)=0.031+0.010u+0.001u^2+0.001u^3$ |
| 0.800,0.042,0.430 | $z(u)=0.371+0.060u-0.011u^2+0.010u^3$ |
| 0.800,0.042,0.430 | $x(u)=0.800-0.046u-0.033u^2-0.020u^3$ |
| | $y(u)=0.042+0.012u+0.002u^2-0.007u^3$ |
| 0.700,0.049,0.471 | $z(u)=0.430+0.068u+0.020u^2-0.046u^3$ |
| 0.700,0.049,0.471 | $x(u)=0.700-0.173u-0.093u^2+0.066u^3$ |
| | $y(u)=0.049-0.005u-0.020u^2+0.009u^3$ |
| 0.500,0.032,0.375 | $z(u)=0.471-0.032u-0.119u^2+0.055u^3$ |
| 0.500,0.032,0.375 | $x(u)=0.500-0.160u+0.105u^2-0.045u^3$ |
| | $y(u)=0.032-0.018u+0.007u^2-0.003u^3$ |
| 0.400,0.017,0.293 | $z(u)=0.375-0.104u+0.047u^2-0.025u^3$ |
| 0.400,0.017,0.293 | $x(u)=0.400-0.084u-0.030u^2+0.015u^3$ |
| | $y(u)=0.017-0.014u-0.002u^2-0.000u^3$ |
| 0.300,0.000,0.200 | $z(u)=0.293-0.085u-0.028u^2+0.020u^3$ |

### 3.3. *Implementation*

The path precompensation method is implemented on a three-link flexible arm as shown in Fig. 16. Links 1 and 2 are the same as the ones presented in Section 2.3. The length $l_3$ of link 3 is 1 m. The mass per unit length is 0.00312 kg/mm. We assumed that link 3 is rigid. The data points used in forming the desired cubic spline trajectory are listed in Table 1. Different curve segments can be obtained for different boundary conditions. The obtained curve segments for an initial $R'=(-30,10,60)$ are listed in Table 2. The flexible arm starts from point $P_1(830,31,371)$ in Fig. 18 to track the desired

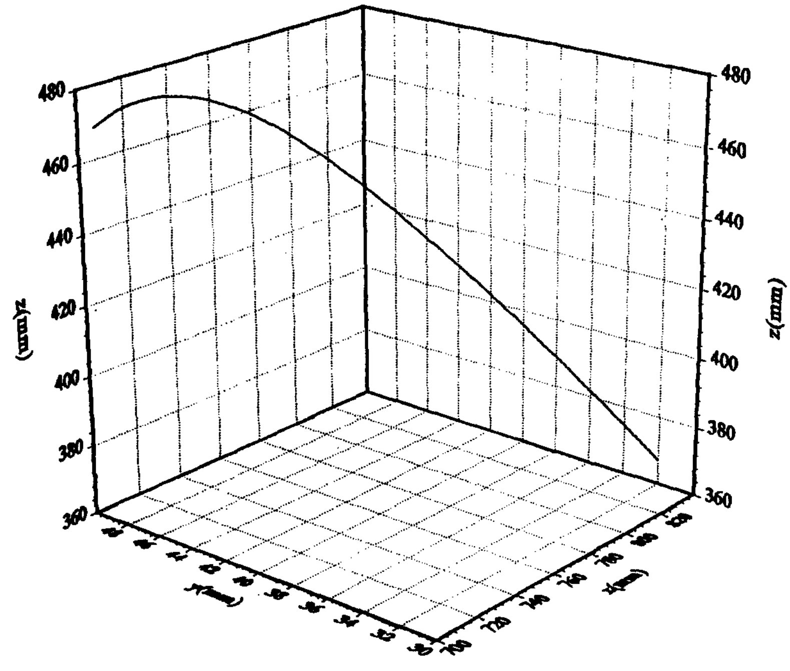

Fig. 18. Planned trajectory corresponding to Table 2.

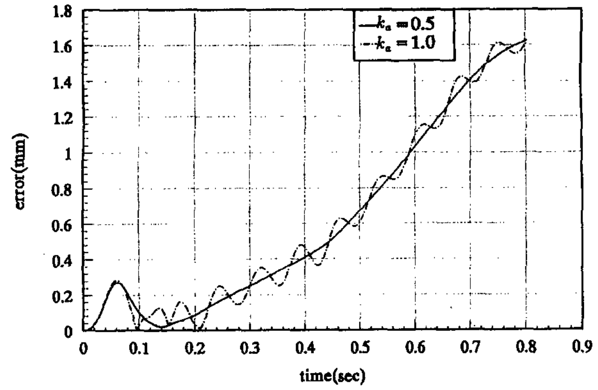

**Fig. 19.** Trajectory error by the open-loop torque computation method with $c=2.5$.

trajectory with different time in accordance with different $c$. The simulation is done for two cases: an open-loop control (torque computation) and a closed-loop control with path precompensation method.

3.3.1. *An open-loop control (torque computation).* The block diagram for the open loop is the same as the one in Fig. 6. The actuator torques obtained by Asada's method are applied to the respective joints. Figure 19 shows the error history of different $k_a$ for tracking with $c=2.5$. Figure 20a and b show the torques profiles of different $k_a$ corresponding to Fig. 19. Note that $k_a=1$ corresponds to Asada's method and $k_a=0.5$ corresponds to the partial deformation compensation proposed in this paper. From the above figures for $c=2.5$, it is seen that using torques to cope fully, i.e. $k_a=1$, with the effects of the elastic deformation creates undesired vibration and somewhat worse trajectory quality. This justifies the use of partial deformation compensation.

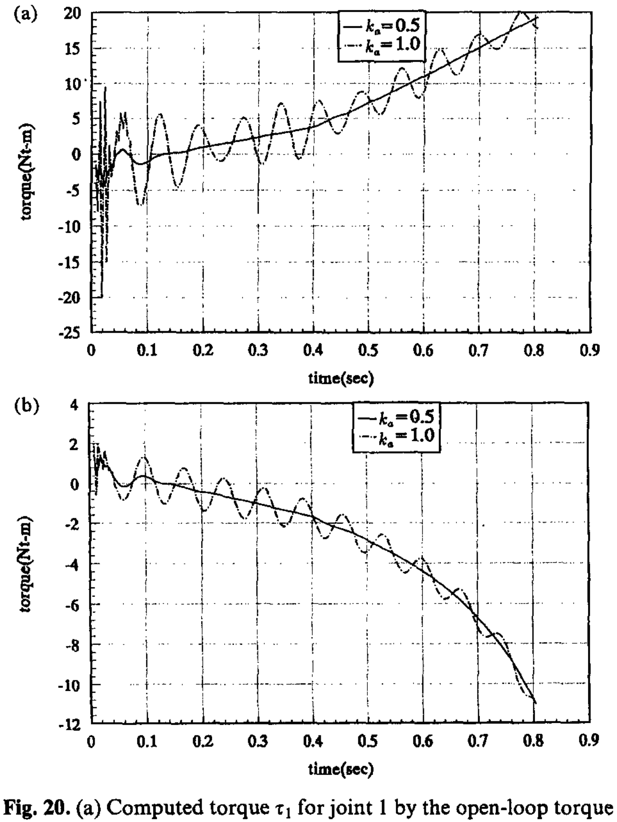

**Fig. 20.** (a) Computed torque $\tau_1$ for joint 1 by the open-loop torque computation method with $c=2.5$. (b) Computed torque $\tau_2$ for joint 2 by the open-loop torque computation method with $c=2.5$.

3.3.2. *Path precompensation method.* The block diagram is the same as the one in Fig. 5. The actuator torques obtained by the path precompensation method in Section 2.3 are applied to the corresponding actuators. Figures 21 and 22 show the error history for $c=2.5$ (duration of tracking: 2 sec) and $c=1$ (duration of tracking: 5 sec) when the tracking trajectory is as shown in Fig. 18.

The following observations can be made from the numerical simulation.

(1) The flexible arm has a better performance with the path precompensation method for both long and short tracking time.
(2) The errors are greater at the segments with bigger curvature.
(3) The error for smaller tracking velocity, i.e. smaller $c$, is smaller at the same time point, but the overall error is larger than for the one with greater velocity. This shows that the errors accumulated. The longer the tracking time, the greater the errors.
(4) There is no deceleration scheme toward the end point of the trajectory in the examples shown. If used, such a deceleration scheme will help reduce the accumulated errors toward the end point.

## 4. CONCLUSIONS

The path precompensation method has been established for a rigid arm robot in the past. This paper addresses the topic of the path precompensation method for a flexible arm robot.

Since the elastic deformation of the flexible arm constitutes one of the major obstacles in its motion control, a torque computation method taking care of the elastic deformation is first proposed and compared with the one proposed by Asada.

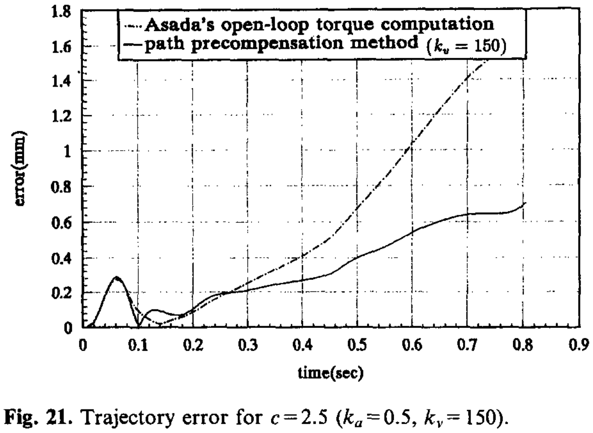

**Fig. 21.** Trajectory error for $c=2.5$ ($k_a=0.5$, $k_v=150$).

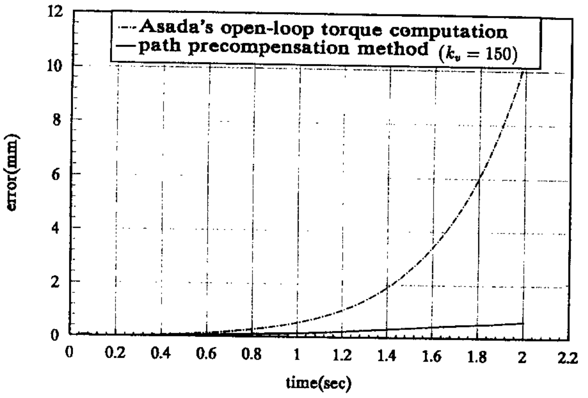

**Fig. 22.** Trajectory error for $c = 1$ ($k_a = 0.5$, $k_v = 150$).

Analysis reveals that the torques computed either by Asada's method or by the proposed method could induce elastic arm vibrations. Based on such observations, a concept of partial compensation for elastic deformation is proposed. It is shown that better torque profiles and trajectory fidelity can be obtained by partial compensation using a factor $k_a$ and $k_f$ smaller than one.

The path precompensation method is finally proposed for a flexible arm to enhance the torque computation by the advantages of closed-loop and path precompensation. It is shown that the proposed closed-loop path precompensation method is efficient in improving the behavior of a flexible arm robot tracking a planar or spatial trajectory. Throughout the study the two-link flexible arm and the three-link flexible arm are given as examples. The proposed concepts and methods in this paper help to advance the knowledge in creating a better motion guidance for flexible arm robot.

## REFERENCES

1. Book, W. J., Recursive Lagrangian dynamics of flexible manipulator arms. *International Journal of Robotics Research*, 1984, **3**, 87–101.
2. Usoro, P. B., Nadira, R. and Mahil, S. S., A finite element/Lagrangian approach to modeling lightweight flexible manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 1986, **108**, 198–205.
3. Sunada, W. and Dubowsky, S., The application of finite element methods to the dynamics analysis of spatial and coplanar linkage system. *ASME Journal of Mechanical Design*, 1981, **103**, 643–651.
4. King, J. O., Gourishankar, V. G. and Rink, R. E., Lagrangian dynamics of flexible manipulators using angular velocities instead of transformation matrices. *IEEE Transactions on Systems, Man, and Cybernetics*, 1987, **SMC-17**, 1059–1068.
5. Cannon, R. H. and Schmitz, E., Initial experiments on the end-point control of a flexible one-link robot. *International Journal of Robotics Research*, 1984, **3**, 62–75.
6. Book, W. J. and Hastings, G. G., A linear dynamic model for flexible robotic manipulators. *IEEE Control Systems Magazine*, 1987, **7**, 61–64.
7. Book, W. J., Maizza-Neto, O. and Whitney, D. E., Feedback control of two joint systems with distributed flexibility. *ASME Journal of Dynamic Systems, Measurement, and Control*, 1975, **97**, 424–431.
8. Ower, J. C. and Van de Vegte, J., Classical control design for a flexible manipulator: modeling and control system design. *IEEE Journal of Robotics and Automation*, 1987, **RA-3**(5), 485–489.
9. Asada, H., Ma, Z.-D. and Tokumaru, H., Inverse dynamics of flexible robot arms: modeling and computation for trajectory control. *ASME Journal of Dynamic Systems, Measurement, and Control*, 1990, **112**, 177–185.
10. Chin, J.-H. and Tsai, H.-C., A path algorithm for robotic machining. *Robotics & Computer-Integrated Manufacturing*, 1993, **10**(3), 185–198.
11. Chin, J.-H. and Lin, T.-C., Cross-coupled precompensation method for the contouring accuracy of CNC machine tools. *International Journal of Machine Tools and Manufacture*. (to appear).
12. Hasegawa, K. and Mizutani, T., Manipulator control using autonomous trajectory generating servomechanism. In *Robotics Research: The Second International Symposium*, ed. Hanafusa, H. and Inoue, H. The MIT Press Series in Artificial Intelligence, 1984, pp. 179–186.
13. Sakaue, S. and Sugimoto, K., Path generation by rate control of manipulators. *Robotics & Computer-Integrated Manufacturing*, 1987, **3**(4), 381–387.
14. Yeung, K. S., Chen, Y. P. Regulation of a one-link flexible robot arm using sliding-mode technique. *International Journal of Control*, 1989, **49**(6), 1965–1978.
15. Baruh, H. and Tadikonda, S. S. K., Issues in the dynamics and control of flexible robot manipulators. *Journal of Guidance*, 1989, **12**(5), 659–670.