ROBUST DIGITAL CONTROLLER FOR ROBOT MANIPULATORS

Tadashi Ishihara

Department of Mechanical Engineering,
Tohoku University, Sendai, Japan

ABSTRACT

Direct digital design of computed torque controllers for a robot manipulator is discussed in this paper. A simple discrete-time model of the robot manipulator obtained by Euler's method is used for the design. Taking account of computation delay in the digital processor, we propose predictor-based designs of the PD and PID type controllers. The PID type controller is designed based on a modified version of the discrete-time integral controller proposed by Mita. For both controllers, the same formulas can be used to determine the feedback gains. A simulation example is presented to compare the robustness of the proposed controllers against physical parameter variations.

1. INTRODUCTION

Computed torque method[1] is well known as an efficient algorithm for trajectory control of a robot manipulator. In this method, the dynamics of a robot manipulator is linearized by a nonlinear compensator and then the conventional linear state feedback law is applied. Since complex on-line computations are required for the nonlinear compensation, it is appropriate to implement the algorithm as a digital controller.

Using the meticulous discrete-time model of a robot manipulator obtained by the trapezoidal approximation, Neuman and Tourassis[2] have proposed a direct digital design of the PD type computed torque controller. However, since the structure of the model is inconvenient for the controller design, they have introduced an additional heuristic approximation. Moreover, in spite of the fact that the computation time is significant in a computed torque controller, their design has ignored the computation delay in the controller.

In this paper, we propose designs of discrete-time computed torque controllers based on a simpler discrete-time model of a robot manipulator obtained by Euler's method[3]. Since Euler's method is the most crude approximation, its use is rarely recommended in numerical solution of differential equations. In our case, the effect of the discretization error is not so serious because the error can be reduced by virtue of feedback.

First, we discuss the design of the PD type controller disregarding the computation delay. Simple formulas for the determination of the feedback gains are obtained. Then we propose a design of the PD type controller which includes a state predictor to compensate the computation delay. The feedback gains can conveniently be determined by use of the formulas for the design disregarding the computation delay. To achieve more robust performance against parameter variations and/or disturbances, we propose a simple and transparent design of the PID type controller accounting the computation delay. This design is based on a modified version of the integral controller proposed by Mita[4]. The formulas for the proposed PD type controllers are still useful to determine the feedback gains. A simulation example is presented to compare the robustness of the proposed controllers against physical parameter variations.

2. DISCRETE-TIME MODEL

Consider an n-degree-of-freedom robot manipulator described by

$$D(\theta)\ddot{\theta} + h(\theta,\dot{\theta}) + g(\theta) = \tau , \qquad (2.1)$$

where τ is the n x 1 vector of joint forces or torques supplied by the actuators, $\theta(t)$ is the n x 1 vector of joint positions, with $\theta(t)=[\theta_1(t)\ \theta_2(t) \cdots \theta_n(t)]'$. The matrix $D(\theta)$ is the n x n mass matrix of the manipulator, $h(\theta,\dot{\theta})$ is the n x 1 vector of centrifugal and Coriolis terms and $g(\theta)$ is the n x 1 vector of gravity terms.

Using Euler's method[3], we construct a discrete-time model of (2.1). Let T denote the sampling period for the discretization. Denoting $\theta(kT)$, $\dot{\theta}(kT)$, $\ddot{\theta}(kT)$ and $\tau(kT)$, where k is an integer, by $\theta(k)$, $\dot{\theta}(k)$, $\ddot{\theta}(k)$ and $\tau(k)$, respectively, we can write (2.1) as

$$\ddot{\theta}(k) = D^{-1}[\theta(k)]\{\tau(k) - h[\theta(k),\dot{\theta}(k)] - g[\theta(k)]\} . \qquad (2.2)$$

Applying Euler's method, we have

$$\theta(k+1) = \theta(k) + T\dot{\theta}(k), \qquad (2.3)$$
$$\dot{\theta}(k+1) = \dot{\theta}(k) + T\ddot{\theta}(k), \qquad (2.4)$$

where we have omitted the discretization errors. Substituting (2.4) into (2.2), we have

$$\dot{\theta}(k+1) = \dot{\theta}(k) + TD^{-1}[\theta(k)]\{\tau(k) - h[\theta(k),\dot{\theta}(k)] - g[\theta(k)]\} . \qquad (2.5)$$

Define

$$q(k) = [\ \theta'(k)\ \ \dot{\theta}'(k)\ ]' . \qquad (2.6)$$

Then the difference equations (2.3) and (2.5) constitute the state space representation of the discrete-time model obtained by Euler's method.

3. PD TYPE CONTROLLER DISREGARDING COMPUTATION DELAY

First, as a preliminary to consider the design accounting the computation delay, we discuss the design disregarding the computation delay. This controller is applicable only when the generation of the control input signal can be completed sufficiently fast compared with the sampling period. The following discussion is almost parallel to the standard derivation of the continuous-time computed torque controller[1].

Nonlinear Compensation

For the discrete-time state space model given by (2.3) and (2.5), we consider the nonlinear compensation given by

$$\tau(k) = h_0[\theta(k), \dot{\theta}(k)] + g_0[\theta(k)] + D_0[\theta(k)]\nu(k), \tag{3.1}$$

where $\nu(k)$ is the new control input vector $h_0[\cdot,\cdot]$, $g_0[\cdot]$ and $D_0[\cdot]$ are estimates of $h[\cdot,\cdot]$, $g[\cdot]$ and $D[\cdot]$, respectively. Then the evolution of the state vector $q(k)$ defined in (2.6) is described by

$$q(k+1) = \Phi_T q(k) + \Gamma_T \nu(k) + \xi_T[q(k)], \tag{3.2}$$

where

$$\Phi_T = \begin{bmatrix} I_n & TI_n \\ 0_n & I_n \end{bmatrix}, \quad \Gamma_T = T\begin{bmatrix} 0_n \\ I_n \end{bmatrix}, \quad \xi_T[q(k)] = \begin{bmatrix} 0_n \\ \eta_T[q(k)] \end{bmatrix}. \tag{3.3}$$

In the matrices defined in (3.3), I_n is n x n identity matrix, 0_n is n x n zero matrix and the vector $\eta_T[q(k)]$, which represents the error arising from the uncertainties of the physical parameters, is defined by

$$\begin{aligned}\eta_T[q(k)] = {} & TD^{-1}[\theta(k)]\{D_0[\theta(k)] - D[\theta(k)]\}\nu(k) \\ & + \{h_0[\theta(k), \dot{\theta}(k)] - h[\theta(k), \dot{\theta}(k)]\} \\ & + \{g_0[\theta(k)] - g[\theta(k)]\}.\end{aligned} \tag{3.4}$$

Apparently, if no modelling error exists, then $\eta_T[q(k)]=0$ and the dynamics of the robot manipulator is completely decoupled and linearized.

Linear Compensation

Let $\theta_d(k)$, $\dot{\theta}_d(k)$ and $\ddot{\theta}_d(k)$ denote the sampled values of the position, the velocity and the acceleration of the reference trajectory, respectively. Introduce the state vector for the reference trajectory as

$$q_d(k) = [\,\theta_d'(k) \quad \dot{\theta}_d'(k)\,]'. \tag{3.5}$$

Assume that the sampled values $\theta_d(k)$, $\dot{\theta}_d(k)$ and $\ddot{\theta}_d(k)$ satisfy the relation of Euler's method, then the state defined by (3.5) satisfies

$$q_d(k+1) = \Phi_T q_d(k) + \Gamma_T \ddot{\theta}(k). \tag{3.6}$$

Using the new control input vector $\nu(k)$ in (3.2), we apply the following state feedback which can be regarded as a PD type control law.

$$\begin{aligned}\nu(k) = {} & \ddot{\theta}_d(k) + K_p[\theta_d(k) - \theta(k)] \\ & + K_v[\dot{\theta}_d(k) - \dot{\theta}(k)]\end{aligned} \tag{3.7}$$

The matrices K_p and K_v in (3.7) are defined as

$$K_p = diag[\,k_{p1}\, k_{p2} \cdots k_{pn}\,], \tag{3.8}$$
$$K_v = diag[\,k_{v1}\, k_{v2} \cdots k_{vn}\,], \tag{3.9}$$

where k_{pi} and k_{vi} are the position and the velocity feedback gains for the i-th joint, respectively. Define the state feedback matrix as

$$F = [\,K_p \quad K_v\,]. \tag{3.10}$$

Then we can rewrite (3.7) as

$$\nu(k) = -F[q(k) - q_d(k)] + \ddot{\theta}_d(k). \tag{3.11}$$

It follows easily from (3.2), (3.6) and (3.11) that the behavior of the tracking error

$$e(k) = q(k) - q_d(k) \tag{3.12}$$

is described by

$$e(k+1) = (\Phi_T - \Gamma_T F)e(k) + \xi_T[q(k)]. \tag{3.13}$$

Using (3.3) and (3.10), we can write the transition matrix in (3.13) as

$$\Phi_T - \Gamma_T F = \begin{bmatrix} I_n & TI_n \\ -TK_p & I_n - TK_v \end{bmatrix}. \tag{3.14}$$

Feedback Gain Determination

In the computed torque method, the feedback gain matrix F is chosen such that the tracking vector $e(k)$ converges to zero as k tends to the infinity assuming that $\xi_T[q(k)]=0$, i.e., no modelling error exists. Employing the special structure of the error transition matrix (3.14), we can obtain a simple formula for the feedback gain matrix F which provides the desired eigenvalue assignment. Note that the characteristic polynomial for the error transition matrix (3.14) can be expressed as

$$\begin{aligned}&\det[zI_{2n} - (\Phi_T - \Gamma_T F)] \\ &= \det[(zI_n - I_n)(zI_n - I_n + TK_v) + T^2 K_p] \\ &= \prod_{i=1}^{n} [z^2 - (2 - Tk_{vi})z + (1 - Tk_{vi} + T^2 k_{pi})].\end{aligned} \tag{3.15}$$

The above expression implies that the response of the i-th joint can be determined independently by the feedback gains k_{pi} and k_{vi}. Note that the characteristic equation for the i-th joint is given by the quadratic equation

$$z^2 - (2 - Tk_{vi})z + (1 - Tk_{vi} + T^2k_{pi}) = 0 . \qquad (3.16)$$

Let z_{i1} and z_{i2} denote the desired eigenvalues for the i-th joint dynamics. It follows easily from (3.16) that the feedback gains in the i-th joint assigning the desired eigenvalues are given by the simple formulas

$$k_{pi} = T^{-2}(1 - z_{i1})(1 - z_{i2}) , \qquad (3.17)$$
$$k_{vi} = T^{-1}(2 - z_{i1} - z_{i2}) . \qquad (3.18)$$

Some information useful for practical design can also be obtained from (3.17) and (3.18). For example, in the case that the desired eigenvalues z_{i1} and z_{i2} are required to be real and positive less than unity, which is necessary for the tracking without overshoot, then the above rule readily gives the range of the feedback gains as

$$0 < k_{pi} < T^{-2}, \quad 0 < k_{vi} < 2T^{-1} . \qquad (3.19)$$

4. PD TYPE CONTROLLER ACCOUNTING COMPUTATION DELAY

In the case that the computation delay is significant, the controller designed by the method proposed in the previous section does not provide desired performance. In this section, we propose a design of the PD type computed torque controller accounting the computation delay.

Predictor-Based Controller

For linear discrete-time systems, it is well known that the state feedback regulator accounting the computation delay can be designed by using the state predictor[4][6]. This idea can be used to design a discrete-time computed torque controller which is essentially nonlinear controller.

For simplicity, we assume that only unit computation exists in the controller, which implies that the necessary computation can be completed within the sampling interval. Let $\theta(k|k-1)$ and $\dot{\theta}(k|k-1)$ denote the one-step predictions of $\theta(k)$ and $\dot{\theta}(k)$, respectively, based on the input and output data available at time k. Using the discrete-time nonlinear state-space model of the robot manipulator (2.3) and (2.5), we can construct the one-step ahead state predictor as

$$\theta(k|k-1) = \theta(k-1) + T\dot{\theta}(k-1) , \qquad (4.1)$$
$$\dot{\theta}(k|k-1) = \dot{\theta}(k-1) - TD_0^{-1}[\theta(k-1)] \{h_0[\theta(k-1),\dot{\theta}(k-1)] + g_0[\theta(k-1)]\} + TD_0^{-1}[\theta(k-1)]\nu(k-1) . \qquad (4.2)$$

Replacing $\theta(k)$ and $\dot{\theta}(k)$ in the nonlinear compensation (3.1) by $\theta(k|k-1)$ and $\dot{\theta}(k|k-1)$, respectively, we can realize the nonlinear compensation using the data available at time k as

$$\tau(k) = h_0[\theta(k|k-1), \dot{\theta}(k|k-1)] + g_0[\theta(k|k-1)] + D_0[\theta(k|k-1)]\nu(k) . \qquad (4.3)$$

If the discretizaion error and the modelling error do not exist, then $\theta(k)=\theta(k|k-1)$ and $\dot{\theta}(k)=\dot{\theta}(k|k-1)$. Consequently, the robot manipulator with the nonlinear compensation (4.3) can simply be described by

$$q(k+1) = \Phi_T q(k) + \Gamma_T \nu(k) . \qquad (4.4)$$

Moreover, in this case, the prediction (4.2) reduces to

$$\dot{\theta}(k|k-1) = \dot{\theta}(k-1) + T\nu(k-1) . \qquad (4.5)$$

For simplicity, we use the formula (4.5) instead of (4.2) to compute the prediction $\dot{\theta}(k|k-1)$ in the nonlinear compensator (4.3). Moreover, if we define

$$\hat{q}(k|k-1) = [\, \theta'(k|k-1) \;\; \dot{\theta}'(k|k-1)]' , \qquad (4.6)$$

then the predicted state can be expressed as

$$\hat{q}(k|k-1) = \Phi_T q(k-1) + \Gamma_T \nu(k-1) . \qquad (4.7)$$

Using the one-step ahead prediction of the state given by (4.7), we replace the linear state feedback law (3.11) by

$$\nu(k) = -F[\hat{q}(k|k-1) - q_d(k)] + \ddot{\theta}_d(k) . \qquad (4.8)$$

The PD type controller accounting the computation delay consists of the nonlinear compensator (4.3), the state predictor (4.7) and the linear state feedback law (4.8).

Feedback Gain Determination

Assume that no modelling error exists. Using (3.6), (3.12), (4.4), (4.7) and (4.8), we can describe the behavior of the tracking error as

$$e(k+1) = \Phi_T e(k) + \Gamma_T \delta(k) , \qquad (4.9)$$
$$\delta(k) = -F[\hat{q}(k|k-1) - q_d(k)] , \qquad (4.10)$$
$$\hat{q}(k+1|k) - q_d(k+1) = \Phi_T e(k) + \Gamma_T \delta(k) , \qquad (4.11)$$

where

$$\delta(k) = \nu(k) - \ddot{\theta}_d(k) . \qquad (4.12)$$

Define the extended tracking error vector as

$$\varepsilon(k) = [\, e'(k) \;\; \delta'(k)]' . \qquad (4.13)$$

Applying the result in [4] to (4.9)-(4.11), we can easily show that the behavior of the extended tracking error vector is described by

$$\varepsilon(k+1) = \Omega_T \varepsilon(k) , \qquad (4.14)$$

where

$$\Omega_T = \begin{bmatrix} \Phi_T & \Gamma_T \\ -F\Phi_T & -F\Gamma_T \end{bmatrix} . \qquad (4.15)$$

If we choose the feedback gain matrix F such that the matrix (4.15) is asymptotically stable, the extended tracking error vector converges to zero as k tends to the infinity. It is interesting to note that the eigenvalues of the error transition matrix (4.15) consist of the eigenvalues of the matrix (3.14) and $2n$ zeros[4]. Consequently, the feedback gains can easily be determined by the formulas for the design disregarding the computation delay discussed in the previous section.

5. PID TYPE CONTROLLER ACCOUNTING COMPUTATION DELAY

The robustness of the PD type computed torque controller can be improved by introducing the integral action. For discrete-time linear systems, Mita[4] has proposed a novel design of integral controller accounting the computation delay. This design can be performed using the solution of the standard regulator problem for a plant. In addition, Mita and Mukaida[5] have proposed a modified version of the design based on heuristic argument. Guo *et al.*[7] have clarified the relation between the two designs and have discussed the extensions to the general computation delay case[8]. Although the purpose of these research is to provide an efficient method for designing a servosystem tracking the step reference signal without steady state error, the resulting algorithm can be used as a state feedback law with the integral action.

Using the modified version of Mita's design discussed in [7] as a state feedback algorithm with the integral action, we propose a simple and transparent design of the PID type computed torque controller accounting the computation delay.

State Feedback with Integral Action

Assuming that no modelling error exists and that the dynamics of the robot manipulator is linearized and decoupled by the nonlinear compensation (4.3) with the state predictor (4.7). Then the behavior of the tracking error (3.12) for arbitrary input $v(k)$ is described by (4.9) and (4.12). To apply the algorithm discussed in [7], we consider the error system

$$e(k+1) = \Phi_T e(k) + \Gamma_T \delta(k), \qquad (5.1)$$

$$\tilde{\theta}(k) = \mathrm{H}e(k), \qquad (5.2)$$

where $\delta(k)$ is defined in (4.12) and

$$\mathrm{H} = [I_n \quad 0_n]. \qquad (5.3)$$

Consider the state feedback law with the integral action defined by

$$\delta(k+1) = -L\delta(k) - M\tilde{\theta}(k) - Ne(k) + s(k), \qquad (5.4)$$

$$s(k+1) = s(k) - M\tilde{\theta}(k), \qquad (5.5)$$

where L, M and N are appropriate feedback gain matrices to be determined and $s(k)$ is the n x 1 vector representing the state of the integrators. Note that the control input $v(k)$ corresponding (5.4) is given by

$$\begin{aligned} v(k+1) = &-N_p[\theta(k) - \theta_d(k)] - N_v[\dot{\theta}(k) - \dot{\theta}_d(k)] \\ &-L[v(k) - \ddot{\theta}_d(k)] + \ddot{\theta}_d(k+1) + s(k+1), \end{aligned} \qquad (5.6)$$

where the matrices N_p and N_v are defined by

$$N = [N_p \quad N_v]. \qquad (5.7)$$

Note that the algorithm (5.6) with (5.6) admits unit step computation delay since the control input depends only on the previous state of the manipulator. The PID type controller consists of the nonlinear compensator (4.3), the state predictor (4.7) and the state feedback control law with the integral action (5.5) and (5.6).

Feedback Gain Determination

To determine the feedback gain matrices L, M and N in (5.4) and (5.5), we apply the method proposed in [7]. By this method, we can achieve the eigenvalue assignment similar to that obtained by the predictor-based PD type controller discussed in the previous section. This design method consists of two steps. First, we choose the feedback gain matrix F defined by (3.10). Then the matrices L, M and N are determined by the linear matrix equations

$$[N + MH \quad M]E = [F\Phi_T^2 \quad I_n + F\Gamma_T + F\Phi_T\Gamma_T], \qquad (5.8)$$

$$L = I_n + F\Gamma_T, \qquad (5.9)$$

where the matrix E is defined as

$$E = \begin{bmatrix} \Phi_T - I_{2n} & \Gamma_T \\ \mathrm{H} & 0_n \end{bmatrix}. \qquad (5.10)$$

Substituting (3.3), (3.10) and (5.7) into (5.8) and (5.10), we obtain

$$\begin{aligned} &[N_p + M \quad N_v \quad M]E \\ &= [K_p \quad 2TK_p + K_v \quad I_n + 2TK_v + T^2K_p], \end{aligned} \qquad (5.11)$$

$$E = \begin{bmatrix} 0_n & -TI_n & 0_n \\ 0_n & 0_n & -TI_n \\ I_n & 0_n & 0_n \end{bmatrix}. \qquad (5.12)$$

Note that the inverse the matrix E is given by

$$E^{-1} = T^{-1}\begin{bmatrix} 0_n & 0_n & TI_n \\ I_n & 0_n & 0_n \\ 0_n & I_n & 0_n \end{bmatrix}. \qquad (5.13)$$

Using (5.13) in (5.11), we can express the matrices M, N_p and N_v as

$$M = K_p, \qquad (5.14)$$

$$N_p = K_p + T^{-1}K_v, \qquad (5.15)$$

$$N_v = T^{-1}I_n + 2K_v + TK_p. \qquad (5.16)$$

In addition, the substitution of (3.10) into (5.9) yields

$$L = I_n + TK_v. \qquad (5.17)$$

Consequently, the feedback gain matrices in (5.4) and (5.5) are given by (5.14)-(5.17) which explicitly include the feedback gain matrices K_p and K_v.

It has been shown that the closed loop eigenvalues achieved by the state feedback law (5.4) and (5.5) with the solution of the matrix equations (5.8) and (5.9) consists of those of the matrix (3.14) and $2n$ zeros[7]. Since the eigenvalue assignment is same as that obtained by the predictor-based PD type controller with the feedback gain matrices K_p and K_v, we can conveniently use the simple formulas (3.16) and (3.17) to determine the feedback gains in the PID type controller.

Although the controller given by (5.4) and (5.5) does not contains the state predictor, we can show that the algorithm with the matrices satisfying (5.8) and (5.9) includes the state predictor. It follows readily from the result given in [7] that the algorithm (5.4) can be rewritten as

$$\delta(k+1) = s(k+1) - N^0[\Phi_T e(k) + \Gamma_T \delta(k)], \qquad (5.18)$$

where the matrix N^0 satisfies the relations

$$N = N^0 \Phi_T, \quad L = N^0 \Gamma_T. \qquad (5.19)$$

The existence of the matrix N^0 satisfying (5.19) is guaranteed[7]. Note that the last term in (5.18) contains the one-step ahead prediction of the tracking error. Consequently, the controller given by (5.4) and (5.5) with the matrices satisfying (5.8) and (5.9) can be regarded as a predictor-based controller.

6. SIMULATION EXAMPLE

To illustrate the performance of the proposed controllers, we present simulation results for a simple robot manipulator. Consider a planar two-link manipulator shown in Fig. 1, which has two revolute joints and the motion is restricted in the X-Y plane. We assume that the direction of the gravity is negative direction of the Y axis. The length and the mass of the each link are given by

$$l_1 = l_2 = 0.4[\mathrm{m}],\ m_1 = 5.4[\mathrm{kg}],\ m_2 = 3.6[\mathrm{kg}]. \qquad (6.1)$$

In this simulation, the dynamic behavior of the robot manipulator is computed by the fourth order Runge-Kutta method[3] with the discretization step 0.1[msec]. We assume that the digital controllers are implemented using a processor with the sampling period T=10[msec].

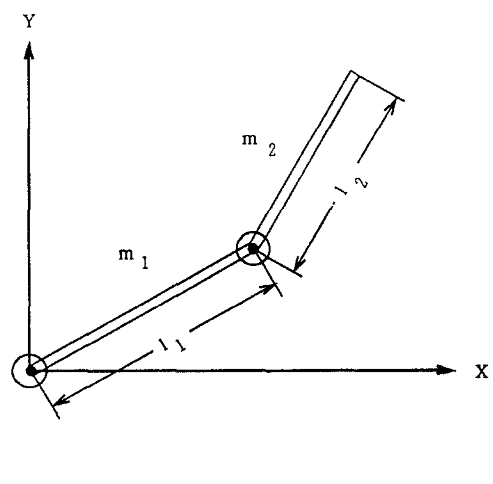

Fig. 1 Two-link planar robot manipulator

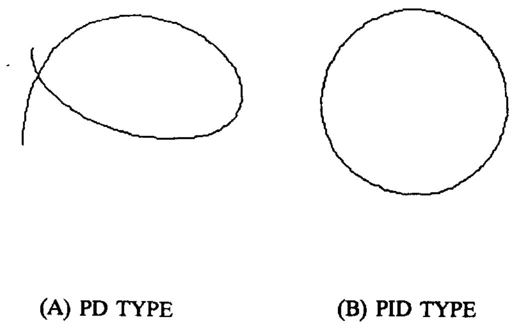

Fig. 2 Comparison of trajectories

The reference trajectory for the tip of the manipulator in the X-Y plane is given by

$$X(t) = 30 - 10\cos\{10V[t - T(1 - e^{-\frac{t}{T}})]\},$$
$$Y(t) = 55 + 10\sin\{10V[t - T(1 - e^{-\frac{t}{T}})]\}, \qquad (6.2)$$

where

$$V = 20\pi/199\ [\mathrm{m/sec}]. \qquad (6.3)$$

The above reference trajectory requires that the tip of the manipulator draws a circle with the radius 10[cm] centered at the point (30, 55)[cm] in 2 seconds.

It is found by numerical simulations that the controllers disregarding the unit computation delay T=10[msec] can not be used effectively under the existence of the delay. Even if the correct physical parameter is used for the design, the controllers disregarding the computation delay can stabilize the closed loop system only for small feedback gains. To implement a digital controller with the rather slow sampling period T=10[msec], the designs accounting the computation delay are required.

The PD and PID type controllers accounting the computation delay are designed using erroneous data. For simplicity, we assume that the errors have uniform percentage for all the physical parameters. In the case that no modelling error exists, the exact tracking to the reference trajectory can be achieved by both controllers. For 10 percent overestimation error of all the physical parameters and the feedback gain matrices

$$K_p = diag\,[20 \quad 20],\quad K_v = diag\,[10 \quad 10], \qquad (6.4)$$

comparison of the trajectories in the X-Y plane and that of the evolution of the tracking errors are shown in Fig. 2 and Fig. 3, respectively. The effect of the parameter error on the maximum tracking errors in the X-Y plane is summarized in Fig. 4 for the feedback gain matrices (6.4) and

$$K_p = diag\,[2 \quad 2],\quad K_v = diag\,[1 \quad 1]. \qquad (6.5)$$

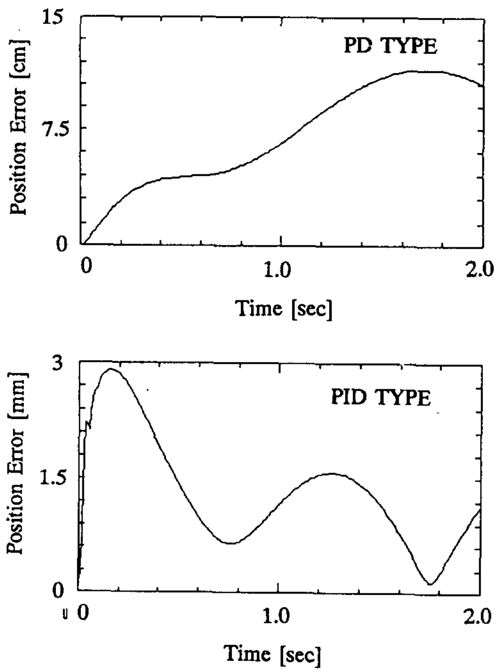

Fig. 3 Evolution of the tracking errors

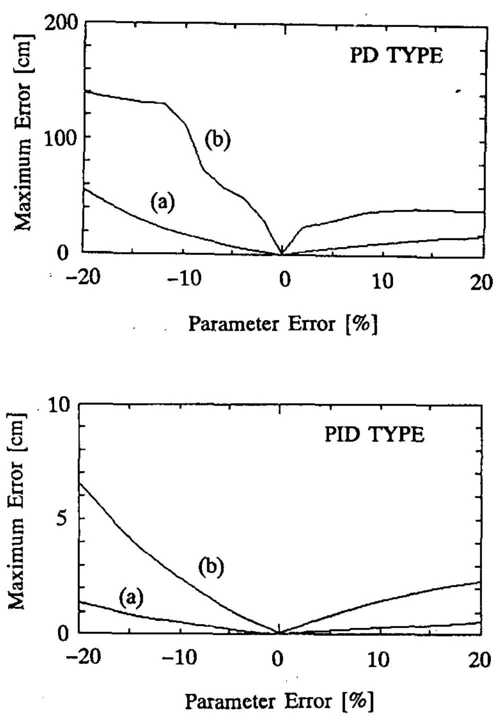

Fig. 4 Parameter error versus maximum position error
(a) K_p=*diag* [20 20], K_v=*diag* [10 10]
(b) K_p=*diag* [2 2], K_v=*diag* [1 1]

Apparently, the PID type controller is superior to the PD type controller in the robustness against the physical parameter variations. Note that both controllers assign the same eigenvalues. The PID type controller designed by use of the feedback gain matrix (6.5) provides more robust performance than the PD controller with the feedback gain matrices (6.4).

7. CONCLUSIONS

Based on the simple discrete-time model of a robot manipulator obtained by Euler's method, we have discussed the direct digital designs of the PD and PID type computed torque controllers. Considering the digital implementation, we have proposed the designs accounting the computation delay. The resulting controllers contain the state predictor to compensate the delay. For both types of controllers, the identical simple formulas are useful to determine the feedback gains. The effectiveness of the proposed designs has been demonstrated by the simulation. In particular, the simulation results have shown the remarkable robustness of the PID type controller against physical parameter variations.

ACKNOWLEDGEMENTS

The author would like to express his thanks to Prof. H. Takeda and Prof. H. Inooka for their continuing encouragement. He is also grateful to H. Kameyama and K. Saito for their assistance in the simulation study.

REFERENCES

[1] J.J. Craig,"Introduction to Robotics Mechanics and Control," Addison-Wesley, 1989.

[2] P. Neuman and D. Tourassis, "Robust discrete nonlinear control for robot manipulator," J. Robotic Systems, Vol. 4, No. 1, pp. 115-143, 1987.

[3] S.D. Conte and Carl de Boor, "Elementary Numerical Analysis," MacGraw-Hill, 1965.

[4] T. Mita, "Optimal digital feedback control systems counting computation time of control laws," IEEE Trans. Automat. Control, Vol. AC-30, No. 6, pp. 542-548, 1985.

[5] T. Mita and M. Mukaida, "On the design of optimal digital servosystem," Trans. Society of Instrument and Control Engineers, Vol. 19, No. 3, pp.7-13, 1983.

[6] T. Ishihara, "Robust stability bounds for a class of discrete-time regulators with computation delays," Automatica, Vol. 24, No.5, pp. 697-700, 1988.

[7] H. J. Guo, T. Ishihara and H. Takeda, "An observer-based design of discrete-time type 1 servosystem," Trans. Society of Instrument and Control Engineers, Vol. 26, No. 2, pp.159-166, 1990.

[8] T. Ishihara, H. J. Guo and H. Takeda, "A design of discrete-time integral controllers with computation delays via loop transfer recovery," to appear in Automatica.