論文 88-25-8-8

Dynamic Modeling and Real-Time Implementation of Control Algorithms for a SCARA Type Robot

(스카라형 로보트의 동적 모델링과 제어 알고리듬들의 실시간 구현)

劉 同 相* 鄭 明 振* 卞 增 男*

(Dong Sang Yoo, Myung Jin Chung and Zeung Nam Bien)

要 約

본 논문에서는 스카라형 로보트를 대상으로 로보트 머니퓰레이터의 동적 특성을 이용한 제어 알고리듬들의 실시간 제어를 구현하였으며 그 효용성을 알아 보았다. 이를 위하여, 스카라형 로보트와 구동기가 조합된 동적 특성을 유도하였다. 실험결과는 동적 특성을 이용한 제어 알고리듬들이 일반적인 PID 제어보다 좋은 결과를 보이고 있으며, 산업용 머니퓰레이터를 제어하는데 효과적으로 이용될 수 있음을 보여 주었다.

Abstract

In this paper, we explore real-time implementation of various dynamic control algorithms, which use the different levels of the information of the dynamics, using a SCARA type robot to show the feasibility and effectiveness of such algorithms. For these purposes, the kinematics of the SCARA type robot is analyzed and the manipulator-actuator dynamic equations based on Lagrange mechanisms are derived. Experimental results indicate that computed torque technique and iterative learning control methods perform better than classical PID control and that these algorithms can be effectively applied to controlling industrial manipulators.

I. Introduction

As robot manipulators are increasingly used for repetitive operations such as arc-welding and cutting, they should have capability to track continuous paths as closely as possible. However, due to the mechanical characteristics of a robot manipulator such as high nonlinearity and coupling effects between joints, it is difficult to control the robot manipulator precisely by conventional control methods in high speed. To overcome these difficulties, various control algorithms using dynamic characteristics of the robot manipulator have been proposed.[1-7] But in practice, due to unknown parameters such as backlash, friction, and nonrigidity, accurate dynamic model can not

*正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electrical Eng., KAIST)

接受日字：1988年 4月 4日

be obtained and also it is quite difficult to implement in real-time because the calculation time increases as the degree of freedom of the robot manipulator increases. Therefore, the performance of dynamic control algorithms have been proved only by computer simulations except a few cases.[11,12]

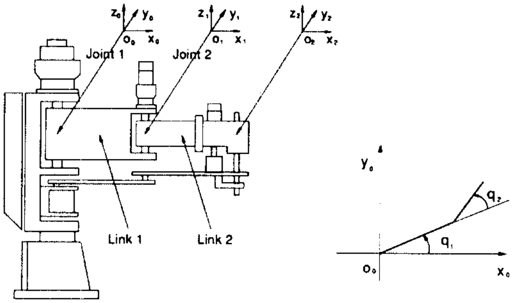
In this paper, using a SCARA type robot, we explore real-time implementation of various dynamic control algorithms developed for the robot manipulator to show the feasibility and the effectiveness of such algorithms. For these purposes, firstly, the kinematics of the SCARA type robot is analyzed. Secondly, the dynamic equations are derived and converted to the equivalent dynamics with reference to the actuator and finally added to the actuator dynamics.[8] Other important terms in dynamics such as viscous friction and coulomb friction, which can not be easily obtained using the above method, are found by the experiments. Thirdly, a hysteresis current controller is used for the torque servo control. This controller utilizes the property that the torque developed by the actuator is proportional to the armature current of the actuator.

In Section 2, the mechanics for the SCARA type robot is derived. Also to compare the performance of various control algorithms, classical PID, computed torque technique, and iterative learning control algorithms are briefly mentioned. Real-time experiment and its results on tracking accuracy under various speed conditions in the Cartesian space are discussed in Section 3 and 4, respectively.

II. Mechanics and Control Algorithms

In this paper, since two degrees of freedom are sufficient to investigate the dynamic effect, only two joints of the SCARA type robot, as shown in Fig. 1-a, are used and its mechanics is studied. The link coordinate systems are established by the Denavit-Hartenberg representation and the corresponding link parameters are derived as shown in Table 1. The homogeneous matrix A^i_{i-1} which specifies the location of the ith coordinate system with respect to the i-1th coordinate system is defined as follows.

$$A^i_{i-1}=\begin{bmatrix} \cos q_i & -\sin q_i & 0 & l_i \cos q_i \\ \sin q_i & \cos q_i & 0 & l_i \sin q_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$



a. SCARA type robot b. Top view

Fig.1. SCARA type robot and coordinate systems.

Table 1. Link parameters.

Joint	q (°)	α (°)	d (mm)	ℓ (mm)
1	0	0	0	370
2	0	0	0	230

1. Kinematics

The kinematics of a robot manipulator is the relationship between the Cartesian space and the joint space without regard to the forces which act or react to it and is classified into two categories, the direct and inverse kinematics.

First, the direct kinematics of the robot manipulator is the representation of manipulator position from the joint space into the Cartesian space. From Fig. 1-b, the direct kinematics of the SCARA type robot is geometrically obtained as follows:

$$\begin{aligned} P_x &= l_2 \cos(q_1+q_2) + l_1 \cos q_1 \\ P_y &= l_2 \sin(q_1+q_2) + l_1 \sin q_1 \end{aligned} \qquad (2)$$

where P_x and P_y are x, y components of the tip of the robot hand in the Cartesian space, respectively. q_1, q_2 are joint values of the joint space. l_1, l_2 are link lengths of the SCARA type robot.

Secondly, the inverse kinematics calculates all possible sets of joint values from the given values of the Cartesian space. In case of the SCARA type robot, the inverse kinematics is algebraically obtained by Eq. (2).

Eq. (2) can be rewritten as follows:

$$l_2 \cos q_1 \cos q_2 - l_2 \sin q_1 \sin q_2 + l_1 \cos q_1 = P_x \qquad (3)$$

$$l_2 \sin q_1 \cos q_2 + l_2 \cos q_1 \sin q_2 + l_1 \sin q_1 = P_y \qquad (4)$$

After multiplying cos q_1, sin q_1 to Eq. (3), (4), respectively, the combination of two equations can be rewritten as follows:

$$l_2\cos q_2 + l_1 = P_x\cos q_1 + P_y\sin q_1 \qquad (5)$$

Similarly, by multiplying sin q_1 cosq_1 to Eq. (3), (4), respectively, another equation can be obtained.

$$l_2\sin q_2 = -P_x\sin q_1 + P_y\cos q_1 \qquad (6)$$

Combining the squares of Eq. (5), (6), we can obtain the cosine and sine functions for q_2

$$\cos q_2 = \frac{P_x^2 + P_y^2 - l_1^2 - l_2^2}{2 l_1 l_2} \qquad (7)$$

$$\sin q_2 = \mathrm{ARM}\sqrt{1 - \cos^2 q_2} \qquad (8)$$

where ARM is the index which represents the arm configuration of the SCARA type robot

$$\mathrm{ARM} = \begin{cases} +1 & \text{if right arm} \\ -1 & \text{if left arm} \end{cases}$$

Hence, from Eq. (7) and (8), q_2 is found to be

$$q_2 = \mathrm{atan2}\left[\frac{\sin q_2}{\cos q_2}\right] \qquad (9)$$

To find q_1, multiplying P_x, P_y to Eq. (5), (6) and combining two equations, the cosine function is as follows:

$$\cos q_1 = \frac{P_x l_2\cos q_2 + P_x l_1 + P_x l_2 \sin q_2}{P_x^2 + P_y^2} \qquad (10)$$

Similarly, the sine function is obtained.

$$\sin q_1 = \frac{P_y l_2\cos q_2 + P_y l_1 - P_x l_2 \sin q_2}{P_x^2 + P_y^2} \qquad (11)$$

Therefore, q_1 is found to be

$$q_1 = \mathrm{atan2}\left[\frac{\sin q_1}{\cos q_1}\right] \qquad (12)$$

2. Dynamics

It is well known that the dynamic equations for a robot manipulator with n degrees of freedom based on Lagrangian formulation can be described as follows:

$$\tau = D(q)\ddot{q} + H(q\cdot\dot{q}) + V\dot{q} + G(q) \qquad (13)$$

where q is an nx1 joint angle vector. D(q) is an n×n symmetric and nonsingular inertia matrix. H(q,$\dot{q}$) is an n×1 vector specifying centrifugal and Coriolis terms. V is an nxn diagonal matrix specifying the viscous friction coefficients. G(q) is an nx1 vector specifying the effects due to gravity and external load. τ is an nx1 generalized input torque/force vector.

In this paper, to derive the dynamic equations of the SCARA type robot, each link is modeled to the I type beam having the uniformly distributed mass (in Fig. 2) and geometric parameters for links are shown in Table 2.

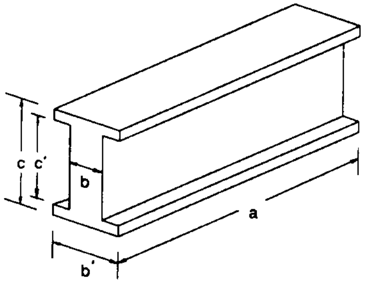


Fig.2. Link model.

Table 2. Geometric parameters for links.

Parameter	Link 1	Link 2
a (mm)	370	230
b (mm)	33	30
b′ (mm)	47	40
c (mm)	145	94
c′ (mm)	101	51
Mass (kg)	7.5	4.0

Since joint axes of the robot arm are parallel to the gravitational force, G(q) can be omitted from Eq. (13) and solving the inertia matrix of each link must be preceded in order to derive D(q) and H(q,$\dot{q}$). The inertia matrix, J_i, of the *i*th link is as follows:

$$J_i = \begin{bmatrix} \rho_i \int_v x_i^2 \, dV_i + m_i \left(\frac{a_i}{2}\right)^2 & 0 & 0 & -m_i\left(\frac{a_i}{2}\right) \\ 0 & \rho_i \int_v y_i^2 \, dV_i & 0 & 0 \\ 0 & 0 & p_i \int_v z_i^2 dV_i & 0 \\ -m_i\left(\frac{a_i}{2}\right) & 0 & 0 & m_i \end{bmatrix} \tag{14}$$

where ρ_i, V_i are the density and the volume of the ith link, respectively. $\rho_i \int_V x_i^2 dV_i$, $\rho_i \int_V y_i^2 dV_i$, and $\rho_i \int_V z_i^2 dV_i$ are the volume integrals about the center of mass of the ith link and are found to be:

$$\begin{aligned} \rho_i \int_v x_i^2 \, dV_i &= \frac{M_i}{12} a_i^2 \\ \rho_i \int_v y_i^2 dV_i &= \frac{M_i b_i^3 c_i' + b_i^{3\prime} (c_i - c_i')}{12 b_i c_i' + b_i' (c_i - c_i')} \\ \rho_i \int_v z_i^2 dV_i &= \frac{M_i \quad M_i \, b_i \, c_i^{3\prime} + b_i' (c_i^3 - c_i^{3\prime})}{12 b_i c_i' + b_i' (c_i - c_i')} \end{aligned} \tag{15}$$

From J_i matrix, the components of $D(q)$, $H(q,\dot{q})$ are obtained as follows:

$$D_{ij} = \sum_{k=\max(i,j)}^{2} Tr(U_{kj} J_k U_{ki}^t) \qquad \text{for } i, j = 1, 2 \tag{16}$$

$$H_i = \sum_{j=1}^{2} \sum_{k=1}^{2} H_{ijk} \dot{q}_j \dot{q}_k \tag{17}$$

$$H_{ijk} = \sum_{m=\max(i,j,k)}^{2} Tr(U_{mjk} J_m U_{mi}^t) \qquad \text{for } i, j, k = 1, 2 \tag{18}$$

where $U_{kj} = \partial A_0^k / \partial q_j$, $U_{mjk} = \partial q_k$, and Tr denotes trace of a matrix. The superscript "t" denotes transpose of a vector or a matrix.

In general, to implement proper control algorithms, the above dynamic equations for the SCARA type robot should be converted to the equivalent dynamics with respect to the actuator.[8,10]

Each joint of the SCARA type robot is driven by a permanent magnet DC servo motor whose characteristics are shown in Table 3.

Let the motor shaft angular position be θ_k, the angular speed be ω_k of the DC servo motor at joint k (k=1,2). The manipulator-actuator dynamics at the actuator shaft can be described as the following equation:

Table 3. Parameters for actuators (PM DC motor).

Item	Joint 1	Joint 2
Rated output(W)	200	60
Rated torque(kg·cm)	6.5	1.95
Rated speed(rpm)	3000	3000
Torque constant(kg·cm/A)	1.21	0.584
Rotor inertia(g·cm·sec²) (superscript)	1.71	0.16
Viscous friction coefficient (g·cm/rpm)	0.27	0.06
Mechanical time constant(msec)	4.8	5.3
Gear ratio	1 : 157	1 : 120

$$T = J_m \dot{\omega} + B_m \omega + F_c(\omega) + \tau^* \tag{19}$$

where

$$\theta = [\theta_1, \theta_2]^t, \quad \omega = [\omega_1, \omega_2]^t,$$
$$T = [T_1, T_2]^t, \quad \tau^* = [\tau_1^*, \tau_2^*]^t$$

are 2x1 vectors representing the angular positions of the motor, the angular velocities of the motor, the torques developed by the motor, and the load torques at motor shaft, respectively. J_m,B_m are 2x2 diagonal matrices whose entries represent the moments of inertia, the viscous friction coefficients of DC motors, respectively. $F_c(\omega)$ is a 2x1 coulomb friction vector, the kth component of which is f_{ck} sign(ω_k) where f_{ck} is constant. Each joint of the SCARA type robot is coupled with the harmonic gear. Because if friction and backlash between the coupled gear teeth are assumed to be negligible, the generalized torque/force of the SCARA type robot, transferred to the motor through the harmonic gear, is equal to the load torque, τ^*, gear ratio nk can be written as follows:

$$n_k = \frac{\tau_k^*}{\tau_k} = \frac{q_k}{\theta_k} \tag{20}$$

Let N = [n_k] (k=1,2) be a 2x2 diagonal matrix, then $\tau^* = N\tau$. On the other hand, the torque developed by the permanent magnet DC motor is directly proportional to the armature current. thus,

$$T = K_T \cdot I_a \tag{21}$$

where

$$I_a = [i_{a1}, i_{a2}]^t$$

is a 2x1 armature current vector. K_T is a 2x2 diagonal matrix and each diagonal element of K_T represents the torque constant. After using the joint angular position vector q as variables in the joint space, the manipulator-actuator dynamic equations can be finally rewritten as follows:

$$I_a = K_T^{-1}\{J\ddot{q} + B\dot{q} + NH(q,\ \dot{q}) + F_c(\dot{q})\} \qquad (22)$$

where

$$J = J_m N^{-1} + ND(q),\quad B = B_m N^{-1} + NV$$

are 2x2 diagonal matrices whose entries represent the moments of inertia, the viscous friction coefficients of the combined system, respectively. On the other hand, because the coulomb friction is solely determined by the direction of the motion, the kth component of $F_c(\dot{q})$ is:

$$f_{ck}\,\mathrm{sign}(\dot{q}_k) = f_{ck}\,\mathrm{sign}(\omega_k) \qquad (23)$$

In most cases, B, $F_c(\dot{q})$ terms are dominant but they can not be easily obtained analytically. These terms are found by simple experiment. If the manipulator is allowed to move only one joint at a time by locking the other joint, the dynamic equation for a single joint is described as follows:

$$i_{ak} = K_{Tk}^{-1}\{J_k\ddot{q}_k + B_k\dot{q}_k + f_{ck}\,\mathrm{sign}(\dot{q}_k)\} \qquad (24)$$

When angular speed reaches steady state for constant current command, Eq. (24) can be reformulated as the following equation:

$$i_{ak} = K_{Tk}^{-1}\{B_k\dot{q}_k + f_{ck}\,\mathrm{sign}(\dot{q}_k)\} \qquad (25)$$

If the joint angular speed is measured, then B_k and f_{ck} are easily obtained from the above equations. According to Eq. (22), the parameter values of the dynamic model for the SCARA type robot are shown in Table 4.

Table 4. Model parameters for the SCARA type robot.

Item	Joint 1	Joint 2
$J\,(\mathrm{N\ m\ sec^2})$	$J_{11} = 3.18\times10^{-2} + 2.16\times10^{-3}\cdot cq_2$ $J_{12} = 4.90\times10^{-4} + 1.08\times10^{-3}\cdot cq_2$	$J_{21} = 5.88\times10^{-4} + 1.37\times10^{-3}\cdot cq_2$ $J_{22} = 2.45\times10^{-3}$
$B\,(\mathrm{N\ m\ sec})$	2.94×10^{-1}	1.47×10^{-1}
$NH(q,\dot{q})\ (\mathrm{N\,m})$	$-(2.17\times10^{-3}\cdot sq_2\cdot\dot{q}_1\cdot\dot{q}_2 + 1.08\times10^{-3}\cdot sq_2\cdot\dot{q}_2^2)$	$1.42\times10^{-3}\cdot sq_2\cdot\dot{q}_1^2$
$F_c(\dot{q})\ (\mathrm{N\,m})$	$8.33\times10^{-2}\ \mathrm{sign}(\dot{q}_1)$	$5.39\times10^{-2}\ \mathrm{sign}(\dot{q}_2)$

$sq_2 = \sin q_2\quad cq_2 = \cos q_2$

3. Trajectory Planning

A common way of causing a robot manipulator to move from one point to another point in a smooth, controlled fashion is to cause each joint to move as specified by a smooth function of time. How to compute exactly these motion functions is the problem of trajectory planning. In order to force the end-effector to follow a geometric shape through the Cartesian space the desired motion must be converted to an equivalent set of joint motions.[4] In this paper, the desired motion is established to a circle in the Cartesian space. The circle trajectory is one of widely used trajectories in which the effects of the coulomb friction can be observed when direction is reversed. Also, we can observe the effects of the moment of inertia which become the dominant term in the dynamic equations when starting and stopping.

The block diagram for the trajectory generation is as in Fig. 3

In this planning, ϕ is a linear function, which is joined with the 5th order polynomial function at the beginning and end of the motion, and means the counter clockwise angle about the circle. $d\phi$ is the increment of ϕ during one sampling time, dt. $d\phi_f$ is the maximum value of $d\phi$ and equal to dt. *speed*/*R* where *R* and *speed* are the radius of circle and the maximum speed in the Cartesian space, respectively.

4. Control Algorithms

The purpose of manipulator control is to maintain the dynamical response of a robot manipulator in accordance with some prespecified system performance and desired goals based on the dynamic model. To achieve these goals, many control algorithms for the robot manipulator have been proposed.[1-7]

In this paper, we briefly discuss three control algorithms which use the different levels of information of the dynamics of a robot manipulator.

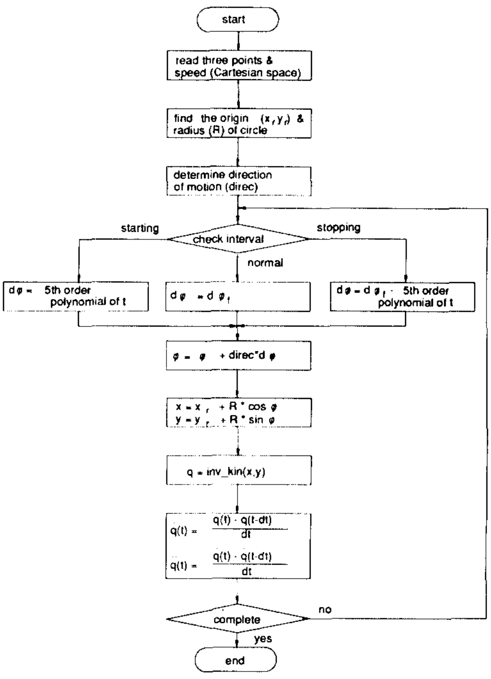

start
read three points & speed (Cartesian space)
find the origin (x_r, y_r) & radius (R) of circle
determine direction of motion (direc)
starting
check interval
stopping
normal
dφ = 5th order polynomial of t
dφ = dφ$_f$
dφ = dφ$_f$ - 5th order polynomial of t
φ = φ + direc*dφ
x = x$_r$ + R * cos φ
y = y$_r$ + R * sin φ
q = inv_kin(x,y)
q̇(t) = (q(t) - q(t-dt)) / dt
q̈(t) = (q̇(t) - q̇(t-dt)) / dt
complete
no
yes
end

Fig.3. Block diagram for the circle trajectory.

1) PID Control Algorithm

Since PID control can provide quick response, good system stability, and small steady-state errors in linear system, this control method is widely used.

Let q^d be the desired joint angle and q be the actual joint angle. Assuming that each joint is completely decoupled and independently controlled, the control input u can be described as follows:

$$u = K_V(\dot{q}^d - \dot{q}) + K_P(q^d - q) + K_I \int (q^d - q)\,dt \quad (26)$$

where K_V, K_p and K_I are 2x2 diagonal matrices which represent the derivative gain, the proportional gain, and the integration gain, respectively.

2) Computed Torque Technique

The problems which lead to the difficulties of controlling a robot manipulator are inherent nonlinearity and high coupling. These mean that much of linear control theory are not directly applicable. Thereby, computed torque techniques which are based on either the Lagrange-Euler formulation or the Newton-Euler formulation have been proposed for robot control. Here, computed torque technique based on the Lagrange-Euler formulation is implemented.

The control input is:

$$u = K_T^{-1}\{J[\ddot{q}^d + K_V(\dot{q}^d - q) + K_P(q^d - q)] + B\dot{q} + NH(q, \dot{q}) + F_C(\dot{q})\} \quad (27)$$

where K_p is a 2x2 proportional gain matrix, and K_V is a 2x2 derivative gain matrix.

3) Iterative learning control algorithms

For the motion control of a robot manipulator, various learning control algorithms have been proposed recently to overcome the difficulty, i.e. modeling error, which appears in the previous control methods. In this paper, the following learning control algorithms are discussed and implemented.

Betterment Process: Arimoto and his colleagues proposed a learning control method called "Betterment Process".[5,6] It consists of a linear PD feedback control which make the robot motion follow the vicinity of given desired trajectory and additional input constructed by learning to reduce the deviation between the desired trajectory and the actual robot motion. Here, the desired trajectory is given in angular velocity profile and the convergency of this algorithm was proved in the sense that the integrations of squared errors decreased through iterative operations.

The input to control the SCARA type robot at the jth operation is given as follows:

$$u^j = K_P(q^d - q^j) + K_V(\dot{q}^d - \dot{q}^j) + v^j \quad (28)$$

where K_p is a 2x2 position feedback gain matrix, and K_V is a 2x2 velocity feedback gain matrix. And the additional input v^j has the following form:

$$v^j = v^{j-1} + \phi(\dot{q}^d - \dot{q}^{j-1}) \quad (29)$$

where ϕ is a 2x2 velocity learning gain matrix.

Model Algorithmic Learning Control: Another type of an iterative learning control method was proposed for dynamic systems with uncertain parameters.[7] The method, which adopted the model algorithmic control concept[13] in the iteration sequence was shown to be convergent for linear time-varying systems and to be applicable for robot control systems.

For the control of SCARA type robot, the control algorithm is as follows:

Define the 4 dimensional state vector X

$$X = (q^t, \dot{q}^t)^t \qquad (30)$$

When X_d is the desired trajectory and the *j*th model state, $X^{j,M}$, is updated by the following method:

$$X^{j,M} = X^{j-1,M} + S\,(X^d - X^{j-1}) \qquad (31)$$

the control input at the *j*th operation is determined as follows:

$$u^j = K_T^{-1}\,\{[B^M, J^M]\,\dot{X}^{j,M} + F_C(\dot{q}^d)\} \qquad (32)$$

where S is a 4x4 constant weighting matrix.

III. Implementation

Fig. 4 represents the hardware structure for evaluating the dynamic control algorithms.

Functions of each block are described as follows:

A 16 bit super-microcomputer, SSM-16, based on MC68000, is used as a host computer. Under the UNIX operating system, the host computer is used for the development of the control algorithms and loads execution program in common memory. Common memory stores the execution program and input-output data, which can be accessed through VME-bus. The size of common memory is 256 Kbyte. An MVME 110-1 single board computer with a floating point coprocessor MC68881 is used as the servo computer for calculating the desired trajectory and the control input by the execution program in common memory.

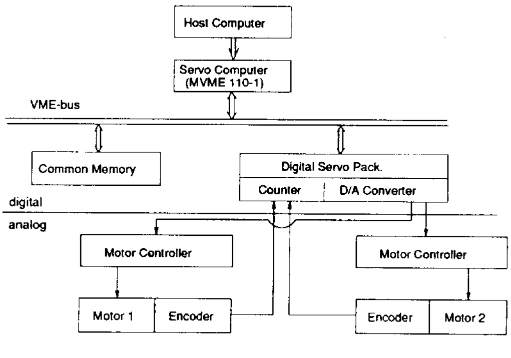
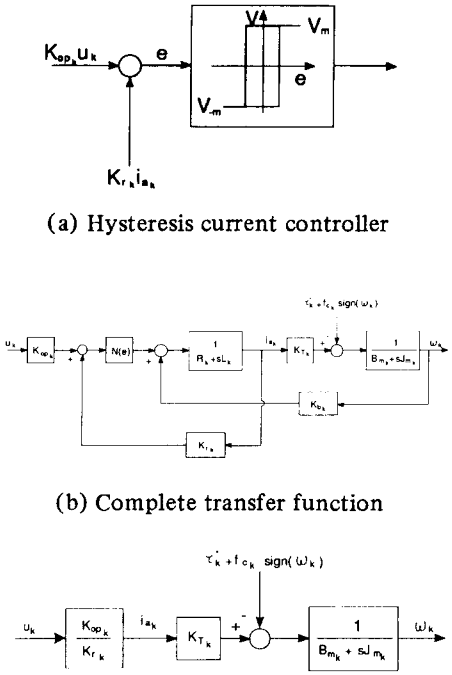


Fig.4. Hardware structure.

A hysteresis current controller[9] which can control and limit the armature current of the motor is used as the motor controller. The hysteresis current controller is a nonlinear element and its output depends on the signal *e* as shown in Fig. 5-a. Let N(e) be the describing function which describes the behavior of the hysteresis current controller. The overall block diagram of the motor and the motor controller is shown in Fig. 5-6. Since N(e) is very large, i_{ak} can be approximated by $u_k K_{opk}/K_{rk}$ where K_{opk} is the operational amplifier gain. As a result, the transfer function can be simplified as shown in Fig. 5-c. If K_{rk} is set to be equal to K_{opk}, then i_{ak} is directly replaced by u_k.

Digital servo package consists of three parts. Those are feedback circuit, D/A conversion circuit, and emergency check circuit. Firstly, feedback circuit counts the pulse trains generated



(a) Hysteresis current controller

(b) Complete transfer function

(c) Simplified transfer function

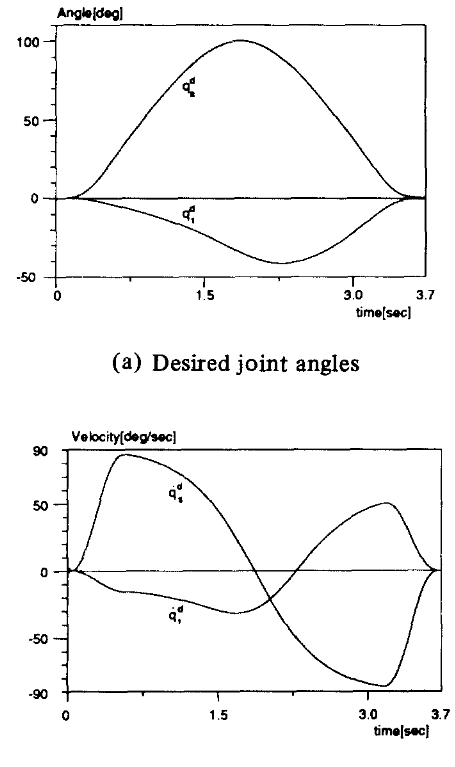
Fig.5. Block diagram of current controller and motor at the *k*th joint.

from the encoder of the motor for motor position sensing and checks the direction of the motor motion. Secondly, the control input calculated by the servo computer is transformed into analog signal by D/A conversion circuit, and is transferred to the motor controller. Thirdly, emergency check circuit checks the emergency signals such as overrun, overcurrent, and overheat which may be generated during the robot motion.

IV. Experiment and Evaluation

Now, in order to evaluate the performance of each control algorithm, the desired trajectory is the counter clockwise circle whose radius is 100 mm, and speed is 200 mm/sec and desired trajectories transformed into the joint space are as shown Fig. 6. The sampling time is set to 15 msec.[14]

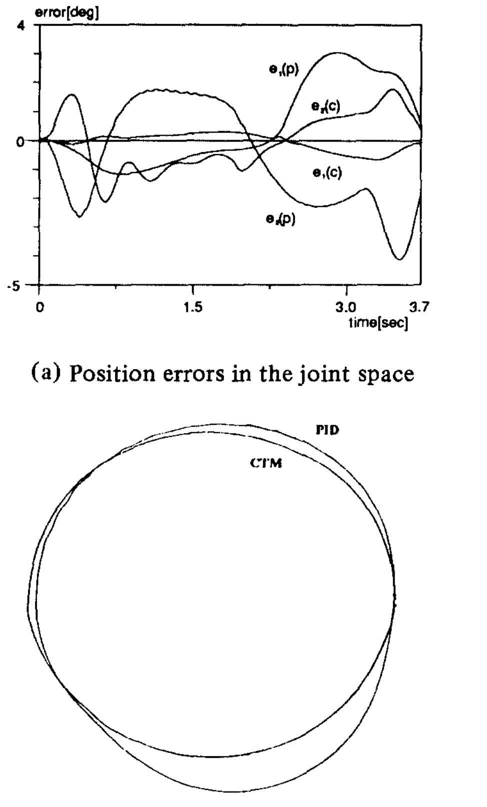
Firstly, experimental results on both classical PID and computed torque technique were compared. The gains for PID control are K_v=diag [11, 20.5], K_p = diag [41, 82], and K_I=diag[70, 110]. And K_p=diag[850, 1200], and K_v=diag [58.3, 69.3] for computed torque technique. Fig. 7-a shows the position error in the trajectory of joint angle when speed is 200 mm/sec in the Cartesian space. The position error of each joint of computed torque technique is smaller than that of classical PID control. The position error of joint 2, which is dominantly influenced by the coupling effects between joints in the SCARA type robot is larger than that of joint 1. The tracking result for the desired circle in the Cartesian space is shown in Fig. 7-b. In cases of the other speeds of the desired circle in the Cartesian space, similar



(a) Desired joint angles

(b) Desired angular velocities

Fig.6. Desired trajectories in the joint space (speed = 200 *mm/sec* in the cartesian space).



(a) Position errors in the joint space

(b) Tracking result in the cartesian space (scale = 0.25)

Fig.7. Experimental results (p : PID control, c: computed torque method).

results are obtained as shown in Table 5. From these results, computed torque technique which uses the dynamic information of the robot manipulator have shown better performance than that of PID control.

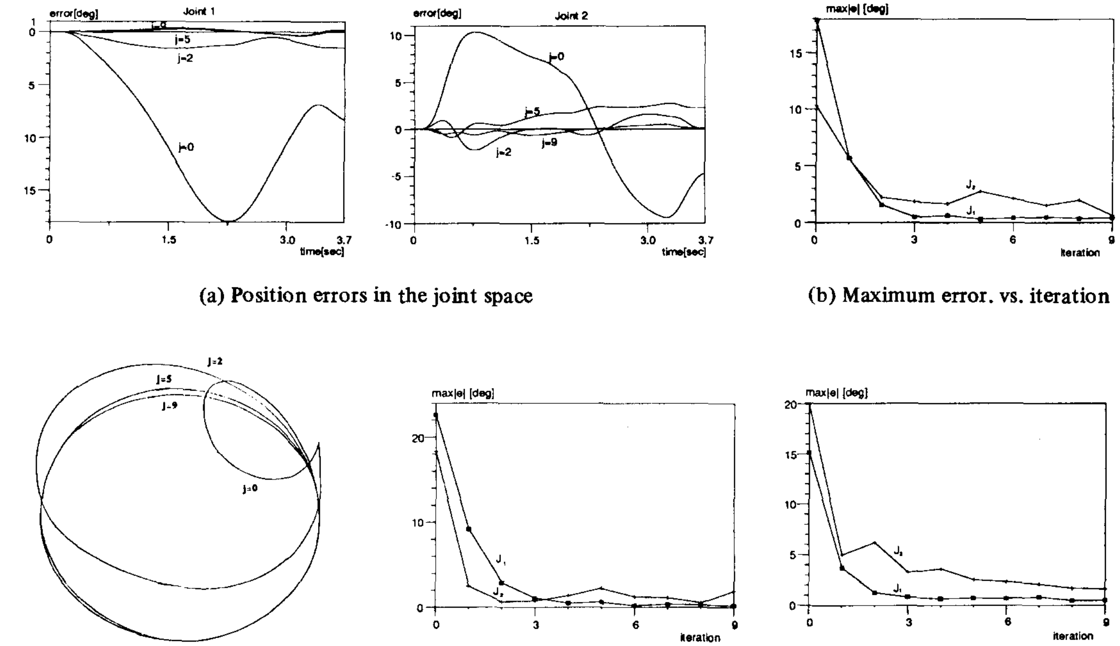
Table 5. Maximum error in the joint space.

	Maximum error (deg)					
Speed (mm/sec)	100		200		300	
Joint	1	2	1	2	1	2
PID	2.0	−1.7	3.0	−4.0	4.6	−7.2
CTM	−0.3	1.5	−0.7	2.2	−1.0	3.5
MALC	−0.5	2.0	−0.5	0.8	−0.7	2.0
BP	2.6	−3.2	2.7	−3.2	2.5	−2.7

Secondly, performance evaluation for iterative learning control algorithms was carried out for various speed conditions. For model algorithmic learning control method, the weighting matrix S is chosen as S = diag[0.95, 0.9, 0.95, 0.9]. And for betterment process, control gains are chosen as K_p=diag[13, 17], K_v=diag[6.5, 8.5], and ϕ =diag[1,3]. In the former case, in spite of inaccuracies of modeling, the robot motion is getting closer to the desired trajectory with fast convergency rate by repeating operations as shown in Figs. 8-a, b and c. In the other cases, the maximum error of each joint is decreasing similarly as in 200 mm/sec with the different convergency rate as shown in Fig. 8-d and e. On the other hand, in betterment process, the actual robot motion converges the desired trajectory as shown in Fig. 9, although the convergency rate is slower than that in the former. The maximum error for each control algorithm in joint space is shown in Table 5. From Table 5, we can find that when using iterative learning control algorithms, the exact dynamic model is not always needed to control a robot manipulator precisely.

V. Conclusion

This paper presented the implementation and experimental result of dynamic control algorithms



(a) Position errors in the joint space

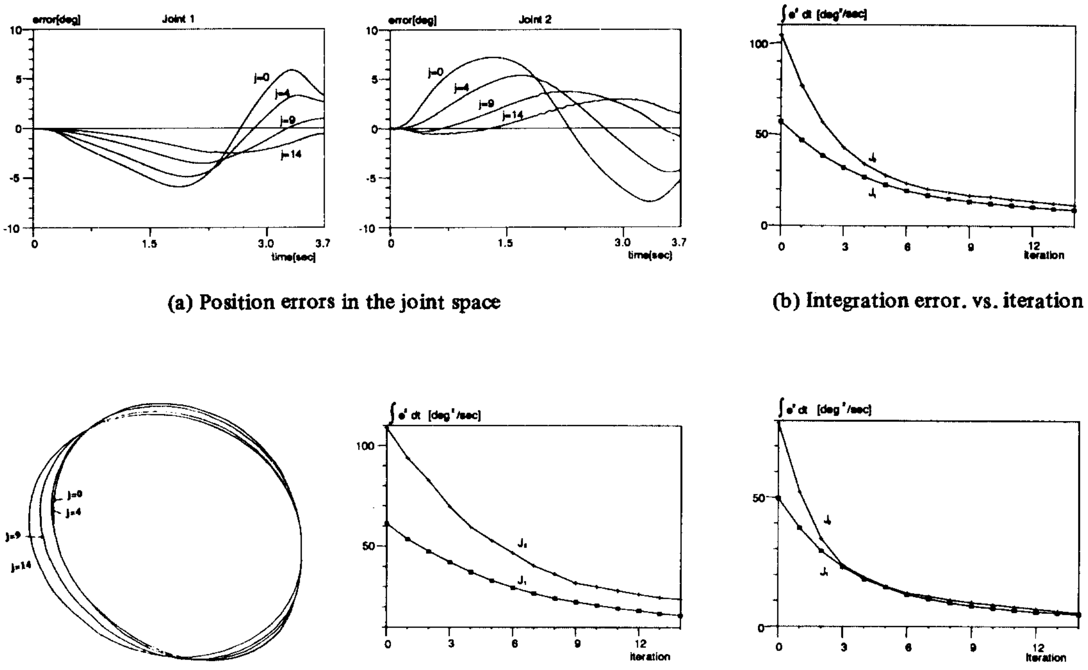
(b) Maximum error. vs. iteration

(c) Tracking result in the Cartesian space (speed = 200 mm/sec, scale = 0.25)

(d) Maximum error. vs. iteration (speed = 100 mm/sec)

(e). Maximum error. vs. iteration (speed = 300 mm/sec)

Fig.8. Experimental results of model algorithmic learning control.



(a) Position errors in the joint space

(b) Integration error. vs. iteration

(c) Tracking result in the Cartesian space (speed = 200 mm/sec, scale = 0.25)

(d) Integration error. vs. iteration (speed = 100 mm/sec)

(e) Integration error. vs. iteration (speed = 300 mm/sec)

Fig.9. Experimental results of betterment process.

using the SCARA type robot. For real-time experiment, the manipulator-actuator dynamics were derived and three control algorithms were selected and compared to show the feasibility and effectiveness of the dynamic control algorithms. From the experimental results, we could confirm that computed torque technique using the dynamic information of the robot manipulator performs better than conventional control methods. Iterative learning control algorithms, which have been proposed to overcome the difficulty of obtaining exact modeling, track the desired continuous path with some accuracy through repetitive operations in various speeds.

Reference

[1] M. Vukobratovic, and D. Stokic, "Is dynamic control need in robotic systems, and, if so, to what extent?," *Int. J. of Robotics Res.*, vol. 2, no. 2, pp. 18-34, Summer 1983.

[2] C.S.G. Lee, R.C. Gonzalez, and K.S. Fu, *Tutorial on robotics*, 2nd Edition, *IEEE Computer Society* Press, 1986.

[3] R.P. Paul, *Robot Mainpulators-Mathematics, Programming, and Control*, MIT Press, 1981.

[4] J.J. Craig, *Introduction to Robotics; Mechanics and Control*, Addison-Wesley Publishing Co., 1986.

[5] S. Arimoto, S. Kawamura, and F. Miyazaki, "Can mechanical robots learn by themselves?," *Proc. of 2nd Int. Symp. Robotics Res.*, pp. 127-134, 1984.

[6] S. Kawamura, F. Miyazaki, and S. Arimoto, "Iterative learning control for robotic systems," *Proc. IECON'84*, Tokyo, Japan, pp. 393-398, 1984.

[7] S.R. Oh, "A study on iterative learning control methods for the robot manipulator," Ph.D. Dissertation, KAIST, 1987.

[8] C.H. Liu, "A comparison controller design and simulation for an industrial mani-

pulator," *IEEE Trans. on IE,* vol. IE-33, no. 1, pp. 58-65, Feb. 1986.

[9] B.H. Kwon, H.J. Park, M.J. Youn, and T.Y. Ahn, "Digital position controller of PM DC motor for improving time optimal control under the inexact load parameter," *Proc. IECON'86,* Milwaukee, Wisconsin, pp. 253-258, 1986.

[10] J.Y.S. Luh, "Conventional controller design forindustrial robots - A tutorial,"*IEEE trans. on SMC,* vol. SMC-13, no. 3, May 1983.

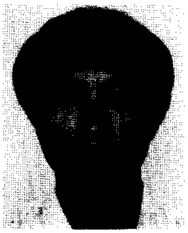
[11] K.P. Valavanis, N.B. Leahy, and G.N. Saridis, "Real-time evaluation control methods," *Proc. of Int. Conf. on Robotics and Automation,* St. Louis, Missouri, pp. 644-649, 1985.

[12] H. Hashimoto, K. Maruyama, and F. Harashima, "A microprocessor-based robot manipulator control with sliding mode," *IEEE Trans. on IE,* vol. IE-34, no. 1, pp. 11-18, Feb. 1987.

[13] R. Rouhani, and R.K. Mehra, "Model algorithmic control (MAC); Basic theoretical properties," Automatica, vol. 18, no. 4, pp. 401-414, 1982.

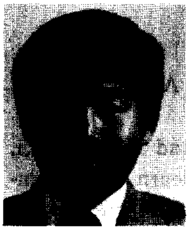
[14] D.S. Yoo, M.J. Chung, and Z. Bien, "Real-time implementation and evaluation of dynamic control algorithms for industrial manipulators," *Proc. IECON'87,* Cambridge, Massachusetts, pp. 26-31, 1987.

著者紹介



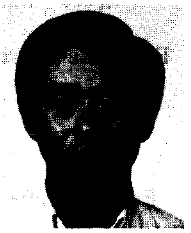
劉 同 相(正會員)

1962年 4月 12日生. 1985年 2月 서울 대학교 전기공학과 공학사학위 취득. 1987年 2月 한국과학기술원 전기 및 전자공학과 공학 석사학위 취득. 1987年 3月～ 현재 한국과학기술원 전기 및 전자공학과 박사과정 재학중. 주관심분야는 로보틱스임.



鄭 明 振(正會員)

1950年 1月 31日生. 1973年 2月 서울대학교 전기공학과 공학사 학위 취득. 1976年 9月～1977年 12月 미국 미시간대학교 전기공학과 공학석사학위 취득. 1978年 1月～1983年 8月 미국 미시간 대학교 전기공학과 공학박사학위 취득. 현재 한국과학기술원 전기 및 전자공학과 조교수. 주관심분야는 로보틱스임.



卞 增 男(正會員)

1943年 10月 11日生. 1969年 2月 서울대학교 전자공학과 공학사 학위 취득. 1972年 Iowa대학 전기과 공학석사학위 취득. 1972年 Iowa 대학 수학과 공학석사학위 취득. 1975年 Iowa 대학 전기과 공학박사학위 취득. 1977年 Iowa대학 전기과 객원조교수. 1981年 한국과학기술원 전기 및 전자공학과 조교수 1982年 Iowa대학 전기과 객원부교수. 현재 한국과학기술원 전기 및 전자공학과 교수. 주관심분야는 자동제어이론, 로보틱스 및 인공지능, 공장 자동화 등임.