Optimum Contouring of Industrial Robot Arms Under Assigned Velocity and Torque Constraints

S. Rohan Munasinghe, *Student Member, IEEE*, Masatoshi Nakamura, *Senior Member, IEEE*, Satoru Goto, and Nobuhiro Kyura, *Member, IEEE*

***Abstract*—This paper presents a general solution to the contouring problem of industrial robot arms, under the constraints of assigned Cartesian velocity and joint torque/acceleration. The proposed solution is an off-line trajectory generation algorithm and, therefore, it possesses significant industrial implications, as no hardware changes are needed for its implementation. According to the proposed method, maximum utility of joint torque/acceleration is guaranteed bound to the assigned velocity constraint. In the proposed method, a realizable trajectory is generated from the objective trajectory and it is compensated for delay dynamics using a forward compensator. The proposed method has been experimented with Performer MK-3s (PMK-3s) industrial robot arm in that optimum contouring has been realized.**

***Index Terms*—Assigned velocity constraint, delay dynamics compensation, industrial robot arm, off-line trajectory generation, torque/acceleration constraint.**

I. Introduction

THE MAIN objective of this paper is to realize the optimum performance of industrial robot arms in contouring operations. Contouring problem under the assigned velocity and torque/acceleration constraints is the subject of this research. The proposed solution is an off-line trajectory generation algorithm so that it could be incorporated with the servo controllers of industrial robot arms without major changes in existing hardware architecture.

The origins of robot manipulators date back to the 1940s, when Walters invented "machina," the first robotic manipulator. Then, Devol invented the first industrial robot, "Unimate" in the late 1950s. Since then, industrial robot arms have proliferated and in today's automated industries, robot arms are well employed in various tasks on the factory floor. There are two application categories where industrial robot arms are usually employed: 1) positioning and 2) contouring. In positioning, the end-effector of the robot arm is moved from a start-point to an end-point, whereas in general, there is no major concern with the course of motion it takes in between the two stated points. A typical positioning application is *pick-and-place*. In contouring, the end-effector is supposed to move along a given Cartesian trajectory, obeying a given course of motion. Some typical contouring applications are welding, cutting, and painting.

Manuscript received October 8, 1999; revised May 7, 2001. This paper was recommended by Associate Editor W. A. Gruver.

S. R. Munasinghe, M. Nakamura, and S. Goto are with the Department of Advanced Systems Control Engineering, Saga University, Honjomachi Saga, Japan (e-mail: rohan@cntl.ee.saga-u.ac.jp).

N. Kyura is with the Department of Electrical and Computer Engineering, Kinki University, Iizuka, Japan.

Publisher Item Identifier S 1094-6977(01)06489-6.

In both categories, high-speed and precise operation is desirable. In positioning, precision refers to how close the end-effector follows the straight-line drawn from the start-point to the end-point; whereas in contouring, precision refers to how close the end-effector follows the realizable trajectory.

In general, robot arm operations are carried out in two phases. The first phase involves off-line trajectory generation, in that, taught data (time-based sequences of joint position) are generated. In the second phase, joint servo motors are simultaneously actuated by the servo controller according to the taught data sequences.

Specifications, constraints, and performance evaluation of industrial robot arms are defined more in Cartesian space than in joint space. Nevertheless, the motion of the robot arm originates at joint servomotors. Furthermore, dynamics are modeled together with torque/acceleration constraints in joint space. Therefore, robot arm operations possess equal considerations in both Cartesian and joint spaces. Coordinate transformation between these two spaces is possible with forward and inverse Jacobian transformations. Using this technique, Witney [1] proposed resolved motion in that joint motions are combined and resolved into separately controllable end-effector motion in Cartesian space. Paul [2] and Taylor [3] explained general trajectory generation methodologies in both Cartesian and joint spaces where knot-point selection, interpolation between knot-points, and issues such as error of interpolation and computational burden in real-time coordinate transformation have been discussed. Luh *et al.* [4] explained trajectory generation with kinematic constraints though it had not been implemented experimentally. In order to avoid computational burden in real-time coordinate transformations, Luh *et al.* [5] proposed approximate joint trajectories in that cubic and quadratic polynomial functions have been used with least square fit so that the approximation error is reduced. In this effort, trajectory approximations are unconstrained, and a sufficient number of knot-points are required in order to maintain the entire operation within the constraints. Trajectory generation algorithms proposed by Shin *et al.* [6], [7], look objectively similar to ours but significantly complex, and require a comprehensive set of link/joint parameters though some of them are not specified for general industrial robot arms.

There are plenty of trajectory generation algorithms devised by many researchers against the contouring problem. Nevertheless, the appropriate industrial constraints have not been sufficiently addressed yet. In [8], we addressed the contouring problem under torque constraint where constraint of assigned velocity had not been considered. In [9], we addressed posi-

tioning control with the constraints of assigned velocity and joint torque/acceleration.

In this paper, we have combined the two stated researches and realized a general solution to the contouring problem of industrial robot arms under the respective constraints. Experimenting with Performer MK-3s (PMK-3s) industrial robot arm, the effectiveness of the proposed method has been demonstrated.

II. Overview of Industrial Robot Arms

A. System Architecture of Industrial Robot Arms

A system schematic of a general industrial articulated robot arm is shown in Fig. 1(a), together with its world coordinate system in Fig. 1(b). In Fig. 1(a), the base, and first three links are shown. The hand of the robot arm possesses few concentrated joints for end-effector orientation control. For positioning and contouring in three-dimensional (3-D) space, only the first three joints are necessary. Fig. 1(b) shows a two-dimensional (2-D) world coordinate system together with a wire-frame of robot arm link configuration. Industrial robot arms are devised with independent joint controllers based on the decoupled joint dynamics, and driven by PID servo systems. Servo systems possess joint servomotors, which are actuated with current or voltage controllers that implement torque control of joint servomotors, according to the resident PID control algorithm. Required input to the servo controller is termed as taught data; $u_j(t); j = 1, 2$ is the joint index. Typically, taught data is the time-based profiles of joint position or velocity. Reference input generator generates taught data and is issued to the servo controller as a varying reference input in real-time servoing.

B. Robot Arm Kinematics

Robot arm kinematics is twofold: forward kinematics and inverse kinematics (or arm solution). Forward kinematic equations determine Cartesian position (x, y) and orientation of the end-effector, given the arm configuration (θ_1, θ_2) in joint coordinates. Kinematics reduces to a unique solution as given by

$$x = L_0 + L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \tag{1}$$

$$y = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \tag{2}$$

for the robot arm shown in Fig. 1(a). Lengths of the first two links are denoted by L_1 and L_2, whereas L_0 is the horizontal displacement of the first joint from the origin of the selected world coordinate frame, as shown in Fig. 1(b).

Inverse kinematics solves for the arm configuration in joint coordinates, given the position and orientation of the end-effector in Cartesian coordinates. Unlike kinematics, inverse kinematics are ill-conditioned and cannot be formulated as a closed explicit solution, except for simple manipulators. Inverse transformation method [10] is a general solution for inverse kinematics and due to the same reason it is extremely time consuming, hence, troublesome in real-time applications. It further requires decision equations to eliminate redundancy. Based on *D–H* representation [10], Paul *et al.* [11] proposed an explicit closed form arm solution for simple robot arms. The simplest among different arm solution schemes is the geometric approach

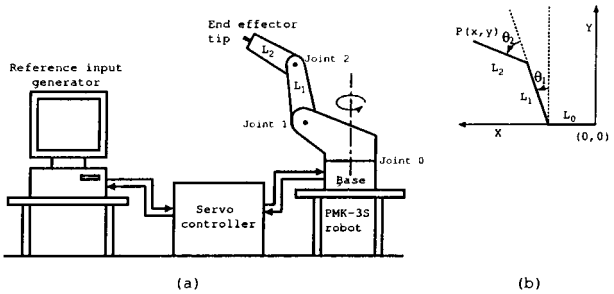


Fig. 1. PMK-3s industrial robot arm. (a) System schematic. (b) World coordinate system in two-dimensions.

proposed by Lee *et al.* [12] which could be applied to most industrial robot arms in the form of an explicit solution. For the robot arm illustrated in Fig. 1, inverse kinematics is given by

$$\theta_1 = \tan^{-1}\left(\frac{c}{y}\right) - \cos^{-1}\left(\frac{L_1^2 - L_2^2 + c^2 + y^2}{2L_1\sqrt{c^2 + y^2}}\right) \tag{3}$$

$$\theta_2 = \pi - \cos^{-1}\left(\frac{L_1^2 + L_2^2 - c^2 - y^2}{2L_1 L_2}\right) \tag{4}$$

where

$$c = x - L_0.$$

C. Joint Dynamics of Industrial Robot Arms

Industrial robot arms are employed in predetermined operations and, therefore, the nonlinear torque disturbances such as centripetal and Coriolis torques as well as gravity loading can be precomputed or estimated. These nonlinearities are usually weakened by way of appropriate mechanical design and/or by specifying operational limits, usually for velocity and acceleration. In industrial robot arms, inertia change due to the change in arm configuration is not considered significant. Instead, a linear relationship between joint torque and joint acceleration is assumed. Based on this assumption, modeling of joint dynamics and kinematic control can be conveniently realized. It further reduces joint torque constraint to its corresponding constraint in joint acceleration.

Almost all industrial robot arms are kinematically controlled. Caccavale *et al.* [13] proposed such a kinematic control scheme in joint space for a preplanned trajectory in Cartesian coordinates. Kinematic control of industrial robot arms is based on the decoupled, linear servo model, which is widely used in today's robotic applications. A comprehensive analysis of this model can be found in [14]. The two-link version of this dynamic model is shown in Fig. 2, together with assigned velocity and joint torque/acceleration constraints. Joint dynamics is governed by the two servo parameters K_{pj} and K_{vj}. These parameters are well known in industry as gains of position loop and velocity loop, respectively. The determination procedures of these parameters can be found in [14]. According to this model, linear joint dynamics $G_j(s)$ is given by

$$G_j(s) = \frac{\Theta_j(s)}{U_j(s)} = \frac{K_{pj}K_{vj}}{s^2 + K_{vj}s + K_{pj}K_{vj}} \tag{5}$$

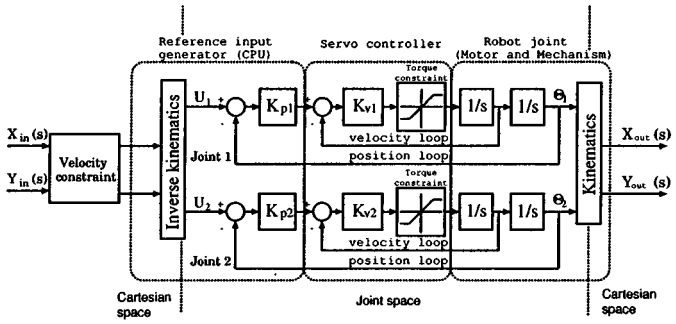


Fig. 2. Joint dynamic model for a two-link industrial robot arm.

in that $\Theta_j(s)$ and $U_j(s)$ are Laplace transforms of $\theta_j(t)$ and $u_j(t)$, respectively. From (5), linear joint dynamics in time domain can be written as

$$\ddot{\theta}_j(t) = K_{vj}\left\{K_{pj}\left(u_j(t) - \theta_j(t)\right) - \dot{\theta}_j(t)\right\} \tag{6}$$

in that $\ddot{\theta}_j(t)$ and $\dot{\theta}_j(t)$ are acceleration and velocity of joint variable $\theta_j(t)$. However, subjected to torque/acceleration saturation shown in Fig. 2, (6) could be rearranged, as given by

$$\ddot{\theta}_j(t) = \text{sat}\left[K_{vj}\left\{K_{pj}\left(u_j(t) - \theta_j(t)\right) - \dot{\theta}_j(t)\right\}\right] \tag{7}$$

where

$$\text{sat}(z) = \begin{cases} \ddot{\theta}_j^{\max} & \text{if } (z \geq \ddot{\theta}_j^{\max}) \\ z & \text{if } (-\ddot{\theta}_j^{\max} < z < \ddot{\theta}_j^{\max}) \\ -\ddot{\theta}_j^{\max} & \text{if } (z \leq -\ddot{\theta}_j^{\max}) \end{cases}$$

in that $\ddot{\theta}_j^{\max}$ is the torque/acceleration limit.

D. Constraints and Criterion of Trajectory Generation

The optimum performance of industrial robot arms should be realized within the limits of set constraints. In most practical cases, these constraints are: 1) maximum joint torque/acceleration $\ddot{\theta}_j^{\max}$ and 2) assigned velocity v_a. The constraint of joint torque/acceleration states that

$$|\ddot{\theta}_j| \leq \ddot{\theta}_j^{\max}. \tag{8}$$

Actually, $\ddot{\theta}_j^{\max}$ refers to the power amplifier current rating. It is also the saturation limit of the servo drive.

Contouring operations are specified with the assigned velocity v_a, the most effective end-effector velocity for the particular application. Assigned velocity is usually determined considering many technical factors and therefore treated as a set specification. Assigned velocity constraint states that

$$|v| \leq v_a \tag{9}$$

in that v stands for the end-effector velocity.

The objective of trajectory generation is to realize accurate contouring performance bound to the constraints (8) and (9). If acceleration saturates in real-time servoing, end-effector deviates from the trajectory it is supposed to trace, and results in poor contouring. To assure accurate contouring, it is therefore necessary that joints be actuated bound to (8). Further, optimum operation could be realized if joints could be actuated at the edge of acceleration limit, more longer.

III. Trajectory Generation for Optimum Contouring Performance of Industrial Robot Arms

A. Generation of Realizable Trajectory

1) Realizable Trajectory for a Corner: The objective trajectory is only a set of Cartesian points; start-point, all corners, and end-point in most cases. It is, therefore, necessary to plan the end-effector motion profile considering the objective trajectory, all specifications and set constraints. A trajectory generated such a way could be used to realize the desired motion of the end-effector; thus, it is termed *a realizable trajectory.*

Objective trajectory could be decomposed into straight-line segments and corners, in Cartesian space. It is proposed that corners and straight-lines be generated separately and merged together to obtain the realizable trajectory. Generation of trajectory corners requires special consideration as contouring at corners always gives rise to poor performance. A sharp corner can never be realized as it is in [8] and is therefore rounded up, as shown in Fig. 3. In Fig. 3, H is a sharp corner. According to the realizable trajectory, end-effector takes the path along the circular arc from R to S. The error at the corner is therefore ϵ. Referring to ΔRGH, $\angle RGH = (\alpha_2 - \alpha_1)/2$. The relationship between radius of arc r and error ϵ is then given by

$$\frac{RG}{GH} = \frac{r}{r+\epsilon} = \cos\{(\alpha_2 - \alpha_1)/2\}. \tag{10}$$

In order to express r in terms of ϵ, (10) could be rearranged as given by

$$r = \frac{\epsilon \cos\{(\alpha_2 - \alpha_1)/2\}}{1 - \cos\{(\alpha_2 - \alpha_1)/2\}}. \tag{11}$$

The radius of arc and tangential velocity v_t could be determined from specifications. The relationship between Centripetal acceleration $\mathbf{a}$, radius of arc r, and tangential velocity v_t is described by the theory of circular motion and is given by

$$|\mathbf{a}| = \frac{v_t^2}{r}. \tag{12}$$

There are two categories of industrial applications. One of them specifies a lower limit, $v_t^{\min}$ for the tangential velocity, while the other specifies a maximum limit $\epsilon^{\max}$ for the corner error. In addition, the maximum Cartesian acceleration of the end-effector, $|\mathbf{a}^{\max}|$ is usually specified by the manufacturer [14].

Case I—$v_t \geq v_t^{\min}$: In this category, having $v_t = v_t^{\min}$ in (12), r could be decreased while $|\mathbf{a}| \leq |\mathbf{a}^{\max}|$. Thus, there exists a lower bound for r as given by

$$r \geq \frac{\{v_t^{\min}\}^2}{|\mathbf{a}^{\max}|}. \tag{13}$$

Then, using (13) on (11), corresponding lower bound for ϵ could be determined as given by

$$\epsilon \geq \frac{\{v_t^{\min}\}^2 \left(1 - \cos\{(\alpha_2 - \alpha_1)/2\}\right)}{|\mathbf{a}^{\max}| \cos\{(\alpha_2 - \alpha_1)/2\}}. \tag{14}$$

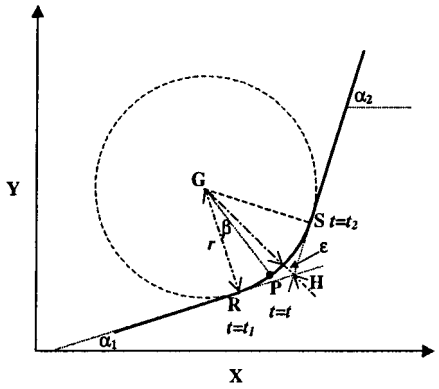


Fig. 3. Realizable trajectory at a sharp corner.

Case II—$\epsilon \leq \epsilon^{\max}$: In this category, according to (11), there exists an upper bound for r as given by

$$r \leq \frac{\epsilon^{\max} \cos\{(\alpha_2 - \alpha_1)/2\}}{1 - \cos\{(\alpha_2 - \alpha_1)/2\}}. \tag{15}$$

Corresponding upper bound for v_t could be determined by substituting (15) on (12) with $|\mathbf{a}| = |\mathbf{a}^{\max}|$, as given by

$$v_t \leq \sqrt{|\mathbf{a}^{\max}| r}. \tag{16}$$

Referring to Fig. 3, end-effector $P(x(t), y(t))$ makes $\beta = v_t(t - t_1)/r$; $t_1 \leq t \leq t_2$ as it moves along the circular curvature from R to S. Since v_t and r are known, the corner could be generated, as given by

$$\begin{aligned} x(t) &= x(t_1) + r\sin\beta\cos\alpha_1 \\ y(t) &= y(t_1) + r\sin\beta\sin\alpha_1. \end{aligned} \tag{17}$$

Using inverse kinematics given in (3) and (4), corner trajectory $P(x(t), y(t))$ from R to S could be transformed into corresponding joint trajectories $(\theta_1(t), \theta_2(t))$.

2) Realizable Trajectory for a Straight-Line: A set of knot-points is located along the straight-line segment. Interknot distance is constant in most practical cases and, it is generally determined using assigned velocity v_a and command time interval t_c. Command time interval is a manufacturer specification, whereas, assigned velocity depends on the particular application.

The set of Cartesian knot-points is transformed into joint space using inverse kinematics given in (3) and (4). Trajectory generation for straight-line segments is carried out in joint space, taking one knot-lap (two consecutive knot-points) at a time, and repeating it along the entire straight-line segment.

It is proposed that at least one joint be driven with the maximum acceleration within a knot-lap. This criterion is known as *maximum joint acceleration criterion (MAC)*. It harnesses the system servo capacity and realizes the maximum utility of joint acceleration so that the assigned end-effector velocity could be achieved very quickly. As MAC criterion is employed, end-effector velocity increases gradually, and assigned velocity is reached at some point on the line. Assigned velocity is the most favorable operating velocity which is specified after considering many aspects. Therefore, once it is reached, it is desirable to maintain it longer. To maintain the assigned Cartesian velocity, *assigned velocity criterion (AVC)* is employed.

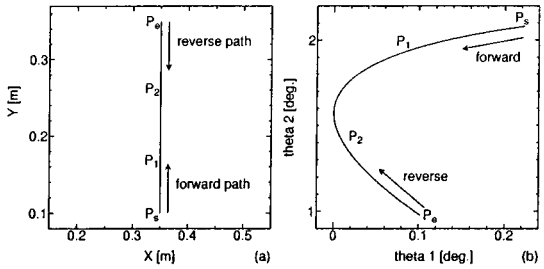


Fig. 4. Realizable straight-line segment. (a) In Cartesian space. (b) In joint space.

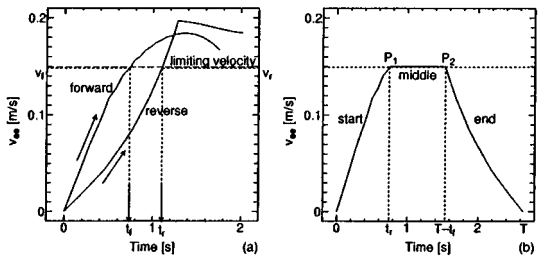


Fig. 5. Velocity profiles of a realizable straight-line segment. (a) Velocity profiles of forward and reverse paths and location of switching points at $t = t_f$ and $t = t_r$. (b) End-effector velocity profile.

However, end-effector has to be decelerated from assigned velocity to tangential velocity as it reaches a corner, or it should be decelerated to final standstill at the end-point. This deceleration is known as *maximum deceleration criterion (MDC)* and it is implemented by applying MAC criterion in the reverse direction. Using MDC criterion, it is possible to locate the point on the straight-line where AVC criterion has to be altered to MDC criterion. Thereby, it is possible to maximize the length of assigned velocity travel.

Fig. 4(a) illustrates a realizable straight-line segment P_sP_e. The path P_sP_1 is known as *forward path* and it is generated with MAC criterion. P_eP_2 is known as *reverse path* and it is generated with MDC criterion. Points P_1 and P_2 are the switching points where trajectory generation criterion is altered. At P_1, MAC criterion changes to AVC criterion. And at P_2, AVC criterion changes to MDC criterion. Fig. 4(b) illustrates the same information in joint space.

Fig. 5(a) shows how end-effector velocity is maintained along the forward and reverse paths, assuming standstill at both ends. End-effector velocity increases in both forward and reverse paths as MAC criterion is applied. Assigned velocity is reached at $t = t_f$ on the forward path and at $t = t_r$ on the reverse path. Trajectory between switching points P_1P_2 is termed *middle path* and it is generated in Cartesian space with AVC criterion. Fig. 5(b) shows end-effector velocity profile after the entire straight-line segment is generated.

Fig. 6 illustrates trajectory generation algorithm for straight-line segments diagrammatically. The forward path and reverse path are generated with MAC criterion applied in forward and reverse directions, respectively. Then, P_1 and P_2 are located on the line where constraint (9) is violated. The trajectory between P_1 and P_2 is generated with AVC criterion. Finally, the three

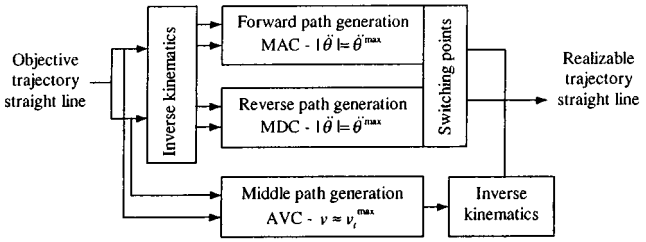


Fig. 6. Generation of realizable straight-line segments.

paths P_sP_1 (start), P_1P_2(middle), and P_2P_e(end) are joined so that the entire straight-line is generated.

Similar approaches have been proposed by Shin *et al.* [6], [7]. However, these reported works have been carried out only for positioning control tasks and at the same time without concerning assigned velocity constraint. Therefore, trajectory generation had been confined to joint space. In our previous work, Munasinghe *et al.* [9] promoted [15] by considering assigned Cartesian velocity constraint. Somehow, it addressed only positioning control and in this research, [9] is further extended to contouring operations.

Criterion of Maximum Joint Acceleration (MAC): The MAC criterion is explained as follows. Assuming uniform maximum acceleration motion the minimum interval $h_j^{\min}(k)$ each joint would take to move from kth knot point to $(k+1)$th knot point (kth lap) is determined by

$$h_j^{\min}(k) = \begin{cases} \dfrac{-\dot{\theta}_j(t_k) + \sqrt{\dot{\theta}_j^2(t_k) + 2\ddot{\theta}_j(k)\,\Delta\theta_j(k)}}{\ddot{\theta}_j(k)\Delta\theta_j(k)}, & \text{if } \ddot{\theta}_j(k) \neq 0 \\ \Delta\theta_j(k)/\dot{\theta}_j(k), & \text{if } \ddot{\theta}_j(k) = 0 \end{cases} \tag{18}$$

where

$$\ddot{\theta}_j(k) = \begin{cases} \ddot{\theta}^{\max}, & \text{if } \Delta\theta_j(k) > 0 \\ -\ddot{\theta}^{\max}, & \text{if } \Delta\theta_j(k) < 0 \end{cases}$$

in that $\dot{\theta}_j(t_k)$ is joint velocity at kth knot point. Angular displacement in kth knot-lap is $\Delta\theta_j(k) = \theta_j(k+1) - \theta_j(k)$ and $\ddot{\theta}_j(k)$ is the uniform joint acceleration in kth lap. It is required that the end-effector be moved as quickly as possible. Yet, none of the joints should lead to acceleration saturation. Therefore, optimum interknot interval $h^{\min}(k)$ for kth lap is set to the maximum of $h_j^{\min}(k)$ as given by

$$h^{\min}(k) = \max_j \left\{ h_j^{\min}(k) \right\}. \tag{19}$$

Then, optimum joint accelerations $\ddot{\theta}_j(k)$ for kth lap are determined by

$$\ddot{\theta}_j(k) = \left. \frac{2\left\{ \Delta\theta_j(k) - \dot{\theta}_j(k) h^{\min}(k) \right\}}{h^{\min}(k)} \right|_{|\ddot{\theta}_j(k)| \le \ddot{\theta}_j^{\max}}. \tag{20}$$

Then kth lap $\theta_j(t,\, k)$; $t \in [t_k,\, t_{k+1}]$ is generated, as given by

$$\theta_j(t,\, k) = \theta_j(k) + \dot{\theta}_j(k)t + \ddot{\theta}_j(k)t^2/2; \qquad t \in \left[0,\, h^{\min}(k)\right] \tag{21}$$

where $t_{k+1} = t_k + h^{\min}(k)$. This procedure is repeated with successive laps so that forward path is generated from P_s to P_1 (Fig. 4). Using the same procedure in the reverse direction, reverse path is generated from P_e to P_2 (Fig. 4). While MAC and MDC criterions are employed, Cartesian position is also calculated using (1) and (2). The end-effector velocity is also calculated differentiating (1) and (2) as given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \mathbf{J} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} \tag{22}$$

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{23}$$

where Jacobian $\mathbf{J}$ is given by

$$\mathbf{J} = \begin{bmatrix} L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2) & L_2\cos(\theta_1+\theta_2) \\ -L_1\sin\theta_1 - L_2\sin(\theta_1+\theta_2) & -L_2\sin(\theta_1+\theta_2) \end{bmatrix}.$$

Then, as shown in Fig. 5(a), switching point $P_1(x(t_f),\, y(t_f))$ is located at $t = t_f$ where $v = v_f \approx v_a$. Similarly, $P_2(x(t_r),\, y(t_r))$ is located at $t = t_r$ where $v = v_r \approx v_a$.

Criterion of Assigned Cartesian Velocity (AVC): The AVC criterion can be explained as follows. The length of the middle path is calculated by $l_{\text{middle}} = \sqrt{\{x(t_r) - x(t_f)\}^2 + \{y(t_r) - y(t_f)\}^2}$ and mean velocity is $v_{\text{middle}} = (v_f + v_r)/2$. Then, according to uniform acceleration motion, end-effector takes time $t_{\text{middle}} = 2l_{\text{middle}}/(v_f + v_r)$ to travel the middle path. The uniform Cartesian acceleration in the middle path is then $|\mathbf{a}_{\text{middle}}| = (v_r - v_f)/t_{\text{middle}}$ and it is fixed as v_f, v_r, and t_{middle} are known. This acceleration is used to move the end-effector within the middle path so that the velocity continuity could be achieved at both forward–middle (P_1) and middle–reverse (P_2) switching points. Then, according to the uniform acceleration motion with $\mathbf{a}_{\text{middle}}$, middle path could be generated as given by

$$x(t) = x(t_f) + \dot{x}(t_f)(t - t_f) + |\mathbf{a}_{\text{middle}}|\cos\phi(t - t_f)^2/2 \qquad (t_f \le t \le T - t_r) \tag{24}$$

$$y(t) = y(t_f) + \dot{y}(t_f)(t - t_f) + |\mathbf{a}_{\text{middle}}|\sin\phi(t - t_f)^2/2 \qquad (t_f \le t \le T - t_r) \tag{25}$$

in that $(\dot{x}(t_f),\, \dot{y}(t_f))$ specifies Cartesian velocity components at $P_1(x(t_f),\, y(t_f))$. Angle $\phi = \sin^{-1}\{(y(t_r) - y(t_f))/l_{\text{middle}}\}$ whereas T is the total travelling time. The "middle" path is then transformed into joint space using (3) and (4).

Finally, all corners and straight-line segments are connected together and that generates the entire realizable trajectory in full.

B. Compensation of the Realizable Trajectory for Delay Dynamics

Delay of joint dynamics causes poor contouring. Therefore, it is proposed that the realizable trajectory be compensated for delay dynamics using a compensator, as illustrated in Fig. 7. Delay compensated trajectory $U_j(s)$ is the taught data which is used as the input to the servo controller in real-time servoing.

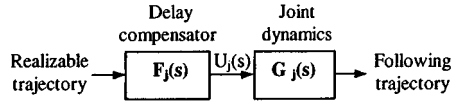


Fig. 7. Delay dynamics compensation with a forward compensator.

With reduced delay in dynamics, the realizable trajectory could practically be realized. With the major concern to the convenience in industrial implementation, an algorithm for delay dynamics compensation has been devised by Goto *et al.* [16] based on the pole placement regulator theory. It has also been synthesized to a forward compensator $F_j(s)$, as shown in Fig. 7. Second order compensator dynamics is given by

$$F_j(s) = -\frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{(s-\mu_{j1})(s-\mu_{j2})(s-\gamma_j)} \tag{26}$$

in that numerator coefficients are

$$\begin{aligned}
a_0 &= -\mu_{j1}\mu_{j2}\gamma_j \\
a_1 &= (K_{vj}+\gamma_j)(\mu_{j1}+\mu_{j2}) + K_{vj}^2 + \mu_{j1}\mu_{j2} + K_{vj}\gamma_j \\
&\quad - \mu_{j1}\mu_{j2}\gamma_j/K_{pj} \\
a_2 &= \frac{1}{K_{pj}}\left\{(K_{vj}+\gamma_j)(\mu_{j1}+\mu_{j2}) + K_{vj}^2 \right. \\
&\quad \left. + \mu_{j1}\mu_{j2} + K_{vj}\gamma_j\right\} - \frac{\mu_{j1}\mu_{j2}\gamma_j}{K_{pj}K_{vj}} \\
a_3 &= \frac{1}{K_{pj}K_{vj}}\left\{(K_{vj}+\gamma_j)(\mu_{j1}+\mu_{j2}) + K_{vj}^2 \right. \\
&\quad \left. + \mu_{j1}\mu_{j2} + K_{vj}\gamma_j\right\}
\end{aligned}$$

where μ_{j1}, μ_{j2} regulator poles, and γ_j is the observer pole. The details of determination procedure of regulator poles μ_{j1}, μ_{j2}, and observer pole γ_j of the compensator can be found in [16].

IV. Implementation of the Proposed Method of Trajectory Generation

A. Conditions for Simulation and Experiment

Objective trajectory was specified by Cartesian points (0.350 m, 0.100 m), (0.410 m, 0.150 m), (0.280 m, 0.300 m), and (0.350 m, 0.350 m). Assigned velocity was set to $v_a = 0.15$ [m/s] and joint acceleration limit was set to $\ddot{\theta}_j^{\max} = 0.72$ [rad/s^2], $\forall j = 1$, 2. Corner error was specified by $\epsilon^{\max} = 0.005$ [m]. Cartesian acceleration limit was set to $|\mathbf{a}^{\max}| = 0.09$ [m/s].

The objective trajectory and realizable trajectory in both Cartesian and joint spaces are shown in Fig. 8. Switching points are labeled with P_i; $i = 1 \cdot 2 \cdots 9$, and segments of the realizable trajectory could be classified according to the generating criterion as P_0P_1, P_3P_4, P_7P_8 (MAC), P_1P_2, P_5P_6, P_8P_9 (MDC), P_4P_5 (AVC), and P_2P_3, P_6P_7 (corner). Time stamping of the realizable trajectory is as follows: 0.000 [s] P_0, 0.712 [s] P_1, 1.182 [s] P_2, 1.758 [s] P_3, 2.294 [s] P_4, 2.848 [s] P_5, 3.344 [s] P_6, 3.894 [s] P_7, 4.490 [s] P_8, and 5.770 [s] P_9. End-effector velocity and joint acceleration profiles of the realizable trajectory are shown in Fig. 9.

A conventional method has been devised for comparison with the proposed method so that the performance could be evaluated. In the conventional method, trajectory is generated assuming $v = 0.04$ [m/s] uniform end-effector velocity along the objective trajectory. No special consideration or treatment is given for trajectory corners and straight-line segments. In addition, no compensation for delay dynamics is incorporated.

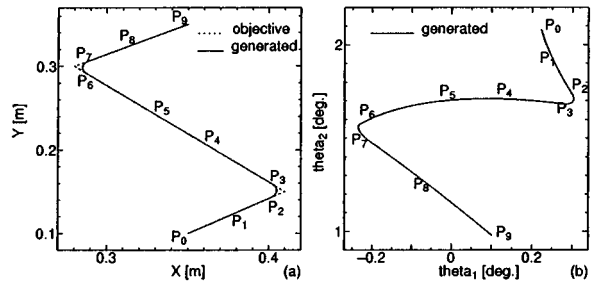
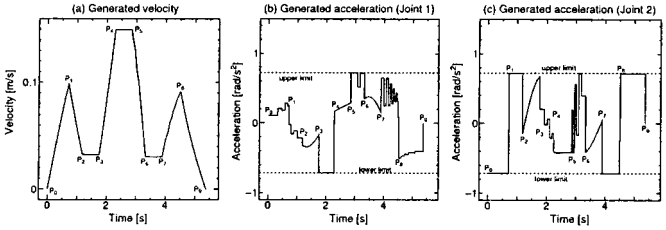


Fig. 8. Experimented trajectory. (a) In Cartesian space. (b) In joint space.



(a) (b) (c)

Fig. 9. Realizable trajectory. (a) End-effector velocity. (b) Joint 1 acceleration. (c) Joint 2 acceleration.

PMK-3s industrial robot arm was selected to carry out the experiment of the proposed method. Referring to Fig. 1, the lengths of PMK-3s arm are $L_0 = 0.135$ [m], $L_1 = 0.250$ [m], and $L_2 = 0.215$ [m]. Referring to Fig. 2, servo parameters of PMK-3s are $K_{pj} = 25$ [1/s] and $K_{vj} = 150$ [1/s], $\forall j = 1, 2$.

B. Results and Evaluation

The results are shown in Fig. 10. The following interpretations can be drawn according to the results obtained for the conventional method.

1) Joint 1 acceleration saturation indicated by $\downarrow$ $A1$ in Fig. 10(c) and that of joint 2 indicated by $\downarrow$ $A4$ in Fig. 10(d) cause end-effector velocity fluctuation indicated by $\downarrow$ $B1$ in Fig. 10(b). It further causes overshoot and hence poor performance at the first corner as indicated by $\downarrow$ $C1$ in Fig. 10(a).
2) Joint 1 acceleration saturation indicated by $\downarrow$ $A2$ in Fig. 10(c) and that of joint 2 indicated by $\downarrow$ $A5$ in Fig. 10(d) cause end-effector velocity fluctuation indicated by $\downarrow$ $B2$ in Fig. 10(b). It further causes overshoot and hence poor performance at the second corner as indicated by $\downarrow$ $C2$ in Fig. 10(a).
3) At the end of the operation, acceleration saturation occurs at both joints as indicated by $\downarrow$ $A3$ in Fig. 10(c) and $\downarrow$ $A6$ in Fig. 10(d). It causes a significant end-effector fluctuation as indicated by $\uparrow$ $C3$ in Fig. 10(a) which lasts for about 3 [s] before the final standstill.

On the contrary, much better performance has been realized by the proposed method as shown in simulation and experimental results in Figs. 10(e) and 10(i), respectively.

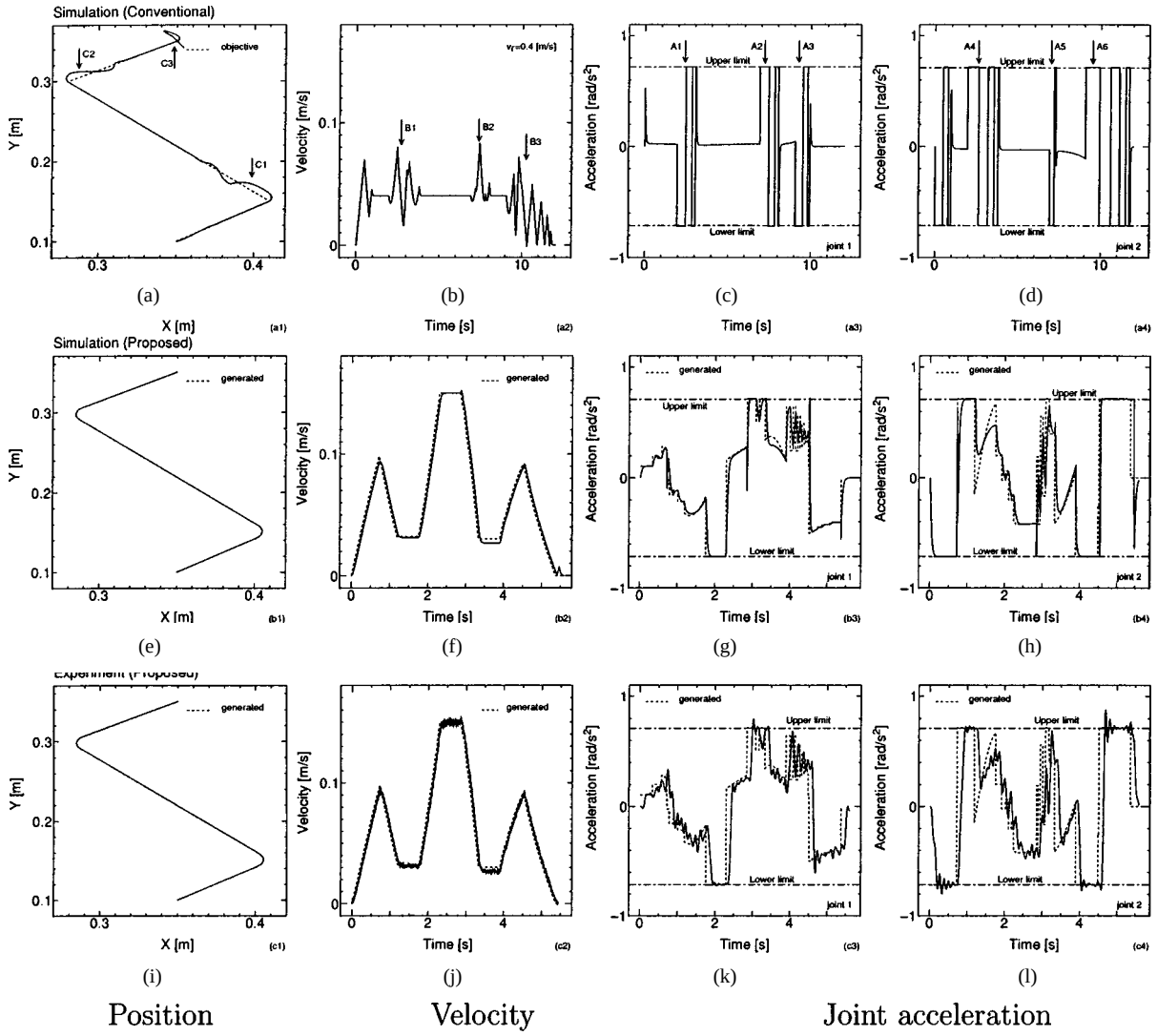


Fig. 10. Contouring results of conventional proposed methods.

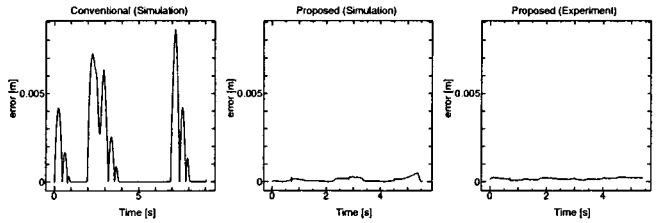


Fig. 11. Contouring error profiles of conventional and proposed methods.

End-effector velocity profiles in Figs. 10(f) and 10(j) show that the assigned Cartesian velocity constraint is not violated. According to the acceleration profiles in Fig. 10(g)–(l), both joints are driven bound to acceleration limits within the entire operation. More importantly, at any given instant, one of the two joints is driven with the limiting acceleration, except at corners and middle paths.

Fig. 11 illustrates contouring error profiles for conventional and proposed methods and Table I summarizes the performance quantitatively, in that e_{RMS} stands for root mean square error (RMSE) and T_{total} is the time requirement for both methods to complete the operation.

According to Fig. 11 and Table I, the proposed method has reduced time requirement significantly to about 46% that of the conventional method as calculated from the simulation results. At the same time, RMSE has been reduced to about 6.3% in the same respect.

C. Discussion

In industry, robot arms are modeled with the decoupled, linear servo model and controlled by PID servo controllers. Therefore, it is required that trajectory generation algorithms be devised on this basis so that the consistency between the proposed method and industrial setup is maintained and, therefore, the proposed method would bring about strong industrial implications. Most similar algorithms found in scientific literature have been devised on the basis of Lagrange–Euler and/or Newton–Euler [10] dynamics, or their extensions. Such models and their control algorithms are relatively complex and require long computational time and high costs for digital signal processing electronics. It is also required that many link and joint parameters of the robot arm be known though some of them are not possible to measure or estimate reliably. Usually, such algorithms are not simple and

TABLE I
COMPARISON OF RESULTS

	conventional (simulation)	proposed (simulation)	proposed (experiment)
e_{RMS} [mm]	2.630	0.164	0.185
T_{total} [s]	12.060	5.600	5.450

robust enough for the rough use and together with the high cost and requirement of skilled labor cause them to be infeasible for most industries to date.

Even if the objective trajectory is modified with considerations to the constraints and desired performance, the delay of the joint dynamics would cause poor performance. In the proposed solution, this problem has been eliminated by incorporation of a delay compensator.

The proposed method is perfectly consistent with the modeling and control methods of most industrial robot arms. It requires link lengths and two servo parameters for each joint, which can be accurately determined by simple tests as explained in [14]. With the proposed method, the speed of the control loop can be raised toward its servo controller limit, as there is no additional computational burden involved. There is no additional cost associated with the proposed method and it is simple enough for the practitioner to comprehend, operate, and trouble shoot. Though illustrated with a two-link robot arm, the proposed trajectory generation method can be extended and applied to a higher number of links as well.

In the middle path, Cartesian acceleration is very small and, therefore, velocity profile is more or less uniform. Hence, proposed method assumes without a further verification that constraint (8) remains satisfied within the middle path. At the middle path, joint accelerations could be computed by

$$\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} = \begin{bmatrix}p_{11} & p_{12}\\ p_{21} & p_{22}\end{bmatrix}\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} + \begin{bmatrix}q_{11} & q_{12}\\ q_{21} & q_{22}\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} \tag{27}$$

in that elements of matrices are given by

$$p_{11} = \frac{\sin(\theta_1+\theta_2)}{L_1\sin\theta_2}$$

$$p_{12} = \frac{\cos(\theta_1+\theta_2)}{L_1\sin\theta_2}$$

$$p_{21} = \frac{-\left[L_1\sin\theta_1 + L_2\sin(\theta_1+\theta_2)\right]}{L_1L_2\sin\theta_2}$$

$$p_{22} = \frac{-L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2)}{L_1L_2\sin\theta_2}$$

$$q_{11} = \frac{\cos(\theta_1+\theta_2)}{L_1\sin\theta_2}\dot{\theta}_1 - \frac{\sin\theta_1}{L_1\sin^2\theta_2}\dot{\theta}_2$$

$$q_{12} = -\left[\frac{\sin(\theta_1+\theta_2)}{L_1\sin\theta_2}\dot{\theta}_1 + \frac{\cos\theta_1}{L_1\sin^2\theta_2}\dot{\theta}_2\right]$$

$$q_{21} = \left[-\frac{L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2)}{L_1L_2\sin\theta_2}\dot{\theta}_1 + \frac{\sin\theta_1(L_2+L_1\cos\theta_2)}{L_1L_2\sin^2\theta_2}\dot{\theta}_2\right]$$

$$q_{22} = \left[\frac{L_1\sin\theta_1 + L_2\sin(\theta_1+\theta_2)}{L_1L_2\sin\theta_2}\dot{\theta}_1 + \frac{\cos\theta_1(L_2+L_1\cos\theta_2)}{L_1L_2\sin\theta_2}\dot{\theta}_2\right].$$

The above relationship could be obtained by differentiating

$$\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} = \mathbf{J}^{-1}\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix}. \tag{28}$$

V. CONCLUSION

The realizable trajectory is the optimum trajectory under assigned velocity and torque/acceleration constraints. The proposed trajectory generation method operates on both joint and Cartesian spaces and generates the realizable trajectory by connecting piece-wise segments, which are generated separately. Therefore, it is possible to cope with almost any constraint and/or specification, irrespective of the space where it actually exists. The proposed trajectory generation algorithm has been synthesized in such a way that it can be an off-line block in the entire robot arm control system. This is of utmost importance since in this configuration it is possible to plug this algorithm into the existing servo controllers straight away. For the same reason, it is expected that the proposed trajectory generation algorithm would bring about strong industrial implications.

REFERENCES

[1] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man Mach. Syst.*, vol. MMS-10, pp. 47–53, June 1969.
[2] R. Paul, "Manipulator Cartesian path control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, pp. 702–711, Nov. 1979.
[3] R. H. Taylor, "Planning and execution of straight line manipulator trajectories," *IBM J. Res. Develop.*, vol. 23, no. 3, pp. 424–436, July 1979.
[4] J. Y. S. Luh and C. S. Lin, "Optimum path planning for mechanical manipulators," *Trans. ASME, J. Dyn. Syst. Meas. Contr.*, vol. 14, pp. 444–451, June 1984.
[5] ——, "Approximate joint trajectories for control of industrial robots along Cartesian paths," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-14, pp. 444–450, May/June 1984.
[6] K. G. Shin and N. D. Mckay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 531–541, June 1985.

[7] ——, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 491–511, June 1986.
[8] M. Nakamura *et al.*, "Enhanced contour control of SCARA robot under torque saturation constraint," *IEEE/ASME Trans. Mechatron.*, vol. 5, pp. 437–440, Dec. 2000.
[9] S. R. Munasinghe, S. Goto, and N. Kyura, "High speed precise control of robot arms with assigned speed under torque constraint by trajectory generation in joint coordinates," in *Proc. IEEE Int. Conf. Syst., Man, Cybern*, Oct. 1999, pp. II854–II859.
[10] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics; Control, Sensing, Vision and Intelligence*. New York: McGraw-Hill, 1987, pp. 82–102.
[11] R. P. Paul, B. Shimano, and G. E. Mayer, "Kinematic control equations for simple manipulators," *IEEE Trans. Syst., Man, Cybern*, vol. SMC-11, pp. 449–455, June 1981.
[12] C. S. G. Lee and M. Zeigler, "Geometric approach in solving inverse kinematics of PUMA robots," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-20, pp. 695–705, Nov. 1984.
[13] F. Caccavale, S. Chiaverini, and B. Siciliano, "Second-order kinematic control of robot manipulators with Jacobian damped least-squares inverse: Theory and experiments," *IEEE/ASME Trans. Mechatron.*, vol. MECH-2, pp. 188–184, Sept. 1984.
[14] M. Nakamura, S. Goto, and N. Kyura, *Mechatronic Servo System Control* (in Japanese). Tokyo, Japan: Morikita Shuppan, 1998, ch. 1 and 4.
[15] M. Nakamura *et al.*, "Upper limit of performance for contour control of industrial articulated robot arm with torque constraints and its realization" (in Japanese), *Trans. SICE*, vol. 35, no. 1, pp. 122–129, Jan. 1999.
[16] S. Goto, M. Nakamura, and N. Kyura, "Modified taught data method for industrial mechatronic servo-controller to achieve accurate contour control performance," in *Proc. IEEE/ASME Int. Conf. AIM*, vol. 525B, June 1997.

S. Rohan Munasinghe (S'01) was born in Colombo, Sri Lanka, in 1970. He received the B.Sc. degree in electrical engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 1996 and the M.Sc. degree in electrical engineering from Saga University, Honjomachi Saga, Japan, in 1997, where he is currently pursuing the Ph.D. degree.

He has held faculty positions in the Department of Electrical Engineering and the Department of Computer Science and Engineering at the University of Moratuwa. His research interests include design of control algorithms for industrial manipulators, tele/space robotics, and intelligent control systems.

Masatoshi Nakamura (SM'99) was born in Fukuoka, Japan, in 1943. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from Kyushu University, Iizuka, Japan, in 1967, 1969, and 1974, respectively.

From 1973 to 1974, he was a Research Associate at Kyushu University. Since 1974, he has been with the Faculty of Science and Engineering, Saga University, Honjomachi Saga, Japan, where he is currently a Professor in the Department of Advanced Systems Control Engineering in the Graduate School. His research interests include systems control theory and its applications, especially power system control, thermal flow-control, robotics, and biomedical engineering.

Dr. Nakamura is a Fellow of the Society of Instrument and Control Engineers of Japan and a Member of the IEEE Control Systems Society, the Institute of Electrical Engineers of Japan, the Robotics Society of Japan, and the Institute of Systems, Control, and Information of Japan.

Satoru Goto was born in Fukuoka, Japan, in 1966. He received the B.S. and M.S. degrees in applied physics in 1988 and 1990, respectively, and the Ph.D. degree in 1995, all from Osaka University, Osaka, Japan.

Since 1995, he has been with the Faculty of Science and Engineering, Department of Advanced Systems Control Engineering, Saga University, Honjomachi Saga, Japan, where he has held faculty positions of Research Associate from 1995 to 1996 and Lecturer from 1996 to 1998. He has been an Associate Professor since 1998 in the same department. His research interests include control theory and its applications to actual systems.

Dr. Goto is a Member of the Society of Instrument and Control Engineers of Japan, the Robotics Society of Japan, and the Institute of Systems, Control, and Information of Japan.

Nobuhiro Kyura (M'00) was born in 1942. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from Kyushu University, Iizuka, Japan, in 1964, 1966, and 1992, respectively.

Since 1964, he has been with Yasakawa Electric Corporation, KitaKyushu, Japan, where he is currently the General Manager of the Research Laboratory. Since 1995, he has been with the Department of Electrical and Computer Engineering, Kinki University, Iizuka, Japan, where he is currently a Professor of electrical engineering. His research interests include motion controller architecture, optimum motion control, and robot manipulator control.

Dr. Kyura is a Member of the Institute of Electrical Engineers of Japan, the Society of Instrument and Control Engineers of Japan, the Japan Society of Precision Engineering, the Robotics Society of Japan, and the Institute of Systems, Control, and Information of Japan.