

Trajectory Planning and Control of Industrial Robot Manipulators

S. R. Munasinghe and Masatoshi Nakamura

1. Introduction

Industrial robot manipulators are used in various applications in order to achieve fast, precise, and quality production. In pick-and-place operations such as part handling, assembly, etc., the end-effector of the manipulator has to travel between two specific points in the workspace, and the path it takes in between is of no concern. In trajectory tracking applications such as welding, cutting, painting, etc., the end-effector has to follow a specific trajectory in 3-space as closely as possible, while maintaining rated velocity as much as possible (Munasinghe, 2001). In the latter case, planning the trajectory can be complex when there are constraints on the end-effector velocity, joint acceleration, and trajectory error. Trajectories planned without proper consideration to these constraints often result in poor performance such as trajectory overshoots, end-effector deviations from the planned trajectory, and undue velocity fluctuations (Nakamura, et. al., 2000). Performance could be even more deteriorated especially at sharp corners in the Cartesian trajectory (Nakamura, 20001). Lot of trajectory planning algorithms have been proposed so far starting from simple Cartesian path control (Paul, 1979) to time optimized trajectories (Shin, 1985). However, the industrial systems experience difficulties accommodating most of these methods because of at least two specific reasons; 1) These techniques often require hardware changes in the existing setup and the manufacturing process has to be interrupted for system reconfigurations, which usually takes a longer period of time, and 2) Many of these methods often consider only one constraint, and often they pay less concern about industrial requirements and actual constraints set by applications. Therefore, they find difficulties in industrial implementation.

In this view, we present a new trajectory planning algorithm which considers end-effector velocity limit, joint acceleration limit, and error tolerance set by the application. These are the actual constraints in most industrial applications. Another technical problem in industrial manipulators is their delay dynamics, which causes the end-effector to overshoot at trajectory corners. To remedy this problem, we have designed a feed-forward compensator (Goto,

1997), which slightly alters the corners of the trajectory so that to make sure that the end-effector actually traces the trajectory even with the presence of the delay dynamics. The new trajectory planning algorithm together with the feed-forward compensator appears as a single front end block in the control system, and it can be easily accommodated to existing industrial manipulator systems without taking the risk and time of hardware reconfigurations.

A trajectory planning algorithm can generate position, velocity, and acceleration profiles for all of the joints of the manipulator. In most industrial manipulators, the system input is the joint position data, which are widely known in the industry as taught data. Paul (Paul, 1979) described how homogeneous transformations (Mittal & Nagrath, 2003) can be used in representing position and orientation of a serial link manipulator in order to control it through a Cartesian trajectory. The work by Shin et al. (Shin et al. 1985) looks similar to ours, however it is difficult to be implemented in industrial systems as it needs to know many link/joint parameters of the manipulator. In industrial manipulator systems, most of these parameters are not precisely known.

In our previous works we have addressed acceleration and velocity constraints for 2-space trajectory planning (Munasinghe, 2001), and in this work we extend it to 3-space, while also considering error tolerance of the trajectory. The proposed method has been tested on a Performer MK-3s industrial manipulator, and its effectiveness has been experimentally verified.

2. Industrial Robot Manipulators

2.1 System Architecture

The industrial robot manipulator Performer MK-3s is shown below.

The reference input generator is a dedicated, or a networked computer which is connected to the servo controller through digital-to-analog (DAC) and analog-to-digital (ADC) converters. Servo controller has motor driver boards to control manipulator joints individually as shown in Fig.1. The reference input generator contains taught data sequences u_j where $j=1, 2, 3$ stands for the joint. Joint position θ_j is fed back to the reference input generator from the servo controller.

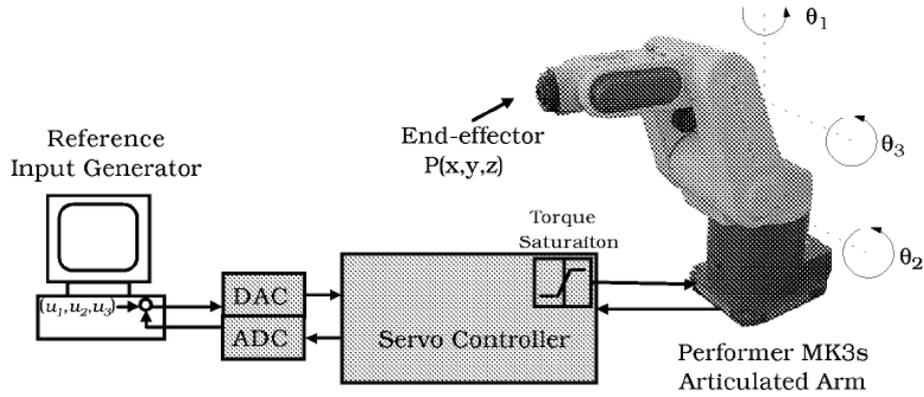


Figure 1. Performer MK-3s industrial robot manipulator

With the taught data and position feedback, reference input generator determines the control commands for each joint, and send those commands to the servo controller, which actuates joint motors accordingly. Referring to Fig.1, kinematics of the manipulator is given by

$$\begin{aligned} x &= [L_1 + L_2 \sin \theta_2 + L_3 \sin(\theta_2 + \theta_3)] \cos \theta_1 \\ y &= [L_1 + L_2 \sin \theta_2 + L_3 \sin(\theta_2 + \theta_3)] \sin \theta_1 \\ z &= L_2 \cos \theta_2 + L_3 \cos(\theta_2 + \theta_3) \end{aligned} \quad (1)$$

where (x, y, z) is the end-effector position, and $\theta_1, \theta_2, \theta_3$ is the corresponding joint configuration. L_j is the length of link . By differentiating (1) it is possible to find the velocity relationship between Cartesian velocity and joint velocity as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathbf{J}(\boldsymbol{\theta}) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (2)$$

where Jacobean is given by

$$\mathbf{J}(\boldsymbol{\theta}) = \begin{bmatrix} -\sin \theta_1 [L_1 + L_2 \sin \theta_2 + L_3 \sin(\theta_2 + \theta_3)] & \cos \theta_1 [L_2 \cos \theta_2 + L_3 \cos(\theta_2 + \theta_3)] & L_3 \cos \theta_1 \cos(\theta_2 + \theta_3) \\ \cos \theta_1 [L_1 + L_2 \sin \theta_2 + L_3 \sin(\theta_2 + \theta_3)] & \sin \theta_1 [L_2 \cos \theta_2 + L_3 \cos(\theta_2 + \theta_3)] & L_3 \sin \theta_1 \cos(\theta_2 + \theta_3) \\ 0 & -L_2 \sin \theta_2 - L_3 \sin(\theta_2 + \theta_3) & -L_3 \sin(\theta_2 + \theta_3) \end{bmatrix}$$

in that $\boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \theta_3]^T$ is the arm configuration.

2.2 Joint Dynamics of Industrial Robot Manipulators

Industrial robot manipulators are designed to meet the performance level required by the application such as welding, cutting, part handling, etc. The specifications in general are limited only to a certain degree of accuracy, velocity, and complexity. Therefore, most industrial robot manipulators are designed with linear proportional-integral-derivative (PID) servo controllers with current limiting power amplifiers. This saturating current determines the acceleration limit of the joint. Furthermore, joints are independently controlled, whereas unknown inertia torques, coriolis and centrifugal torques, and torques due to friction and gravity are treated as disturbances to be rejected by the controller. To support this assumption, manipulator links are designed with low inertia, and joints are driven through high gear reductions (Sage et. al. 1999). These controllers are simple, and also provide sufficient robustness. Figure 2 illustrates three degree of freedom decoupled joint dynamic model of an industrial manipulator.

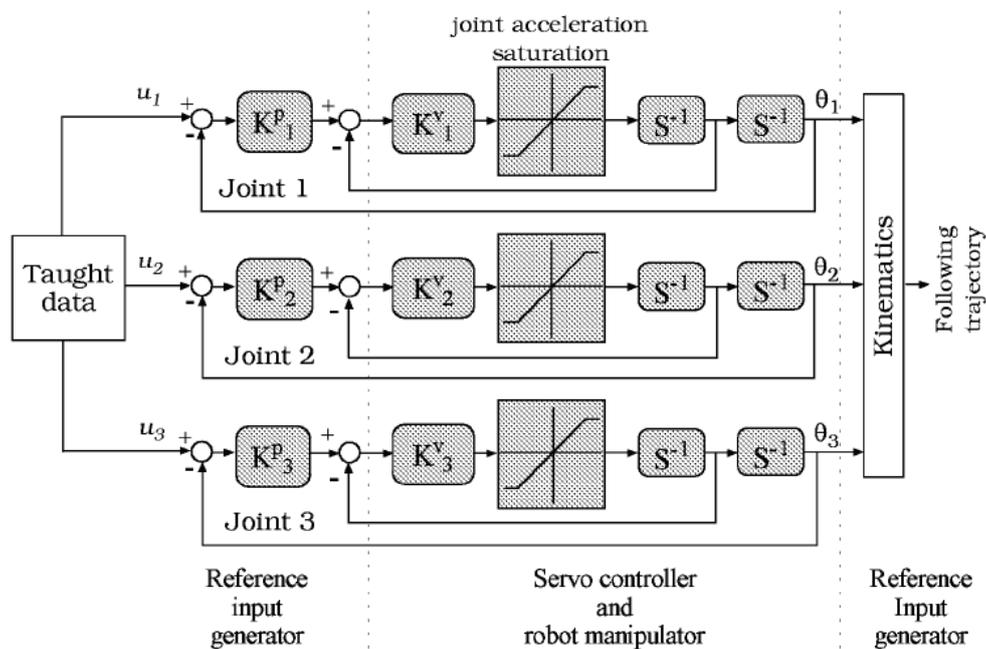


Figure 2. Three degree of freedom joint dynamic model of an industrial robot manipulator

This model also includes power amplifier saturation of joint actuators. K_j^p and K_j^v are the servo controller gains in the position loop and velocity loop of joint j , and these gains are periodically tuned by the trained operators to maintain the level of performance. As only two tuning parameters are involved, controller tuning process is quite simple. Within the linear region of joint ac-

celeration, joint dynamics is given by

$$\frac{\Theta_j(s)}{U_j(s)} = \frac{K_j^p K_j^v}{s^2 + 2K_j^v s + K_j^p K_j^v} \quad (3)$$

And, the joint dynamics when joint acceleration saturates is given by

$$\ddot{\theta}_j(t) = \text{sat}[K_j^v \{K_j^p (u_j(t) - \theta_j(t)) - \dot{\theta}_j(t)\}] \quad (4)$$

where

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

in that $\ddot{\theta}_j^{\max}$ is the maximum acceleration of joint j . In this view, the objective of the trajectory planning is to make the best use of joint acceleration capability, while avoiding saturation.

2.3 Problem Statement

In this work, we consider the following three major issues which are practically applicable in industrial robot manipulator applications.

$$|\ddot{\theta}_j| \leq \ddot{\theta}_j^{\max} \quad (5)$$

$$v = \begin{cases} \leq v_r & \text{along straight line} \\ \leq v_t^{\max} & \text{at corners} \end{cases} \quad (6)$$

$$e \leq \rho \quad (7)$$

where v , v_r , and v_t^{\max} are end-effector velocity, rated velocity, and maximum tangential velocity (at a rounded corner), respectively. e , and ρ are the trajectory error and error tolerance. Constraint (5) describes the linear region for joint acceleration, within which linear dynamics (3) is maintained. A violation of this constraint results in nonlinear joint dynamics (4), which causes the end-effector to deviate from the planned trajectory. Constraint (6) specifies the velocity limit while end-effector moves along straight lines and through corners. Rated velocity of the joint ω_r is given by $\omega_r = 2\pi.RPM_r / (60.N_G)$, where RPM_r is the rated RPM (revolutions per minute) of the joint and N_G is the gear reduction ratio. Then, the rated velocity $v_r = \omega_r L$, where L is the link length.

At trajectory corners the tangential velocity is lowered heuristically to maintain centripetal acceleration within constraint (5), and it can also be theoretically determined as described in (Munasinghe & Nakamura, 2002)

3. Trajectory Planning

3.1 The Algorithm

The proposed trajectory planning algorithm is illustrated in Fig. 3. The objective trajectory $O(s)$ is specified by the application, and it is segmented into a sequence of a) corners and b) straight line segments. Corners are planned in Cartesian space using specified tangential velocity $v = v_t^{\max}$, and transformed into joint space using inverse kinematics. Straight line segments are generated in joint space as piecewise in that every straight line segment has three pieces; forward (acceleration), middle (uniform velocity), and reverse (deceleration). Forward/reverse pieces are planned in such a way that at least one joint moves with its maximum acceleration/deceleration as long as end-effector velocity constraint (6) is not violated (b1 and b2 in Fig.3).

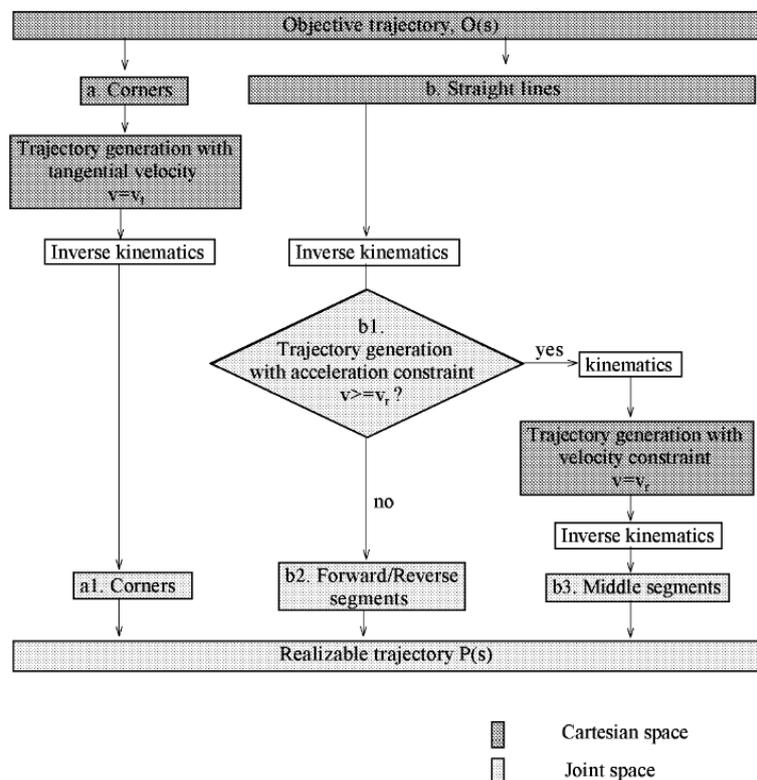


Figure 3. Proposed trajectory planning algorithm

Middle segment is planned in Cartesian space maintaining uniform end-effector velocity $v = v_r$, and then it is transformed into joint space (b3 in Fig. 4). Finally, all corners and straight line segments in joint space are merged in the correct sequence. This trajectory in joint space is called the realizable trajectory $P(s)$.

3.2 Trajectory Planning for a Corner

Figure 4(a) illustrates a sharp corner of the objective trajectory $O(s)$ with the error tolerance (dashed line). Error tolerances are quite common in industrial applications, and it can be used to effectively plan the realizable trajectory $P(s)$. It is required however, to make sure that the realizable trajectory is contained within the error tolerance. Referring to Fig. 4, the largest possible circular arc should pass through point R , and it should be tangential to the section of the tolerance cylinder on the plane of ΔABC . In order to construct this curve, points A' , B' , and C' are determined from A , B , and C according to the following procedure:

$$\text{In } \Delta ABC \quad AB = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2},$$

and $\beta = \cos^{-1}\{(AB^2 + BC^2 - AC^2)/2.AB.BC\}$. A' is located using point coordinates of A and C . For example, x coordinate of A' can be determined by $x_{A'} = (x_A - x_C)\varepsilon_{A'}/AC + x_C$, where $\varepsilon_{A'} = \rho/\sin \hat{A}$. Adopting the same procedure B' could be located with point coordinates of F , B , and $\varepsilon_{B'} = \rho/\sin(\beta/2)$. C' could also be located with point coordinates of A , C , and $\varepsilon_{CB} = \rho/\sin \hat{C}$.

F is located with point coordinates

$$A, C, \text{ and } \varepsilon_F = BC \tan \hat{C} \tan(\beta/2)/\{\tan \hat{C} + \tan(\beta/2)\}.$$

Figure 4(b) illustrates the circular arc constructed at a corner with radius r is given by

$$r = 2\rho/\{1 - \sin(\beta/2)\} \quad (8)$$

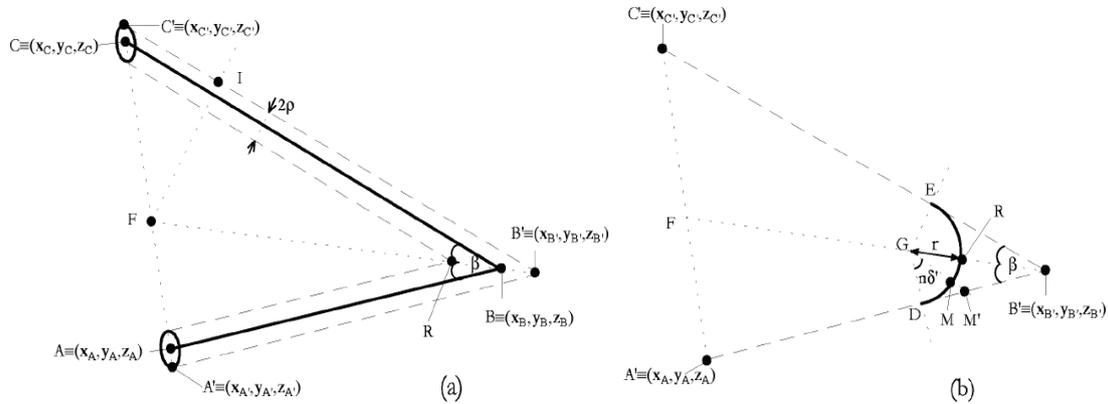


Figure 4 (a) A sharp corner of the objective trajectory, and (b) A planned corner of the realizable trajectory

D and E are the terminal points of the circular arc and they could be located on $A'B'$ and $B'C'$ as $DB' = EB' = \{r + 2\rho / \sin(\beta/2)\} \cos(\beta/2)$. Along the circular arc, from D to E trajectory is sampled at each δ as marked by M. Sampling angle is given by

$$\delta = v_t^{\max} t_s / r \tag{9}$$

where t_s is the sampling interval. Number of sampling points is determined by upward rounding of $N = (\pi - \beta) / 2\delta$ to the closest whole number. Then, sampling angle is readjusted by $\delta' = (\pi - \beta) / 2N$. M' could be located on BD' as $DM' = r \tan(n\delta')$. M could be located on GM' since the ratio r / GM' is known.

3.3 Trajectory planning for a Straight Line

3.3.1 Forward and Reverse Segments

Figure 6 illustrates details of straight line trajectory planning. P_1P_2 is the straight line segment of the objective trajectory, for which a realizable trajectory has to be planned. The two end points of the straight line are either start and end points of the objective trajectory, or terminal points of a circular arc. Either way, position and velocity at these points are known. From P_1 to P_2 , trajectory is segmented by equidistance via points indexed by $k = 0, 1, 2, \dots$. The forward trajectory is planned from P_1 to P_2 , whereas the reverse trajectory is generated from P_2 to P_1 . Both segments are planned in joint space in the two directions using the same algorithm described below.

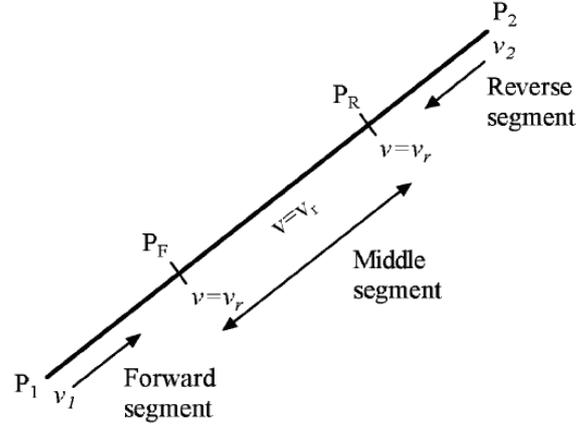


Figure 5. Trajectory planning for straight line segments

The minimum time $t_j^{\min}(k)$ each joint takes to move from k th via point to $(k+1)$ th via point is given by

$$t_j^{\min}(k) = \begin{cases} \frac{\sqrt{\dot{\theta}_j^2(k) + 2\ddot{\theta}_j^{\max}\Delta\theta_j(k)} - \dot{\theta}_j(k)}{\ddot{\theta}_j^{\max}} & \text{if } \Delta\theta_j(k) > 0 \\ \frac{\sqrt{\dot{\theta}_j^2(k) + 2\ddot{\theta}_j^{\max}\Delta\theta_j(k)} + \dot{\theta}_j(k)}{\ddot{\theta}_j^{\max}} & \text{if } \Delta\theta_j(k) < 0 \end{cases} \quad (10)$$

where $\Delta\theta_j(k) = \theta_j(k+1) - \theta_j(k)$. The minimum feasible time between the two via points without letting any of the joints to saturation is the longest t_j^{\min} considering all of the joints as follows.

$$t^{\min}(k) = \max_j \{t_j^{\min}(k)\} \quad (11)$$

Using $t^{\min}(k)$ for planning the trajectory between the two via points guarantees minimum time motion, and resulting joint accelerations are

$$\ddot{\theta}_j(k) = \frac{2(\Delta\theta_j(k) - \dot{\theta}_j(k)t^{\min}(k))}{\{t^{\min}(k)\}^2} \quad (12)$$

Then, the trajectory is planned in joint coordinates as follows.

$$\dot{\theta}_j(k, t) = \dot{\theta}_j(k) + \ddot{\theta}_j(k)t \quad ; kt < t < (k+1)t \quad (13)$$

$$\theta_j(k, t) = \theta_j(k) + \dot{\theta}_j(k)t + 0.5\ddot{\theta}_j(k)t^2 \quad ; kt < t < (k+1)t \quad (14)$$

This algorithm continues as via point advances $k= 0, 1, 2\dots$, and in the same time end-effector velocity is calculated using (2) together with (13) and (14). When the end-effector velocity reaches rated velocity the algorithm terminates (b1 in Fig. 3). As illustrated in Fig. 5, end-effector reaches the rated velocity at P_F in the forward direction, and at P_R in the reverse direction.

3.3.2 Middle Segment

Referring to Fig. 5 $P_F P_R$ is the middle segment of the straight line. This segment is planned in Cartesian space by maintaining rated velocity v_r as follows.

$$\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} v_r^x \\ v_r^y \\ v_r^z \end{bmatrix} + \begin{bmatrix} x(P_F) \\ y(P_F) \\ z(P_F) \end{bmatrix} \quad (15)$$

where v_r^x , v_r^y , and v_r^z are the velocity components of v_r along major axes, and $x(P_F)$, $y(P_F)$ and $z(P_F)$ are the cartesian position coordinates of P_F . Middle segment is transformed into joint coordinates using inverse kinematics.

3.4 Compensation of Delay Dynamics

Planned corners and straight lines are merged to form the realizable trajectory. As shown in Fig. 3, taught data is obtained by compensating realizable trajectory for delay dynamics. In (Goto et. al., 1997) pole placement with linear state feedback were used to develop a feed-forward delay compensator as described by

$$F_j(s) = -\frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{(s - \mu_1)(s - \mu_2)(s - \gamma)} \quad (16)$$

Where

$$\begin{aligned} a_0 &= -\mu_1 \mu_2 \gamma \\ a_1 &= (K_j^v + \gamma)(\mu_1 + \mu_2) + K_j^{v^2} + \mu_1 \mu_2 + K_j^v \gamma - \mu_1 \mu_2 \gamma / K_j^p \\ a_2 &= \frac{1}{K_j^p} \left\{ (K_j^v + \gamma)(\mu_1 + \mu_2) + K_j^{v^2} + \mu_1 \mu_2 + K_j^v \gamma \right\} - \frac{\mu_1 \mu_2 \gamma}{K_j^p K_j^v} \\ a_3 &= \frac{1}{K_j^p K_j^v} \left\{ (K_j^v + \gamma)(\mu_1 + \mu_2) + K_j^{v^2} + \mu_1 \mu_2 + K_j^v \gamma \right\} \end{aligned}$$

in that μ_1 and μ_2 are the regulator poles, and γ is the observer pole. These poles can be tentatively tuned for better performance. A theoretical determination of compensator poles can be found in (Munasinghe & Nakamura, 2003)

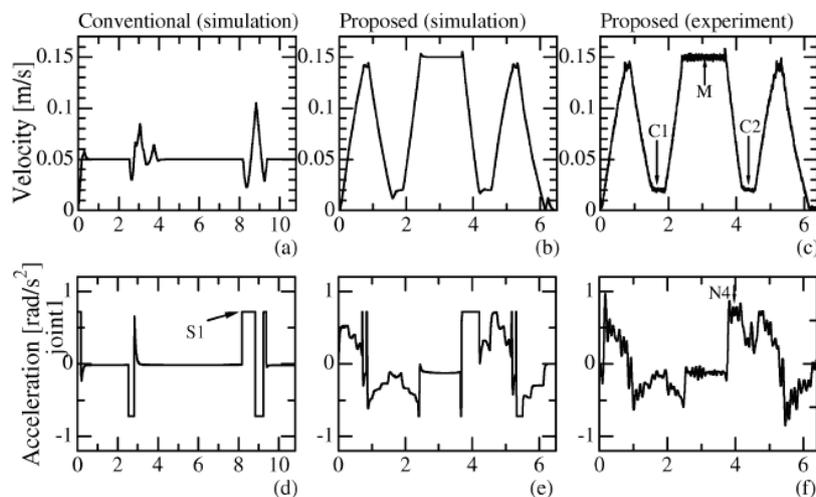
4. Results and Discussion

4.1 Experimental Conditions

The objective trajectory was set as follows: start (0.35, 0, 0.1)[m], first corner(0.41, 0.1, 0.15)[m], second corner (0.28, -0.1, 0.3)[m], and end (0.35, 0, 0.35)[m]. Rated velocity and tangential velocity were set to $v_r = 0.15$ [m/s²] and $v_t^{\max} = 0.02$ [m/s²]. Maximum joint acceleration for all joints were set to $\ddot{\theta}_j^{\max} = 0.72$ [rad/s²]. Trajectory error tolerance was set to $\rho = 0.001$ [m]. Servo controller gains were set to $K_j^p = 15$ [1/s] and $K_j^v = 15$ [1/s]. In the delay compensator, regulator poles were set to $\mu_1 = \mu_2 = -60$, and the observer pole was set to $\gamma = -200$. In order to compare performance of the new method, we simulated a conventional trajectory planning algorithm in that uniform end-effector velocity of 0.05 [m/s] was used to plan the trajectory in Cartesian space through the mentioned objective trajectory.

4.2 Evaluation of Results

Figure 6 illustrates the results.



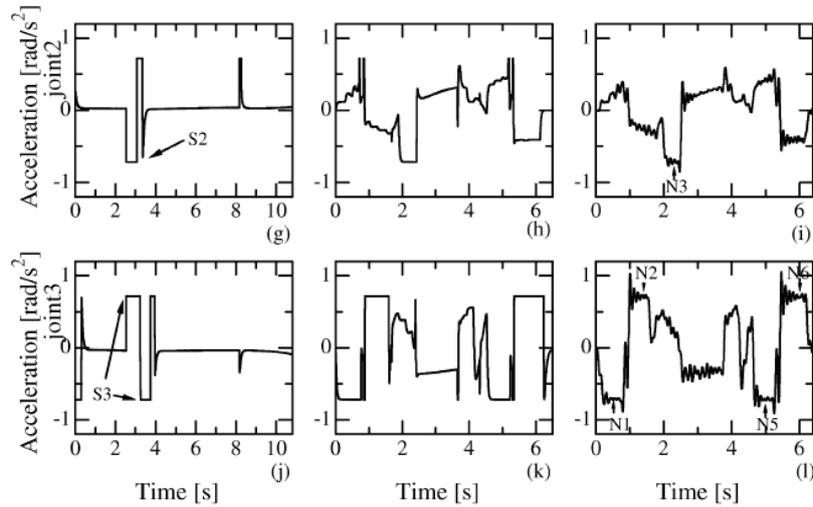


Figure 6. End-effector velocity and joint acceleration profiles under the control of conventional and proposed methods.

One important observation to be made is the close similarity between simulation and experimental results of the proposed method, i.e., the experiment produces end-effector velocity and joint acceleration profiles that are very similar to what is obtained by the simulation under the assumption of linear decoupled dynamics. This conveys the validity of the trajectory planning and delay compensation used in the proposed method.

The arrow sequence $\uparrow N1, \downarrow N2, \uparrow N3, \downarrow N4, \uparrow N5, \downarrow N6$ confirms that at least one of the three joints moves with its maximum acceleration or deceleration within the entire motion, except at corners C1, C2, and middle segment M. End-effector velocity has been kept on or below the rated velocity within the entire motion. On the contrary, the conventional method shows in its simulation a significant saturation in joint acceleration profiles as indicated by S1, S2, and S3.

Figure 7 shows the motion of the end-effector in 3-space with projections to X-Y, Y-Z, and Z-X planes. In Fig. 7(b), huge trajectory errors are observed at corner C1 as a result of acceleration saturation in joint 2 and joint 3 as indicated by S2 and S3. Similar errors are resulted at corner C2 due to acceleration saturation in joint 1 as indicated by S1. On the other hand, proposed method has made the end-effector accurately follow the objective trajectory.

4.3 Discussion

The proposed trajectory planning algorithm takes the crude objective trajectory and the constraints for velocity, acceleration and error tolerance, and plans the realizable trajectory. The realizable trajectory is compensated for delay dynamics. The proposed method brings the best possible performance as it

always maintains at least one of the given constraints (5), (6), or (7) within the entire motion.

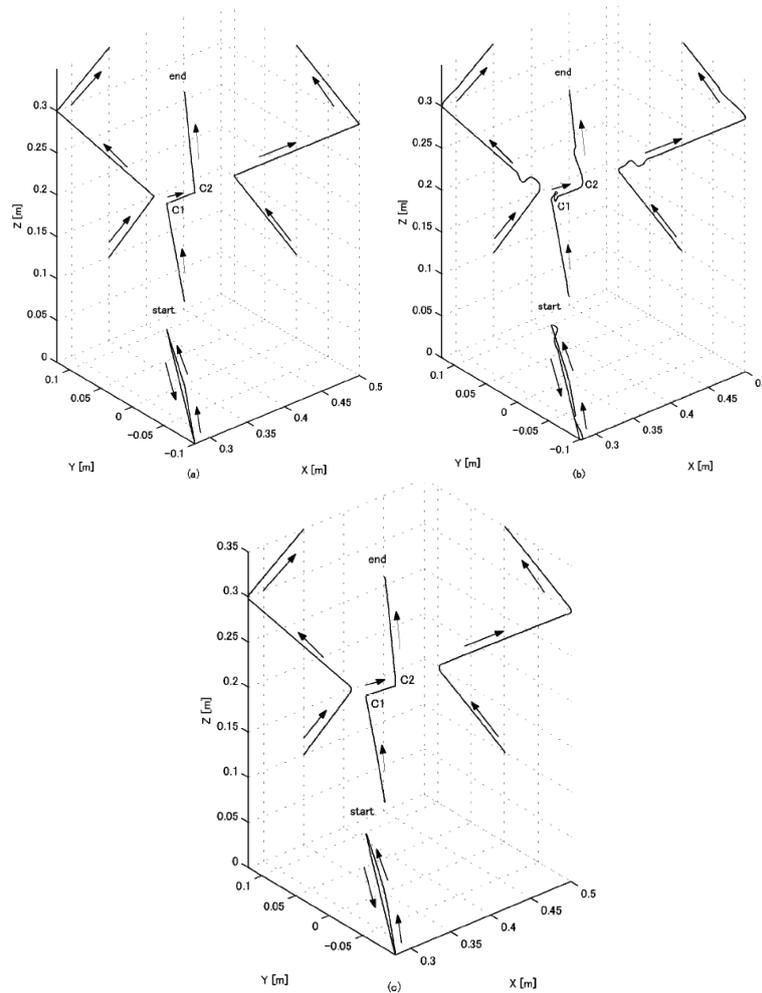


Figure 7. Profiles of the end-effector in 3-space, (a) objective trajectory, (b) resulting motion under conventional method (simulation), and (c) resulting motion under proposed method.

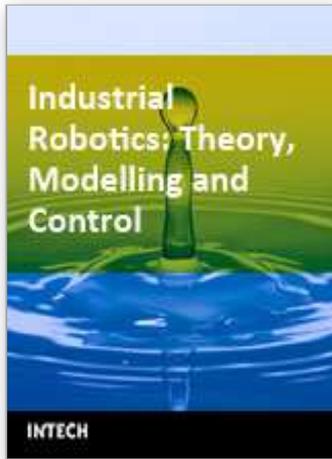
5. Conclusion

This chapter presented a new trajectory planning algorithm for industrial robot manipulators. This algorithm considers joint acceleration constraint, rated end-effector velocity, and trajectory error tolerance, and plans the realizable trajectory accordingly so that all these constraints are maintained in the best possible manner during the entire motion. A feed-forward compensator is also used to compensate the realizable trajectory against delay dynamics of the joints. The method was successfully tested on Performer MK-3s industrial robot manipulator using a complex three dimensional trajectory, in that very ac-

curate motion was realized without violating any of the constraints. The proposed method appears as a single feed-forward block in the control system, therefore, it could be conveniently incorporated into existing industrial manipulators without undertaking a significant cost or risk.

6. References

- Goto, S.; Nakamura, M., & Kyura, N.; (1997). Modified taught data method for industrial mechatronic servo-controller to achieve accurate contour control performance, *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 525B, June 1997
- Mittal, R. K. & Nagrath I. J. (2003). Chapter 2, In: *Robotics and Control*, 35-69, Tata McGraw Hill, 0-07-048293-4, New Delhi
- Munasinghe, S. R. & Nakamura, M. (2001). Optimum contouring of industrial robot arms under assigned velocity and torque constraints. *IEEE Transactions on Systems, Man, and Cybernetics-Part C*, Vol. 31, No. 2., (May, 2001) 159-167
- Munasinghe, S. R. & Nakamura, M.; (2002). Determination of maximum tangential velocity at trajectory corners in robot manipulator operation under torque constraint, *Proceedings of the Annual Conference of the Society of Instrumentation and Control Engineers(SICE)*, MA17-2, 1170-1175, August, 2002
- Munasinghe, S. R., Nakamura, M., Goto S., & Kyura, N., (2003). Pole selection of feedforward compensators considering bounded control input of industrial mechatronic systems. *IEEE Transactions on Industrial Electronics*, Vol. 50, No. 6, (December, 2003) 1199-1206
- Nakamura, M., Munasinghe, S. R., Goto, S., and Kyura, N., (2000). Enhanced contour control of SCARA robot under torque saturation constraint. *IEEE/ASME Transactions on Mechatronics*, Vol. 5, No. 4, (December, 2000) 437-440
- Paul, R. (1979). Manipulator Cartesian path control. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 11., (Nov., 1979) 702-711
- Sage, H. G., De Mathelin M. F., & Ostertag F., (1999). Robust control of industrial manipulators: a survey. *International Journal of Control*, Vol. 72, No. 16, (Nov., 1999) 1498-1522
- Shin, K. G., & Mckay N. D., (1985). Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, Vol. 30, No. 6, (Jun., 1985) 531-541



Industrial Robotics: Theory, Modelling and Control

Edited by Sam Cubero

ISBN 3-86611-285-8

Hard cover, 964 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, December, 2006

Published in print edition December, 2006

This book covers a wide range of topics relating to advanced industrial robotics, sensors and automation technologies. Although being highly technical and complex in nature, the papers presented in this book represent some of the latest cutting edge technologies and advancements in industrial robotics technology. This book covers topics such as networking, properties of manipulators, forward and inverse robot arm kinematics, motion path-planning, machine vision and many other practical topics too numerous to list here. The authors and editor of this book wish to inspire people, especially young ones, to get involved with robotic and mechatronic engineering technology and to develop new and exciting practical applications, perhaps using the ideas and concepts presented herein.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

S. R. Munasinghe and Masatoshi Nakamura (2006). Trajectory Planning and Control of Industrial Robot Manipulators, *Industrial Robotics: Theory, Modelling and Control*, Sam Cubero (Ed.), ISBN: 3-86611-285-8, InTech, Available from:
http://www.intechopen.com/books/industrial_robotics_theory_modelling_and_control/trajectory_planning_and_control_of_industrial_robot_manipulators

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2006 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.