High Speed Precise Control of Robot Arms with Assigned Speed
Under Torque Constraint by Trajectory Generation in Joint
Co-ordinates

S.R. Munasinghe*, M. Nakamura*, S. Aoki*, S. Goto* and N. Kyura**
*Department of Advanced Systems Control Engineering,
Saga University, Honjomachi, Saga 840-8502, Japan
**Department of Electrical Engineering,
Kinki University (in Kyushu), 11-6 Kayanomori, Iizuka 820-8555, Japan
*E-mail: rohan@cntl.ee.saga-u.ac.jp

Abstract
This paper presents a methodology for off-line trajectory
generation that guarantees high speed and precise end-
effector point-to-point (positioning) control of industrial
robot arms, within the bounds of the most relevant con-
straints. The constraints being addressed are the satura-
tion of joint torque and the limit of Cartesian velocity. In
the proposed methodology, shortest path of travel and max-
imum joint torque/acceleration utility is guaranteed so that
the true minimum-time trajectory is generated. Minimum-
time trajectory is then compensated for the delay dynam-
ics using a feed-forward compensator. An illustrative im-
plementation is also presented which has been carried out
with Performer MK-3s robot arm, by which attractive re-
results have been obtained. Being an off-line algorithm, the
proposed method can be conveniently introduced into the
existing servo control systems of industrial robot arms, and
to enhance their performance.

Keywords: Robot arm, Servo controller, Off-line
trajectory generation, Torque and velocity constraints,
Minimum-time trajectory, Delay dynamics compensation

1. INTRODUCTION
In today’s roboticized industries, robot arms carry out var-
ious tasks in the factory floor. Tasks such as welding,
cutting, sealing and painting require trajectory tracking
whereas tasks such as object handling require position-
ing control. In either case, high-speed and precise operation is
desired.
The main objective of this research focuses on high-speed
and precise operation for positioning of industrial robot
arms, and the solution be an off-line trajectory generation
algorithm so that it could be implemented in existing robot
arm controllers without a significant change in hardware
configuration.
In general, robot arm operations are carried out in two
phases, the first phase involves off-line trajectory genera-
tion, in that, time-based input position sequences, either
in Cartesian or joint co-ordinates are determined according
to a specific criterion. The second phase is the simultaneous
servoing of joint motors, by independent joint servo
systems for each joint, according to the joint position se-
quencies.
Luh et al. [1] explained trajectory generation with kine-
matic constraints though it had not been implemented ex-
perimentally. Trajectory generation algorithms proposed
by Shin et al. [2] [3], look objectively similar to that of
ours but significantly complex, in that, they require many
link parameters, though some of them are not known in
case of industrial robot arms.

Despite plenty of trajectory generation algorithms devised
by many researchers for positioning control, the consid-
eration of realistic industrial constraints and their appro-
priateness with regard to applications have not yet been
sufficiently addressed. In our previous work [4], we have
addressed high-speed end-effector positioning control of in-
dustrial robot arms, with emphasis to joint torque con-
straint, and in this paper we extend it giving provisions to
Cartesian velocity constraint. The effectiveness of the pro-
posed method has been verified by its implementation on
Performer MK-3s robot arm.

2. AN OVERVIEW OF CONTROL SYSTEM OF
INDUSTRIAL ROBOT ARMS

PROBLEM STATEMENT; CONSTRAINTS AND
CRITERION OF TRAJECTORY GENERATION
The optimum performance should be realised within the
limits of prevailing constraints which are 1. joint torque
limit and 2. Cartesian velocity limit. Within the entirety
of operation, torque of no joint can exceed its machine limit,
as constrained by
\[ |\tau_i| \leq \tau_{i,\text{max}} \]
(1)
where \(\tau_i\) and \(\tau_{i,\text{max}}\) stand for torque and its limiting value
of the \(i^{th}\) joint. These limits actually reflect power ampli-
der current ratings and consequent saturation of joint servo
drives. Industrial robot arms are also specified with velocity
rating \(v_{\text{max}}\) and thus, the end-effector motion is also
constrained by
\[ |v_{ee}| \leq v_{\text{max}} \]
(2)
where \(v_{ee}\) is the end-effector velocity in Cartesian co-
ordinates. The objective of trajectory generation is to
minimise traveling time and tracking error, bound to
the constraints (1) and (2). Minimum time control re-
quires 1. shortest following path, and 2. maximum joint
torque/acceleration utility.

ROBOT ARM SYSTEM ARCHITECTURE
A schematic of Performer MK-3s industrial robot arm and
its 2D world co-ordinate system are shown in Fig. 1(a) and
Fig. 1(b). Out of the five joints of the robot arm, the first
three joints from the base are shown. The end-effector is attached to the hand of the arm, that consists of three concentrated joints which are manipulated for orientation control of the end-effector tool.

Fig. 1(b) shows 2D world co-ordinate system and link placement where $L_0=0.135$ [m], $L_1=0.250$ [m], and $L_2=0.215$ [m]. Industrial robot arms are devised with independent joint controllers based on the decoupled joint dynamics, and typically driven by PID servo systems. Servo systems have joint motors which are actuated with current or voltage controllers that implement torque control of joint motors, according to the resident PID control algorithm. The required input, i.e., the time-base sequences of either position or velocity is given by the reference input generator that generates the joint trajectories of all joints, according to a specific trajectory generation algorithm.

**ROBOT ARM KINEMATICS**

Robot arm kinematics is two-fold, forward kinematics and inverse kinematics (or arm solution). Forward kinematic equations determine Cartesian position and orientation of the end-effector, given the arm configuration in joint co-ordinates. Kinematics always result in a unique solution as given by:

$$z = L_0 + L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad (3)$$

$$y = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2). \quad (4)$$

for the end-effector position of the Performer MK-3s robot arm shown in Fig. 1. Inverse kinematics solves for the arm configuration in joint co-ordinates, given the position and orientation of the end-effector in Cartesian co-ordinates. Unlike kinematics, inverse kinematics is ill conditioned and cannot be formulated as a closed explicit solution, except for simple manipulators. The simplest among different arm solution schemes is the geometric approach proposed by Lee et al. [5] which could be applied to most industrial robot arms to obtain a closed form arm solution as given by:

$$\theta_1 = \tan^{-1}\left(\frac{c}{y} \right) - \cos^{-1}\left(\frac{L_1^2 - L_2^2 + c^2 + y^2}{2L_1 \sqrt{c^2 + y^2}} \right) \quad (5)$$

$$\theta_2 = \pi - \cos^{-1}\left(\frac{L_1^2 + L_2^2 - c^2 - y^2}{2L_1 L_2} \right) \quad (6)$$

where $c = x - L_0$, for the Performer MK-3s robot arm shown in Fig. 1.

**DYNAMICS AND CONTROL OF INDUSTRIAL ROBOT ARMS**

Almost all industrial robot arms are kinematic controlled. Kinematic control of industrial robot arms are based on the decoupled, linear joint servo model which has been thoroughly established in [6]. The two-link version of this dynamic model is given in Fig. 2, with the additional Cartesian velocity constraint.

Cartesian velocity constraint requires that the input Cartesian trajectory be bound to maximum velocity, within the entirety of operation. Following dynamics is governed by the two servo parameters, $K_{pi}$ and $K_{vi}$ known as position loop gain and velocity loop gain, respectively. The determination procedures of these parameters can be found in [6], and those for the Performer MK-3s robot arm are $K_{pi}=25$ [1/s] and $K_{vi}=150$ [1/s], $i=1,2$. Joint dynamics is given by:

$$G_j(s) = \frac{\Theta_j(s)}{U_j(s)} = \frac{K_{pi} K_{ej}}{s^2 + K_{vi} s + K_{pj} K_{ej}} \quad (7)$$

which was derived from Fig. 2. However, when joint acceleration saturates, non-linear dynamics will be resulted. In kinematic control, torque is addressed through joint acceleration, in that, torque constraint is represented by a limiting acceleration. Making provisions to torque saturation be included, joint dynamics in (7) can be written in time domain by:

$$\ddot{\theta}_j(t) = \text{sat}\left[K_{pj}(u_j(t) - \dot{\theta}_j(t)) - \dot{\theta}_j(t)\right] \quad (8)$$

where

$$\text{sat}(z) = \begin{cases} A_{j,max} & \text{if } (z > A_{j,max}) \\ z & \text{if } (-A_{j,min} \leq z \leq A_{j,min}) \\ -A_{j,max} & \text{if } (z < -A_{j,min}) \end{cases}$$

and $A_{j,max}$ represents the limiting acceleration of $j^{th}$ joint. A thorough explanation of joint dynamics of industrial robot arms and their appropriateness is found in [6]. Typically, $K_{pi}$ and $K_{vi}$ are determined in such a way that to realise critically damped joint dynamics.

Industrial robot arms are employed in pre-determined tasks, in that the workspace and non-linear effects on joint torque due to centrifugal, Coriolis and gravity loading are quantifiable. Thus, a global maximum joint accelerations can be set for all joints to guarantee linear torque profiles within the entirety of a given operation. Therefore, we assume torque saturation be represented by a corresponding saturation in joint acceleration.

---

**Fig. 1** Performer MK-3s articulated robot arm (a) Control system schematic, (b) Convention of robot arm 2D co-ordinate system

**Fig. 2** Joint dynamic model of two link industrial robot arm
3. TRAJECTORY GENERATION BOUND TO INDUSTRIAL CONSTRAINTS

METHODOLOGY OF TRAJECTORY GENERATION

The objective Cartesian trajectory is inscribed by a set of equispaced Cartesian points \( \{(x_k, y_k) \mid k = 1, 2, \ldots, N \} \) known as "Cartesian knot set", where \( N \) is the number of knots. The corresponding "joint knot set" \( \{\{\theta_{k,h,k,b} \mid k = 1, 2, \ldots, N\} \) is determined by (5) and (6).

Based on the knot sets, the trajectory is generated by parts as shown in Fig. 3, in that three segments are separately generated, the start segment \( P_sP_1 \), middle segment \( P_1P_2 \) and the end segment \( P_2P_e \) and then, they are connected on the same time base. In the start segment, at least one joint is driven with its acceleration limit, within two successive knot points. End segment is the segment within which at least one joint is driven with its maximum deceleration, until zero velocity is attained. In the middle segment, end-effector travels with almost its limiting velocity in Cartesian co-ordinates.

In Fig. 3, "forward path" and "reverse path" are the trajectories generated along \( P_sP_1 \) and \( P_1P_2 \) respectively, with at least one joint be driven with the limiting acceleration or deceleration, either appropriate without concern to Cartesian velocity constraint. But along all forward and reverse paths, instantaneous end-effector velocity is determined to locate switching points \( P_1 \) and \( P_2 \) on the respective trajectories where end-effector reaches its velocity limit. The instantaneous end-effector velocity \( v_{ee} \) is calculated by

\[
\begin{align*}
\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} &= J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \\
v_{ee} &= \sqrt{\dot{x}^2 + \dot{y}^2}
\end{align*}
\]

where

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

Middle trajectory is generated in Cartesian co-ordinates, with Cartesian velocity limit. In this way, it is certain that the generated trajectory be the true minimum-time trajectory, bound to the constraints (1) and (2). Robot mechanism possesses a delay in dynamics which causes tracking deteriorations and other control problems such as system stability. The minimum-time trajectory is therefore compensated by a feed-forward compensator which is shown as \( F_f(s) \) in Fig. 3. Detailed explanation of minimum-time trajectory generation and its compensation for delay dynamics are explained in the following section.

GENERATION OF MINIMUM-TIME TRAJECTORY

Trajectory Generation in Joint Co-ordinates

The procedure of minimum-time trajectory generation is illustrated step-by-step in Fig. 4. Fig. 4(a) shows the objective trajectory in Cartesian space where \( P_s \), \( P_e \), \( P_1 \), and \( P_2 \) stand for start point, end point, forward path switching point, and reverse path switching point, in that respect. Fig. 4(b) shows the same information in joint co-ordinates. Middle trajectory is generated in Cartesian co-ordinates, with Cartesian velocity limit. In this way, it is certain that the generated trajectory be the true minimum-time trajectory, bound to the constraints (1) and (2). Robot mechanism possesses a delay in dynamics which causes tracking deteriorations and other control problems such as system stability. The minimum-time trajectory is therefore compensated by a feed-forward compensator which is shown as \( F_f(s) \) in Fig. 3. Detailed explanation of minimum-time trajectory generation and its compensation for delay dynamics are explained in the following section.

Fig. 4 Generation of minimum-time trajectory (a) Objective trajectory in Cartesian space, (b) Objective trajectory in joint space, (c) End-effector velocity along forward and reverse paths, and (d) End-effector velocity along the minimum-time trajectory.
path trajectory generation. The both is carried out in joint co-ordinates, as explained below. The minimum time $h_{j,k}$ of the motion of $j^{th}$ joint from $\theta_{j,k}$ to $\theta_{j,k+1}$ is determined according to uniform acceleration motion $\dot{\theta}_{j,k+1} = \dot{\theta}_{j,k} + A_j h_{j,k}/2$ and is given by

$$h_{j,k} = \frac{-\dot{\theta}_{j,k} + \sqrt{\dot{\theta}_{j,k}^2 + 4 A_j (\theta_{j,k+1} - \theta_{j,k})}}{2 A_j}$$

(11)

where

$$A_j = \begin{cases} A_j, \text{max} & \text{if } \theta_{j,k+1} > \theta_{j,k} \\ -A_j, \text{max} & \text{if } \theta_{j,k+1} < \theta_{j,k} \\ 0 & \text{otherwise} \end{cases}$$

The symbol $h_{j,k}$ is the minimum time of the motion of $j^{th}$ joint between $k$th and $k+1$th knot points, without saturation of joint torque/acceleration. In order to generate the trajectory between the two knot points, it is required that the minimum time be determined ensuring none of the joints leads to torque/acceleration saturation. Therefore, the maximum of $h_{j,k}$ is selected according to

$$h_k = \max_j (h_{j,k})$$

(12)

as the inter-knot time $h_k$ for the trajectory generation between $k^{th}$ and $k+1^{th}$ joints. Then, accelerations of all joints are calculated by rearranging $\dot{\theta}_{j,k+1} = \dot{\theta}_{j,k} + A_j h_{j,k}^2/2$ and is given by

$$\dot{\theta}_{j,k} = \frac{2(\theta_{j,k+1} - \theta_{j,k} - \dot{\theta}_{j,k} h_k^2)}{h_k^2}$$

(13)

where $\dot{\theta}_{j,k}$ is calculated using (11). The trajectory between the two knot points is then generated according to uniform acceleration motion, as given by

$$\theta_{j,k}(t_k + t) = \theta_{j,k} + \dot{\theta}_{j,k} t + \frac{\dot{\theta}_{j,k} t^2}{2} ; t \in [0, t_{k+1} - t_k]$$

(14)

That concludes joint space trajectory generation between $k^{th}$ and $k+1^{th}$ joints. Knot position is advanced by one and the procedure continues so that forward path is generated from $P_s$ to $P_e$ and reverse path is generated from $P_e$ to $P_s$. While forward path and reverse path are generated in joint co-ordinates, Cartesian position and end-effector velocity are calculated by (3),(4) and (9),(10), respectively, and switching points $P_1(x(t_f), y(t_f))$ and $P_2(x(t_0), y(t_0))$ are located when the end-effector have approached its limiting velocity. Then the part of the forward path from $P_s$ to $P_1$ is chosen to be the start segment. Similarly, the part of the reverse path from $P_2$ to $P_e$ is chosen to be the end segment.

Trajectory Generation in Cartesian Co-ordinates

In the middle segment, i.e. between switching points, trajectory is generated according to uniform acceleration motion in Cartesian space. Straight line length of this segment $l_c$ is determined by

$$l_c = \sqrt{(x(t_r) - x(t_s))^2 + (y(t_r) - y(t_s))^2}$$

(15)

With $l_c$ and uniform mean velocity $\left((v_f + v_r)/2\right)$, travelling time of the middle segment is calculated by

$$t_c = 2l_c/(v_f + v_r).$$

(16)

Uniform Cartesian acceleration that is required to change end-effector velocity from $v_f$ to $v_r$ within $t_c$ is determined by

$$A_c = (v_r - v_f)/t_c.$$ (17)

Then, according to uniform acceleration motion, trajectory in the middle segment is generated by

$$x(t) = x(t_f) + x(t_f)(t - t_f) + A_c \cos(\alpha(t-t_f))t/2$$

(18)

$$y(t) = y(t_f) + y(t_f)(t - t_f) + A_c \sin(\alpha(t-t_f))/2$$

(19)

where $\alpha = \frac{\sin^{-1}((y(t_s) - y(t_f))/L)}{2}$. The middle segment in joint co-ordinates is determined by transformation of $(x(t), y(t))$ into joint space by inverse kinematics given in (5) and (6).

**DELAY DYNAMICS COMPENSATION**

Response delay of robot mechanism for a time-based input position sequence causes poor tracking performance and control problems such as stability. With the major concern to the convenient industrial implementation, an algorithm for delay dynamics composition has been devised by Goto et al. [7] based on the pole placement regulator theory which has been synthesised in the form of a feed-forward compensator $F_j(s)$ as shown in Fig. 3. The second order compensator dynamics is given by

$$F_j(s) = \frac{-A_0 s^2 + A_1 s + A_0}{(s - \mu_1)(s - \mu_2)(s - \mu_3)}$$

(20)

where

$$a_0 = -\mu_1 \mu_2 \mu_3$$

$$a_1 = (K_{e1} + \gamma_1)(\mu_1 + \mu_2) + K_{e2} \mu_3 + K_{e3} \mu_3$$

$$a_2 = \frac{1}{K_{e1}} \left\{(K_{v1} + \gamma_1)(\mu_1 + \mu_2) + K_{v2} \mu_3 + K_{v3} \mu_3 \right\}$$

$$a_3 = \frac{1}{K_{e2} K_{e3}} \left\{(K_{v1} + \gamma_1)(\mu_1 + \mu_2) + K_{v1} \mu_3 \mu_3 + K_{v2} \mu_3 + K_{v3} \mu_3 \right\}.$$ (21)

Time-based co-ordinate profiles $u_i(t)$ of all joints are compensated by its delay dynamics compensator $F_j(s)$ in the time domain. The determination procedure of the poles $\mu_1, \mu_2$, and $\mu_3$ of the compensator is a mere tuning process, and based on heuristics with regard to actual industrial applications, though appropriate analytical procedures can also be developed.

4. EVALUATION OF PROPOSED TRAJECTORY GENERATION METHOD

**CONDITIONS FOR SIMULATION AND EXPERIMENT**

Objective trajectory was specified by a 25 [cm] long vertical straight line from $P_s(0.35, 0.1)$ to $P_e(0.35, 0.35)$ in the XY plane as shown in Fig. 4(a). Velocity limit was set to 0.15 [m/s] and the global acceleration limit of torque saturation was set to 0.72 [rad/s^2], for all joints. In the industry, knot-points are selected equispaced in Cartesian space. In the experiment with the Performer MK-3s robot arm, the sampling interval $T_s$ was set to 2 [ms] and 66 [ms] and inter-knot distance was selected as 1 [cm].
y & y switching point Pz((0.35, 0.28257 [m])) Cartesian with \( I_1 = 0.10539 \text{ [m]} \), I.108 [s] with ZI1 = 0.14994 [m/s] (recall Fig. 4(c)).

Coefficients of the delay dynamics compensator were evaluated for Performer MK-3s robot arm dynamics, poles of the delay dynamics compensators were set to \( \mu_1 = \mu_2 = -250 [1/{\text{s}}] \) and \( \mu_3 = -300 [1/{\text{s}}] \) according to [4]. Corresponding coefficients of the delay dynamics compensator were evaluated to \( a_0 = 1.875 \times 10^7, a_1 = 8.65 \times 10^5, a_2 = 9.6 \times 10^3 \), and \( a_3 = 5.007 \times 10^1 \).

Minimum time trajectory was compensated by delay dynamics compensators and applied on the Performer MK-3s robot arm.

### RESULTS AND EVALUATION

The obtained results are shown in Fig. 5. The first and second rows illustrate simulation and experimental results of the proposed method whereas the third and fourth rows illustrate simulation results of the conventional methods, for two different assigned velocities. About columns, the first column shows tracking results and the second column shows the velocity of the end-effector. The third and fourth columns show joint accelerations.

As for the comparison requirements, a trajectory was generated from \( P_1 \) to \( P_2 \) according to a conventional method in which none of the trajectory generation strategies have been included. In this conventional method, a uniform end-effector velocity was assigned from the start to the end, and no compensations have been introduced as for the prevailing constraints and delay dynamics. Table 1 shows quantitative evaluation of results, in terms of root mean square error \( e_{RMS} \) and total travelling time \( T_{total} \) as defined by

\[
\text{error} = \sqrt{\sum (x_{\text{out}} - x_{\text{in}})^2 + (y_{\text{out}} - y_{\text{in}})^2}
\]

\[
T_{total} = \int dt = t_e - t_s, \text{ where } t_e \text{ and } t_s \text{ stamps the start and end of the manipulation, and symbol } i \text{ is the index of data sampled at } 2 \text{ [ms].}
\]

\( e_{RMS} \) and \( T_{total} \) prove, in a way, the validity of the selected joint dynamic model, though it is excessively simplified and linear. The acceleration profiles of the proposed method were obtained from filtering the original data by a fourth order chebyshev infinite impulse response (IIR) low pass filter (LPF). As can be seen in the results of "conventional 1" method in Fig. 5, conventional trajectory causes robot arm significantly oscillate at the end.

Such oscillations can only be alleviated by reducing the assigned velocity in expense of long travelling time, as indicated by the results of "conventional 2" method in Fig. 5. In contrary, proposed method does not cause such behavior as it has provisions for velocity reduction as the end-effector approaches the end point. According to the experimental results of the proposed method, 1. end-effector travels with the limiting velocity during the middle segment and 2. at least one joint is driven with the limiting torque/acceleration within start and end segments. Hence, it confirms the true minimum-time manipulation of the robot arm, within the prevailing constraints. According to the experimental tracking results of the proposed method, end-effector moves almost perfectly, along the objective trajectory. The error has been assessed to 0.00273 [m].

### DISCUSSION

Most trajectory generation algorithms found in scientific literature have been devised on the basis of Lagrange-Euler and/or Newton-Euler dynamics or on an extended model of either of them. Such models and their control algorithms are relatively complex and consume long computation time and high cost for digital signal processing electronics. It is also required that all physical quantities of the robot arm be known though some of them are not possible to measure or estimate reliably. More often they are not simple and robust enough for rough use in most industries too. The high cost and lack of practitioner friendliness causes such algorithms infeasible for industrial implementation.

The proposed method is perfectly consistent with the dynamic model and control method 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 [6]. With the proposed method, the speed of the control loop can be raised up towards its servo controller limit as their is no computation burden inside the reference input generator. 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 any number of links.

### 5. CONCLUSIONS

Stick to the linear decoupled servo model of industrial robot arms, a off-line trajectory generation algorithm has been devised which guarantees high speed and precise positioning control of the end-effector. Joints are actuated with its torque limits when Cartesian velocity constraint is not in effect, and when it is in effect, it is maintained within the joint torque limits. It outlines the ultimate trajectory generation that exploits maximum servo capacity within the prevailing constraints. The selected constraints are applied in most actual industrial applications and therefore, this method would cer-
certainly have strong industrial implications. The proposed trajectory generation algorithm is a mere off-line feed-forward block in the robot control system so that it will be easily conceived by both the system analyst as well as the practitioner, without exerting much effort. It further confirms that no hardware changes or considerable reconfiguration is required for the implementation of the proposed method on the existing industrial robot arm control systems.

6. REFERENCES

