Precise, Jerk-free Contouring of Industrial Robot Arms with Trajectory Allowance Under Torque and Velocity Constraints

S.R. Munasinghe, M. Nakamura, T. Iwanaga, S. Goto and N. Kyura

Department of Advanced Systems Control Engineering, Saga University
Honjomachi 1, Saga 840-8502, Japan
roham@cntl.ee.saga-u.ac.jp

Abstract—This paper addresses the contouring problem of articulated industrial robot arms and proposes a practical solution to realize precise, jerk-free servoing. Proposed solution is an off-line trajectory construction algorithm that addresses the relevant industrial constraints and specifications namely, velocity constraint, torque/acceleration constraint and trajectory allowance. Spline approximation is used to eliminate jerking. A forward compensator is used to compensate for delay dynamics. Proposed method has been experimented with a Performer Mk3s industrial robot arm in that improved performances have been demonstrated.

I. INTRODUCTION

In industrial robotics, contouring applications such as welding, cutting etc., specify a Cartesian path to be traced by the end-effector. The specified end-effector motion is realized by solving it to joint trajectories of the articulated arm and actuating joints accordingly [1]. However, direct resolution is not appropriate in most practical cases as constraints exist in both Cartesian and joint spaces and it is needed to construct piece-wise trajectories in both spaces while considering respective local constraints. Finally, all piece-wise trajectories are transformed into joint space and merged together in correct sequence.

Nevertheless, merged joint trajectories most likely possess discontinuities in velocity and acceleration at merged locations and cause unnecessary jerking in real-time servoing. Such discontinuities could be eliminated using approximation techniques such as what is proposed by Luh et al. [2] and Lin et al. [3].

Trajectory construction for two dimensional applications have been addressed considering torque constraint [4], torque and velocity constraints [5]. In [6], trajectories were successfully constructed for three dimensional applications with consideration to trajectory allowance. The maximum end-effector velocity at a corner was theoretically derived in [7] and better trajectories were constructed.

In this research, we further improve [7] incorporating jerk reduction by spline approximation.

II. INDUSTRIAL ROBOT ARM

A. System Architecture and Joint Dynamics

A system schematic of a typical industrial robot arm is shown in Fig. 1. End-effector position $P$ in Cartesian co-ordinates is denoted by $(x, y, z)$ and its corresponding joint position is denoted by $(\theta_1, \theta_2, \theta_3)$. In real time servoing, joint motors are actuated by the servo controller according to the taught data, $U(s)$. Taught data is the time based joint position sequences of all joints. Reference input generator is a PC which runs the user written control algorithm.

![System schematic of a typical industrial robot arm](image)

Fig. 1. System schematic of a typical industrial robot arm

According to the articulated arm shown in Fig. 1, coordinate transformation from joint space to Cartesian space is given by

$$
x = [L_1 + L_2 \sin \theta_2 + L_3 \sin (\theta_2 + \theta_3)] \cos \theta_1
\ny = [L_1 + L_2 \sin \theta_2 + L_3 \sin (\theta_2 + \theta_3)] \sin \theta_1
\nz = L_2 \cos \theta_2 + L_3 \cos (\theta_2 + \theta_3)
$$

(1)

where $L_i; j = 1, 2, 3$ stands for the length of $j^{th}$ link. The inverse transformation of (1) can be formulated as given by

$$
\theta_1 = \tan^{-1} \left( \frac{y}{x} \right)
\theta_2 = \frac{x}{c} \tan^{-1} \left( \frac{c^2 + z^2 - L_2^2 - L_3^2}{2L_2c^2 - L_3^2} \right)
\theta_3 = \cos^{-1} \left( \frac{z^2 + c^2 - L_3^2}{2L_2c^2 - L_3^2} \right)
$$

(2)

where $c = \sqrt{x^2 + y^2}$. The end-effector velocity $v$ in terms of its components $(\dot{x}, \dot{y}, \dot{z})$ along major axes can be resolved into corresponding angular velocities
\[ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \]  
(3)

in that, Jacobian, \( J \) is

\[
\begin{bmatrix}
-s_1 [L_1 + L_2 s_2 + L_3 s_2 s_3] & c_1 [L_2 c_2 + L_3 c_2 c_3] & L_3 c_1 c_3 \\
s_1 [L_2 c_2 + L_3 c_2 c_3] & s_1 [L_2 c_2 + L_3 c_2 c_3] & -L_3 s_1 c_3 \\
0 & -L_2 s_2 & -L_3 s_2 c_3
\end{bmatrix}
\]

where \( s_i = \sin(\theta_i) \), \( c_m = \cos(\theta_m) \), \( s_{1,m} = \sin(\theta_1 + \theta_m) \), and \( c_{1,m} = \cos(\theta_1 + \theta_m) \).

Industrial robot arms are actuated by independent servo drives in each joint. Servo drives operate on position and/or velocity error whereas whatever nonlinear effects exist considered as disturbances. Such servo drives are designed for the worst case so that sufficient robustness and stability are guaranteed as long as servoing is implemented within specifications. On this basis, an appropriate linear model for joint dynamics can be constructed as shown in Fig.2.

![Servo dynamic model of the industrial robot arm in 3-DOF](image)

where \( K_p^j \) and \( K_d^j \) are position and velocity loop gains. These parameters are tuned and set by practitioners in order to realize optimum performance. Tuning process is based on trial and error and it is not troublesome, as only two servo parameters exist. This model gives sufficient robustness in industrial applications and it is more appropriate when high gearing and low link inertia exist [8].

Second order linear servo dynamics \( G_j(s) \) is given by

\[ G_j(s) = \frac{\Theta_j(s)}{U_j(s)} = \frac{K_p^j K_d^j}{s^2 + 2K_p^j s + K_d^j} ; \]  
(4)

where \( U_j(s) \) and \( \Theta_j(s) \), stand for taught data and joint position.

**B. Problem Statement**

There are two major constraints addressed in this research. 1. torque/acceleration constraint and 2. velocity constraint. In kinematic control [9] torque is assumed proportional to acceleration and a global limit for acceleration is specified to make sure that torque does not saturate in real time servoing. End-effector velocity is also limited to an upper limit to make sure that the rated RPM(revolutions per minute) of the servo motor is not exceeded. These two constraints can be formulated as

\[ \ddot{\theta}_j \leq \ddot{\theta}_j^{\text{max}} \]  
(5)

\[ v \leq \begin{cases} v_a & \text{for straight lines} \\ v_t^{\text{max}} & \text{for corners} \end{cases} \]  
(6)

where \( \ddot{\theta}_j \) and \( \ddot{\theta}_j^{\text{max}} \) stand for joint acceleration and its limit. Assigned velocity and maximum tangential velocity of the end-effector at a corner are denoted by \( v_a \) and \( v_t^{\text{max}} \).

In contouring operations, objective trajectory is specified with an allowance \( \rho \) so that any contour is accepted as long as it falls within the allowance.

**III. Trajectory Construction for Industrial Robot Arms Bound to Velocity and Torque Constraints**

**A. Proposed Methodology**

The proposed methodology for trajectory construction is illustrated in Fig.3. Objective trajectory \( \mathcal{O}(s) \) is specified by a set of key Cartesian points including start point, all corners and end point. Constraints for torque/acceleration and velocity and trajectory allowance are also specified together.

![Proposed trajectory construction algorithm](image)

Objective trajectory cannot be realized due to the existence of constraints. Thus, special treatments are necessary to assure that all constraints are maintained within the entirety of operation. The purpose of trajectory construction is to create a realizable trajectory \( \mathcal{R}(s) \) considering given constraints (5), (6) and trajectory allowance \( \rho \).

Different constraints exist in different parts of the objective trajectory. Therefore, realizable trajectory is constructed piece-wise, and merged in correct sequence. When merging takes place, acceleration dis-
continuity (jerk) occurs at merging points. To eliminate jerk, realizable trajectory is approximated with spline interpolations and an approximated trajectory \( Q(s) \) is constructed. Approximated trajectory is compensated for delay dynamics using a forward compensator to obtain taught data \( U(s) \). Taught data is used to actuate the robot in real time servoing.

### B. Construction of Realizable Trajectory

The construction procedure of realizable trajectory is illustrated in Fig. 4. First, objective trajectory is solved into a. corners and b. straight lines. Corners are constructed in Cartesian space according to maximum tangential velocity criterion (MTCV) in that \( v = \nu_{t \text{max}} \) along the curvature (Fig. 4a1). Then, it is transformed into joint space using inverse kinematics described in (2) (Fig. 4a2).

Straight lines are transformed into joint space and trajectory is constructed in three pieces, namely, accelerating segment, cruising segment, and decelerating segment (Fig. 4b1, b4). The three segments are constructed using maximum acceleration criterion (MAC), assigned velocity criterion (AVC) and maximum deceleration criterion (MDC), respectively. These criterions will be described in subsection IIIB2 in more detail. The three segments are merged and the entire straight line is constructed in joint space.

#### B.1 Trajectory Corners

The construction procedure for a sharp Cartesian corner \( \triangle ABC \) in the objective trajectory is shown in Fig. 5. Fig. 5(a) shows the corner with given allowance \( \rho \) and corner angle \( \beta \). Sharp corners cannot be realized and therefore it is proposed to introduce a circular arc as shown in Fig. 5(b). However, to comply with the given allowance, the curvature should pass through point \( H \), tangential to \( A'B' \) and \( B'C' \). To satisfy this condition, radius of curvature \( r \) is determined by

\[
r = \frac{2\rho}{1 - \sin(\beta/2)}
\]

where \( \beta = \arccos((AB^2 + BC^2 - AC^2)/(2 \cdot BC \cdot AC)) \). The curvature from \( D \) to \( E \) through \( H \) is constructed with maximum tangential velocity criterion (MTVC), where, the end-effector travels with maximum tangential velocity \( \nu_{t \text{max}} \). Determination of \( \nu_{t \text{max}} \) is carried out as follows.

By differentiating (3), we obtain

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{z}
\end{bmatrix} = \mathbf{J}_x
\begin{bmatrix}
\ddot{\theta}_1 \\
\ddot{\theta}_2 \\
\ddot{\theta}_3
\end{bmatrix} + \mathbf{H}_x
\begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_2 \\
\dot{\theta}_3
\end{bmatrix}
\]

in that Jacobian \( \mathbf{J}_x \) and Hessian \( \mathbf{H}_x \) are given by

\[
\mathbf{J}_x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}
\]

and \( \mathbf{H}_x = \begin{bmatrix}
-c_1(L_1 + L_2s_2 + L_3s_3) & -s_1(c_2L_2 + c_3s_3) & -s_3s_2c_3 \\
-s_1(L_2c_2 + L_3c_3) & -s_1s_2c_3 & -s_3c_2c_3 \\
0 & 0 & 1
\end{bmatrix}
\]

columns \( \dot{x}, \dot{y}, \dot{z} \) and \( (\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3) \) stand for acceleration in Cartesian and joint spaces. Matrices \( \mathbf{J}_x \) and \( \mathbf{H}_x \) are Jacobian and Hessian of \( \ddot{x} \). In industrial applications, end-effector velocity is lowered at trajectory corners and it is reasonable to neglect velocity terms in (8)[7]. Then, (8) could be approximated to

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{z}
\end{bmatrix} = \mathbf{J}_x
\begin{bmatrix}
\ddot{\theta}_1 \\
\ddot{\theta}_2 \\
\ddot{\theta}_3
\end{bmatrix}
\]

and it could be rearranged to

\[
\begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_2 \\
\dot{\theta}_3
\end{bmatrix} = \mathbf{J}_x^{-1}[\mathbf{a}]
\begin{bmatrix}
\cos \beta_x \\
\cos \beta_y \\
\cos \beta_z
\end{bmatrix}
\]

in that \( \mathbf{a} \) and \( [\cos \beta_x, \cos \beta_y, \cos \beta_z]^T \) are Cartesian acceleration and column of directional cosines of \( \mathbf{a} \). By expanding (10),

\[
\dot{\theta}_j = \phi_j[a]
\]

is obtained where,

\[
\phi_j = (h_{j,1} \cos \beta_x + h_{j,2} \cos \beta_y + h_{j,3} \cos \beta_z)
\]

in that \( h_{j,i} \) is the element of \( \mathbf{J}_x^{-1} \) in \( j \)-th row and \( i \)-th column. Then, a relationship for maximum accelerations \( \mathbf{a}_{\text{max}} \) could be determined as given by

\[
|\mathbf{a}_{\text{max}}| = \min_{j} \left\{ \frac{\dot{\theta}_{j \text{max}}}{\phi_j} \right\}
\]

minimum value of the right hand side is selected so that all three joints move within their acceleration limits when the end effector moves with \( \mathbf{a}_{\text{max}} \). According to centripetal acceleration, \( \nu_{t \text{max}} \) is given by

\[
\nu_{t \text{max}} = \sqrt{|\mathbf{a}_{\text{max}}|}t
\]

Then, using sampling interal \( t_s \), corner curvature could be determined at each \( \delta \)

\[
\delta = \nu_{t \text{max}}t_s/t
\]
B.2 Straight Lines

Trajectory construction procedure for straight lines is shown in Fig. 6. First, a series of knot points \( P_k = (x_k, y_k, z_k) \) is located along \( P_1 P_2 \) where \( k = 0, 1, 2, \ldots \) is the knot index. Then, starting from \( P_1 \) \((v = v_1 < v_2)\), accelerating segment is constructed taking two consecutive knot points at a time, according to MAC procedure. The MAC procedure is explained as follows. The minimum time interval each joint would take to travel between the two knot points is given by

\[
\tau_{\text{min}}^{(j)}(k) = \begin{cases} 
\frac{\sqrt{\theta_j^2(k) + 2\tau_{\text{max}}^{(j)}} \Delta \theta_j(k) - \dot{\theta}_j(k)}{\Delta \theta_j(k)} & \text{if } \Delta \theta_j(k) > 0 \\
\frac{\sqrt{\theta_j^2(k) + 2\tau_{\text{max}}^{(j)}} \Delta \theta_j(k) + \dot{\theta}_j(k)}{\Delta \theta_j(k)} & \text{if } \Delta \theta_j(k) < 0
\end{cases} \tag{16}
\]

where \( \Delta \theta_j = \theta_j(k + 1) - \theta_j(k) \). The maximum of all \( \tau_{\text{min}}^{(j)}(k) \) is selected as the minimum possible interval \( \bar{\tau}_{\text{min}}(k) \) for trajectory construction between the two knot points as given by

\[
\bar{\tau}_{\text{min}}(k) = \max_j \{ \tau_{\text{min}}^{(j)}(k) \} \tag{17}
\]

Using \( \bar{\tau}_{\text{min}}(k) \), joint accelerations for all joint are recalculated as given by

\[
\ddot{\theta}_j(k) = \frac{2(\Delta \theta_j(k) - \dot{\theta}_j(k) \bar{\tau}_{\text{min}}(k))}{\bar{\tau}_{\text{min}}(k)^2} \tag{18}
\]

Then, the trajectory between the two knot points is constructed in joint co-ordinates by

\[
\begin{align*}
\dot{\theta}_j(k, t) &= \dot{\theta}_j(k) + \ddot{\theta}_j(k) t; \quad [kt < t < (k + 1)t] \\
\theta_j(k, t) &= \theta_j(k) + \dot{\theta}_j(k) t + 0.5\ddot{\theta}_j(k) t^2; \quad [kt < t < (k + 1)t]
\end{align*}
\tag{19}
\]

This process continues as knot index \( k \) advances. However, end-effector velocity increases gradually as MAC procedure applies and at point \( P_F \) assigned velocity is reached (Fig. 6). Therefore, accelerating segment terminates at \( P_F \).

The decelerating segment \( P_R P_2 \) is constructed by applying the same procedure starting from \( P_2 \), advancing in the direction of \( P_2 P_1 \). This is the MDC process and it is the reverse application of MAC process. According to MDC, end-effector velocity increasing until it reaches the assigned velocity at point \( P_R \). Therefore, \( P_R P_2 \) is the decelerating segment.

From \( P_F \) to \( P_R \) is the cruising segment, where end-effector travels with the assigned velocity. Cruising segment is constructed as given by

\[
\begin{bmatrix}
x(t) \\
y(t) \\
z(t)
\end{bmatrix} = 
\begin{bmatrix}
v_x^F \\
v_y^F \\
v_z^F
\end{bmatrix} \cdot t + 
\begin{bmatrix}
x(P_F) \\
y(P_F) \\
z(P_F)
\end{bmatrix}
\tag{21}
\]

where \([v_x^F, v_y^F, v_z^F]^T\) is the column of velocity components of \( P_0 \) along major axes as the end-effector moves along \( P_F P_R \). Point co-ordinates \((x(P_F), y(P_F), z(P_F))\) represents point \( P_F \). Accelerating segment, cruising segment and decelerating segment are merged together to construct the entire straight line segment. Finally, all straight lines and corners are merged and that constructs the entire realizable trajectory.

\section*{C. Jerk Reduction with Spline Approximation}

It has been observed that unnecessary jerking occurs at merging points of the realizable trajectory. Therefore, it is proposed to apply spline approximation for the realizable trajectory. Spline approximation is a piece-wise polynomial approximation of the realizable trajectory with acceleration continuity at merging points. The approximation process is as follows.

Samples \((t^0, r^0)\) are taken from the realizable trajectory at each \( t^i \) where \( i = 0, 1, \ldots \) is the index of piece-wise polynomial. Then, \( i^{th} \) polynomial is defined with its acceleration \( \ddot{r}_i^j(t), \quad t_{j-1}^i < t < t_j^i \) such that

\[
\ddot{r}_i^j(t) = \eta_{j-1}^{i-1} (t^i - t)/h^i + \eta_j^i (t - t^{i-1})/h^i
\tag{22}
\]

where \( \eta_j^i \) is the acceleration at \( t^i \) and \( h^i \) is the interval of \( i^{th} \) piece-wise polynomial. By integrating (22), ve-
locity of \(i^{th}\) piece-wise polynomial could be obtained. The approaching and receding velocities at \(i^{th}\) merging point is given by

\[
\begin{align*}
q'_j(t^-) &= h^j n'_j / 6 + h^j n'_j / 3 + (r^j_j - r^j_{j-1}) / h^j \\
q'_j(t^+) &= -h^j n'_j / 3 - h^j n'_j / 6 + (r^j_j - r^j_{j-1}) / h^j 
\end{align*}
\]

For velocity continuity, \(q'_j(t^-) = q'_j(t^+)\). Then, it follows that

\[
K_1^j n^{i-1}_j + 2n^j_j + K_2^j n^{i+1}_j = K^j_3 
\]

where

\[
\begin{align*}
K_1^j &= 1 - K_2^j \\
K_2^j &= h^{i+1} / (h^i + h^{i+1}) \\
K_3^j &= 6(K^{i+1}_j - K^j_j) / (h^i + h^{i+1}) \\
K_3^j &= (r^j_j - r^j_{j-1}) / h^j 
\end{align*}
\]

Since the boundary conditions are known \(n^0_j = n^n_j = 0\), where, \(n\) is the total number of piece-wise polynomials. Then, spline interpolation could be carried out and the approximated trajectory is constructed by

\[
q^j_j(t) = n^{i-1}_j (t - t^-)^3 / 6 h^i + n^j_j (t - t^-)^2 / 6 h^i + (f^i_j - n^{i-1}_j (h^i)^2 / 6) (t - t^-) / h^i + (f^i_j - n^j_j (h^i)^2 / 6) (t - t^-) / h^i
\]

The differential form of the realizable trajectory and spline approximated trajectory are compared in Fig.7 Spline approximation eliminates huge differential po-

sitions at merging points of the realizable trajectory. Therefore, a reduction in jerk could be expected.

D. Compensation for Delay Dynamics

Spline approximated trajectory is compensated using a forward compensator \[10\] as shown in Fig 8. Compensator transfer function is given by

\[
F_j(s) = \frac{\mu_j^2}{K_j^2 K_j^2} \frac{(s^2 + K_j^2 s + K_j^2 K_j^2)}{(s + \mu_j)^2} \quad (26)
\]

Forward compensator \(F_j(s)\) is a pole-zero cancellation filter designed based on notch-filter theory. It allows system poles to be arbitrarily assigned by the operator. Compensator gain \(\mu_j^2 / (K_j^2 K_j^2)\) is adjusted to maintain zero steady state error. Compensated trajectory is the taught data and it is used in real time servoing of the robot manipulator.

IV. RESULTS AND DISCUSSION

A. Conditions for Simulation and Experiment

Objective trajectory was specified by Cartesian points \((0.35, 0.00, 0.10)[m], (0.41, 0.10, 0.15)[m], (0.28, -0.10, 0.30)[m] and (0.35, 0.00, 0.35)[m].\) Assigned velocity was set by \(v_0 = 0.15[m/s]\). Maximum joint acceleration was set by \(a^{max} = 0.72[rad/s^2]\). Maximum tangential velocities for the two corners were calculated to be 0.03[m/s] and 0.024[m/s]. For the experiment, Performer MK-3s robot manipulator was used and its servo parameters were \(K_j^2 = 23\) and \(K_j^2 = 150\). Compensator pole was set by \(\mu_j = 90\).

For evaluation of the proposed method, a conventional methods was used in that a uniform end-effector velocity is maintained within the entire trajectory. In the conventional method, no special trajectory construction is used and sampled objective trajectory was used as taught data.

B. Results and Evaluation

Using taught data constructed by the proposed method, experiment was carried out with Performer MK-3s industrial robot manipulator. Fig 9 shows the obtained results of end-effector velocity and joint accelerations. In the conventional method, due to stability concertrus, very slow speeds are allowed, as it causes sudden fluctuations in end-effector speed as indicated by A1 in Fig.9(a), which is a result of excessive torque saturation as indicated by B1,B2 and B3 in Fig.9(d),(g), and (j). It further causes poor cornering as indicated by C1 and C2 in Fig.10(b). On the contrary, proposed method has a well tailored velocity profile as shown in Fig.9(b) which is accurately realized in the experiment as shown in Fig.9(c). Joint acceleration profiles as shown in Fig.9(e),(h) and (k) are closely comparable to that of experimenal results shown in Fig.9(f),(i), and (l) and they all obey the limits [-0.72,0.72]. The ultimate performance of the proposed method can be evaluated by examining Fig.10(a) and (c) in that precise operation has been guaranteed. Fig.10 shows the contouring performance in three dimension. It gives the final evaluation of the proposed method in terms of precise conctouring performance. In the conventional method, fol-
Fig. 10. Contouring results in three dimensions. (a) Objective trajectory, (b) Following trajectory with the Conventional method (simulation) and (c) Following trajectory with the proposed method (experiment).

Fig. 9. Results of end-effector velocity and joint accelerations following trajectory is deteriorated at corners as shown in Fig.10(b) whereas the proposed has made the manipulator follow the objective trajectory precisely as shown in Fig.10(c).

V. Conclusions

The proposed method considers acceleration and velocity constraints of industrial robot manipulators. It also considers trajectory allowance. Such consideration is very much appropriate and was consulted from actual roboticized industry. Proposed trajectory construction algorithm has been designed off-line so that it does not demand any hardware changes in the existing systems. Thus, testing and implementation of the proposed method is convenient and risk-free. Precise contouring performance has been obtained bound to the relevant constraint and therefore, proposed method could expect wide industrial implications.

REFERENCES


