Enhanced Contour Control of SCARA Robot Under Torque Constraint

Masatoshi Nakamura, Rohan Munasinghe
and Satoru Goto
Dept. of Advanced Systems Control Engineering
Saga University
Honjomachi, Saga 840-8502
Japan

Nobuhiro Kyura
Dept. of Electrical Engineering
Kinki University (in Kyushu)
11-6 Kawayamori, Iizuka 820-8555
Japan

Abstract

Contour control of a robot arm is an act of the end-effector being moved along a reference Cartesian path, with an assigned velocity. While contouring sharp corners and, at the start and the end of the operation, a substantial torque demand is originated from the reference input generator. It causes power amplifier saturation of the servo controller and hence the saturation of joint torque. Once torque saturation occurs, and last for some time, contour deteriorations are resulted. In addition, system delay dynamics also causes contour deteriorations. In this research, optimum avoidance of torque saturation and delay dynamics compensation are achieved by an off-line trajectory generation algorithm and modified taught data algorithm.

1 Introduction

The main objective of this research is to enhance contour control performance of selective compliance assembly robot arm (SCARA), under torque constraint. Its implementation on SCARA AR-M270 robot arm is also presented. Contour control is a two-fold task. It involves (1) off-line trajectory generation and (2) on-line trajectory tracking. Trajectory generation starts with trajectory segmentation in which, a set of “way points” is declared to describe trajectory segments. Trajectory segments are approximated with either straight lines and/or some functions as proposed by Lin et al. [1]. Finally, the segments are linked in a sequence, which forms the generated trajectory.

There are two major causes of contour deteriorations. (1) torque saturation and (2) system delay dynamics. When torque saturates, desired and actual joint dynamics become significantly different. These differences transform into Cartesian space as contour deteriorations. System delay dynamics causes contour deteriorations, usually in the form of overshoots at trajectory corners. Both the above causes become significant at high tracking velocities [2]. There is no widely practiced discipline in the industry for contour control of robot arms. Instead, contour deteriorations are partially compensated by expert practitioners, by incorporation of various ad-hoc methods.

In this paper, an application oriented trajectory generation methodology is presented for contour control of industrial robot arms. Both Cartesian and joint constraints affiliated to actual industrial applications are addressed. It has been shown from experimental results that this method significantly improves contour control performances of SCARA robot arms. No significant changes are required to implement this methodology as it has been synthesized to an off-line trajectory generation algorithm.

2 System Architecture of an Industrial Robot Arm

2.1 Overview of the Robot Arm System

A schematic of SCARA system is shown in Fig. 1(a). The overall system could be resolved physically into (1) reference input generator (CPU), (2) servo-controller and (3) robot, as shown in Fig. 1(a). Reference input generator generates the sampled Cartesian trajectory and transforms into sampled joint trajectories. Sampled joint trajectories are written on to the servo controller in real time operation. In closed position loop servo controllers, the reference input generator remains off-line and not supposed to involve in
on-line corrective tasks. Instead, time series of sampled trajectory or an equivalent (based on the particular robot arm) is synchronously released to the servo controller. Servo controller reads joint trajectory data from the reference input generator and generates power amplifier currents necessary to actuate joint motors. The maximum current that the power amplifier can handle corresponds to the saturation limit of joint torque. When power amplifiers saturate, robot arm shows non-linear joint dynamics, which causes undesirable contour deteriorations.

2.2 Description of the Robot Arm

As shown in Fig. 1(a), SCARA AR-M270 (Hirata Co. Ltd.) includes four vertical axis joints. In this research, SCARA robot is operated in 2-degree-of-freedom by locking third (prismatic) and fourth joints. In the 2-degree-of-freedom configuration, robot arm parameters are $l_1=0.27[\text{m}]$, $b_2=0.24[\text{m}]$, $m_1=20[\text{kg}]$ and $m_2=15[\text{kg}]$, where $m_i$ and $l_i$ denote the mass and length of $i^{th}$ link. The two selected joints are direct driven, hence friction and backlash can be reasonably neglected. Referring to Fig. 1(b), robot arm kinematics of co-ordinate transformation from Cartesian to joint space is given by

$$x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2)$$

$$y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2).$$

Corresponding inverse kinematics transforms Cartesian co-ordinates to joint co-ordinates as described by

$$\theta_1 = \sin^{-1}\left(\frac{y}{\sqrt{x^2 + y^2}}\right) - \sin^{-1}\left(\frac{l_2 \sin \theta_2}{\sqrt{x^2 + y^2}}\right)$$

$$\theta_2 = \pm \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right).$$

2.3 Robot Arm Dynamic Model

It is assumed that each joint operates independently and coupling between links is neglected. Also neglected are the Coriolis and centripetal torques, which cause significant contributions to contouring performance only at high speeds. It is also assumed that the joint torque and acceleration are proportionally related i.e.,

$$\tau_i = \frac{\tau_1}{\tau_2} \alpha \left(\frac{\theta_1}{\theta_2}\right),$$

where $\tau_i$ represents the torque of the $i^{th}$ joint motor. This relationship has been used extensively in modeling of contouring robot arms [2] and consequently, torque saturation phenomena can be reasonably modeled by proportional acceleration saturation. The mathematical description of joint dynamics is given by

$$\ddot{\theta}_i(t) = \text{sat}[K_{vi}(K_{pi}(U_i(t) - \theta_i(t)) - \dot{\theta}_i(t))]$$

$$\text{sat}(z) = \begin{cases} \dot{\theta}_{i,max} & (z > \dot{\theta}_{i,max}) \\ \dot{\theta}_{i,max} & (-\dot{\theta}_{i,max} \leq z \leq \dot{\theta}_{i,max}), \\ -\dot{\theta}_{i,max} & (z < -\dot{\theta}_{i,max}) \end{cases}$$

$U_i(t)$ represents the position command in joint co-ordinates and $\dot{\theta}_{i,max}$ represents the angular acceleration limit of the $i^{th}$ joint. The servo controller parameters $K_{p1}=17[1/\text{s}], K_{p2}=15[1/\text{s}], K_{v1}=102[1/\text{s}]$ and $K_{v2}=90[1/\text{s}]$ represent SCARA AR-M270 system. When the operation is within the bounds of torque saturation, dynamics of each independent joint is given by

$$\frac{\dot{\theta}_i(s)}{U_i(s)} = G(s) = \frac{K_{pi}K_{vi}}{s^2 + K_{vi}s + K_{pi}K_{vi}}.$$  

3 Enhanced Contour Control Method Under Torque Constraint

3.1 Methodology of Trajectory Generation

3.1.1 Sharpness Elimination at Corners

Sharp corners of the objective trajectory are rounded up as shown in Fig. 2(a). Referring to $\Delta RST$, the trigonometric relationship between the radius of arc $r$ and the introduced error $\epsilon$ is given in [3] and also given here as

$$r = \frac{\epsilon \cos \left(\frac{(\alpha_1 - \alpha_2)/2}{2}\right)}{1 - \cos \left(\frac{(\alpha_1 - \alpha_2)/2}{2}\right)}.$$
The entire trajectory is represented by segments as marked by \( t_i \) where \( i = 0, 1, \ldots, 7 \). According to the well-defined trajectory geometry and tracking velocity profile shown in Fig. 2(a) and (b), the generated trajectory can be mathematically described by \((x(t), y(t))\) such that

\[
x(t) = \begin{cases} 
A_{\text{max}} t^2 \cos \alpha_1 / 2 & (t_0 \leq t < t_1) \\
x(t_1) + v(t - t_1) \cos \alpha_1 & (t_1 \leq t < t_2) \\
x(t_2) + [v(t - t_2) - A_{\text{max}} (t - t_2)^2 / 2] \cos \alpha_1 & (t_2 \leq t < t_3) \\
x(t_3) + r \sin(\alpha_1 + v_c (t - t_3) / r) - \sin \alpha_1 & (t_3 \leq t < t_4) \\
x(t_4) + [v_c (t - t_4) + A_{\text{max}} (t - t_4)^2 / 2] \cos \alpha_2 & (t_4 \leq t < t_5) \\
x(t_5) + vt \cos \alpha_2 & (t_5 \leq t < t_6) \\
x(t_6) + v(t - t_6) - A_{\text{max}} \cos \alpha_1 (t - t_6)^2 / 2 & (t_6 \leq t \leq t_7) 
\end{cases}
\]

and

\[
y(t) = \begin{cases} 
A_{\text{max}} t^2 \sin \alpha_1 / 2 & (t_0 \leq t < t_1) \\
y(t_1) + v(t - t_1) \sin \alpha_1 & (t_1 \leq t < t_2) \\
y(t_2) + [v(t - t_2) - A_{\text{max}} (t - t_2)^2 / 2] \sin \alpha_1 & (t_2 \leq t < t_3) \\
y(t_3) - [\cos \alpha_1 - \cos(\alpha_1 + v_c (t - t_3) / r)] & (t_3 \leq t < t_4) \\
y(t_4) + [v_c (t - t_4) + A_{\text{max}} (t - t_4)^2 / 2] \sin \alpha_2 & (t_4 \leq t < t_5) \\
y(t_5) + vt \sin \alpha_2 & (t_5 \leq t < t_6) \\
y(t_6) + v(t - t_6) - A_{\text{max}} \sin \alpha_2 (t - t_6)^2 / 2 & (t_6 \leq t \leq t_7) 
\end{cases}
\]

Proposed trajectory described by (9) and (10) is calculated with variable \( A \), which is increased gradually until acceleration/torque for any joint approach the condition for saturation, i.e., \( \max(\ddot{\theta}_i(t)) = \ddot{\theta}_{i,\text{max}} v(\theta_1, \theta_2) \), along the trajectory. Then, \( A_{\text{max}} = A \) is determined. The procedure is described by

\[
A_{\text{max}} = \max(A) \quad \text{such that} \quad \left( \frac{\ddot{\theta}_i \leq \ddot{\theta}_{i,\text{max}}}{\ddot{\theta}_2 \leq \ddot{\theta}_{2,\text{max}}} \right), \forall (\theta_1, \theta_2).
\]

3.1.3 Construction of Trajectory Corners for Different Conditions

Construction of corners is application oriented. The specifications for different applications are of different nature. Nevertheless, there are two major categories of applications found in most industries. In first category, the maximum tracking error \( \epsilon_{\text{max}} \) is pre-specified. In second category, maximum tracking velocity denoted by \( v_{c,\text{min}} \) is specified.

Category 1 (given \( \epsilon_{\text{max}} \))

Tracking error at the corner should be less than the given maximum \( \epsilon_{\text{max}} \) i.e.,

\[
\epsilon \leq \epsilon_{\text{max}}.
\]

Then, from (8), radius of the circular arc is constrained according to

\[
r \leq \frac{\epsilon_{\text{max}} \cos\left(\frac{(\alpha_1 - \alpha_2)}{2}\right)}{1 - \cos\left(\frac{(\alpha_1 - \alpha_2)}{2}\right)}.
\]

Centripetal acceleration along the circular arc is \( v_c^2 / r \).

The maximum Cartesian acceleration, \( A_{\text{max}} \), is already known. Therefore, the maximum centripetal acceleration is given by

\[
v_c^2 / r = A_{\text{max}}.
\]

Tangential velocity \( v_c \) is then derived by

\[
v_c = \sqrt{A_{\text{max}} r}.
\]
Category 2 (given $v_{c,\text{min}}$) Given constraint for tracking velocity is
\[ v_c \geq v_{c,\text{min}}. \]  
(16)
From (8) and (16), the radius of the circular arc is constrained according to
\[ r \geq \frac{v_{c,\text{min}}^2}{A_{\text{max}}}. \]  
(17)
From (8) and (17), $\epsilon$ is given by
\[ \epsilon \geq \frac{v_{c,\text{min}}^2}{A_{\text{max}} \cos\left\{ \frac{(\alpha_1 - \alpha_2)/2}{\alpha_1 - \alpha_2}/2 \right\}}. \]  
(18)
Proposed trajectory given by (9) and (10) can be numerically realised with the values determined for $A_{\text{max}}$, $r$, and $v_c$, as described above.

3.2 Compensation for System Delay Dynamics

The trajectory described by (9) and (10) is transformed into joint co-ordinates by inverse kinematics described in (3) and (4). Joint co-ordinate trajectories are then compensated for system delay dynamics, by modified taught data (MTD) algorithm [4]. MTD algorithm is a feed forward compensator, which improves transient performance of joint dynamics. MTD algorithm processes joint co-ordinate trajectories before it is sent to the servo controller. The processing is carried out in such a way that the modified input $P_i(t)$ generates fast joint dynamics. The compensator dynamics is
\[ F_i(s) = -\gamma(s + K_{ps}) \]  
(19)
where $\gamma$ is the pole of the compensator [4]. The pole $\gamma$ can be arbitrarily selected by the practitioner.

4 Implementation of the Proposed Contour Control Method

4.1 Virtual Torque Saturation

Virtual torque saturation (VTS) is proposed, which processes modified servo controller input $P_i(t)$ so that the system output would depict as if it was realised by a servo controller with virtually specified torque limits. VTS enables a robot arm to simulate different torque saturation phenomena, even though the operation is within the actual torque limits. It explains an effective option to simulate torque saturation phenomena without risking the servo hardware. When $P_i(t)$ is input to the virtual robot arm, the output obtained is $\dot{\theta}_i(t)$, bound to virtual torque saturation. The same output is obtained from the actual robot arm for the input $P_i^*(t)$.

The task of VTS algorithm is to transform $P_i(t)$ into $P_i^*(t)$ so that the saturation characteristics can be physically realised on the actual robot arm, without actually being saturated. Mathematical expression for VTS can be derived from (5) and (6), subjected to saturation and non-saturation conditions and is given by
\[ P_i^*(t) = \begin{cases} 
P_i(t) & \text{if } |\dot{\theta}_i(t)| < \dot{\theta}_{i,\text{max}} \\
(\text{sgn} \dot{\theta}_i(t))\dot{\theta}_{i,\text{max}} - \dot{\theta}_i(t)/K_{ps}K_{vi} & \text{otherwise}
\end{cases} \]  
(20)
where
\[ \text{sgn} \dot{\theta}_i(k) = \begin{cases} 
1 & \dot{\theta}_i > 0 \\
-1 & \dot{\theta}_i < 0
\end{cases} \]  
(21)

4.2 Generated Trajectory for Contour Control Experiment

The equivalent pulse encoder setting of the acceleration limits of the two joints is $140,000 [/\text{Pulse/s}^2]$. It refers to joint acceleration limits $\ddot{\theta}_{1,\text{max}}=1.432[/\text{rad/s}^2]$ and $\ddot{\theta}_{2,\text{max}}=1.732[/\text{rad/s}^2]$, respectively. Right angular corner with vertices (0.2,0.2), (0.25,0.2) and (0.25,0.25) was selected as the objective trajectory. Maximum error $e_{\text{max}}$ was set to 0.5[m]. According to category 1, (15) yielded $A_{\text{max}}=0.035[\text{m/s}^2]$. Bound to (17), $r=1.2[\text{mm}]$ was determined. Then, (19) yielded $v_c=0.05[\text{m/s}]$. Corresponding trajectory segmentation was realised with $t_0 = 0.000[\text{s}]$, $t_1 = 0.100[\text{s}]$, $t_2 = 0.988[\text{s}]$, $t_3 = 1.038[\text{s}]$, $t_4 = 1.110[\text{s}]$, $t_5 = 1.160[\text{s}]$, $t_6 = 2.048[\text{s}]$ and $t_7 = 2.148[\text{s}]$.

4.3 System Setup of the Experiment

The schematic of the contour control experimental set-up is illustrated in Fig. 3. Objective Cartesian trajectory is subjected to trajectory generation according to (9) and (10), which realises sharpness elimination at corners and tracking velocity control. Inverse kinematics transforms Cartesian trajectory into joint trajectories as described by (3) and (4). Joint trajectories are subjected to delay dynamics compensation according to (19). Compensated joint trajectories, further processed by VTS algorithm, as described by (20) and (21) will be the input to the servo controller, for real time servoing. Servo controller, motor and mechanism
operate in joint co-ordinates and are modeled by (5) and (6). The following joint trajectories are transformed into Cartesian co-ordinates by kinematics, as described by (1) and (2).

5 Results and Discussion

5.1 Contouring Performances

Fig. 4(a) and (b) illustrate contouring results of conventional and proposed methods. In the conventional method, reference input generator carries out only sampling of the objective trajectory and the rated tracking velocity is considered constant over the entire operation. Simulation and experimental results are comparable for both methods. Under conventional control, a significant overshoot occurs as indicated by →A in Fig. 4(a). This overshoot can be approximated to 2[mm]. In addition, significant contouring fluctuations are observed near the corner. In contrary, under proposed control, overshooting is fully eliminated and the existing contour deteriorations are minute. Experimental data of joint accelerations are not accessible at the servo controller and can not be constructed from the available position data either, because of the noise contamination. Instead, simulation results of joint accelerations are illustrated in Fig. 5(a),(b),(c) and (d). Under conventional control, joint 1 shows torque saturation indicated by →B in Fig. 5(a), which lasts approximately 0.2[s] whereas, joint 2 saturates for 0.1[s] as indicated by →C in Fig. 5(b). In contrary, proposed control causes no torque saturation, except momentary demand of maximum torque at the trajectory corner, as indicated by →D in Fig. 5(c).

5.2 Analysis of System Robustness

Behavior of contouring error $E_c$ was obtained by simulation study of contour performance, in response to a change of servo parameter settings. Delay compensator was set to nominal servo parameter settings whereas servo controller settings of the two joints were changed in 5% steps from $-70\%$ to $130\%$ that of the nominal settings. Corresponding contouring performances were obtained as shown in Fig. 6(a). Each solid line illustrates the following trajectory obtained for a particular offset of servo parameters $K_{pi}$, in steps of 5%. The offset of servo parameters is denoted by $\Delta K_p$ and is defined by

$$\Delta K_p = \frac{K_{pi} - K_{pi}^*}{K_{pi}}$$  \hspace{1cm} (22)
where $K_{pi}$ and $K_{pi}^*$ refer to the position loop gains of the modification term and servo controller, respectively. When $K_{pi} < K_{pi}^*$, following trajectory shifts inward as can be seen in Fig. 6(a). Therefore, under this condition, contouring performance can be evaluated by the amount of inward shift from the particular trajectory obtained with the nominal settings. On the other hand, when $K_{pi} > K_{pi}^*$, following trajectory shifts outward, resulting an overshoot. Under this condition, contouring performance is evaluated by the amount of overshoot.

Fig. 6(b) illustrates corresponding contouring error variation that was extracted from Fig. 6(a), for $\Delta K_p < 0$. It also shows variation of overshoot for $\Delta K_p > 0$. Contouring error increases with the decrease of servo parameters. Overshoot increases with the increase of servo parameters.

5.3 Discussion

Most trajectory generation algorithms found in scientific literature are based on computed torque methods developed on Lagrange/Euler or Newton/Euler dynamics and D-H representation of the kinematic chain [1] [5]. Though highly accurate, these models are extremely complex. They consume long computation time and high cost for signal processing electronics. These algorithms require many parameters of the robot arm too, though not all of them could be determined reliably in case of an industrial robot arm. Therefore, those algorithms could hardly be used in the industry.

The proposed method has been developed on the linear decoupled servo dynamics given in (7), which is extensively used in today’s robotic industry. The method proposed is simple and provides sufficiently high performance. To implement, it requires only two servo parameters to be known, which could be determined optimally as described in [2].

6 Conclusions

Contour control performances of SCARA type industrial robot arms have been significantly enhanced by the proposed method, under the constraint of torque saturation. Optimum avoidance of torque saturation and compensation for delay dynamics have been successfully incorporated and synthesised into a single off-line trajectory generation algorithm. Hence, the proposed method can be conveniently implemented on existing SCARA type robot arms for contour control applications. The necessary hardware changes are minute and possible. It is further emphasized that the proposed method could be easily customized and it is sufficiently robust for most industrial application to date. The proposed method has its merits based on its simplicity and sufficiently high performance valid for most industrial applications, where most of the other methods fail to provide with.

References


