Abstract: Robots are highly nonlinear and chaotic in position control. The present paper mainly presents the position control of PUMA-560 Robot manipulator. Computed torque controller (CTC) is one of the solutions for position control of robot manipulators. The main drawback of controller is that it fails to operate under different dynamic operating conditions. To overcome this difficulty, intelligent controllers have gained importance. In this paper a novel approach, design of prisoner’s dilemma-based fuzzy C-means controller to control the position of robot manipulator is presented. This controller is employed at the inputs of computed torques for obtaining the desired position. Fuzzy C-means controller with computed torques is realised by validating the clusters to choose most contributed rules. Thus the unfired rules are eliminated from the actual rule-base. Hence, a compact fuzzy controller with minimum rule-base, fuzzy C-means computed torque controller (FCMCTC), is designed. The concept of prisoner’s dilemma is introduced in this paper to improve the fuzzy strategy. The interrelations between inputs and outputs of a fuzzy linguistic model are assigned using payoff matrix through prisoner’s dilemma. Simulation results prove the efficacy of proposed controller when compared to proportional derivative computed torque controller (PD-CTC), normal FLC and that of the reference signal.

Keywords: PUMA-560 robot manipulator; fuzzy C-means; fuzzy clustering technique; phase-plane plot; prisoner’s dilemma; clustering; phase-plane plot.


Biographical notes: Ch Ravi Kumar completed his MTech degree from Jawaharlal Nehru Technological University, and is currently pursuing his PhD degree from Andhra University, Visakhapatnam.
1 Introduction

The term robot has been applied to wide variety of mechanical devices. PUMA-560 (Piltan and Yarmahmoudi, 2012) is one of the important classes of robot manipulators. It is a six degree of freedom robot manipulator. The end-effector or of the robot arm can reach a point within its workspace from any direction. The robot can determine its final position from the given feedback information. These robot manipulators are highly nonlinear, time variant and multiple input multiple output (MIMO) in nature. One of the most important challenges in the field of robotics is robot manipulator control with acceptable performance. For this purpose many controllers are suggested in literature such as LQG, $H_\alpha$, input shaping as well as singular perturbation, feedback linearisation, manifolds and output redefinition techniques (Bernstein and Haddad, 1989; Tzafestas and Papanikolopoulos, 1990). Computed torque controller (CTC) (Piltan and Yarmahmoudi, 2012) is one of the model-based nonlinear controller, widely used in the control of robot manipulators. CTC is a significant nonlinear controller to certain systems which it is based on feedback linearisation and computes the required arm torques using the nonlinear feedback control law. When all dynamic and physical parameters are known, CTC works satisfactorily; practically most of the nonlinear systems have a large amount of uncertainties and have variation in dynamic parameters. Hence this controller has no acceptable performance. Because of this, use of the intelligent controllers has gained importance (Moudgal et al., 1994).

In this paper a new prisoner’s dilemma-based fuzzy C-means (FCMs) controller (Ravi Kumar et al., 2013) is proposed and employed at the inputs of computed torques for minimising the error present in the joint angles. A key challenge in the analysis of fuzzy controller is finding out the exact rule-base (Tanaka and Wang, 2001). Obviously, it is difficult for human experts to examine all the input/output data from a complex system to find the proper fuzzy rule-base (Taniguchi et al., 2000; Palm, 1992) for the system. Such an approach requires a large number of repetitions and is therefore tedious and time consuming. This will increase the number of fuzzy rules, complexity and also impact the quality of the control. FCMs is a method of clustering of one piece of data belongs to two or more clusters and this minimises the rule-base and also further improves the performance of the system as normal fuzzy controller do. Further it is necessary to systematically tune the parameters used in FCM controller to get the desired and optimum response under all dynamic variations and uncertainties such as variation in inertia and gravitational constants. These parameters are commonly determined by trial and error method, which is rather time consuming and does not guarantee an optimal control. The present work concentrates on proper tuning of FCM controller parameters using prisoner’s dilemma. The fuzzy sets are introduced to prisoner’s dilemma model by
generalising the payoffs to invoke fuzzy goal sets and to deem the strategies as fuzzy sets. The paper proposes an effort to assess the optimal gains of FCMs controller by updating payoff matrix of players as inputs. Simulation results prove the effective performance of the proposed controller in minimising the error in joint angles when compared to proportional derivative computed torque controller (PD-CTC) (Subudhi et al., 2010), normal fuzzy logic controller (FLC) and that of the reference signal. The rest of the paper is organised as follows: the nonlinear model of PUMA-560 manipulator is explained in Section 2, FCMs technique is explained in Section 3, prisoner’s dilemma-based optimal FCMCTC is explained in Section 4, evolutionary prisoner’s dilemma is presented in Section 5, results and discussions are presented in Section 6.

2 Nonlinear model of PUMA-560 manipulator

The nonlinear model of PUMA-560 robot manipulator (Armstrong et al., 1986) is derived by using Lagrange-Euler formulation based on the Newton’s second law and d’Alembert principle, considering three links among the total six links, such that $q_4 = q_5 = q_6 = 0$. The joint torques required to cause time-dependent motions to realise a trajectory is computed using the recursive Newton-Euler dynamic equations of motion (Piltan and Yarmahmoudi, 2012).

$$\tau = A(q)\ddot{q} + B(q)[\dot{q}\dot{q}] + C(q)[q^2] + G(q)$$

where

$q$ $n \times 1$ position vector
$A(q) \ n \times n$ inertia matrix of the manipulator
$G(q) \ n \times 1$ vector of gravity terms
$\tau \ n \times 1$ vector of torques
$B(q) \ n \times (n-1)/2$ matrix of Coriolis torques
$C(q) \ n \times n$ matrix of centrifugal torques
$\ddot{q}$ $n$ vector of acceleration
$[\dot{q}\dot{q}]$ and $[q^2]$ notation for $n(n-1)/2$ vector of velocity products and the $n$-vector of squared velocities respectively.

The desired trajectory of the manipulator is obtained by changing the proportional derivative gains, which changes the torques computed accordingly. This forms PD-CTC.
CTC is a powerful nonlinear controller which it widely used in control of robot manipulators. It is based on feedback linearisation and computes the required arm torques using the nonlinear feedback control law (Piltan and Yarmahmoudi, 2012). This controller works very well when all dynamic and physical parameters are known but when the robot manipulator has variation in dynamic parameters, this controller has no acceptable performance. In practice, most of physical systems (e.g., robot manipulators) parameters are unknown or time variant, therefore, computed torque like controller used to compensate dynamic equation of robot manipulator. However both controllers have been used in feedback linearisation, but predictive strategy gives better result as a performance. A computed torque control with non-parametric regression models have been presented for a robot arm. It has assumed that the desired motion trajectory for the manipulator, as determined, by a path planner. Defines the tracking error as:

\[ e(t) = q_d(t) - q_a(t) \]

where \( e(t) \) error of the plant is, \( q_d(t) \) is desired input variable, that in our system is desired displacement, \( q_a(t) \) is actual displacement.

### 3 Fuzzy C-means

For the complex systems such as PUMA-560 modelling, where traditional methods are difficult to apply, fuzzy logic (Yousef and Al-Abri, 2014; Piltan et al., 2011) became an active research area. The fuzzy logic model is empirically-based, relying on operator’s
experience rather than their technical knowledge of the system. For complex systems where little and ambiguous information may be available, fuzzy reasoning provides a way to understand the system behaviour by allowing us to interpolate approximately between observed input and output situation (Rao and Sudha, 2004; Johnson et al., 2005).

Fuzzy systems implement the crisp input and output (Ying, 2000) and produce a nonlinear functional mapping. Depending on the system, it may not be necessary to evaluate every possible input combination since some may rarely or never occur. Also it is difficult for human experts to examine all the input/output data from a complex system to find the proper fuzzy rule-base for the system. Such an approach requires a large number of repetitions and is therefore tedious and time consuming. This will increase the number of fuzzy rules, complexity and also impact the quality of the control. To cope up with this difficulty and to identify the optimal fuzzy rule-base the fuzzy clustering technique is adopted in this paper. FCMs is a method of clustering which allows one piece of data belongs to two or more clusters.

3.1 Design procedure of FCM

Step 1 The normal fuzzy controller is designed heuristically with fuzzy linguistic rules.

Step 2 The FCMs controller is tuned to the normal fuzzy controller.

Step 3 The phase-plane plot of the input space is obtained.

Step 4 The input space is divided into clusters using FCMs and the cluster centers are identified.

Step 5 The sequence of rules of the original fuzzy controller is super imposed onto the phase-plane plot of the input space with cluster centres.

Step 6 Hence the required rules are identified and the non-cooperative rules are eliminated.

3.1.2 Clustering

Clustering (Chiu, 1994; Luciano et al., 1998) is the unsupervised classification of patterns into groups. The main aim of clustering is to divide data set in a way that the data belonging to one cluster is as similar as possible. The idea is to identify the number of sub classes of clusters in a universe. One of the easiest methods is the distance between the observed data and if one is able to measure the distance between all the observed data then it can be expected that the distance between the points in the same cluster will be less when compared to the distance between the points in different clusters. Based on the requirements the user can decide the number of clusters needed that can best suited to the given purpose. It is done based on minimisation of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \| x_i - c_j \|^2$$

where $m$ is any real number greater than 1, $u_{ij}$ is the degree of membership of $x_i$ in the cluster $\mu$, $x_i$ is the $i^{th}$ of d-dimensional measured data, $c_j$ is the d-dimension center of the cluster and $\| * \|$ is any norm expressing the similarity between any measured data and the center.
Fuzzy partitioning is carried out through an iterative optimisation of the objective function shown above, with the update of membership $u_{ij}$ and the cluster centers $c_j$ by:

$$u_{ij} = \frac{1}{\sum_{j=1}^{c} \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{1}{m-1}}} \quad \text{where} \quad \frac{\sum_{j=1}^{N} u_{ij}^m x_i}{\sum_{j=1}^{N} u_{ij}^m}$$

This iteration will stop when $\{|u_{ij}^{(k+1)} - u_{ij}^k|\} < \varepsilon$, where $\varepsilon$ is a termination criterion between 0 and 1, whereas $k$ is the iteration step. This procedure converges to a local minimum or a saddle point of $J_m$. In a batch mode operation, FCM determines the cluster centers $c_j$ and the membership matrix $U$ using the following steps:

**Step 1** Set the number of clusters $c$. Initialise the membership matrix $U$ with random values between 0 and 1 such that the summation of degrees of belongingness of a data point to all clusters is always equal to unity.

**Step 2** Calculate $c$ fuzzy cluster centers, $1$, $c_i$ where $i = 1, 2, \ldots, c$, using equation (3).

**Step 3** Compute the objective function according to equation (2). Stop if either it is below a certain tolerance value or its improvement over previous iteration is below a certain threshold.

**Step 4** Compute a new $U$ using equation (3). Go to Step 2.

### 3.1.3 Phase-plane analysis

The phase-plane plots (Sudha et al., 2012) of the inputs then given to the fuzzy controller are used to achieve the optimal rule-base. This requires characterisation of the relation between the rules and state space associated with the dynamic system under control, this relation is based on the relative influence of each rule of the rule base on the control action produced by fuzzy inference engine. A closed loop trajectory can be mapped on the position space, the sequence of rules obtained according to the order in which they are fired forms the solution called linguistic trajectory. This corresponds to a certain system trajectory and provides guidelines to obtain the required rule-base from the phase-plane plots of the inputs given to the fuzzy controller. The closed-loop trajectory is mapped on position space of the inputs. The clusters are formed in entire position space of the inputs using FCMs. The cluster centers are identified and marked on the phase-plane plot. These cluster centers are mapped with the closed-loop trajectory and the required rules are identified, non-cooperative or unfired rules are thus eliminated.

### 3.2 Design of proposed fuzzy controller using FCM algorithm

The normal fuzzy controller is designed heuristically with all possible rules. The two normalised input variables $e, \dot{e}$ are fuzzified first and then the fuzzy sets are derived as follows:

$$\mu_{\epsilon}(\cdot) = \begin{cases} \lfloor x(.) + 1 \rfloor & \text{if } x(.) \in [-1, 0] \\ -\lfloor x(.) - 1 \rfloor & \text{if } x(.) \in [0, 1] \end{cases}$$

(4)
The FCMs controller is tuned to the normal fuzzy controller (Chiu, 1997; Jain and Dubes, 1998). The phase-plane plot of the input space is obtained. The input space is divided into clusters and the cluster centers are identified. The sequences of the original fuzzy controller rules are super imposed onto the phase-plane plot of the input space with cluster centers as shown in Figure 2. Hence, the required rules are identified and the non-cooperative rules (Cheng et al., 1995; Kusiak and Chow, 1987) are eliminated as shown in Table 1.

**Figure 2** Phase-plane plot of the input space with cluster centers (see online version for colours)

**Table 1** Rules affected at joint 3

<table>
<thead>
<tr>
<th>e</th>
<th>PB</th>
<th>NS</th>
<th>NB</th>
<th>NS</th>
<th>NB</th>
<th>PS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\dot{e})</td>
<td>PS</td>
<td>PB</td>
<td>PS</td>
<td>NS</td>
<td>NB</td>
<td>NB</td>
</tr>
<tr>
<td>(U)</td>
<td>NB</td>
<td>NS</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>NM</td>
</tr>
</tbody>
</table>

### 4 Prisoner’s dilemma-based optimal FCMCTC

The main objective of the controller is to sense information from robot manipulator and to improve the systems performance by achieving a small tracking error. CTC is used to compensate dynamic equation of robot manipulator tracking response in uncertain environment. The basic structure of FLC is as shown in Figure 2.

#### 4.1 Fuzzy sets associated with FCMCTC design

In the present formulation, the structure of the proposed FCMCTC model is as shown in the Figure 3. In proposed design; two variables \(e, \dot{e}\) are used as input signals. The coefficients \(K_p, K_d\) which are called scaling factors, transform the scaled real values to required value in decision limit. The output signal coefficient \(K_u\) is injected to the summing point. The normalised inputs of the proposed controller are \(A_1\) and \(A_2\) which are equal to \(K_pe\), \(K_d\dot{e}\) respectively. The scaling factors \(K_p, K_d\) are tuned using the prisoner’s
dilemma. The two similar fuzzy sets defining the two inputs of the proposed FCMCTC is given by:

\[ K_{pe} = K_{pe} = \{NB(Negative\ Big),\ NM(Negative\ Medium),\ NS(Negative\ Small),\ Z(Zero),\ PS(Positive\ Small),\ PM(Positive\ Medium),\ PB(Positive\ Big)\} \]

Figure 3  Structure of FCMCTC

The triangular membership functions are considered and partitioned within the universe of discourse in the range \([-6, +6]\) for the inputs and outputs. For instance, the mathematical model of membership function is given as follows:

\[
\begin{align*}
\mu_{ZF} &= \frac{7}{20}x + 1, \quad \text{if } 0 < x < 2.85 \\
\mu_{ZF} &= \frac{7}{20}x + 1, \quad \text{if } -2.85 < x < 0 \\
\mu_{ZF} &= 0 
\end{align*}
\]

(7)

Otherwise

\[
\begin{align*}
\mu_{ZF} &= -(7/20)x + 1, \quad \text{if } 0 < x < 2.85 \\
\mu_{PS} &= (7/20)x + 1, \quad \text{if } -2.85 < x < 0 \\
\mu_{PS} &= 0 
\end{align*}
\]

(8)

4.2 IF-then rules

The decisions in fuzzy logic-based approach are made by forming series of rules which relate the inputs to outputs by IF-THEN statements. In the proposed case the number of control rules to cover all the possible combinations of the seven membership functions of each input variable is \(7 \times 7\) (i.e., 49).

4.3 Defuzzification

In this paper centroid defuzzification method is adopted to calculate the output. The equation represents the final output \(U\) by computing the centroid of the area of the possibility distribution.
Membership value of the input output corresponding to the membership of the input
\[ U = \frac{\sum_{i,j=1}^{n} \{ Membership \ value \ of \ the \ input \times output \ corresponding \ to \ the \ membership \ of \ the \ input \}}{\sum_{i,j=1}^{n} \{ membership \ value \ of \ input \}} \]

5 Evolutionary prisoner’s dilemma

The prisoner’s dilemma (Borges and Pacheco, 1997) is one of the classical games. The interactions of the players in the prisoner’s dilemma are generally described by a 2 × 2 payoff matrix (Gao et al., 2009) of player A as in equation (9),

\[
P_A = \begin{pmatrix} C & D \\ R & S \end{pmatrix}
\]

\[
P_A = \begin{pmatrix} C & R \\ D & T \end{pmatrix}
\]

C and D, cooperation and defection are two strategies that can be selected by each player in each round and \( T > R > P > S \). For example, two (A1 and A2) players are chosen to play, the entries of the payoff matrix are interpreted as follows (Ishibuchi et al., 2009; Számadó et al., 2008):

- if A1 and A2 choose cooperation, then both gets reward \( R \)
- if A1 and A2 choose defection, then both deserves punishment \( P \)
- if A1 or A2 defects and A2 or A1 cooperates, then A1 or A2 (defector) obtains payoff \( T \) (temptation); while A2 or A1 (sucker) the sucker gets payoff \( S \).

Therefore with respect to the above, every player should tend to defect because it will get more total payoff irrespective of the strategy of its opponent \((T > R \ and \ P > S)\). Hence, one only gets the penalty \( P \). Distinctly, when they both choose cooperation \((2R > T + S)\), they will get higher total payoffs in the long run. This is the dilemma. Based on the literature, a rescaled form about this matrix where \( T > 1 \), \( R = 1 \), \( P = S = 0 \) is generally adopted. The strategies of prisoner’s dilemma can be updated using MAX payoff strategy updating technique (Perc and Szolnoki, 2008; Szabó and Szolnoki, 2009), where, a player in each site plays against its neighbours including itself. Each individual implement the strategy of the player who gains the highest total payoff. Hence, the total payoff \( P_i \) of \( i^{th} \) player is calculated as follows:

\[
P_i = \sum_{j=1}^{\Psi_i} X_i^T A x_j
\]

where \( \Psi_i \) is the neighbourhood of \( i^{th} \) player including itself, \( A \) denotes the payoff matrix equation (10), \((\cdot)^T\) denotes the transpose. And \( X_i, x_j \) satisfy the following requirements; if \( i^{th} \) player chooses defection \( x_i = (-1, 1)^T \); if \( i^{th} \) chooses cooperation \( x_i = (1, -1)^T \). Sometimes a player does not know the exact payoff of its opponents and even its own, in this case it is difficult to decide its updating strategy. Fuzzy logic is a good approach to deal such approximate uncertainty. The present work adopts a fuzzy linguistic rule...
Design of prisoner’s dilemma-based FCMCTC

model-based strategy updating scheme. And we obtain a series of reasonable simulation results 49-fuzzy-rule-base. This paper presents an approach to derive the optimal parameters of PD FCMCTC. The problem is defined as a tournament between two player’s payoffs as fuzzy inputs, the winning possibility as fuzzy outputs (Perc and Szolnoki, 2008; Szabó and Szolnoki, 2009). Through fuzzy reasoning, it is possible to get the possibility of one player forcing its strategy on its opponent in the strategy update (Chong and Yao, 2006).

5.1 Updating payoff to obtain optimal parameters of PD FCMCTC

Figure 4 shows the flow chart to update the payoff matrix of PD FCMCTC; the updated strategy (Andreoni and Miller, 1993; Gedeon et al., 2002) is introduced as inputs to the fuzzy controller. The payoff matrix is derived by optimising the equation (9) to reduce the error. \( K_p \) and \( K_d \) are the controller gains that are derived (Yamauchi et al., 2011; Deng et al., 2011) by updating payoff matrix to acquire the desired response. These gains are introduced as scaling factor of fuzzy logic CTC. Hence a new prisoner’s dilemma-based PD FCMCTC is derived and proposed.

**Figure 4** Flow chart to update payoff matrix
6 Results and discussions

The proposed prisoner’s dilemma-based FCMs controller applied to PUMA-560 robot manipulator has been tested for both step and ramp inputs and compared with PD-CTC, FLCTC and reference signal. The six-degree-of-freedom robotic arm was simulated to evaluate the performance of the controller. The difference between the desired location and current location is an input vector to the controller that generates joint rate commands. The performance of the proposed controller is tested by incorporating at each joint of PUMA-560 robot manipulator without and with uncertainties such as inertia and gravitational constants. The results presented in this paper prove the effective performance of the proposed controller. The development of the control algorithm, simulation testing, results and the performance of the controller are reported. From Figures 5–8 it can be observed that the error in the theta value for the link3 joint is minimised with FCMs controller when both ramp and step inputs are given. The numerical data analysis of link 3 is shown in Tables 2–5. The similar analysis was done for the remaining three links. The minimised fuzzy rule-base for link3 is shown in Table 1. This work concludes that the FCMs-based controller outperformed the other controllers. The peak time, delay time, rise time, settling time and the peak overshoot are reduced considerably.

Figure 5  Error in theta3 of link3 for ramp input without uncertainties (see online version for colours)

<table>
<thead>
<tr>
<th>Control technique</th>
<th>$T_p$ (sec)</th>
<th>$T_d$ (sec)</th>
<th>$T_r$ (sec)</th>
<th>$M_p$</th>
<th>$e_{pm}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD-CTC</td>
<td>3.4</td>
<td>1.75</td>
<td>8.95</td>
<td>2.982</td>
<td>0.241</td>
</tr>
<tr>
<td>Fuzzy</td>
<td>3.23</td>
<td>1.58</td>
<td>3.5</td>
<td>2.935</td>
<td>0.225</td>
</tr>
<tr>
<td>Fuzzy C-means</td>
<td>3.18</td>
<td>1.45</td>
<td>3.22</td>
<td>2.814</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 6  Error in \( \theta_3 \) of link3 for ramp input with uncertainties (see online version for colours)

![Error in \( \theta_3 \) of link3 for ramp input with uncertainties](image)

Figure 7  Error in \( \theta_3 \) of link3 for step input without uncertainties (see online version for colours)

![Error in \( \theta_3 \) of link3 for step input without uncertainties](image)

Table 3  Error in \( \theta_3 \) of link3 for ramp input with uncertainties

<table>
<thead>
<tr>
<th>Control technique</th>
<th>( T_p ) (sec)</th>
<th>( T_d ) (sec)</th>
<th>( T_i ) (sec)</th>
<th>( T_r ) (sec)</th>
<th>( M_p )</th>
<th>( e_{ul} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD-CTC</td>
<td>3.48</td>
<td>1.78</td>
<td>6.7</td>
<td>3.2</td>
<td>0.22</td>
<td>0.21</td>
</tr>
<tr>
<td>Fuzzy</td>
<td>3.26</td>
<td>1.57</td>
<td>3.5</td>
<td>3.21</td>
<td>0.225</td>
<td>0.04</td>
</tr>
<tr>
<td>Fuzzy C-means</td>
<td>3.21</td>
<td>1.45</td>
<td>3.22</td>
<td>2.89</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 8  Error in theta3 of link3 for step input with uncertainties (see online version for colours)

Table 4  Error in theta3 of link3 for step input without uncertainties

<table>
<thead>
<tr>
<th>Control technique</th>
<th>$T_p$ (sec)</th>
<th>$T_d$ (sec)</th>
<th>$T_s$ (sec)</th>
<th>$T_r$ (sec)</th>
<th>$M_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD-CTC</td>
<td>3.74</td>
<td>1.68</td>
<td>4.53</td>
<td>2.82</td>
<td>-0.157</td>
</tr>
<tr>
<td>Fuzzy</td>
<td>3.17</td>
<td>2.08</td>
<td>3.21</td>
<td>2.94</td>
<td>-0.043</td>
</tr>
<tr>
<td>Fuzzy C-means</td>
<td>2.11</td>
<td>1.53</td>
<td>2.16</td>
<td>1.94</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5  Error in theta3 of link3 for step input with uncertainties

<table>
<thead>
<tr>
<th>Control technique</th>
<th>$T_p$ (sec)</th>
<th>$T_d$ (sec)</th>
<th>$T_s$ (sec)</th>
<th>$T_r$ (sec)</th>
<th>$M_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD-CTC</td>
<td>3.74</td>
<td>1.68</td>
<td>4.53</td>
<td>2.82</td>
<td>-0.157</td>
</tr>
<tr>
<td>Fuzzy</td>
<td>3.18</td>
<td>2.06</td>
<td>3.21</td>
<td>2.94</td>
<td>-0.043</td>
</tr>
<tr>
<td>Fuzzy C-means</td>
<td>2.13</td>
<td>1.54</td>
<td>2.16</td>
<td>1.94</td>
<td>0</td>
</tr>
</tbody>
</table>

7 Conclusions

Here in this article a novel approach, designing of fuzzy C-means computed torque controller (FCMCTC) based on prisoner’s dilemma is presented. It is a fuzzy rule-based approach for robot motion control to eliminate the computational complexity associated with the conventional mathematical algorithm. The errors in the joint angles of PUMA-560 manipulator are minimised considerably. FCMs-based FLCs are placed at the inputs of the proportional derivative controllers and the gains are made adaptive. In the present paper fuzzy CTC with minimum rules is obtained by validating the clusters to choose most contributed rules. Prisoner’s dilemma is employed to systematically tune the gains of the controller. The interrelations between inputs and outputs of a fuzzy linguistic model are assigned using payoff matrix through prisoner’s dilemma. In this paper FCMs concept is used to systematically derive the rule-base. For rule minimisation, the fuzzy
Design of prisoner’s dilemma-based FCMCTC

clustering technique in addition with the phase-plane plot of the inputs of the fuzzy controller is utilised and finally required rules are identified, the non-cooperative or unfired rules are thus eliminated. From the simulation results, Figure 5, it is observed that the response of prisoner’s dilemma-based FCM controller settled to its final value at 3.22 seconds, whereas the responses of fuzzy controller and PD-CTC settled at 3.5 and 8.95 seconds respectively. From Figures 5 to 8, it is observed that the responses of prisoner’s dilemma-based FCM controller outperformed the responses of other controllers. The corresponding numerical analysis, i.e., peak time, rise time, delay time, settling time, peak overshoot and steady state error is tabulated in Tables 2 to 5. The numerical analysis prove the effectiveness of the proposed controller in minimising the error in joint angles when compared to PD-CTC, normal FLC and that of the reference signal.

References


