Robotica (2015) volume 33, pp. 2045–2064. © Cambridge University Press 2014
doi:10.1017/S0263574714001258

A novel self-adaptive modified bat fuzzy sliding mode control of robot manipulator in presence of uncertainties in task space

Mohammad Veysi†, Mohammad Reza Soltanpour† and Mohammad Hassan Khooban‡*

†*Department of Electrical Engineering, Shahid Sattari Aeronautical University of Science and Technology, Tehran, Iran*
‡*Young Researchers Club, Garmsar Branch, Islamic Azad University, Garmsar, Iran*

(Accepted April 24, 2014. First published online: May 22, 2014)

SUMMARY
In this paper, an optimal fuzzy sliding mode controller has been designed for controlling the end-effector position in the task space. In the proposed control, feedback linearization method, sliding mode control, first-order fuzzy TSK system and optimization algorithm are utilized. In the proposed controller, a novel heuristic algorithm namely self-adaptive modified bat algorithm (SAMBA) is employed. To achieve an optimal performance, the parameters of the proposed controller as well as the input membership functions are optimized by SAMBA simultaneously. In this method, the bounds of structural and non-structural uncertainties are reduced by using feedback linearization method, and to overcome the remaining uncertainties, sliding mode control is employed. Mathematical proof demonstrates that the closed loop system with the proposed control has global asymptotic stability. The presence of sliding mode control gives rise to the adverse phenomenon of chattering in the end-effector position tracking in the task space. Subsequently, to prevent the occurrence of chattering in control input, a first-order TSK fuzzy approximator is utilized. Finally, to determine the fuzzy sliding mode controller coefficients, the optimization algorithm of Self-Adaptive Modified Bat is employed. To investigate the performance of the proposed control, a two-degree-of-freedom manipulator is used as a case study. The simulation results indicate the favorable performance of the proposed method.

KEYWORDS: Robot manipulator; Task space; Chattering; Optimal fuzzy sliding mode control; TSK method; SAMB algorithm; Uncertainty.

1. Introduction

In recent years, automatic industrial robots have found widespread use in industrial processes. Industrial robot manipulators have completely non-linear dynamic equations in the form of multi-input, multi-output as well as structured and unstructured uncertainties. Finding a precise model in such systems is not easy. Hence, designing a controller with a favorable performance which is based on the system model is difficult. Robot manipulator position control is usually established by the method of control in the joint space; however, due to the effects of uncertainties, the favorable control in the joint space does not result in favorable control in the task space.

In joint space control, feedbacks from the joint space are given to control the system while tracking a desired trajectory. This control system does not detect the position error of the end-effector in the work space. Even if a precise tracking of joint positions is achieved, a desired tracking in the task space is not provided by the use of an imperfect transformation of the control space. Thus, due to detecting tracking error of the end-effector, the task space tracking control of a normal-cost robot is

* Corresponding author. E-mail: dr.mohammadhassan.khooban@ieee.org

superior to that of the joint space control. It means that we can expend less cost to achieve a desired performance by a task space control of a normal-cost robot in replacement of the joint-space control of an expensive robot. However, obtaining feedbacks from the task space is not as convenient as the joint-space. The joint positions are measured suitably by optical encoders while the end-effector position may be detected using vision systems.[1,2]

There is a challenge in robot control to overcome uncertainties, nonlinearities and couplings from different aspects in the field of robust control as surveyed in ref. [3]–[6]. The robust control provides stability under uncertainties with a tradeoff between tracking performance and bounds of uncertainties. This control approach was extensively presented in the joint space while controlling a robot in the task space is still a control problem. Recently, several regulating controllers were proposed for the task space to overcome parametric uncertainties.[7] The approximate Jacobian controllers were proposed with the task space damping for the set-point control of robot with uncertain kinematics and dynamics.[8] Moreover, an adaptive Jacobian controller was proposed for the trajectory tracking control of robot manipulators in the task space under parametric uncertainties.[9] The controller does not require exact knowledge of Jacobian matrix and dynamic parameters. Moreover, an adaptive task space tracking control method was proposed using visual task space information to overcome the parametric uncertainties in model including actuators.[2] Thus, adaptive control of robot in the task space is successful to overcome parametric uncertainties; however, unstructured uncertainties are remained to consider.

In the paper,[10–15] non-linear robust control, adaptive control and a combination of these two methods have been adopted to control the robot manipulator in the task space. In the paper,[10] dynamic uncertainties are only included in design of the proposed control. Hence, by occurrence of an uncertainty in Jacobian matrix, the stability of closed loop system is not guaranteed. In the papers,[11–13] non-linear robust control has been utilized to control the robot manipulator in the task space. The robot manipulator with the proposed method has global asymptotic stability in the presence of all dynamic and kinematic uncertainties. However, due to the use of sliding mode technique and Lyapunov redesign method in the suggested controllers, chattering in the control input is inevitable. Although a number of solutions are provided in the aforementioned papers to eliminate chattering, the robot manipulator with the modified controllers has uniform ultimate boundedness stability. In the papers,[14–15] a combination of non-linear robust control and adaptive control has been used to design the controller. The proposed methods have desired performance in overcoming structured and unstructured uncertainties as well as external disturbances in the robot manipulator dynamics in the task space. However, due to the presence of several adaptive laws in the control input, these controller have substantial number of calculations. Therefore, in case of delay in control input calculations, the stability of closed loop system faces problems.

Researchers in recent years have shown that the application of fuzzy logic in control of non-linear systems with uncertainty has led to excellent results.[16] In as much as the robot manipulator is completely non-linear and with uncertainty, the researchers have used fuzzy logic and neural network in the robot manipulator position control.[17,18] Although these methods proved capable of dealing with uncertainties in the dynamics of a robot manipulator, all these controllers are designed in the joint space and due to parametric uncertainties in Jacobian matrix, tracking precision in the joint space does not guarantee the tracking precision in the task space.

In this paper, to design the proposed control, first, the equations of robot manipulator is transferred to the task space. Next, using these dynamic equations as well as using feedback linearization method, a sliding mode controller is designed to control the robot manipulator position in the task space. Mathematical proof shows that a closed loop system with this controller has global asymptotic stability. Afterwards, to eliminate the adverse phenomenon of chattering in the control input, a first-order TSK fuzzy approximator is designed. Then, to determine the coefficients of the fuzzy sliding mode controller, SAMBA optimization algorithm is used which considerably reduces the tracking error of the position. Finally, to display the performance of the proposed control, the simulations in three steps are conducted on a two-degree-of-freedom robot manipulator.

2. Dynamic Equations of a Robot Manipulator in the Joint Space

Dynamic equation of a robot manipulator in the joint space is a nonlinear, multi-input, multi-output and second order differential equation which is expressed as

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

follows:[19]

$$D(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) + T_d + u \tag{1}$$

In which $D(q) \in R^{n\times n}$ represents the inertia matrix, $V(q, \dot{q}) \in R^{n\times n}$ is a matrix including sections related to Coriolis and centrifugal forces, $G(q) \in R^n$ stands for the gravitation vector, $T_d \in R^n$ is a vector including disturbances or un-modeled dynamics, $q(t) \in R^n$ is designated as the vector of joint positions, $\dot{q}(t) \in R^n$ is assigned as the vector of joint velocities, $\ddot{q}(t) \in R^n$ is the vector of joint accelerations, and $u \in R^n$ is the vector of robot manipulator input torque.

To simplify Eq. (1), the following equation is defined:

$$H(q, \dot{q}) = V(q, \dot{q})\dot{q} + G(q) + T_d \tag{2}$$

By substituting (2) in (1) we obtain:

$$D(q)\ddot{q} + H(q, \dot{q}) = u \tag{3}$$

Relation (1) has the following specifications:

Specifications 1: inertia matrix D(q) is symmetric and positive-definite.

3. Dynamic Equations of a Robot Manipulator in Task Space

To design controller in the task space, the dynamic equation of robot manipulator in the task space is used. For this purpose, Eq. (3) can be simplified as follows:

$$\ddot{q} = D^{-1}(q)(u - H(q, \dot{q})) \tag{4}$$

To obtain the velocity of end-effector in the task space, the following equation is used:[19]

$$\dot{X} = J(q)\dot{q} \tag{5}$$

In which $J(q) \in R^{n\times n}$ represents the Jacobian matrix, $\dot{q}(t) \in R^n$ is the vector of joint velocities, and $\dot{X}(t) \in R^n$ is the vector of velocity in the task space. Differentiating with respect to time in Eq. (5), we obtain:

$$\ddot{X} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \tag{6}$$

Assumption 1: Smoothness of the desired trajectory is condition of existence $\dot{J}(q)$. Assuming that the task space trajectory is free from singularities, by substituting Eq. (4) in (6), we obtain:

$$\ddot{X} = J(q)D^{-1}(q)(u - H(q, \dot{q})) + \dot{J}(q)\dot{q} \tag{7}$$

Equation (7) is rewritten as:

$$D(q)J^{-1}(q)\ddot{X} + H(q, \dot{q}) - D(q)J^{-1}(q)\dot{J}(q)\dot{q} = u \tag{8}$$

$J^{-1}q$ is inverse Jacobian matrix.

Assumption 2: We assume that the robot is operating in a finite task space such that the Jacobian matrix is full rank.

For transmission of torque space to force space, the following equation can be used:[19]

$$u = J^T(q)F(t) \tag{9}$$

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

Where $J^T(q)$ is Jacobian matrix transpose and $F(t) \in R^n$ is a force vector acting on the end-effector of the robot. Equation (9) in (8) is substituted and arranged as:

$$J^{-T}(q)D(q)J^{-1}(q)\ddot{X} + J^{-T}(q)H(q, \dot{q}) - J^{-T}(q)D(q)J^{-1}(q)J(q)\dot{q} = F(t) \tag{10}$$

According to Eqs. (2) and (10), the following equations are defined as:

$$\begin{cases} D_x(q) = J^{-T}(q)D(q)J^{-1}(q) \\ V_x(q, \dot{q}) = J^{-T}(q)(v(q, \dot{q}) - D(q)J^{-1}(q)\dot{J}(q)\dot{q}) \\ G_x(q) = J^{-T}(q)G(q) \end{cases} \tag{11}$$

In the above equations, analogous to the joint space quantities, $D_x(q) \in R^{n\times n}$ is the Cartesian mass matrix, $v_x(q, \dot{q}) \in R^{n\times n}$ is a vector of velocity terms in Cartesian space and $G_x(q) \in R^n$ is a vector of gravity terms in Cartesian space. $H_x(q, \dot{q})$ is defined as:

$$H_x(q, \dot{q}) = V_x(q, \dot{q})\dot{q} + G_x(q) + T_{dx} \tag{12}$$

According to the Eqs. (10) and (12), the dynamic equations of robot manipulator in the task space can be obtained as follows:

$$D_x(q)\ddot{X} + H_x(q, \dot{q}) = F(t) \tag{13}$$

In Eqs. (12) and (13), $X(t) \in R^n$ is an appropriate Cartesian vector representing position and orientation of the end-effector,[20] $\dot{X}(t) \in R^n$ is the velocity vector of end-effector in Cartesian space, $\ddot{X}(t) \in R^n$ is the vector of end-effector acceleration in Cartesian space and $T_{dx} \in R^n$ is a vector including disturbances or un-modeled dynamics in Cartesian space.

Definition 1: Sylvester's law of inertia: If $A \in R^{n\times n}$ is a symmetric square matrix and $C \in R^{n\times n}$ is non-singular matrix, then the number of positive, negative and zero eigenvalues of matrix A and matrix C^TAC are the same, where C^T is the transpose of C.[21]

According to the equation $D_x(q) = J^{-T}(q)D(q)J^{-1}(q)$ and due to the non-singularity of $J^{-1}(q)$ and in view of the specifications 1 expressed in Section 2, using Sylvester's law of inertia, the specifications 2 can be deduced.

Specifications 2: Cartesian mass matrix $D_x(q)$ is a positive-definite matrix.

4. Design of Sliding Mode Controller for Robot Manipulator in Task Space

To design sliding mode control, the sliding surface vector is defined as:[22]

$$S = (d/dt + \lambda)^{n-1}e \tag{14}$$

In Eq. (14), $e = X - X_d$ represents the tracking error vector in which $X = [x_1 x_2 \dots x_n]^T$ is the vector of end-effector position and $X_d = [x_{1d} x_{2d} \dots x_{nd}]^T$ is the vector of desired trajectory in Cartesian space and $\lambda = diag\,[\lambda_1, \lambda_2, \dots, \lambda_n]$ is a diagonal matrix in which $\lambda_1, \lambda_2, \dots, \lambda_n$ are constant and positive coefficients.

Generally, to design sliding mode controller, the variable $x_r^{(n-1)}$ is defined as:

$$x_r^{(n-1)} = x^{(n-1)} - s \tag{15}$$

Since the robot manipulator is expressed by the second order differential equation, Eq. (15) with $n = 2$ is determined as:

$$\dot{x}_r = \dot{x} - s \tag{16}$$

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

Differentiating Eq. (16), we obtain:

$$\ddot{x}_r = \ddot{x} - \dot{s} \tag{17}$$

Point 1: Since x, $\dot{x}$, $\ddot{x}$ and S are n × 1 vectors, thus $\dot{x}_r$ and $\ddot{x}_r$ are n × 1 vectors.

To design the sliding mode controller, with respect to Eqs. (16) and (17), Eq. (13) is changed into:

$$D_x(q)\,\ddot{x}_r + D_x(q)\,\dot{s} + H_x(q,\dot{q}) = F(t) \tag{18}$$

Next, the control law is proposed as:

$$F(t) = \hat{F}(t) - Ksgn(s) - As \tag{19}$$

In which sgn(s) is the sign function and $\hat{F}(t)$ is selected as:

$$\hat{F}(t) = \hat{D}_x(q)\,\ddot{x}_r + \hat{H}_x(q,\dot{q}) \tag{20}$$

In Eqs. (19) and (20), $\hat{D}_x(q)$ and $\hat{H}_x(q,\dot{q})$ are estimations of $D_x(q)$ and $H_x(q,\dot{q})$ respectively and $K = \mathrm{diag}[k_1, k_2, \ldots, k_n]$ is a positive-definite diagonal matrix and $A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix}$ is a positive-definite matrix. Substituting Eqs. (19) and (20) in (18), we obtain:

$$D_x(q)\,\ddot{x}_r + D_x(q)\,\dot{s} + H_x(q,\dot{q}) = \hat{D}_x(q)\,\ddot{x}_r + \hat{H}_x(q,\dot{q}) - Ksgn(s) - As \tag{21}$$

Equation (21) is simplified as:

$$D_x(q)\,\dot{s} = \left(\hat{D}_x(q) - D_x(q)\right)\ddot{x}_r + \left(\hat{H}_x(q,\dot{q}) - H_x(q,\dot{q})\right) - As - Ksgn(s) \tag{22}$$

For the sake of simplicity of the aforementioned equations, $\Delta D_x(q) = \hat{D}_x(q) - D_x(q)$, $\Delta H_x(q,\dot{q}) = \hat{H}_x(q,\dot{q}) - H_x(q,\dot{q})$ and $\Delta f = \Delta D_x(q)\,\ddot{x}_r + \Delta H_x(q,\dot{q})$ are defined and Eq. (22) is simplified as:

$$D_x(q)\,\dot{s} = \Delta D_x(q)\,\ddot{x}_r + \Delta H_x(q,\dot{q}) - As - Ksgn(s) = \Delta f - As - Ksgn(s) \tag{23}$$

Point 2: $\Delta f \in R^n$ is a vector including all parametric, non-structural uncertainties as well as unmodeled dynamics.

4.1. Proof of closed-loop system stability

To prove the closed-loop system stability of Eq. (22) with respect to the dynamic features of robot manipulator as mentioned in Section 3, Lyapunov function candidate is proposed as:

$$V(s) = \frac{1}{2}s^T D_x(q)\,s \tag{24}$$

Differentiating with respect to time in Eq. (24), we obtain:

$$\dot{V}(s) = s^T D_x(q)\,\dot{s} + \frac{1}{2}s^T \dot{D}_x(q)\,s \tag{25}$$

Differentiating with respect to time of all entries of matrix $D_x(q)$ and $\dot{D}_x(q)$ is defined as:

$$\dot{D}_x(q) = \begin{bmatrix} \dot{D}_{11} & \cdots & \dot{D}_{1n} \\ \vdots & \ddots & \vdots \\ \dot{D}_{n1} & \cdots & \dot{D}_{nn} \end{bmatrix} \tag{26}$$

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

With respect to Eqs. (23) and (26), Eq. (25) is rewritten, and to understand them easier, the equations are presented in matrix form:

$$\dot{V}(s) = [s_1 s_2 \dots s_n] \times \left(\begin{bmatrix} \Delta f_1 \\ \Delta f_2 \\ \vdots \\ \Delta f_n \end{bmatrix} - \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} - \begin{bmatrix} k_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & k_n \end{bmatrix} \begin{bmatrix} sgn(s_1) \\ sgn(s_2) \\ \vdots \\ sgn(s_n) \end{bmatrix} \right)$$
$$+ \frac{1}{2} [s_1 s_2 \dots s_n] \begin{bmatrix} \dot{D}_{11} & \cdots & \dot{D}_{1n} \\ \vdots & \ddots & \vdots \\ \dot{D}_{n1} & \cdots & \dot{D}_{nn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \tag{27}$$

After simplifying Eq. (27), in three steps, the following equations can be concluded:

$$\dot{V}(s) = [s_1 s_2 \dots s_n] \times \left(\begin{bmatrix} \Delta f_1 \\ \Delta f_2 \\ \vdots \\ \Delta f_n \end{bmatrix} - \begin{bmatrix} \sum_{i=1}^{n} s_i A_{1i} \\ \sum_{i=1}^{n} s_i A_{2i} \\ \vdots \\ \sum_{i=1}^{n} s_i A_{ni} \end{bmatrix} - \begin{bmatrix} k_1 sgn(s_1) \\ k_2 sgn(s_2) \\ \vdots \\ k_n sgn(s_n) \end{bmatrix} \right)$$
$$+ \frac{1}{2} \begin{bmatrix} \sum_{i=1}^{n} s_i \dot{D}_{i1} & \sum_{i=1}^{n} s_i \dot{D}_{i2} & \dots & \sum_{i=1}^{n} s_i \dot{D}_{in} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \tag{28}$$

$$\dot{V}(s) = [s_1 s_2 \dots s_n] \times \begin{bmatrix} \Delta f_1 - \sum_{i=1}^{n} s_i A_{1i} - k_1 sgn(s_1) \\ \Delta f_2 - \sum_{i=1}^{n} s_i A_{2i} - k_2 sgn(s_2) \\ \vdots \\ \Delta f_n - \sum_{i=1}^{n} s_i A_{ni} - k_n sgn(s_n) \end{bmatrix}$$
$$+ \frac{1}{2} \left(\sum_{i=1}^{n} s_i s_1 \dot{D}_{i1} + \sum_{i=1}^{n} s_i s_2 \dot{D}_{i2} + \dots + \sum_{i=1}^{n} s_i s_n \dot{D}_{in} \right) \tag{29}$$

$$\dot{V}(s) = \sum_{i=1}^{n} (s_i \, (f_i - k_i sgn(s_i))) - \sum_{i=1}^{n} \sum_{j=1}^{n} s_i s_j A_{ij} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} s_i s_j \dot{D}_{ij} \tag{30}$$

In Eq. (30), s_i is i^{th} entries of sliding surface vector S, Δf_i is i^{th} entries of vector Δf, K_i is i^{th} entries of the main diameter of matrix k, A_{ij} is entries in i^{th} rows and j^{th} columns of matrix A; in addition, $\dot{D}_{ij}$ is entries in i^{th} rows and j^{th} columns of matrix $\dot{D}_x(q)$. To prove the closed-loop system stability, Eq. (30) must be less than zero, that is:

$$\dot{V}(s) = \sum_{i=1}^{n} (s_i \, (\Delta f_i - k_i sgn(s_i))) - \sum_{i=1}^{n} \sum_{j=1}^{n} s_i s_j A_{ij} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} s_i s_j \dot{D}_{ij} < 0 \tag{31}$$

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

The aforementioned equation is satisfied if:

$$K_i > ||\Delta f_i|| \tag{32}$$

$$||A_{ij}|| > \left\| \frac{\dot{D}_{ij}}{2} \right\| \tag{33}$$

Thus by selecting appropriate K which satisfies Eq. (32) and also by selecting appropriate A which satisfies Eq. (33), the closed-loop system will possesses the global asymptotic stability.

5. Design of Fuzzy Sliding Mode Controller for a Robot Manipulator in Task Space

A first-order fuzzy TSK system is delineated by fuzzy if-then rules show the relations between inputs and outputs. Generally, first-order fuzzy TSK control system rules are defined as:

$$if\, x_1 \, is\, A_1^i \, and \ldots and\, x_n \, is\, A_n^i \, then\, y^i = a_0^i + a_1^i x_1 + \ldots + a_n^i x_n \tag{34}$$

In which i = 1, 2, . . . , M and M is the number of fuzzy rules. y^i's are the output of these M fuzzy rules and $a_0^i, a_1^i, \ldots, a_n^i$ are constant coefficients.

To design the sliding mode controller, Eq. (19) can be stated as:[18]

$$\begin{cases} F_p = \hat{F} + K - As, & s < 0 \\ F_n = \hat{F} - K - As, & s > 0 \end{cases} \tag{35}$$

With respect to Eq. (35), controller fuzzy rules can be stated as:

$$\begin{array}{l} if\, s \, is\, A_1^1 \, and\, F_p \, is\, A_2^1 \, and\, F_n \, is\, A_3^1 \, then\, y^1 = a_0^1 + a_1^1 s + a_2^1 u_p + a_3^1 u_n \\ if\, s \, is\, A_1^2 \, and\, F_p \, is\, A_2^2 \, and\, F_n \, is\, A_3^2 \, then\, y^2 = a_0^2 + a_1^2 s + a_2^2 u_p + a_3^2 u_n \end{array} \tag{36}$$

In the aforementioned relation, $a_0^1 = a_0^2 = a_1^1 = a_2^2 = a_1^2 = a_3^1 = 0$ and $a_2^1 = a_3^2 = 1$ and the membership functions will be defined as:

$$A_1^1 = \begin{cases} 1, & s \le -\gamma_1 \\ 1 - 2(s + \gamma_1)^2, & -\gamma_1 \le s \le 0 \\ 2(s - \gamma_1)^2, & 0 \le s \le \gamma_1 \\ 0, & s \ge \gamma_1 \end{cases} \tag{37}$$

$$A_1^2 = \begin{cases} 0, & s \le -\gamma_2 \\ 2(s + \gamma_2)^2, & -\gamma_2 \le s \le 0 \\ 1 - 2(s - \gamma_2)^2, & 0 \le s \le \gamma_2 \\ 1, & s \ge \gamma_2 \end{cases} \tag{38}$$

In Eqs. (37) and (38), γ_1 *and* γ_2 are positive constants.

$$A_2^1 = A_2^2 = 1, \quad lower\, bound\, of\, F \le F_p \le upper\, bound\, of\, F \tag{39}$$

$$A_3^1 = A_3^2 = 1, lower\, bound\, of\, F \le F_n \le upper\, bound\, of\, F \tag{40}$$

Point 3: To design the controller for the robot manipulator, designers need to have access to the information of dynamic equations of robot. In this case, the uncertainties bound of the dynamic equations of robot manipulator is determined. Therefore, for the desirable performance of robot manipulator, the bound of exerted force to end-effector is determined.

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

Assuming $x = [s, F_p, F_n]^T$ to be input vector of fuzzy TSK system, its output will be calculated based on the combination of fuzzy rules (36) and is expressed as follows:

$$y = \frac{\sum_{i=1}^{2} f^i(x) y^i(\mathrm{x})}{\sum_{i=1}^{2} f^i(x)} \tag{41}$$

$\mathrm{f^i(x)}$ is the firing strength of the i^{th} rule, which is obtained from the following equation:

$$f^i(x) = \mu_{A_1^i}(x_1) * \mu_{A_2^i}(x_2) * \mu_{A_3^i}(x_3) \tag{42}$$

$*$ is the indicator of a t-norm and $\mu_{\mathrm{A_j^i}}(\mathrm{x_j})$ indicates the membership degree of the input $\mathrm{x_j}$ in the membership function $\mathrm{A_j^i}$ from the ith rule.

6. Self-Adaptive Modified Bat Algorithm (SAMBA)

In this section, a new optimization algorithm based on bat algorithm (BA) is proposed.

6.1. Original bat algorithm

BA is a met heuristic population based algorithm which simulates the searching behavior of bat animals for food. The main concept behind the BA is constructed using three simple and basic ideas:[23]

1. Each bat animal with the position X_i has the velocity of V_i producing an especial pulse with the frequency and loudness of f_i and A_i respectively.
2. Echolocation phenomenon is sued to distinguish between the food and prey.
3. Loudness A_i alters from a large value to a low value.
4. During the optimization process, the frequency f_i and rate r_i of each pulse is regulated automatically.

Similar to the other evolutionary optimization algorithms, the BA also starts its search using a random population. The process of updating the position of bats is as follows:

$$\begin{aligned} V_i^{new} &= V_i^{old} + f_i\,(Gbest - X_i), \quad i = 1, \ldots, N_{Bat} \\ X_i^{new} &= X_i^{old} + V_i^{new}, \qquad\qquad i = 1, \ldots, N_{Bat} \\ f_i &= f_i^{min} + \varphi_1 \left(f_i^{max} - f_i^{min}\right), \quad i = 1, \ldots, N_{Bat} \end{aligned} \tag{43}$$

Where $Gbest$ is the best bat; N_{Bat} is the size of the population; f_i^{max}/f_i^{min} are the maximum / minimum frequency of the i^{th} bat and φ_1 is a random value in the range [0,1].

In the BA, another random movement is also simulated. Therefore, a random number β is generated randomly. If this random value is larger than r_i, a new solution around the bat X_i is produced:

$$X_i^{new} = X_i^{old} + \varepsilon A_{mean}^{old}, \quad i = 1, \ldots, N_{Bat} \tag{44}$$

Where ε is a random value in the range of $[-1,1]$ and A_{mean}^{old} is the mean value of the bats' frequency loudness. On the other hand, if the random value β is less than r_i, a new solution X_i^{new} is generated randomly. The new solution X_i^{new} can be accepted if the two conditions below are satisfied:

$$\beta < A_i f(X_i) < f(Gbest) \tag{45}$$

Meanwhile, the loudness and rate parameter are renewed as follows:

$$A_i^{new} = \alpha A_i^{old} r_i^{Iter+1} = r_i^0[1 - \exp(-\gamma \times Iter)] \tag{46}$$

Here α and γ are constant values and $Iter$ is the number of the iteration number.

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

6.2. Self-adaptive modification method

In this section, a new self-adaptive modification method is proposed to advance the total search ability of the BA efficiently. The key point behind this modification mechanism is to make use of an adaptive structure to allow the bats to have the choice of selection between two different modifications. In fact, the proposed modification method consists of two modification methods which are described at below.

6.3. Sub-modification method 1

The purpose of the first modification is to increase the diversity of the bat population using the crossover and mutation operators. The significant role of this modification is to improve the performance of the optimization algorithms which is demonstrated in the literature.[24,25] In this regard, for each bat X_i three bats X_{m1}, X_{m2} and X_{m3} are chosen randomly such that $m_1 \neq m_2 \neq m_3 \neq i$. Next, by the use of mutation operator, a test solution is generated:

$$\begin{aligned} X_{Test} &= X_{m1} + \varphi_1(X_{m2} - X_{m3}) \\ X_{Test} &= [x_{Test,1}, x_{Test,2}, \ldots, x_{Test,n}] \end{aligned} \tag{47}$$

Where n is the length of the control vector. Then, the crossover operator is employed to produce two new promising optimal solutions as follows:

$$\begin{aligned} X_{Test1} &= \begin{cases} x_{i,j}, & \varphi_2 < \varphi_3 \\ gbest_j, & ,\varphi_3 \geq \varphi_2 \end{cases} \\ X_{Test2} &= \begin{cases} x_{Test,j}, & \varphi_3 < \varphi_4 \\ gbest_j, & \varphi_4 \geq \varphi_3 \end{cases} \\ X_i &= [x_{i,1}, x_{i,2}, \ldots, x_{i,n}] \\ Gbest &= [gbest_1, gbest_2, \ldots, gbest_n] \end{aligned} \tag{48}$$

In the above equations, φ_1 to φ_4 are random values in the range [0,1].

6.4. Sub-modification method 2

This modification method is sued to update the parameter α in Eq. (46) during the optimization adaptively.

$$\alpha^{new} = (1/2Iter)^{1/Iter}\alpha^{old} \tag{49}$$

This formulation is obtained experimentally by several running of the algorithm.

At the commencement, a probability parameter is defined for the sub-modification methods (called Pr_θ for θ^{th} sub-modification method). It is initially assumed that the probability parameters of both modification methods are equal; i.e. $Pr_\theta = 0.5 \,\&\, \theta = 1, 2$. As mentioned before, the idea of this adaptive modification is to give the bats the choice of preference. Nevertheless, it is the successful performance of each sub-modification which can increase or decrease their probability. It is evident that bigger Pr_θ shows more chance for θ^{th} modification to be chosen as the proper sub-modification method by the bats.

In each iteration, the bat population is sorted in descending order. Next, better bat solution will take a higher weighting factor:

$$WT_j = \frac{\mathrm{Log}\,(N - j + 1)}{\sum_{i=1}^{n} Log(i)}; \quad j = 1, \ldots, N \tag{50}$$

Where N is the number of bats in the population. Next, the probability success of each sub-modification method is updated as follows:

$$Pr_\theta = Pr_\theta + \frac{WT_l}{n_{Mod_\theta}}; \quad l = 1, \ldots, n_{Mod_\theta}, \quad \theta = 1, 2 \tag{51}$$

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

Begin:
  For i=1:N_{Bat} where N_{Bat} is size of bat population
    If $_1 \leq Prb_1^{Iter+1}$
    Select modification method 1 for the bat solution i
    ElseIf $Prb_1^{Iter+1} < rand_i \leq Prb_1^{Iter+1} + Prb_2^{Iter+1}$
    Select modification method 2 for the bat solution i
    End If
  End For i
End

Fig. 1. Pseudo code for choosing θ^{th} modification method by RWM.

Here,n_{Mod_θ} indicates the number of bats that have chosen the θ^{th} sub-modification method. At the end of each iteration, the probability success parameters are updated as follows:

$$Pr_\omega = \frac{Pr_\omega}{\sum_{\omega=1}^{2} Pr_\omega} \tag{52}$$

In order to keep the random characteristics of the algorithm, we make use of roulette wheel mechanism for selection of the proper modification method by each bat. This process is shown in Fig. 1.

Generally, the heuristic algorithm such as SAMBA only requires checking the cost function for guidance of its search and no longer requiring information regarding the system.[26–31] Therefore, in this paper, the Mean of Root of Squared Errors (MRSE) is considered as follows:

$$MRSE = E(K) = {}^1\!/_N \sum_{i=1}^{N} |e(i)| \tag{53}$$

Where, $e(i)$ is the trajectory error of ith sample for the object, N is the number of sample, i is the iteration number, $u(i)$ is the control signal.

7. Implementation of Optimal Fuzzy Sliding Mode Control

To implement the proposed control, the following should be done step by step:

1. Based on the available information in the robot manipulator dynamics, determine the estimated values of $\hat{D}_x(q)$ and $\hat{H}_x(q, \dot{q})$.
2. By determining tracking error vectore and its differential $\dot{e}$, find sliding surface vector s.
3. By using Eq. (17), determine $\ddot{X}_r$.
4. By using Eqs. (32) and (33) and the information contained on the uncertainties in the robot manipulator dynamics, determine the allowable range of the matrices K and A.
5. In this step, to design the first-order TSK fuzzy system, use singleton fuzzifier.
6. Since the values of F_p and F_n are determined until the fourth step, determine the rules base of fuzzy system by using Eqs. (36), (37), (38), (39) and (40).
7. By using the SAMBA algorithm, determine the optimal values of entries of the matrices K and A, the coefficients in the sliding surface vectors and coefficients of γ_1 and γ_2.

7.1. Advantage of optimal fuzzy sliding mode control

In the design of the proposed control, a number of factors have been considered which easily makes the control to be implemented. The most important factors are as follows:

1. The use of the feedback linearization method causes the uncertainty bounds to reduce. Hence, in determining the control input coefficients, we can act such that the amplitude of control input is in a desired range.
2. Sliding mode control is capable of overcoming all uncertainties in the robot manipulator dynamics. On the other hand, the fuzzy theory has proved its ability in controlling systems with uncertainties.

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

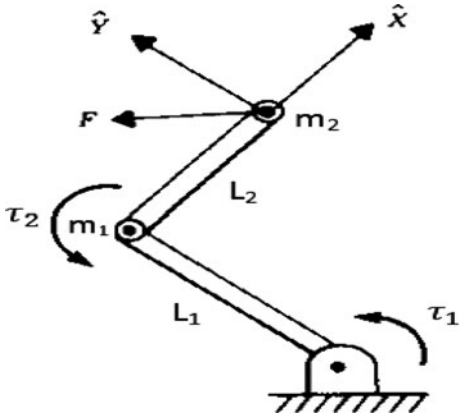


Fig. 2. Robot manipulator with two revolute joints.

Therefore, using a combination of sliding mode control and fuzzy logic theory makes the proposed control capable overcome structured and unstructured uncertainties as well as external disturbances and hence, makes it more flexible.

3. One of the problems in practical implementation of sliding mode controllers is the occurrence of control input chattering. In the proposed control, this problem is tackled by using the TSK fuzzy system.
4. One of the most important issues in practical implementation of controllers is the number of calculations of the control input. Since the rules base of fuzzy inference engine in the proposed control has only two rules, the number of calculations of the control input is low.
5. In many of the controllers presented for the robot manipulator position tracking, trial and error is used to determine the control input coefficients. Hence, the use of non-optimal coefficients lead to an increase in control input amplitude. In this case, the actuators of robot manipulator go to saturation mode. For such cases, the designers use actuators with high power. As a result, the cost of practical implementation of the controllers increases. Since the input coefficients of the proposed control is obtained from the SAMBA algorithm, the amplitude of control input is the optimal and the problem of actuator saturation is resolved.
6. Since the proposed control is designed in the task space, kinematic uncertainties of the robot manipulator has no effect on the performance of the proposed controller.

8. A Case Study on Revolute Double-Joint Robot Manipulator

The controllers which have been designed and scrutinized in this paper are conducted on the revolute double-joint robot manipulator of Fig. 2.

Dynamic equations of this robot are as follows:[19]

$$\mathrm{D}_x\,(\mathrm{q})\,\ddot{\mathrm{X}} + \mathrm{V}_\mathrm{x}(\mathrm{q},\dot{\mathrm{q}})\dot{\mathrm{q}} + \mathrm{G}_\mathrm{x}(\mathrm{q}) + \mathrm{T}_{\mathrm{dX}} = \mathrm{F}(\mathrm{t}) \tag{54}$$

In which:

$$D_x\,(q) = \begin{bmatrix} m_1 + \frac{m_1}{(sinq_2)^2} & 0 \\ 0 & m_2 \end{bmatrix} \tag{55}$$

$$V_x\,(q,\dot{q}) = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \tag{56}$$

$$V_{11} = -\left(m_2 L_1\,(cosq_2) + m_2 L_2)\dot{q}_1 - 2m_2 L_2 + m_2 L_1\,(cosq_2) + m_1 L_1 \frac{cosq_2}{(sinq_2)^2}\right)\dot{q}_2 \tag{57}$$

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

Table I. Parameters of revolute double-joint robot.

$\hat{L}_1 = 1.1\,\text{m}$	$L_1 = 1\,\text{m}$
$\hat{m}_1 = 9.5\,\text{kg}$	$m_1 = 10\,\text{kg}$
$\hat{L}_2 = 0.9\,\text{kg}$	$L_2 = 0.8\,\text{m}$
$\hat{m}_2 = 7.5\,\text{kg}$	$m_2 = 8\,\text{kg}$
$g = 9.8\,{}^{m}/_{s^2}$	$T_{d_x} = T_{d_y} = 5$

Table II. Controlling parameters in revolute double-joint robot manipulator.

$k_2 = 200$	$k_1 = 100$
$\lambda_2 = 100$	$\lambda_1 = 50$
$A_{12} = 0$	$A_{11} = 90$
$A_{22} = 100$	$A_{21} = 0$
$\gamma_2 = 0.5$	$\gamma_1 = 0.5$

$$V_{12} = -m_2 L_2 \dot{q}_2 \tag{58}$$

$$V_{21} = m_2 L_1 (sinq_2)\,\dot{q}_1 + m_2 L_1 (sinq_2)\,\dot{q}_2 \tag{59}$$

$$V_{22} = 0 \tag{60}$$

$$G_x(q) = \begin{bmatrix} m_1 g \frac{cosq_1}{sinq_2} + m_2 g\,(sinq_1)\,(sinq_2) \\ m_2 g\,(cosq_1)\,(cosq_2) \end{bmatrix} \tag{61}$$

$$T_{dX} = \begin{bmatrix} T_{dx} \\ T_{dy} \end{bmatrix} \tag{62}$$

In each link, the mass distribution is considered as point particle and the center of mass of each link is considered to be determined at the end. L_1 represents the length of the first link, L_2 is designated as the length of the second link, m_1 is assigned as the mass of the first link, m_2 is the mass of the second link, g is the gravity, T_{dx} is the disturbance or un-modeled dynamic and F is the force exerted on the end-effector.

The quantities for the robot which are utilized in this simulation have been presented in Table I.

Point 4: $\hat{L}_1$, $\hat{m}_1$, $\hat{L}_2$ and $\hat{m}_2$ are the estimations from the actual quantities of L_1, m_1, L_2 and m_2 which have been utilized in calculation of $\hat{F}$.

The quantities of controlling parameters in controller (19) which have been utilized in this simulation are presented in Table II.

Point 5: Quantities k_1 and k_2 are calculated based on Eq. (32) and also quantities A_{11},A_{12}, $A_{21}$and A_{22} are calculated based on Eq. (33).

It is worth mentioning that these values are obtained in their allowable range by trial and error. The values of Table II is only used in the first and second steps of simulations.

By the parameters mentioned in Tables I and II, the relation (19) is applicable. Matrix $D_x(q)$ is calculated as:

$$\dot{D}_x(q) = \begin{bmatrix} \frac{-m_1(2(sinq_2)(cosq_2))}{(sinq_2)^4} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \dot{D}_{11} & \dot{D}_{12} \\ \dot{D}_{21} & \dot{D}_{22} \end{bmatrix} \tag{63}$$

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

In Eq. (63), upper bound of $\dot{D}_{11}$ is specified. Thus considering Lyapunov function candidate as Eq. (24), we can conclude Eq. (30) for the double-link robot as:

$$\dot{V}(s) = \sum_{i=1}^{2}(s_i\,(\Delta f_i - k_i sgn(s_i))) - \sum_{i=1}^{2}\sum_{j=1}^{2} s_i s_j A_{ij} + \frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2} s_i s_j \dot{D}_{ij} \tag{64}$$

With respect to quantities of $\dot{D}_{11}$, $\dot{D}_{12}$, $\dot{D}_{21}$ and $\dot{D}_{22}$, Eq. (64) is simplified as:

$$\dot{V}(s) = \sum_{i=1}^{2}(s_i\,(\Delta f_i - k_i sgn(s_i))) - s_1^2 A_{11} - s_1 s_2 A_{12} - s_2 s_1 A_{21} - s_2^2 A_{22} + \frac{1}{2} s_1^2 \dot{D}_{11} \tag{65}$$

To prove the closed-loop system stability, Eq. (65) must be less than zero, that is:

$$\dot{V}(s) = \sum_{i=1}^{2}(s_i\,(\Delta f_i - k_i sgn(s_i))) - s_1^2 A_{11} - s_1 s_2 A_{12} - s_2 s_1 A_{21} - s_2^2 A_{22} + \frac{1}{2} s_1^2 \dot{D}_{11} < 0 \tag{66}$$

To satisfy the above equation, the following equations must be established:

$$K_i > \|\Delta f_i\|; \quad i = 1, 2 \tag{67}$$

$$\|A_{11}\| > \left\| \frac{\dot{D}_{11}}{2} \right\| \tag{68}$$

In addition, quantities of A_{12}, A_{21} and A_{22} are determined such that the matrix A to be positive-definite. Therefore, we can conclude global asymptotic stability for the closed-loop system. The Jacobian matrix is in the form of:

$$J(q) = \begin{bmatrix} L_1 sinq_2 & 0 \\ L_1 cosq_2 + L_2 & L_2 \end{bmatrix} \tag{69}$$

To investigate the weaknesses of sliding mode controller (19) and indicating the favorable operation of the proposed fuzzy sliding mode control, the simulations are performed in three steps:

Step 1 of simulation: Sliding Mode Control (SMC) input is simulated for revolute double-joint robot in the task space. In this step, control input of Eq. (23) is simulated for the revolute double-joint robot.

After performing the simulation, the desired and actual trajectories in Cartesian space for end-effector have been shown in Fig. 3.

According to Fig. 3, tracking errors of the end-effector position in Cartesian space for X and Y axes are shown in Fig. 4.

As evident in Figs. 3 and 4, the maximum tracking error of the end-effector position is 14×10^{-6} meters for X axis and 32×10^{-5} meters for Y axis. Oscillations around the zero will occur in the X and Y axes of tracking error of the end-effector position.

Figure 5 shows the exerted control input to the joints 1 and 2.

It is evident that the exerted control input has a chattering domain in the range of 46 to 786 Newton meters for the joint 1 in most time intervals. This domain is from 24 to 358 Newton meters for the joint 2. This chattering can lead to the activation of dynamic modes of the robot manipulator.

Step 2 of simulation: Fuzzy Sliding Mode Control (FSMC) input is simulated for revolute double-joint robot in the task space.

After execution of the simulation, the tracking error of the end-effector position on X and Y axes have been indicated in Fig. 6.

According to this figure, the tracking error of the end-effector position on X axis will reach zero after 0.189 s; thereafter, tracking will continue with no errors and oscillations. In the event that

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

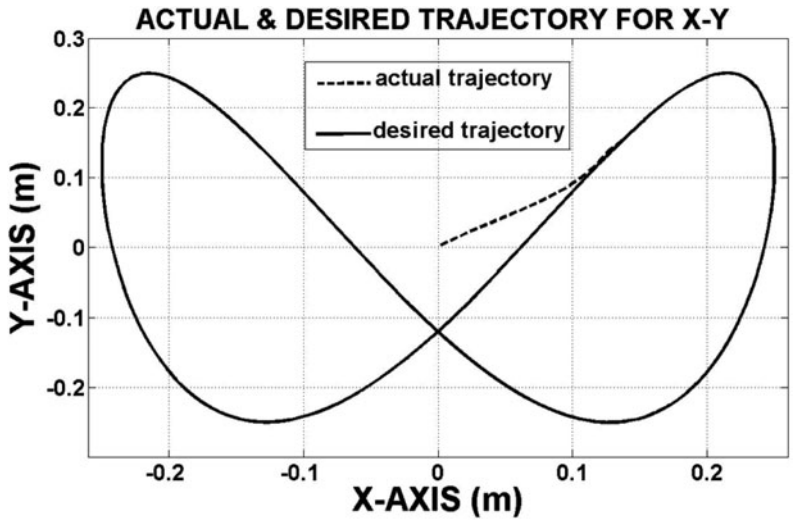


Fig. 3. The desired and actual trajectories in Cartesian space for end-effector.

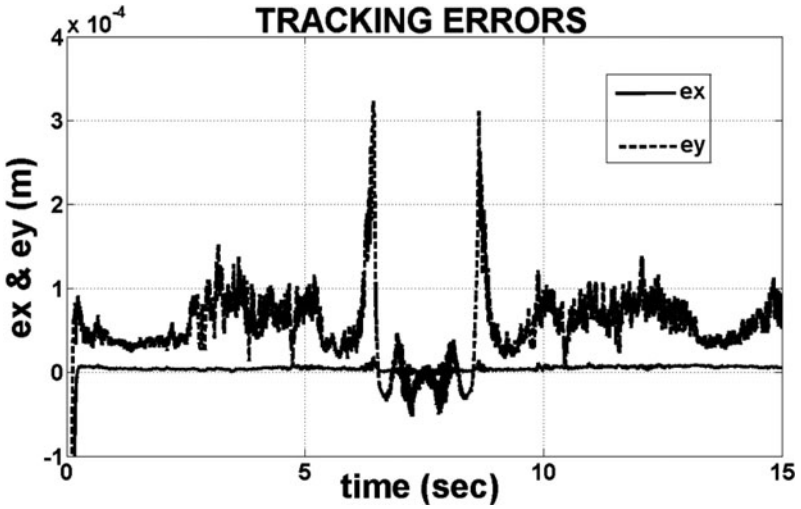


Fig. 4. Tracking error of the end-effector position in Cartesian space.

Table III. Controlling parameters in revolute double-joint robot manipulator.

$k_2 = 158$	$k_1 = 84$
$\lambda_2 = 69$	$\lambda_1 = 32$
$A_{12} = 0$	$A_{11} = 57$
$A_{22} = 71$	$A_{21} = 0$
$\gamma_2 = 0.2621$	$\gamma_1 = 0.2783$

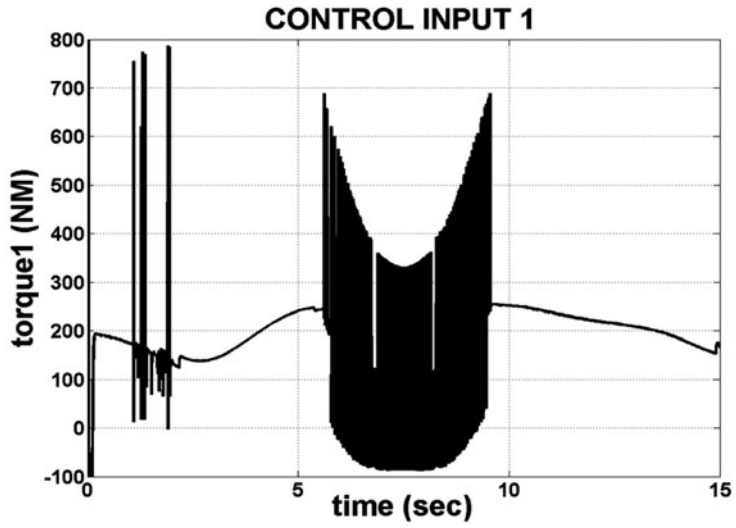
the tracking error on Y axis will reach zero after 7.464 s; moreover, it never remains zero and the maximum tracking error on Y axis is 8×10^{-7} meters.

Figure 7 shows the control inputs for the joints 1 and 2.

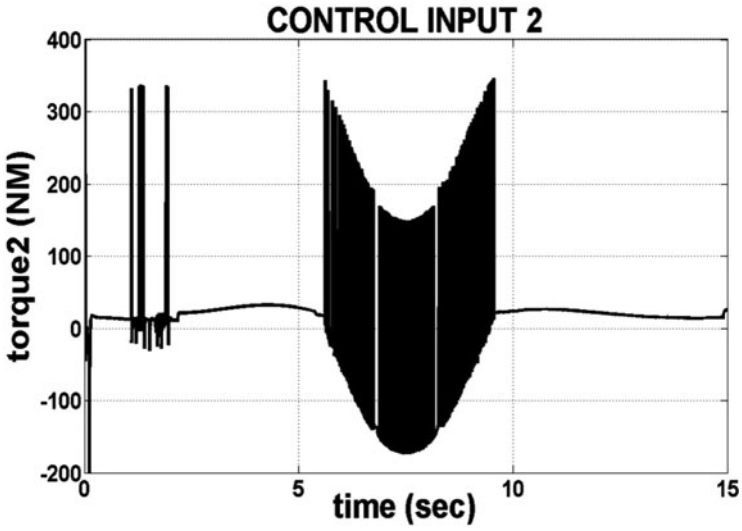
As it is understood from Fig. 7, the control inputs for the joints 1 and 2 have no chattering.

Step 3 of simulation: In this step of the simulation, the fuzzy sliding mode controller parameters are searched and adjusted in the allowable range by the SAMBA algorithm. The values are presented in Table III.

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press



(a) The exerted control input to joint 1



(b) The exerted control input to joint 2

Fig. 5. Exerted control inputs to joints 1 and 2 (a) The exerted control input to joint 1 (b) The exerted control input to joint 2.

The Optimal Fuzzy Sliding Mode Control (OFSMC) input is simulated for the revolute double-joint robot in the task space. Block diagram of this simulation is shown in Fig. 8.

So far in all steps of simulation, a constant quantity of disturbance, according to Table I, is applied to the robot. In this step of the simulation, to test the robustness of the suggested control system against disturbances, the control system is challenged and disturbances are applied to the robot as shown in Fig. 9.

After execution of simulation, the tracking error of the end-effector position on X and Y axes have been indicated in Fig. 10.

According to this figure, the tracking error of the end-effector position in X and Y axes is very negligible and limited to zero. With a little care and comparing the tracking errors in the previous steps of simulations, significant reduction of the tracking error of the end-effector position on X and Y axes is visible.

Figure 11 shows the control inputs for the joints 1 and 2.

As it is observed from Fig. 11, the control inputs for joints 1 and 2 have no chattering. In addition, with comparing Figs. 8 and 11, it is understood that in the timescales ranging, the size of control inputs

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

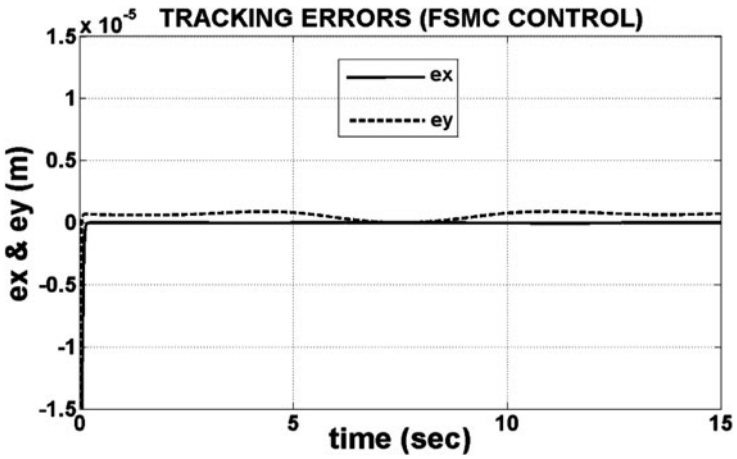
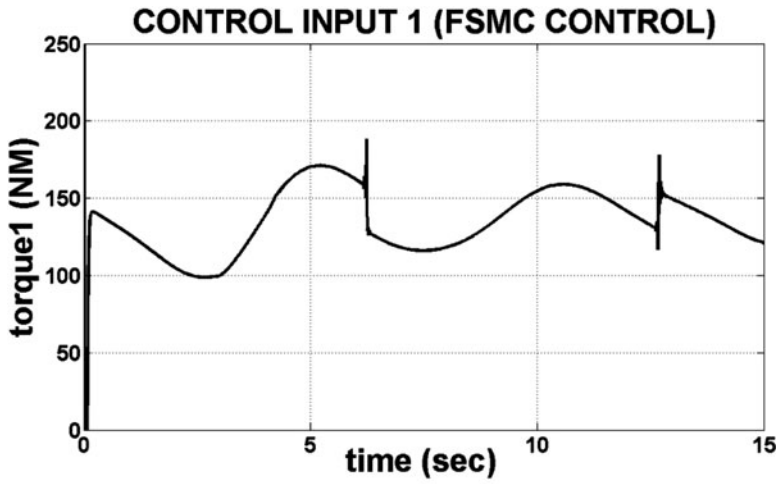
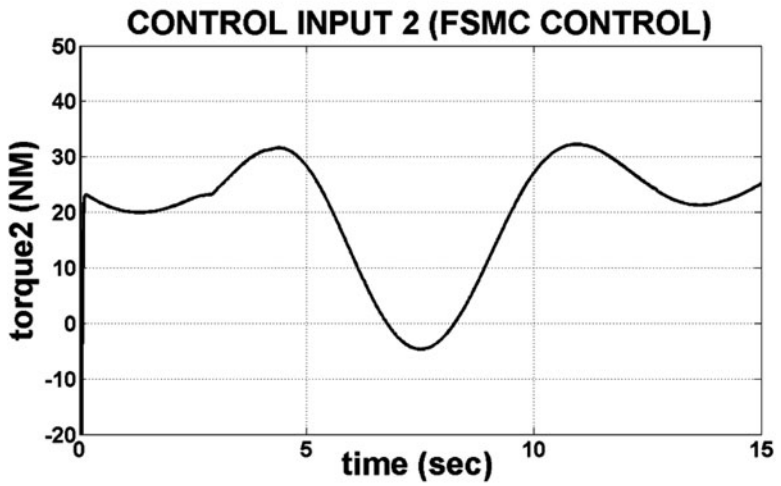


Fig. 6. Tracking error of the end-effector position on X and Y axes.



(a) Exerted control input to joint 1



(b) Exerted control input to joint 2

Fig. 7. Exerted control inputs to joints 1 and 2 (a) Exerted control input to joint 1 (b) Exerted control input to joint 2.

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

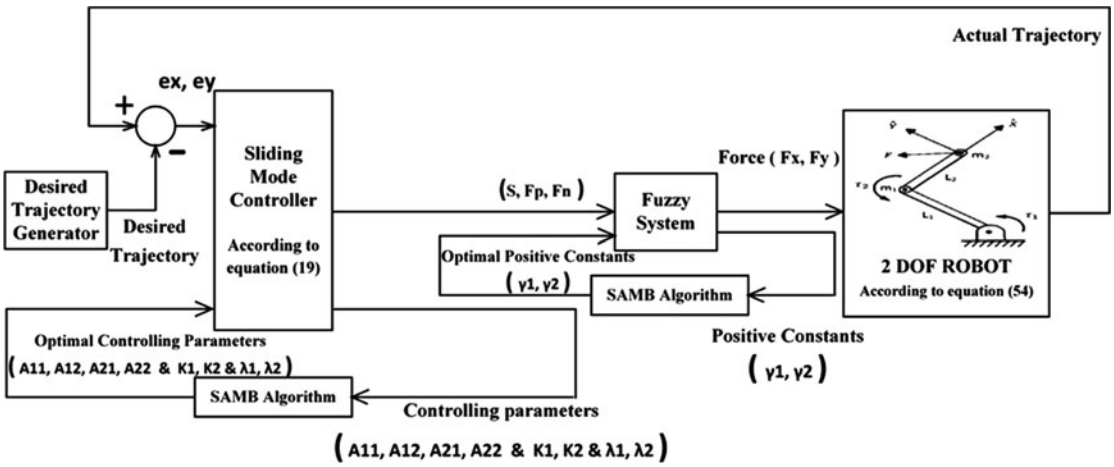


Fig. 8. Block diagram of Optimal Fuzzy Sliding Mode Control (Step 3 of simulation).

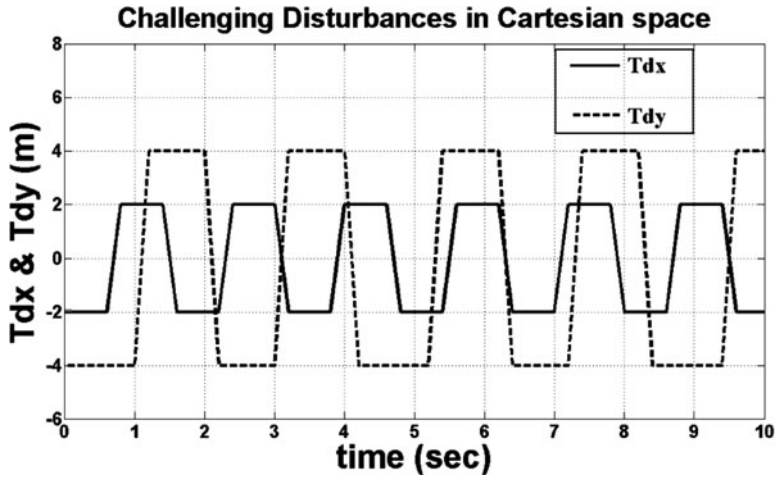


Fig. 9. Exerted challenging disturbances to the revolute double-joint robot in task space.

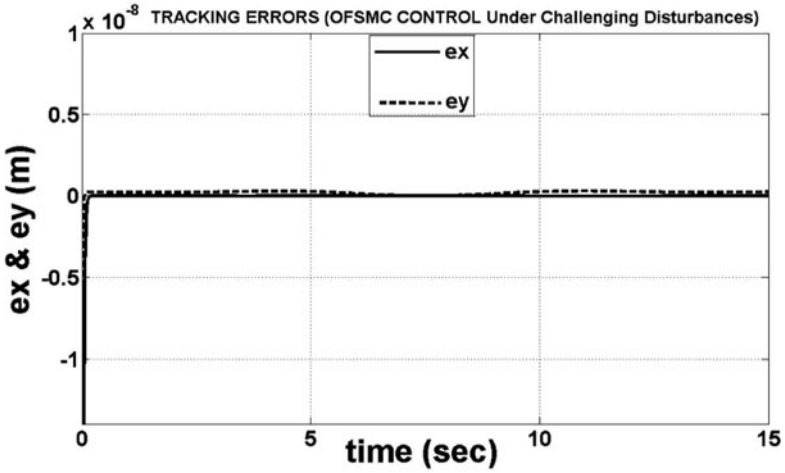
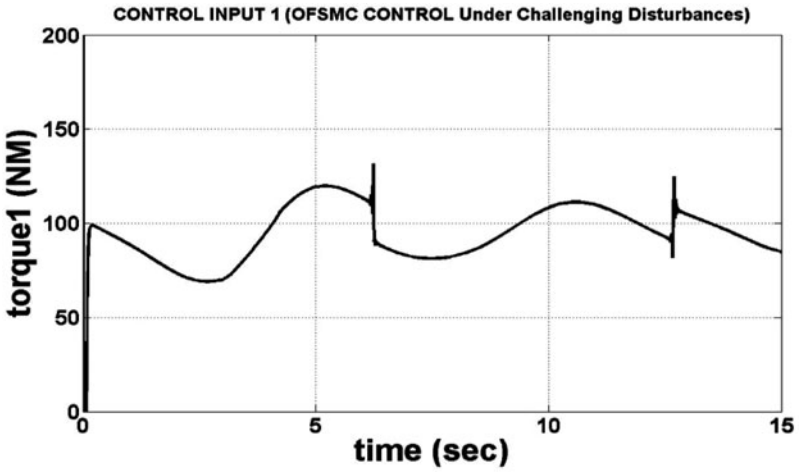
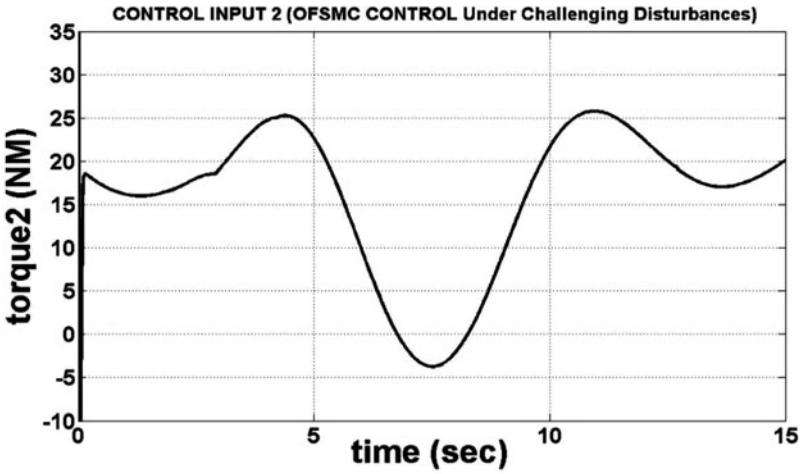


Fig. 10. Tracking error of the end-effector position on X and Y axes by applying disturbances are shown in Fig. 9.

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press



(a) Exerted control input to joint 1



(b) Exerted control input to joint 2

Fig. 11. Exerted control inputs to joints 1 and 2 by applying disturbances are shown in Fig. 9 (a) Exerted control input to joint 1 (b) Exerted control input to joint 2.

declined so that the reduction of control inputs 1 and 2 are 42 to 57 and 4.3 to 7.2 Newton meters, respectively. Given Figs, 10 and 11 and comparing them with the results of step 2 of simulation, we understand that despite exerting disturbances to the revolute double-joint robot as shown in Fig. 9, we have achieved our control objectives which had very negligible position tracking error and free-of-chattering control inputs with ability to implement practically.

9. Conclusions

In this paper, the sliding mode controller has been presented to track robot manipulator position in the task space. In the design of this controller, a combination of feedback linearization method and sliding mode control has been utilized. The mathematical proof demonstrated that the closed loop system with the proposed control has global asymptotic stability in the presence of all structured and unstructured uncertainties as well as external disturbances. Further, to eliminate the problem of chattering in the control input, a first-order TSK fuzzy approximator was designed by using the fuzzy theory. The simulations of the second step showed that the presented fuzzy sliding mode controller has no chattering and possesses all the advantages of sliding mode control. Ultimately, the SAMBA algorithm was employed to adjust the input coefficients of the fuzzy sliding mode controller. The results of the simulations of the third step revealed that the optimal fuzzy sliding mode controller has

https://doi.org/10.1017/S0263574714001258 Published online by Cambridge University Press

a favorable performance in overcoming all the uncertainties in the robot manipulator. In design and modification of the proposed controllers, we considered factors which are discussed in the Section 7 of the paper. The simulations which were performed in the three steps clearly demonstrated the problems of sliding mode control and the advantages of fuzzy sliding mode control and the optimal fuzzy sliding mode control.

References

1. W. Yu and M. A. Moreno-Armendariz, "Robust visual servoing of robot manipulators with neuro compensation," *J. Franklin Inst*. **342**, 824–838 (2005).
2. C. Liu, C. C. Cheah and J.-J. E. Slotine, "Adaptive Jacobian tracking control of rigid-link electrically driven robots based on visual taskspace information," *J. Autom*. **42**, 1491–1501 (2006).
3. J. M. Valenzuela and L. G. Hernández, "Operational space trajectory tracking control of robot manipulators endowed with a primary controller of synthetic joint velocity," *ISA Trans.* **50**(1), 131–140 (2011).
4. S. P. Guzman, J. M. Valenzuela and V. Santibáñez: "Adaptive neural network motion control of manipulators with experimental evaluations," *Sci. World J.* Vo. **2014**, 1–13 (2014).
5. Z. Qu and D. M. Dawson, *Robust Tracking Control of Robot Manipulators* (IEEE Press, Inc., New York, USA 1996).
6. H. G. Sage, M. F. De Mathelin and E. Ostertag, "Robust control of robot manipulators: A survey," *Int. J. Control* **72**(6), 1498–1522 (1999).
7. W. E. Dixon, "Adaptive Regulation of Amplitude Limited Robot Manipulators with Uncertain Kinematics and Dynamics," *Proceedings of American Control Conference*, Boston, MA (2004) pp: 3844–3939.
8. C. C. Cheah, M. Hirano, S. Kawamura and S. Arimoto. : "Approximate Jacobian control with task-space damping for robot manipulators," *J. IEEE Trans. Autom. Control* **49**(5), 752–757 (2004).
9. C. C. Cheah, M. Hirano, S. Kawamura and S. Arimoto, "Approximate Jacobian control for robots with uncertain kinematics and dynamics," *J. IEEE Trans. Robot. Autom.* **19**(4), 692–702 (2003).
10. M. R. Soltanpour, M. M. Fateh and A. R. Ahmadifard, "Nonlinear tracking control on a robot manipulator in the task space with uncertain dynamics," *J. Appl. Sci. Asian Netw. Sci. Inf.* **8**(23), 4397–4403 (2008).
11. M. R. Soltanpour and M. Siahi, "Robust control of robot manipulator in task space," *Int. J. Appl. Comput. Math.* **8**(2), 227–238 (2009).
12. M. M. Fateh and M. R. Soltanpour, "Robust task-space control of robot manipulators under imperfect transformation of control space," *Int. J. Innovative Comput. Inf. Control* **5**(11)(A), 3949–3960 (2009).
13. M. R. Soltanpour and S. E. Shafiei, "Robust backstepping control of robot manipulator in task space with uncertainties in kinematics and dynamics," *Electron. Electr. Eng. J. Autom. Robot.* **96**(8), 75–80 (2009).
14. M. R. Soltanpour and M. M. Fateh, "Adaptive robust control of robot manipulators in the task space under uncertainties," *Aust. J. Basic Appl. Sci.* **1**(3), 308–322 (2009).
15. M. R. Soltanpour and S. E. Shafiei, "Robust adaptive control of manipulators in the task space by dynamical partitioning approach," *Int. J. Electron. Electr. Eng. J. Autom. Robot.* **101**(5), 73–78 (2010).
16. M. R. Soltanpour, B. Zolfaghari, M. Soltani and M. H. Khooban, "Fuzzy sliding mode control design for a class of nonlinear systems with structured and unstructured uncertainties," *Int. J. Innovative Comput. Inf. Control* **9**(7), 2713–2726 (2013).
17. S. E. Shafiei and M. R. Soltanpour, "Robust neural network control of electrically driven robot manipulator using backstepping approach," *Int. J. Adv. Robot. Syst.* **6**(4), 285–292 (2009).
18. M. Veysi and M. R. Soltanpour, "Eliminating chattering phenomenon in sliding mode control of robot manipulators in the joint space using fuzzy logic," *J. Solid Fluid Mech. Shahrood University of Technology* **2**(3), 45–54 (2012).
19. J. J. Craig, *Introduction to Robotics: Mechanics and Control* (Addision-Wesley, 2005).
20. O. Khatib, Dynamic Control of Manipulators in Operational Space. Sixth IFTOMM Congress on Theory of Machines and Mechanisms, New Dehli (1983).
21. Strang . Gilbert . : Linear algebra and its application, Massachusetts Institute of Technology, saunders College Publishing (1986).
22. J. J. E. Slotine and W. Li, *Applied Nonlinear Control* (Prentice-Hall 1991).
23. X.-S. Yang, A. H. Gandomi and B. Algorithm: "A novel approach for global engineering optimization," *Eng. Comput.* **29**(5), 464–483 (2012).
24. A. Baziar, A. Kavoosi-Fard and J. Zare. "A novel self adaptive modification approach based on bat algorithm for optimal management of renewable MG," *J. Intell. Learn. Syst. Appl.* **5**(1) (2013).
25. T. Niknam, A. Kavousifard and A. Seifi, "Distribution feeder reconfiguration considering fuel cell/wind/photovoltaic power Plants," *Renew. Energy* **37**(1), 213–225 (2011).
26. M. H. Khooban, M. R. Soltanpour, D. Nazari and Z. Esfahani, "Optimal intelligent control for HVAC systems," *J. Power Technol.* **92**(3), 192–200 (2012).
27. M. H. Khooban, A. Alfi and D. N. M. Abadi, "Teaching–learning-based optimal interval type-2 fuzzy PID controller design: a nonholonomic wheeled mobile robots," *Robotica* **31**, 1059–1071. doi:10.1017/S0263574713000283, (2013).
28. M. H. Khooban, A. Alfi and D. N. M. Abadi, "Control of a class of nonlinear uncertain chaotic systems via an optimal type-2 fuzzy PID controller," *IET Sci. Meas. Technol.* **7**(1), 50–58 (2013).

29. M. R. Soltanpour, M. H. Khooban and M. Soltani. "Robust fuzzy sliding mode control for tracking the robot manipulator in joint space and in presence of uncertainties," Robotica, available on CJO2013. doi:10.1017/S0263574713000702, (2013).
30. M. H. Khooban, "Design an intelligent proportional-derivative (PD) feedback linearization control for nonholonomic-wheeled mobile robot," *J. Intell. Fuzzy Syst*. **26**(4), 1833–1843. Doi: 10.3233/IFS-130863, (2014).
31. T. Niknam, M. H. Khooban, A. Kavousifard and M. R. Soltanpour, "An optimal type II fuzzy sliding mode control design for a class of nonlinear systems," *Nonlinear Dyn.* **75**(1–2), 73–83 (2014).