



# Optimal fuzzy PD control for a two-link robot manipulator based on stochastic fractal search

The Anh Mai<sup>a</sup> and Thai Son Dang

School of Engineering and Technology, Vinh University, Vinh, Nghe An, Vietnam

Received 1 September 2021 / Accepted 30 October 2021 / Published online 13 November 2021  
© The Author(s), under exclusive licence to EDP Sciences, Springer-Verlag GmbH Germany, part of Springer Nature 2021

**Abstract** This paper focuses on the design and optimization of Fuzzy PD controllers for a two-link robot manipulator system. The dynamic equations of the two-link robot manipulator are constructed by the Lagrangian approach. The accurate motion of the robot manipulator to the desired position is implemented by fuzzy PD controllers based on the relational model and the computed torque control method. The fuzzy PD controllers are optimized by the stochastic fractal search method. The proposed control system is also evaluated by comparison with some different systems in the literature such as the genetic algorithm (GA) and the modified neural network algorithm (MNNA) with PID controllers, where its abilities are demonstrated. Numerical simulations are implemented in Matlab using the classical fourth-order Runge-Kutta method to approximate the solution of the initial-value problem. The results have shown advantages of the proposed optimal fuzzy PD controllers in terms of performance such as small error, rapid response, and high stability.

## 1 Introduction

The robot has been researched and applied in many decades; it can complete required tasks and follow the object with more effectiveness and a short time than the human. Robot manipulators have been extensively used in different fields of human life and industry to perform many tasks that are pernicious or too costly for humans such as underwater activities, in space, activities in nuclear power plants [1–5]. Robot manipulators have been also used to complete tasks that are difficult for humans to achieve the required precision or quality in repetitive works such as assembly, welding, spray painting, materials handling, and other applications with such repetitive works [6–10].

A robot manipulator is a coupled multi-input multi-output (MIMO) system with the possession of highly non-linear and uncertain dynamics, where the proliferation of links will be the system further complicated. Classical PID controllers have been researched and widely used in control applications for robot manipulators. In Ref. [11], a PID controller with particle swarm optimization algorithm is used to control the position of the flexible manipulator, in which the optimal PID parameters found with PSO control the position of the flexible manipulator and ensure low vibration. A PID control scheme based on modeling compensation techniques is presented in Ref. [12] to solve the tracking problem of robot manipulators with revolute joints. All

nonlinearities were merged into a single function to generate the modeling error function and estimated via a reduced-order observer, which is used in the feedback loop to compensate for nonlinearities. In research [13], an adaptive PID control through the equivalence relationship between the two controls with simple structure, model-free, and robust against payload variations is implemented to control a WAM robot holding a baseball bat with the high levels of stability and performance using the adaptive time-delay control. But robot manipulators are complex nonlinear coupled systems; PID controllers are not really effective for control action which is shown in a review article by Kumar [14]. It means that the classical PID controllers are most effective for linear systems but not really suitable for nonlinear and complex systems.

To better match the complex nonlinear coupled properties of robot manipulators, several nonlinear control techniques are studied in many articles. A practical nonsingular terminal sliding mode control approach for robot manipulators using time-delay estimation is introduced in Ref. [15], in which the time-delay estimation is used to guarantee fast convergence and reduce computational complexity involving the robot dynamics. In research [16], the force/position hybrid control method is used to control the operation of the redundant actuated parallel robot where the cascade control strategy based on a combination of the proportional-integral control and model predictive control in the redundant branch of the robot to significantly improve the tracking ability of the system. Integrating adap-

<sup>a</sup> e-mail: [theanh@vinhuni.edu.vn](mailto:theanh@vinhuni.edu.vn) (corresponding author)

tive backstepping control approach into fractional-order controller design for an  $n$ -degree of freedom ( $n$ -DOF) robotic manipulator is presented in Ref. [17] to resolve the problem of finite-time stabilization and control the operation of the robot under interfering factors such as model nonlinearity, uncertainties, and external disturbances. An adaptive backstepping nonsingular fast terminal sliding mode control is constructed in Ref. [18] based on an integration of integral nonsingular fast terminal sliding mode surface, a backstepping control strategy, and an adaptive technique that requires no prior knowledge of the bound value of the disturbance and uncertainties. In Ref. [19], a robust control technique is designed for robot manipulators with the fast and finite-time tracking capability of the torque signals along with the desired torque commands, and a fast nonsingular terminal sliding mode control algorithm is developed to guarantee the fast convergence of the joint positions to their desired values. The paper [20] introduced a theorem for robust finite-time stability and derived a finite-time  $H_\infty$  controller for robotic manipulators that achieves high-precision tracking performance without requiring the solution of a Hamilton-Jacobi equation or a Riccati equation is derived using the backstepping method.

To implement the nonlinear control techniques such as sliding mode control, model reference adaptive control, backstepping, etc. for robot manipulators, exact plant mathematical model are required. However, dynamics of robot manipulators are highly nonlinear and complicated, and exact models of the robot systems are not always obtainable.

Many researchers always tried to find out the solution for efficient control design of nonlinear systems based on intelligent control techniques like fuzzy control, neural network, genetic algorithm, etc. In intelligent control technologies, fuzzy control is a promising solution that is widely used to design intelligent controllers. Fuzzy control is implemented based on the concept of linguistic variables which can take actions on the basis of logics and rules investigated by expert knowledge, thus it can give a powerful tool to design intelligent controller. In the paper [21], an adaptive fuzzy control system is constructed for a dual-arm robot, where an approximate Jacobian matrix and a decentralized fuzzy logic controller are used in combination to address the uncertain kinematic control and compensate for uncertain dynamics of the control object. In Ref. [22], the control structure for a robot manipulator is designed and implemented by a fractional-order fuzzy PD+I controller (FOFPD+I) which uses fuzzy logic and non-dominated sorting genetic algorithm-II for its nonlinear gains to give an adaptive capability and preserves the linear structure of fractional order PD+I controller. The papers [23, 24] propose a fuzzy sliding mode control (FSMC) scheme using the complementary cooperation of the traditional sliding mode control (SMC) and the fuzzy logic control (FLC) for the trajectory tracking control of a two-link robot manipulator where a fuzzy tuning scheme is presented to improve the performance of the control system. In Ref. [25], a kind

of adaptive sliding mode control system for tracking control of robot manipulator has structured uncertainties and unstructured uncertainties where MIMO fuzzy logical system is used as a compensator in the control law to compensate all the uncertainties. Although there have been many studies on fuzzy control applications for robot manipulators. However, fuzzy controllers are often used in combination or at a high level for adaptive control. Besides, the fuzzy controller design is based on experience and expert knowledge of the system. There have also been many optimization algorithms used for fuzzy controller optimization such as genetic algorithm, particle swarm optimization, cross-entropy, etc. but the fuzzy controller optimization problem is still a challenging problem [26–28].

Stochastic fractal search (SFS) is a new powerful optimization algorithm based on randomized fractals using the diffusion-limited aggregation (DLA) method which is developed with an effort to deal with the shortcomings such as premature convergence and poor solution quality in existing metaheuristics [29]. SFS algorithm carry out remarkable improvement in solution quality and convergence speed by finding a solution that has the minimal error within a number of small iteration in comparison with other optimal solutions. Enhancing the performance of the SFS optimization algorithm can be also implemented simply by tuning only the maximum diffusion number (MDN). The SFS method has been studied and employed in various applications in recent years. The SFS algorithm is presented in Ref. [30] to adjust the PID parameters for model identification and vibration control of a flexible beam system. In papers [31, 32], the hybrid version of the SFS algorithm is used to optimize the controller parameters in an automatic generation control of power systems. In Ref. [33], the variant SFS is implemented for optimizing PID and fuzzy PD controller parameters to enhance the performance of a simple twin-rotor control system. In the study, a small MDN is also used for the variant SFS which makes its convergence rate better than the original SFS. In Ref. [34], the SFS method is used to tune controller gains in an automatic generation control of wind incorporated hydrothermal system. The parameters of the integral minus proportional derivative with first-order filter (I-PDF) controller are tuned by the SFS method and other population-based optimization techniques in comparison where the results affirm the supremacy of the SFS against other metaheuristics. The researches have shown that SFS algorithm is good at finding a solution with remarkable improvement in solution quality and convergence speed. However, the SFS algorithm is often employed for optimization of conventional controllers such as PID controllers. Application of the SFS algorithm for fuzzy control systems should be also reviewed and studied, especially for complexity nonlinear objects such as robot manipulators.

To improve responses and simplify the control structure for complex nonlinear objects such as robot manipulators, this paper proposes the use of the fuzzy PD control system. The fuzzy PD control system is designed based on the relational model and optimized by the

stochastic fractal search algorithm to reduce the complexity, suitable for real-time applications [35–37]. The main contribution of the paper can be summarized as follows:

1. Design and implementation of the optimal fuzzy control system for a two-link robot manipulator based on relation models and Lyapunov stable analysis with a minimum number of membership functions (MFs).
2. Propose and improve the performance of the designed fuzzy control system based on an optimal algorithm using random fractals.
3. Carry out comparison and evaluation simulations to show the advantages of the proposed optimal fuzzy control system.

The rest of this paper is arranged as follows: After an introduction in Sect. 1, a brief of the fractal search algorithm is presented in Sect. 2. System description and mathematical model of a two-link robot manipulator are constructed in Sect. 3. The design and optimization of the proposed fuzzy control system are implemented in Sect. 4. In Sect. 5, simulation scenarios and discussion is used for validation of results. Finally, conclusions and next directions of the proposed work are drawn in Sect. 6.

## 2 Fractal search algorithm

Random fractals are generated through modifying the iteration process according to a stochastic rule using the Diffusion Limited Aggregation method which is the key operation used for the diffusion phase [38, 39].

The optimization algorithm based on random fractals is constructed by incorporation of fractal growths through the DLA method and potential theory where particles are generated according to the following rules [40]:

1. Each particle contains its own particular electrical potential energy.
2. Each particle is subject to the random diffusion process using the levy flight, Gaussian distribution, or incorporating the advantages of both of them to generate other random particles, where the energy of the seed particle is distributed among the generated particles.
3. In successive generations, many unsuccessful particles are dropped from the population, only a few of the best particles have the ability to survive and grow through successive generations.

Stochastic Fractal Search algorithm is an optimization algorithm with powerful capabilities and simple implementation which is constructed from two main processes of the diffusion process and the updating process [41]. In order to implement the SFS algorithm, the

parameters such as the position of each particle ( $P$ ), the lower bound ( $LB$ ), and upper bound ( $UB$ ) of search space, including the search space dimension ( $N$ ), the amount of initial particle ( $M_i$ ), maximum diffusion of each particle allowed ( $D$ ), and maximum iteration ( $G$ ) need to be defined and initialized. And then the cost function also is determined in advance to spread every initial particle in the search space using (1)

$$P_j = LB + \varepsilon \times (UB - LB), \quad j = 1, 2, \dots, M_i \quad (1)$$

The diffusion process is presented by mathematical equations (2) and (3) to produce new particles that conduct searches in defined search spaces. A Gaussian distribution is used in the SFS algorithm to carry out the diffusion process. This process is often called the Gaussian Walks ( $GW1$  &  $GW2$ ) that form a pattern of fractal growth.

$$GW_1 = Gaussian(\mu_{BP}, \sigma) + (\varepsilon \times BP - \varepsilon' \times P_i), \quad i = 1, 2, \dots, N \quad (2)$$

$$GW_2 = Gaussian(\mu_P, \sigma) \quad (3)$$

The average value ( $\mu$ ) and standard deviation ( $\sigma$ ) are defined:

$$\begin{aligned} \mu_{BP} &= |BP| \\ \mu_P &= |P_i| \\ \sigma &= \left| \frac{\log(g)}{g} \times (P_i - BP) \right| \end{aligned} \quad (4)$$

where  $BP$  is the best particle obtained by using the cost function to test each particle;  $P_i$  is the  $i$ th particle which implements diffusion; and  $g$  is the number of iterations that have been implemented when the particle is diffusing. The parameters  $\varepsilon$  and  $\varepsilon'$  are random numbers that are uniformly distributed in the range of values from 0 to 1.

A probability value to each particle based on its fitness value is used to begin the first updating process. This process is used for the purpose of providing the highest probability value to the particle with the best fitness value. Further, this process will increase the chance of the worst particle being replaced. The probability value to each particle ( $Pa_i$ ) is denoted as follows:

$$Pa_i = \frac{rank(P_i)}{N} \quad (5)$$

where  $N$  is the total number of particles in a population

After being given a probability value, the particle that has a value of  $Pa_i < \varepsilon$  will be updated as follows:

$$P'_i(j) = P_r(j) - \varepsilon \times (P_t(j) - P_i(j)) \quad (6)$$

where  $P'_i$  is the particle in new population,  $P_i$  is the previous particle in old population,  $P_r$  and  $P_t$  are randomly selected particles among each population.

After the first updating process is completed, the second updating process will be continued for all particles in the population. Again, all particles are given by probabilities according to (5), and then an update is implemented by (7)

$$P'_t = \begin{cases} P'_i - \hat{\varepsilon} \times (P'_t - BP), & \text{if } \hat{\varepsilon} \leq 0.5 \\ P'_i - \hat{\varepsilon} \times (P'_t - P'_r), & \text{if } \hat{\varepsilon} > 0.5 \end{cases} \quad (7)$$

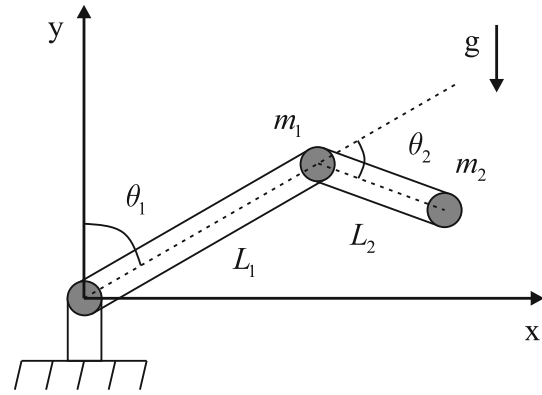
where  $P''_t$  is the particle in newest population,  $P'_r$  and  $P'_t$  are randomly selected particles based on the result of the first updating process. The complete SFS algorithm can be summarized and implemented by Algorithm 1.

```

%Initialize all particles based on (1) %Find the best
particle (BP);
%Generate a diffusion function;
function Pnew = Diffusion(Pi, g, D)
    for i = 1 : length(D) do
        %for each particle
        if condition to use GW1 then
            | generate a new particle using GW1
        else
            | %condition to use GW2;
            | generate a new particle using GW2
        end
    end
end
return;

Main algorithm
for g = 1 : length(G) do
    %Do diffusion for each particle;
    Diffusion (Pi, g, D);
    %update the value of RP ;
    %Starting the first updating process ;
    %each particle given a probability value based on
    (5) ;
    for i = 1 : length(totalparticle) do
        %for each particle Pi ;
        if ε ≥ Pai then
            | update the particle position based on (6)
        else
            | do nothing
        end
    end
    %update the best particle (RP) ;
    %Starting the second updating process ;
    %each particle given a probability value based on
    (5) ;
    for i == 1 : length(totalparticle) do
        %for each particle P'i ;
        if ε ≥ Pai then
            | update the particle position based on (7)
        else
            | do nothing
        end
    end
    %update the best particle (BP);
end
    
```

**Algorithm 1:** SFS Pseudo Algorithm



**Fig. 1** The coordinate system for a simplified model of a two-link robot manipulator

### 3 Dynamic model of two-link manipulators

Consider a 2-DOF robot manipulator with two links as shown in Fig. 1.

Referring to Fig. 1 the kinematic relations for each mass of the robot can be written [42, 43]:

$$\begin{cases} x_1 = L_1 \sin \theta \\ y_1 = L_1 \sin \theta \\ x_2 = L_1 \sin \theta + L_2 \sin(\theta_1 + \theta_2) \\ y_2 = L_1 \cos \theta + L_2 \cos(\theta_1 + \theta_2) \end{cases} \quad (8)$$

The kinetic and potential energies of the system can be given by:

$$\begin{cases} T = \frac{1}{2}m_1(\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2}m_2(\dot{x}_2^2 + \dot{y}_2^2) \\ V = m_1gL_1 \cos \theta + m_2gL_2 \cos(\theta_1 + \theta_2) \end{cases} \quad (9)$$

The definitions for kinetic energy and potential energy can be considered by Lagrange dynamics [44–48]:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} + \frac{\partial V}{\partial q_j} = Q_j \quad (10)$$

The dynamic modeling of a 2-DOF robot manipulator can be expressed as follows:

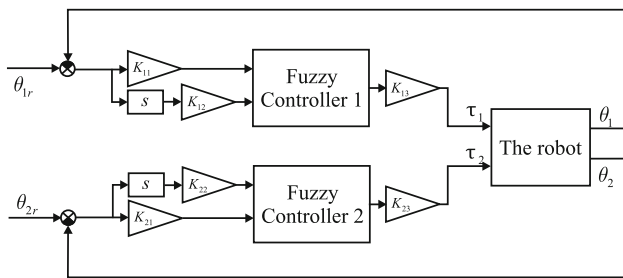
$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \quad (11)$$

where  $q = [\theta_1 \ \theta_2]^T$  is the state variable;  $u = [\tau_1 \ \tau_2]^T$  is torque exerted by actuators (servo motors) at each joint;  $M$ ,  $C$ ,  $g$  are the inertia matrix (symmetric positive definite), the Coriolis and centrifugal matrix, and the gravity vector, respectively, which are given by:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

$$M_{11} = (m_1 + m_2) L_1^2 + m_2 L_2^2 + 2m_2 L_1 L_2 \cos \theta_2;$$

$$M_{12} = m_2 L_2^2 + m_2 L_1 L_2 \cos \theta_2;$$



**Fig. 2** The proposed block diagram of the fuzzy control system for a two-link robot manipulator

$$M_{21} = m_2L_2^2 + m_2L_1L_2\cos\theta_2; \quad M_{22} = m_2L_2^2$$

$$C = \begin{bmatrix} -m_2L_1L_22\dot{\theta}_2\sin\theta_2 & -m_2L_1L_2\dot{\theta}_2\sin\theta_2 \\ 0 & -m_2L_1L_22\dot{\theta}_1\sin\theta_2 \end{bmatrix}$$

$$g = \begin{bmatrix} -(m_1 + m_2)gL_1\sin\theta_1 - m_2gL_2\sin(\theta_1 + \theta_2) \\ -m_2L_2g\sin(\theta_1 + \theta_2) \end{bmatrix}.$$

### 4 Design and optimize the fuzzy system for a two-link manipulator

In this section, the design and optimization of the fuzzy control system for the two-link robot manipulator is implemented. The block diagram of the design control system is presented in Fig. 2 where two fuzzy controllers are used for controlling the movement of the joints in the separate loops to ensure the robot can reach the desired position. The design protocol and optimization of the proposed control system is divided into 2 steps as follows:

- Step 1: Design of the fuzzy controllers based on relation models and the Lyapunov stability analysis. The rules and the parameters of the relation matrixes ( $R_1, R_2$ ) of the fuzzy controllers are designed and selected to minimize the number of the input and output linguistic variables, and their membership functions.
- Step 2: Optimize the designed fuzzy controllers using the SFS algorithm where the performance of the designed system can be implemented by adjusting the input scale factor of the fuzzy controllers and found the optimal parameters.

#### 4.1 Fuzzy controllers design

Fuzzy control systems are developed based on rule-based or knowledge-based. Fuzzy control systems are designed by fuzzy inferences using linguistic variables to depict the state of the system with given conditions. In fuzzy systems, linguistic variables use the MFs to dene the degree of measurement belong to a particular state of the linguistic variable. The fuzzy rule base is a set of fuzzy inferences “IF-THEN”, where the  $i$ th rule

( $L_i$ ) can be represented as follows [48]:

$$L_i : \text{If } X_1 \text{ is } A_1^i, X_2 \text{ is } A_2^i \dots \text{and } X_n \text{ is } A_n^i$$

$$\text{then } Y \text{ is } G_c^i$$

$$i = 1, 2, \dots, N \tag{12}$$

where  $X = [X_1, \dots, X_n]^T$  and  $Y$  are the input and output linguistic variables,  $A_j^i$  and  $G_c^i$  are fuzzy sets.

To design the fuzzy controllers, a single-input 2th-order nonlinear system for each joint of the robot is considered as the following form:

$$\dot{x} = f(x) + b(x)u \tag{13}$$

where  $x = [x_1, x_2]^T$  is state vector;  $f(x) = [f_1(x), f_2(x)]^T$ ,  $b(x) = [b_1(x), b_2(x)]^T$  are functions which present the dynamics of the plant;  $u$  is the control signal from the design fuzzy controller.

To ensure the stability of the control system, the Lyapunov stability theory will be used for designing the fuzzy controllers. Each fuzzy controller is designed with two inputs, and we assume that the control object may have really two degrees of freedom referred to as  $x_1$  and  $x_2 = \dot{x}_1$ , respectively; the states  $x_1$  and  $x_2$ , are the only measurable variables; the 2th-order derivation of error  $\dot{x}_2$  is proportional to the control signal  $u$  ( $u$  increases when  $\dot{x}_2$  increases, and vice versa); the initial conditions  $x(0) = (x_1(0), x_2(0))^T$  belong to the set  $\mathfrak{N} = \{x \in \mathfrak{R} : \|x - x_d\| \leq \varepsilon\}$ , with  $x_d$  is the desired equilibrium point. The fuzzy controller is designed using the rule-based to generate the control signal which ensure the stability of the proposed system. The Lyapunov function is defined as follow:

$$V(x_1, x_2) = \frac{1}{2} (x_1^2 + x_2^2) \tag{14}$$

The derivative of  $V(x_1, x_2)$  can be written:

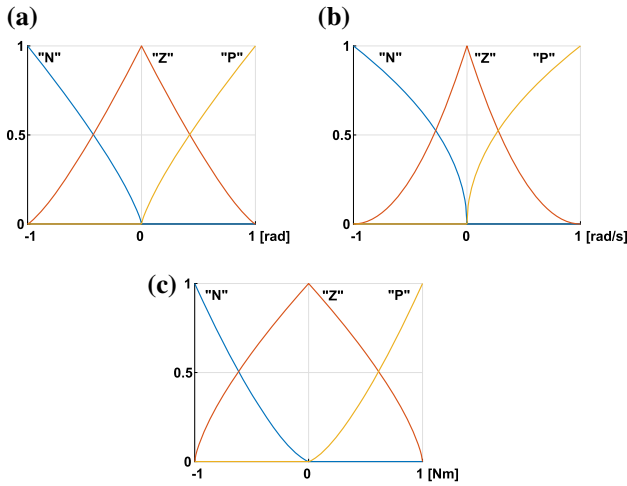
$$\dot{V}(x_1, x_2) = x_1\dot{x}_1 + x_2\dot{x}_2 = x_1x_2 + x_2\dot{x}_2 \tag{15}$$

The system must be guaranteed stabilization at the desired equilibrium point  $x_d = (x_1^d, x_2^d)^T$ , so:

$$\dot{V}(x_1, x_2) = x_1x_2 + x_2\dot{x}_2 < 0 \tag{16}$$

The inequality (16) will be satisfied with sufficient conditions which can be written as: If  $x_1$  and  $x_2$  have opposite signs, then  $x_1x_2 < 0$  and (16) will hold if  $\dot{x}_2 = 0$ ; if  $x_1$  and  $x_2$  are positive, then (16) will hold if  $\dot{x}_2 < -x_1$ ; if  $x_1$  and  $x_2$  are negative, then (16) will hold if  $\dot{x}_2 > -x_1$ . On the other hand, we can notice that  $\dot{x}_2$  is proportional to  $u$ , so these conditions can be translated into the following fuzzy rule-base for the stabilizing controller:

1. If  $x_1$  is negative and  $x_2$  is negative then  $u$  must be positive big;



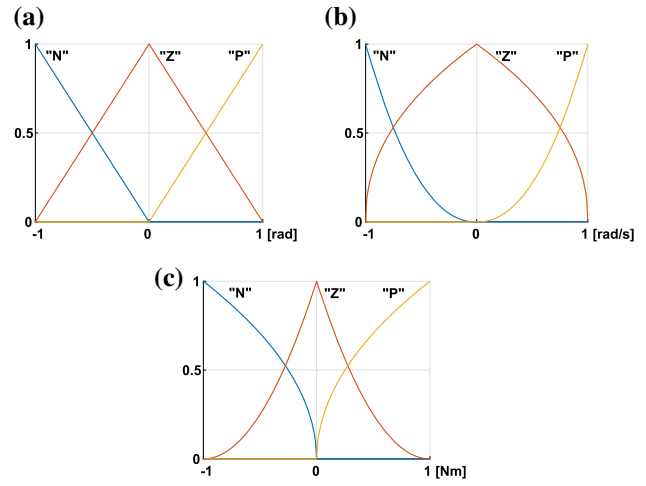
**Fig. 3** The MFs for the input linguistic variables (a), (b) and the output linguistic variables (c) of the fuzzy controller 1, respectively

2. If  $x_1$  is negative and  $x_2$  is zero then  $u$  must be zero;
3. If  $x_1$  is negative and  $x_2$  is positive then  $u$  must be negative;
4. If  $x_1$  is zero and  $x_2$  is negative then  $u$  must be positive;
5. If  $x_1$  is zero and  $x_2$  is zero then  $u$  must be zero;
6. If  $x_1$  is zero and  $x_2$  is positive then  $u$  must be negative;
7. If  $x_1$  is positive and  $x_2$  is negative then  $u$  must be positive;
8. If  $x_1$  is positive and  $x_2$  is zero then  $u$  must be zero;
9. If  $x_1$  is positive and  $x_2$  is positive then  $u$  must be negative big;

The fuzzy controller for the joints of the robot are designed by using relation models with two inputs and an output that are presented by error, rate of the error, and the control signal, respectively. The scale factors for the fuzzy controllers is designed by trial and error method using experimental. The MFs for the linguistic variables of the fuzzy controllers are presented in Figs. 3 and 4, respectively.

The control rule of the designed fuzzy controllers for the joints of the robot is presented by the relation matrix  $R_1$  and  $R_2$ , respectively. The parameters of the relation matrix  $R_1$  and  $R_2$  are selected to satisfy the fuzzy rule-base with the stabilizing conditions which are shown above [49–51]:

$$R_1 = \begin{matrix} & \begin{matrix} N & Z & P \end{matrix} \\ \begin{matrix} N \\ N \\ N \\ Z \\ Z \\ P \\ P \\ P \end{matrix} & \begin{bmatrix} 0 & 0.1 & 0.9 \\ 0 & 1 & 0 \\ 0.65 & 0.35 & 0 \\ 0 & 0.15 & 0.85 \\ 0 & 1 & 0 \\ 0.85 & 0.15 & 0 \\ 0 & 0.35 & 0.65 \\ 0 & 1 & 0 \\ 0.9 & 0.1 & 0 \end{bmatrix} \end{matrix} \quad (17)$$



**Fig. 4** The MFs for the input linguistic variables (a), (b) and the output linguistic variables (c) of the fuzzy controller 2, respectively

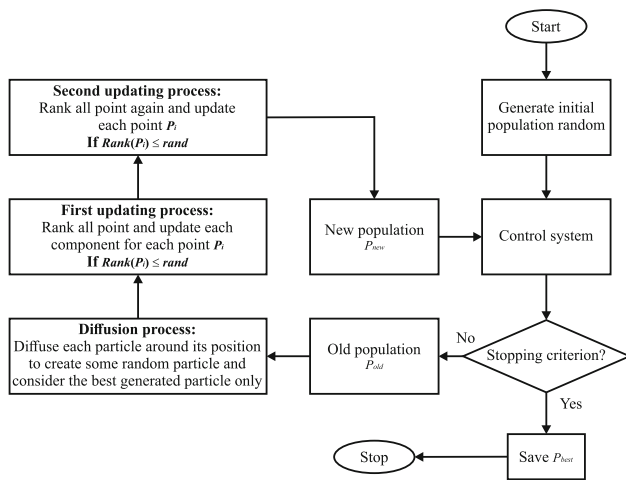
$$R_2 = \begin{matrix} & \begin{matrix} N & Z & P \end{matrix} \\ \begin{matrix} N \\ N \\ N \\ Z \\ Z \\ P \\ P \\ P \end{matrix} & \begin{bmatrix} 0 & 0.02 & 0.98 \\ 0 & 1 & 0 \\ 0.7 & 0.3 & 0 \\ 0 & 0.18 & 0.82 \\ 0 & 1 & 0 \\ 0.82 & 0.18 & 0 \\ 0 & 0.7 & 0.3 \\ 0 & 1 & 0 \\ 0.98 & 0.02 & 0 \end{bmatrix} \end{matrix} \quad (18)$$

### 4.2 SFS optimization

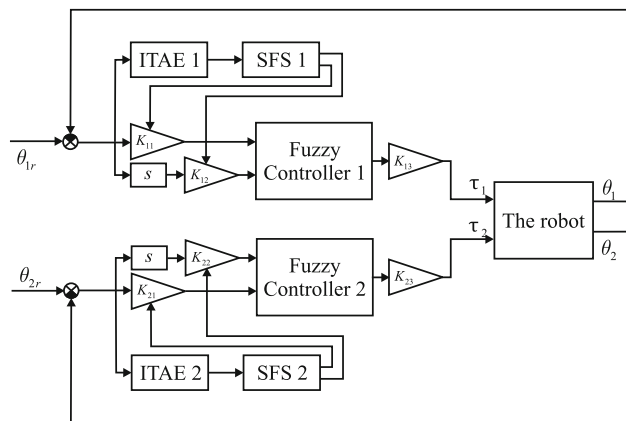
The proposed fuzzy controllers with two degrees of freedom will be optimized using SFS based on tuning the input scale factors. The optimization will be implemented and tested through simulation studies, to control the motion of the robot. The block diagram of the control system using SFS method is illustrated in Fig. 5.

The optimization process is similar for both fuzzy controllers. To use the SFS method, the input scale parameters of the fuzzy controllers are first encoded into a particle such that  $P_i = [K_{i1}, K_{i2}]$  ( $i = 1, 2$ ) where each member is represented by a real number. Then, an objective function is defined, and through the SFS phases, the particles are optimized by minimizing the value of this objective function. The objective function is critical for optimization and selecting this objective function should be realized depending upon the problem nature to obtain the fuzzy controllers with their parameters optimized as desired. In this paper, the ITAE formula is adopted as objective function  $J$  which is given by:

$$J = ITAE = \int_0^{t_{sim}} t|e(t)| dt \quad (19)$$



**Fig. 5** The block diagram of the optimization process using the SFS method for the fuzzy controllers



**Fig. 6** SFS implementation applied for optimizing the designed fuzzy system

In the proposed approach, a random population of 20 particles is selected for the initial population. By using digital simulation in Matlab environment with simulation time of  $t_{sim} = 1(s)$ , each particle is fed into the control system to obtain the response of the joints. Next, fitness values are calculated for each particle by using (19), and the particles are modified by optimization module using Algorithm 1 for the next iteration. The execution is repeated until a specified number of iterations are reached. After finishing the optimization process, the best particle that corresponds to a minimum value of ITAE will be received. The comprehensive flowchart of SFS implementation for optimizing the designed fuzzy control system is shown in Fig. 6.

The SFS algorithm will be implemented to tune the input scale factor of the fuzzy controllers simultaneously. The SFS parameters used in the implementation are shown in Table 1.

**Table 1** SFS parameters

Parameters	Value
$LB$	- 5
$UB$	5
$M_i$	20
$D$	1
$G$	250
$GW_1$	1.5
Search space dimension	2

### 5 Main results

To confirm the efficacy of the designed control system, the simulation will be implemented and the results discussed in conjunction with the GA-PID controller [52] and MNNA-PID controller [53], where the search space for the input scale factors and the system parameters are all assumed the same. The nominal parameters of the 2-DOF robot manipulator in the simulations are  $m_1 = m_2 = 0.1$  [kg],  $L_1 = 0.8$  [m],  $L_2 = 0.4$  [m], and  $g = 9.81$  [m/s<sup>2</sup>] [52, 53]. Various test scenarios are implemented to affirm the efficiency and robustness of the proposed optimal fuzzy system. These scenarios are arranged with the designed parameters which are shown in Table 2 to check with desired reference angles and trajectory experiments for the position of each joint.

In the first simulation, a reference angle is applied for each joint at system nominal parameters. The angular error of each joint of the robot manipulator to follow a reference angle is presented in Figs. 7 and 8. Torque for the joint of the robot manipulator is shown in Figs. 9 and 10. We can see that the designed optimal fuzzy system is more stable than other controllers.

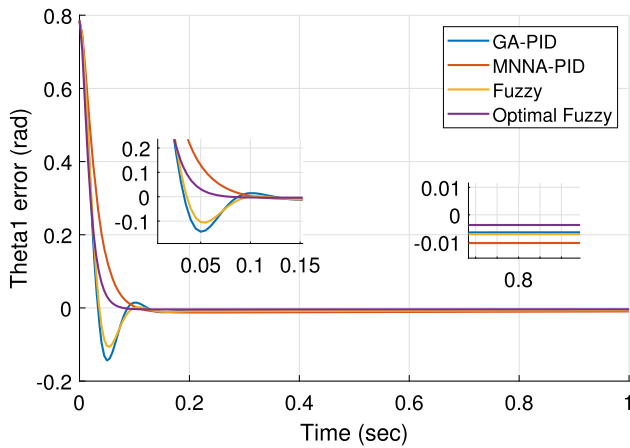
In the second simulation, the performance of the control systems with a reference trajectory will be investigated. Straight-line from point  $A(0, 1.2)$  to  $B(0.3, 0.7)$  will be considered for the desired trajectory of the robot. A cubic position trajectory and a parabolic velocity will be applied for the motion on links of the robot which is clarified in Figs. 11 and 12.

The output response of the links achieved by the different controllers in the situation of the cubic position trajectory test is shown in Figs. 13 and 14 where the efficiency of the optimal fuzzy control system is clearly demonstrated, the proposed algorithm can control links of the robot follow the cubic position trajectory more effectively.

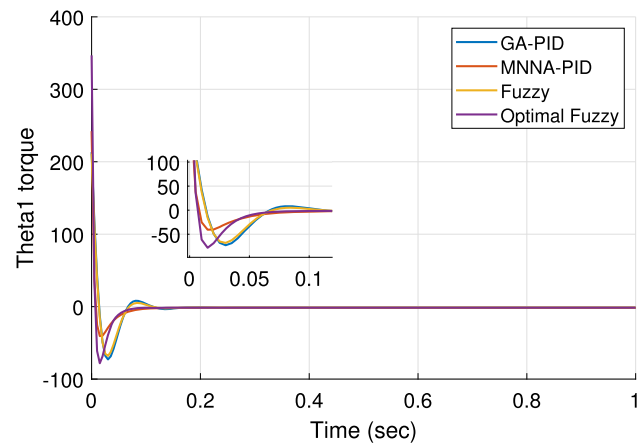
The simulation scenarios showed that the optimal fuzzy controllers can offers better response curves in terms of minimum settling time, negligible overshoots and stability. The designed fuzzy control system can also follow the reference trajectory with fast response and negligible steady-state error compared with other techniques such as GA-PID or MNNA-PID.

**Table 2** SFS parameters

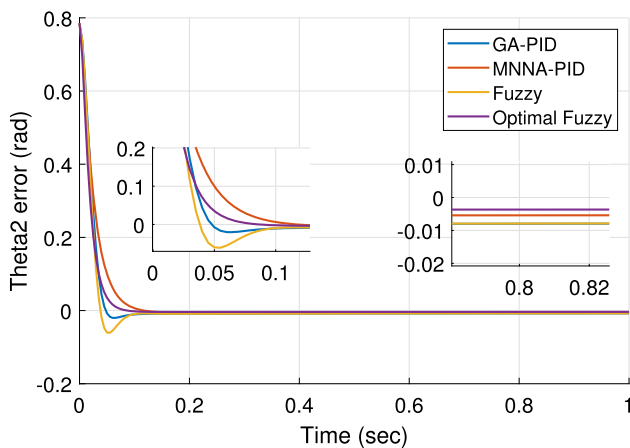
	GA-PID	MNNA-PID	Fuzzy	Optimal fuzzy
Link-1	$K_P = 184.76$ $K_I = 49.68$ $K_D = 8.94$	$K_P = 250$ $K_I = 0.1792$ $K_D = 11.820$	$K_{11} = 1$ $K_{12} = 2$ $K_{13} = 600$	$K_{11} = 1.252$ $K_{12} = 1.87$ $K_{13} = 600$
Link-2	$K_P = 11.46$ $K_I = 16.54$ $K_D = 0.2$	$K_P = 244.524$ $K_I = 0$ $K_D = 6.306$	$K_{11} = 1$ $K_{12} = 3.5$ $K_{13} = 250$	$K_{11} = 2.82$ $K_{12} = 3.276$ $K_{13} = 250$
ITAE	1.1758	0.04196	0.9018	0.0273



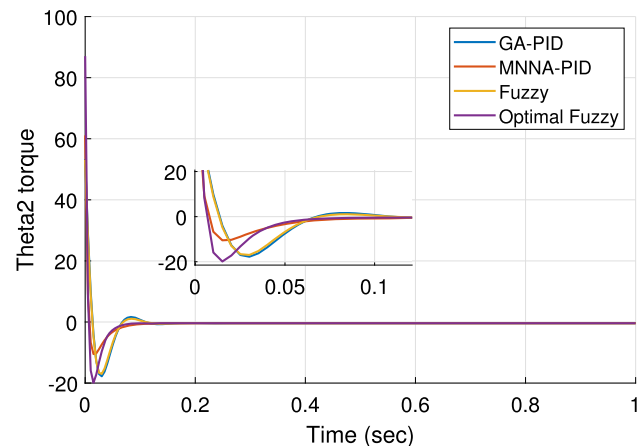
**Fig. 7** The response curves of the position error for the angle  $\theta_1$



**Fig. 9** Torque of the angle  $\theta_1$



**Fig. 8** The response curves of the position error for the angle  $\theta_2$



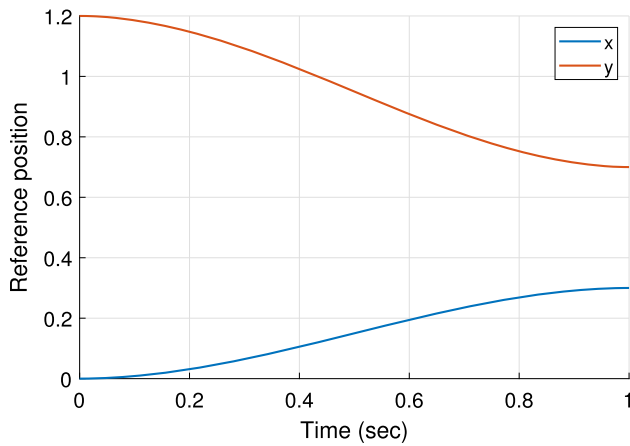
**Fig. 10** Torque of the angle  $\theta_2$

## 6 Conclusion

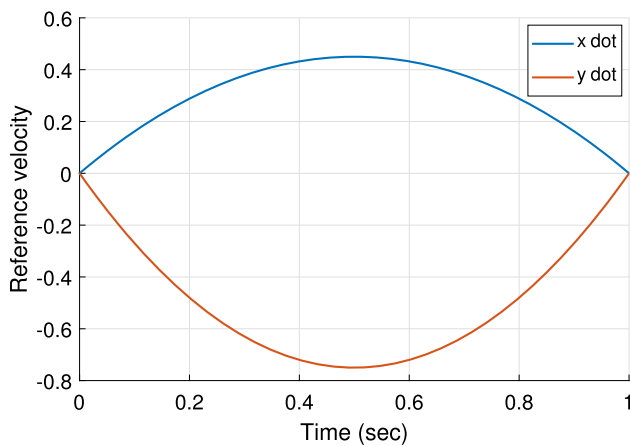
In the paper, an intelligence system using fuzzy controllers based on relation model has been proposed for a two-link robot manipulator. Besides, the SFS method is introduced to optimize the input scale factors of the designed fuzzy controllers by minimizing integral of time-weighted absolute error in control signal. The optimal fuzzy control system is successfully tested in simulation environment to effectively control a non-

linear, multi-input and multi-output, complex and coupled two-link robot manipulator. The performance of the proposed fuzzy control system is also compared with some previous researches such as GA-PID method or MNNA-PID method. The simulation scenarios have been verified the advantages of the proposed fuzzy control system in terms of minimum settling time, negligible overshoots and stability. The proposed fuzzy control system can control the two-link robot manipulator to follow the cubic position trajectory more effectively with fast response and negligible steady-state error.

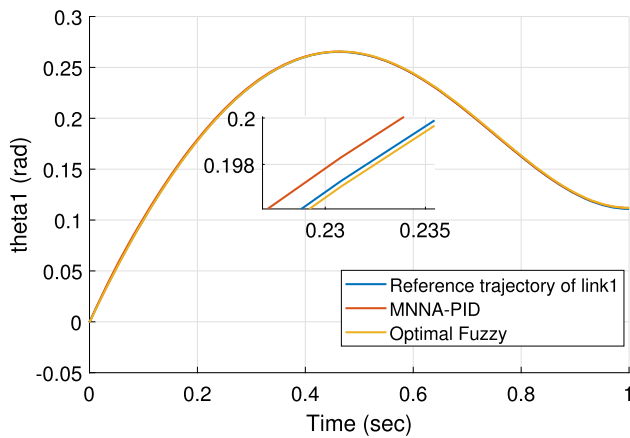




**Fig. 11** Reference cubic position trajectory

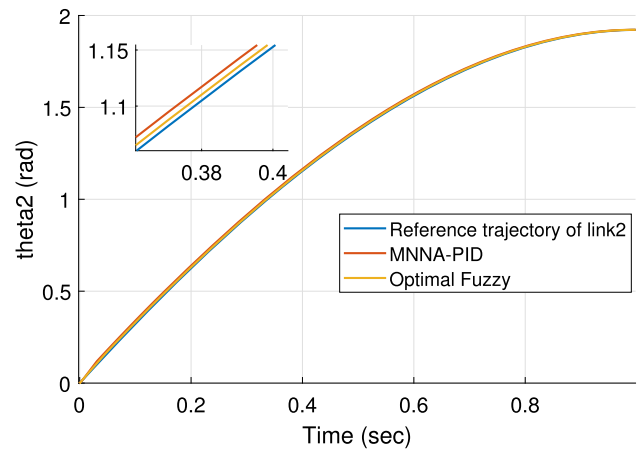


**Fig. 12** Reference parabolic velocity



**Fig. 13** The output response of link-1 for cubic position trajectory

In the future, received simulation results are proposed to be experimentally verified in real-time systems. Besides, the design of the fuzzy system is implemented based on experiences and knowledge of experts, not require a mathematical model of control plants. However, designing and optimizing several fuzzy con-



**Fig. 14** The output response of link-2 for cubic position trajectory

trollers together increase the degree of nonlinearity and complexity in the optimization process, and selecting optimal parameters becomes challenging. These problems will be also our future research directions.

**Declarations**

**Conflict of interest** The authors declare that there is no conflict of interest regarding the publication of this article.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**References**

1. M. Pogliani et al., Security of controlled manufacturing systems in the connected factory: the case of industrial robots. *J. Comput. Virol. Hack. Techn.* **15**(3), 161–175 (2019)
2. V. Kumar, K.P.S. Rana, D. Kler, Efficient control of a 3-link planar rigid manipulator using self-regulated fractional-order fuzzy PID controller. *Appl. Soft Comput.* **82**, 105531 (2019)
3. J. Lee, P.H. Chang, M. Jin, Adaptive integral sliding mode control with time-delay estimation for robot manipulators. *IEEE Trans. Ind. Electron.* **64**(8), 6796–6804 (2017)
4. M. Jin et al., Robust control of robot manipulators using inclusive and enhanced time delay control. *IEEE/ASME Trans. Mechatron.* **22**(5), 2141–2152 (2017)
5. F.J. Torres et al., Master-slave synchronization of robot manipulators driven by induction motors. *IEEE Latin Am. Trans.* **14**(9), 3986–3991 (2016)
6. Y. Su, Comments on a new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans. Ind. Electron.* **67**(8), 7116–7120 (2020)
7. J.E. Lavín-Delgado et al., Trajectory tracking control based on non-singular fractional derivatives for the PUMA 560 robot arm. *Multibody Syst. Dyn.* **50**(3), 259–303 (2020)

8. A. Coronel-Escamilla et al., On the trajectory tracking control for an SCARA robot manipulator in a fractional model driven by induction motors with PSO tuning. *Multibody Syst. Dyn.* **43**(3), 257–277 (2018)
9. C. Tsai et al., A hybrid switched reactive-based visual servo control of 5-DOF robot manipulators for pick-and-place tasks. *IEEE Syst. J.* **9**(1), 119–130 (2015)
10. K. Lochan, B.K. Roy, in *Control of Two-link 2-DOF Robot Manipulator Using Fuzzy Logic Techniques: A Review*. Proceedings of Fourth International Conference on Soft Computing for Problem Solving. (Springer India, New Delhi, 2015)
11. H. Hizarci, S. İkizoğlu, in *Position Control of Flexible Manipulator using PSO-tuned PID Controller*. 2019 Innovations in Intelligent Systems and Applications Conference (ASYU) (2019)
12. I. Cervantes, J. Alvarez-Ramirez, On the PID tracking control of robot manipulators. *Syst. Control Lett.* **42**(1), 37–46 (2001)
13. J. Lee et al., An adaptive PID control for robot manipulators under substantial payload variations. *IEEE Access* **8**, 162261–162270 (2020)
14. A. Kumar, V. Kumar, Evolving an interval type-2 fuzzy PID controller for the redundant robotic manipulator. *Exp. Syst. Appl.* **73**, 161–177 (2017)
15. M. Jin et al., Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control. *IEEE Trans. Ind. Electron.* **56**(9), 3593–3601 (2009)
16. S. Wen et al., The study of model predictive control algorithm based on the force/position control scheme of the 5-DOF redundant actuation parallel robot. *Robot. Auton. Syst.* **79**, 12–25 (2016)
17. N. Nikdel et al., Fractional-order adaptive backstepping control of robotic manipulators in the presence of model uncertainties and external disturbances. *IEEE Trans. Ind. Electron.* **63**(10), 6249–6256 (2016)
18. M. Van, M. Mavrouniotis, S.S. Ge, An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. *IEEE Trans. Syst. Man Cybern. Syst.* **49**(7), 1448–1458 (2019)
19. T.X. Dinh et al., Disturbance observer based finite time trajectory tracking control for a 3 DOF hydraulic manipulator including actuator dynamics. *IEEE Access* **6**, 36798–36809 (2018)
20. H. Liu et al., Finite-time  $H_\infty$  control for high-precision tracking in robotic manipulators using backstepping control. *IEEE Trans. Ind. Electron.* **63**(9), 5501–5513 (2016)
21. C. Yang et al., Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics. *IEEE Trans. Fuzzy Syst.* **27**(3), 574–588 (2019)
22. H. Chhabra et al., Robust nonlinear fractional order fuzzy PD plus fuzzy I controller applied to robotic manipulator. *Neural Comput. Appl.* **32**(7), 2055–2079 (2020)
23. M. Veysi, M.R. Soltanpour, Voltage-base control of robot manipulator using adaptive fuzzy sliding mode control. *Int. J. Fuzzy Syst.* **19**(5), 1430–1443 (2017)
24. H.Q. Thinh Ngo, J.-H. Shin, W.-H. Kim, Fuzzy sliding mode control for a robot manipulator. *Artif. Life Robot.* **13**(1), 124–128 (2008)
25. M. Wan, Q. Tian, M. Wang, Adaptive sliding mode control of robot manipulator based on second order approximation accuracy and decomposed fuzzy compensator. *Wirel. Pers. Commun.* **103**(2), 1207–1218 (2018)
26. T.A. Mai, et al., in *Fuzzy-PID Controller for Two Wheels Balancing Robot Based on STM32 Microcontroller*. 2019 International Conference on Engineering Technologies and Computer Science (EnT) (2019)
27. T.A. Mai et al., A combined backstepping and adaptive fuzzy PID approach for trajectory tracking of autonomous mobile robots. *J. Braz. Soc. Mech. Sci. Eng.* **43**(3), 156 (2021)
28. D.N. Anisimov et al., Design and implementation of fuzzy-PD controller based on relation models: a cross-entropy optimization approach. *Eur. Phys. J. Spec. Topics* **226**(10), 2393–2406 (2017)
29. H. Salimi, Stochastic fractal search: a powerful meta heuristic algorithm. *Knowl. Based Syst.* **75**, 1–18 (2015)
30. T.A. Rahman, A. As'array, N.A.A. Jalil, Active vibration control of a flexible beam structure using chaotic fractal search algorithm. *Proc. Eng.* **170**, 299–306 (2017)
31. R. Sivalingam, S. Chinnamuthu, S.S. Dash, A hybrid stochastic fractal search and local unimodal sampling based multistage PDF plus (1+PI) controller for automatic generation control of power systems. *J. Frankl. Inst.* **354**, 4762–4783 (2017)
32. A. Padhy, S. Panda, A hybrid stochastic fractal search and pattern search technique based cascade PI-PD controller for automatic generation control of multi-source power systems in presence of plug in electric vehicles. *CAAI Trans. Intell. Technol.* **2**, 12–25 (2017)
33. T.A.Z. Rahman, N.A.A. Jalil, R.M.K.R. Ahmad, Chaotic Fractal Search Algorithm for Global Optimization with Application to Control Design, in *IEEE Symposium on Computer Applications and Industrial Electronics; April 24–25*. (Langkawi, Malaysia, 2017), pp. 111–116
34. D. Saha, L.C. Saikia, Performance of FACTS and energy storage devices in a multi area wind-hydro-thermal system employed with SFS optimized I-PDF controller. *J. Renew. Sustain. Energy* **9**(2), 024103 (2017)
35. S. Banerjee, D. Easwaramoorthy, A. Gowrisankar, Fractal Functions, Dimensions and Signal Analysis (2021)
36. N.A.A. Fataf, A. Gowrisankar, S. Banerjee, In search of self-similar chaotic attractors based on fractal function with variable scaling approximately. *Physica Scripta* **95**(7), 075206 (2020)
37. S. Banerjee, S. Mukherjee, M.K. Hassan, A. Gowrisankar, *Fractal Patterns in Nonlinear Dynamics and Applications*, 1st edn. (CRC Press, 2020). <https://doi.org/10.1201/9781315151564>
38. D. Liu et al., Fractal simulation of flocculation processes using a diffusion-limited aggregation model. *Fractal Fract.* **1**(1), 12 (2017)
39. Q. Xia, D. Unger, Diffusion-limited aggregation driven by optimal transportation. *Fractals* **18**(02), 247–253 (2010)
40. H. Salimi, Stochastic fractal search: a powerful meta-heuristic algorithm. *Knowl Based Syst* **75**, 1–18 (2015)

41. N.H. Awad, et al., in *Differential evolution with stochastic fractal search algorithm for global numerical optimization*. 2016 IEEE Congress on Evolutionary Computation (CEC) (2016)
42. P.J. Gaidhane et al., Design of interval type-2 fuzzy precompensated PID controller applied to two-DOF robotic manipulator with variable payload. *ISA Trans.* **89**, 169–185 (2019)
43. X.-J. Liu, Q.-M. Wang, J. Wang, Kinematics, dynamics and dimensional synthesis of a novel 2-DoF translational manipulator. *J. Intell. Robot. Syst.* **41**(4), 205–224 (2005)
44. G. Zhang, D. Cheng, Adaptive fault-tolerant guaranteed performance control for euler-lagrange systems with its application to a 2-link robotic manipulator. *IEEE Access* **8**, 184160–184171 (2020)
45. A.O. Pizarro-Lerma et al., Experimental evaluation of a sectorial fuzzy controller plus adaptive neural network compensation applied to a 2-DOF robot manipulator \*\* This work was supported in part by Programa para el Desarrollo Profesional Docente (PRODEP-Mexico) under grant ITSON-126 and Tecnológico Nacional de Mexico (TecNM) projects. *IFAC-PapersOnLine* **52**(29), 233–238 (2019)
46. X. Wang, B. Hou, Trajectory tracking control of a 2-DOF manipulator using computed torque control combined with an implicit lyapunov function method. *J. Mech. Sci. Technol.* **32**(6), 2803–2816 (2018)
47. X. Dong et al., Geometric approach for kinematic analysis of a class of 2-DOF rotational parallel manipulators. *Chin. J. Mech. Eng.* **25**(2), 241–247 (2012)
48. A. Sanz, V. Etxebarria, Experimental control of a two-Dof flexible robot manipulator by optimal and sliding methods. *J. Intell. Robot. Syst.* **46**(2), 95–110 (2006)
49. T.A. Mai et al., Development of a microcontroller-based adaptive fuzzy controller for a two-wheeled self-balancing robot. *Microsyst. Technol.* **24**(9), 3677–3687 (2018)
50. D.N. Anisimov, A. Mai Tkhe, Dynamic properties of the fuzzy control systems based on the relational models. *Mekhatronika Avtomatizatsiya Upravlenie* **18**(5), 298–307 (2017). <https://doi.org/10.17587/mau.18.298-307> (In Russian) <https://doi.org/10.17587/mau.18.298-307>
51. D.N. Anisimov, A. Mai The, A two-level fuzzy system for control of dynamic objects. *MPEI Vestnik* **4**, 101–109 (2017). <https://doi.org/10.24160/1993-6982-2017-4-101-109> (In Russian) <https://doi.org/10.24160/1993-6982-2017-4-101-109>
52. H.V. Hultmann Ayala, Coelho L. dos Santos, Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Exp. Syst. Appl.* **39**(10), 8968–8974 (2012)
53. M. Elsis et al., An improved neural network algorithm to efficiently track various trajectories of robot manipulator arms. *IEEE Access* **9**, 11911–11920 (2021)