Performance of Two-Link Robotic Manipulator Estimated Through the Implementation of Self-Tuned Fuzzy PID Controller

Aditi Saxena, Rishabh Chaturvedi, and Jitendra Kumar

Abstract Robot manipulators become intensified major segments in the production sectors that are utilized in numerous applications such as welding, granulating, mechanical dealing, and assembling because of velocity, accuracy, and repeatability. These applications necessitate a legitimate path plan, appropriate generation of direction, and above all a control design. In a manipulator system, the links are expanded based on efficiency and versatility is upgraded alongside the complexity. This controller tuned to produce the least weighted sum of integral of absolute error and component of positive change in controller output. The controller has the advantage of examining and controlling a highly non-linear and rigid manipulator system. To optimize the system, a genetic algorithm is being used to provide an actual and desired result by incorporating a combination of a genetic algorithm with self-tuning fuzzy. PID is composing the controller much more superior and reliable.

Keywords Self-tuned fuzzy PID · Manipulator · Genetic algorithm

1 Introduction

The term robot is basically a Czech word that means slaved laborers, and it came into existence in 1920. Robots acts like a machine and is dominated by a controller in a robot. Further, the structure of the robot is replicated as similar to human body parts like a head, two arms, and two legs, waist, and so on. Industry-specific robots play out a few tasks, for example, picking and fixing objects, and movement adjusted from seeing how comparable manual tasks are taken care of by a completely working human arm. Such automated arms are otherwise called robotic manipulators. The proposed work aims to have innovation is a multifaceted concept that includes modifications to products, alterations to organizational aspects dependent on innovations, strategies, rebuilding of organizational culture, and revisions to production processes. Accordingly, such optimized controlled robotic manipulator is

A. Saxena (✉) · R. Chaturvedi · J. Kumar
IET Department of Electronics and Communication Engineering, GLA University, Mathura 281406, India
e-mail: aditeesaxenaa29@gmail.com

appropriate for process innovation since they authorize organizations to concentrate on unique processes for the production of products and services. The second factor is the organizations today are confronted with expanding work costs and a deficiency of laborers, and are thus spending in robotics. Robots nevermore demand promotions and can work nonstop. The third factor is the pressure to increase the production rates to compete the market, and the fourth factor is the increased productivity. The fifth factor explains about the repeatability where the robot's drive product quality or consistency and reduces waste. The last factor represents the speed in which the robotics assist in the increased production and hence reduction in the waiting time.

This controller can easily accomplish all these factors, where the controller is used for underwater controlling, etc. The proposed model can move and grip devices with guidance which can be operated through various programmed motions as it is both programmable and multifunctional devices [1]. Robotics is a combination of two branches [2, 3] that are electronics through the field of control, manufacturing, designing, and kinematics which help in positioning and orientation of the manipulator devices having multi-degree liberty can be positioned with the help of these manipulators. High and improved control strategies are required to achieve accuracy in the trajectory tracking of the manipulator, dynamic model of a robotic manipulator also play a very important role; it is responsible for the performance of a robotic manipulator. To achieve the desire performance and response of the system, a new design is needed for an accurate mathematical model, and after this, the various control strategies are applied to get precise trajectory tracking [4]. For industry purpose, the manipulators that are being used are highly non-linear and coupled; a second-order non-linear differential equation is being used to design a dynamic model for two links rigid robotic manipulator [5]. If the number of links increased simultaneously complexity also increased in terms of position and velocity of the manipulator. To overcome this challenging task, a control system is employed to monitor the manipulator [6]. Especially after designing a dynamic model, it is difficult to attach a precise and powerful controller to the plant that could be able to handle the non-linearities and many other unknown parameters. The controller is managed by adopting a tuning algorithm to have flexibility over the automation, where the entire world is searching for automatic processing in every aspect of life. Many different types of the algorithm are available in current trends such as particle swarm optimization (PSO), optimization method of ant colony [7], cuckoo search algorithm (CSA) [8, 9], genetic algorithm (GA), and so on. In the proposed system, genetic algorithm is used to produce high-quality solutions for optimization and search issues by depending on the process of natural selection such as crossover, mutation, crossover, and selection. The main aim is to reduce the amount of error. The controller will operate with the self-tuned fuzzy logic controller to have the least value of error and can also supervise the non-linearities [10–12]. The limitations of the proposed work are that this controller could be able to control a two-link manipulator having a payload of 0.5 kg [13] (Fig. 1 and Table 1).

2   Dynamic Model for Manipulator

In this study, the configurations of the robotic manipulator is explained and dynamic model was designed by studying the equations of motions for manipulator arm which was given by Lin [14],

$$
\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} \theta_{11} \\ \theta_{22} \end{bmatrix} + \begin{bmatrix} P_{11} \\ P_{21} \end{bmatrix} + \begin{bmatrix} f_{r1} \\ f_{r2} \end{bmatrix} + \begin{bmatrix} fn_{1p} \\ fn_{2p} \end{bmatrix} = \begin{bmatrix} \tau f_{1p} \\ \tau f_{2p} \end{bmatrix}
$$

where

$$
\begin{aligned}
S_{11} = {} & I_{1p} + I_{1p} + m_{11}l_{c1}^2 + m_{22}l_{c2}^2 + 2m_{22}l_{11}l_{c2}\cos\theta_{22} \\
& + m_{vp}l_{11}^2 + m_{vp}l_{22}^2 \\
& + 2m_{vp}l_{11}l_{22}\cos\theta_{22}
\end{aligned} \tag{1}
$$

$$
S_{12} = I_{2P} + m_{22}l_{c2}^2 + m_{22}l_{11}l_{c2}\cos\theta_{22} + m_{vp}l_{22}^2 + m_{vp}l_{11}l_{22}\cos\theta_{22} \tag{2}
$$

$$
S_{21} = S_{12} \tag{3}
$$

$$
S_{22} = I_{2P} + m_{22}l_{c2}^2 + m_{vp}l_{22}^2 \tag{4}
$$

$$
P_{11} = -m_{22}l_{11}l_{c2}(2\theta_{11} + \theta_{22})\theta_{22}\mathrm{Sin}\theta_{22} - m_{vp}l_{11}l_{22}(2\theta_{11} + \theta_{22})\theta_{22}\sin\theta_{22} \tag{5}
$$

$$
P_{21} = m_{22}l_{11}\,\theta_{11}^2 l_{c2}\sin\theta_{22} + m_{vp}l_{11}\,\theta_{11}^2\, l_2 \sin\theta_{22} \tag{6}
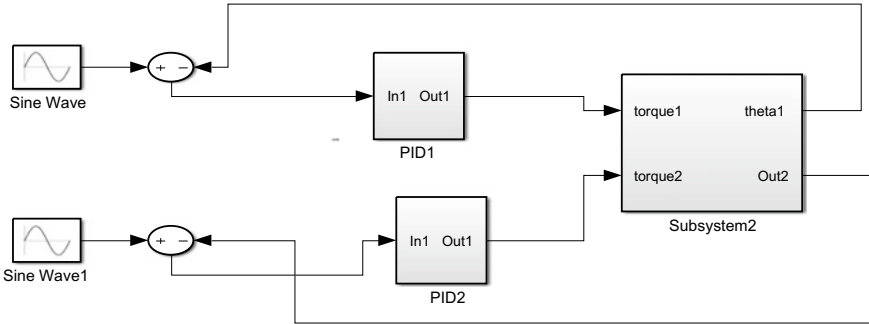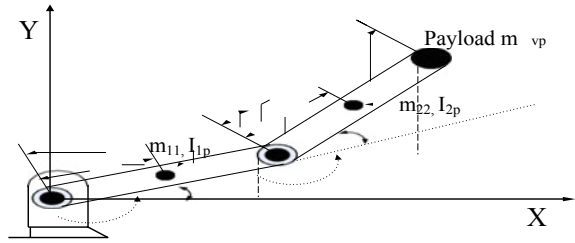$$

$$
f_{r1} = b_{1vp}\,\theta_{11} \tag{7}
$$

$$
f_{r1} = b_{1vp}\,\theta_{22} \tag{8}
$$

$$
\begin{aligned}
fn_{1p} = {} & m_{11}l_{c1}g\cos\theta_{11} + m_{22}g(l_{c2}\cos(\theta_{11} + \theta_{22}) \\
& + l_{11}\cos\theta_{11} + m_{vp}g(l_{22}\cos(\theta_{11} + \theta_{22}) \\
& + l_{11}\cos\theta_{11}\ (9)
\end{aligned} \tag{9}
$$

$$
fn_{2p} = m_{22}l_{c2}g\cos(\theta_{11} + \theta_{22}) + m_{vp}l_2g\cos\,(\theta_{11} + \theta_{22}) \tag{10}
$$

$$
\theta_{11} = \frac{\tau f_{1p} - fn_{1p} - f_{r1} - P_{11} - S_{12} * \theta_{22}}{S_{11}} \tag{11}
$$

Fig. 1 Diagram of two link
robotic manipulator

Fig. 2 PID structure

$$\theta_{22} = \frac{\left(\tau f_{2p} - f_{n2p} - f_{r2} - P_{21} - S_{12} * \theta_{11}\right)}{S_{22}} \tag{12}$$

where θ_{11} and θ_{22} shows the links position; τf_{1p} and τf_{2p} are the generated torque; f_{r1} and f_{r2} are the dynamic friction coefficients; m_{11} and m_{22} shows the masses. l_{11} and l_{22} shows the length, I_{1p}, I_{2P} shows inactivity.

Initially, the mathematical model for each equation is constructed as $S_{11}, S_{12}, S_{21}, S_{22}, P_{11}, P_{21}, f_{r1}, f_{r2}, f n_{1p}, f n_{2p}$. Further, the mathematical model is again converted into a dynamic model for further processing. On the MATLAB SIMULINK, a subsystem for these equations is created and appended with the equations shown in no 11 and 12. This whole mathematical model accommodates various controllers named subsystem 2 that are given in Fig. 2 (Figs. 3, 4, 5 and 6).

2.1 Various Controller Structure

2.1.1 PID Controller Structure

This controller structure is designed using the MATLAB SIMULINK. The design of the simple PID controller is constructed by connecting link 1 and link 2. These
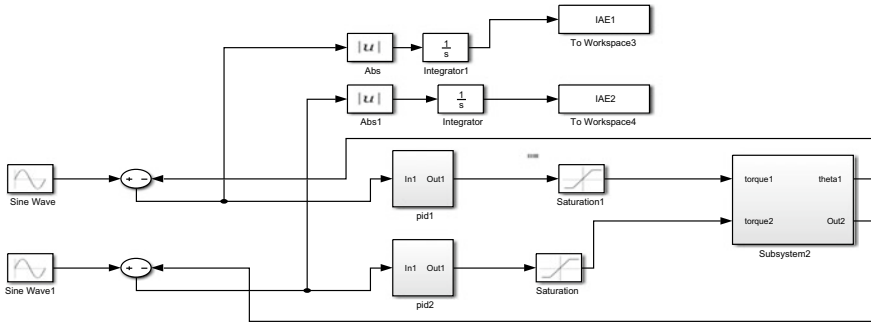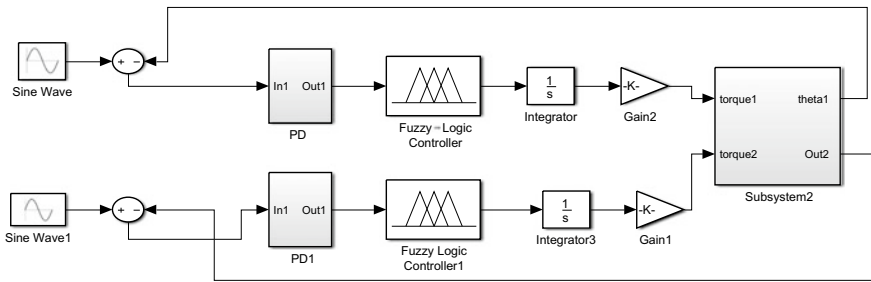
Fig. 3 Optimized PID structure
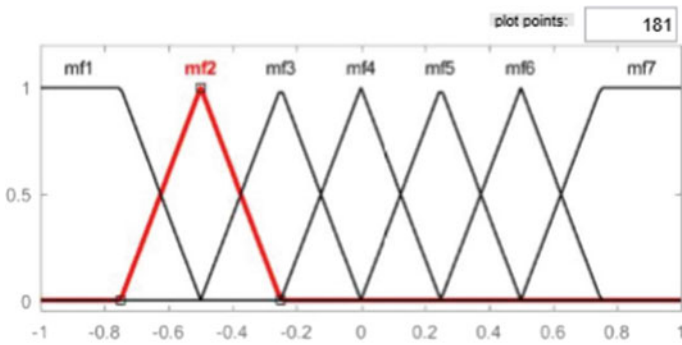

Fig. 4 Fuzzy PD structure

Fig. 5 Membership function plot

links reflect the operations of the equations defined in the mathematical model. Subsequently, a subsystem is being created of the model. The control boundaries of the PID structure are namely, proportional term, integral term, and derivative term. The proportional term refers to a general control activity proportional to the error signal through the steady gain factor. The integral term aims to decrease consistent
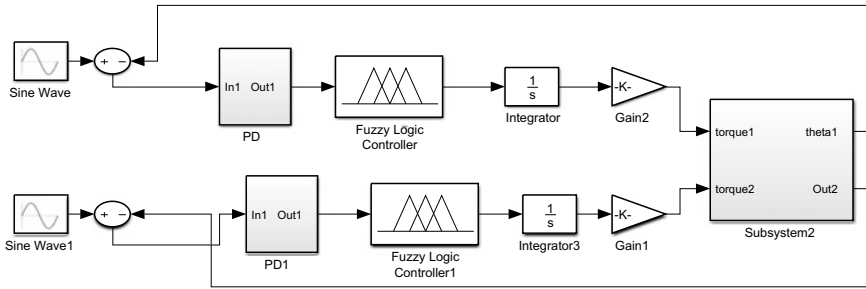
Fig. 6 Fuzzy PI structure

state errors through low-frequency remuneration by an integrator. The derivative term aid to improve transient reaction through high-frequency remuneration by a differentiator. Each link has its own PID controller connected for generating the optimal output. The inputs supplied for each link are in terms of the sine wave, and the controller supports in all sorts of processing, and the final output is matched with the mathematical model equations. If any mismatch occurs between the corresponding desired value and the actual value, then the controller is again fine-tuned towards the gain factor. The genetic algorithm is employed to enhance time and achieve higher precision accuracy.

3 Genetic Algorithm for Optimization

Nature has consistently been an incredible wellspring of motivation for all humankind. The genetic algorithm is associated with search calculations based on the ideas of selection and genetics. In GAs, a pool or a populace of potential clarifications are given for the specified issue. These clarifications at that point go through recombination and transformation (like in genetics), delivering new solutions, and the cycle is rehashed over different circumstances. Every value is specified as a fitness value, and the fitter is given a higher opportunity to yield more fitter. An optimistic approach such as integral absolute error (IAE) is determined to eradicate the number of errors. To operate with the IAE functionality, a small piece of code is written in the command window. The next process is to assign a sim function inside the MATLAB, which is being inbuilt under the optimization tool. By executing the command as optimtool in the command window a pop-up window gets activated, and inside the textbox type the term solver. Finally, run the GA commands with the number of variables associated with the gains. The sequence of steps involved in the genetic algorithm is (1) Population, (2) Fitness scaling, (3) Selection, (4) Reproduction, (5) Mutation, (6) Crossover, and (7) Migration. The population comprises of all the data. The scaling function converts raw fitness scored returned by the fitness function in a given range. The selection function determines parents for next-generation based

on their scaled values. The reproduction essentially demonstrates the functionality of creating children for each new generation then progressed by the mutation function, which makes a small random change in the individual which provides genetic diversity. The crossover consolidates two individuals or parents to form a new individual for the next-generation. The values for each operation are assigned using the two commands such as assignin and sim assignin which are defined inside the workspace. The sim function returns a single simulink (simulation output object) that contains all the simulation output. The obtain results aids to plot the output signal value against the time.

3.1 PID Optimized Controller Structure

The genetic algorithm technique that is discussed in the previous section is being used in this model for optimization. This is the reason for the optimized PID controller design; IAE is being attached to the controller input. The rest of the whole structure is the same as the PID controller structure, where it contains only two blocks, and the initial block consists of absolute, and the other is integrator over time. The IAE integrates absolute error value and provides the least value of error.

3.2 Fuzzy PD Controller Optimized Structure

PID controller is alone insufficient to handle all the non-linearities, so there exists a need for adding the fuzzy logic functionalities. The fuzzy controller or logic consists of membership functions which are described in Fig. 7. The membership functions are being shown each has their respective ranges as fuzzy only accept 0 or 1; so accordingly, the range is being decided with the help of a rule base a fuzzy logic. The design is created by typing fuzzy in the command window, a fuzzy logic appears and it is possible to call this logic to the workspace where our structure has been
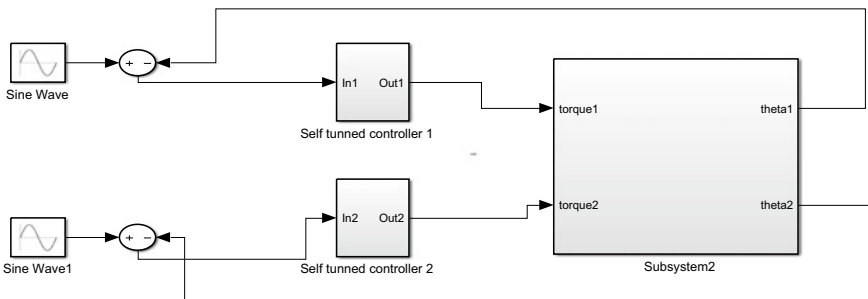

Fig. 7 Optimized self-tuned (FUZZY PID) model in SIMULINK

designed. Further, the fuzzy logic is being combined with PD controller and termed as fuzzy PD controller (Fig. 8).

3.3 Fuzzy PI Optimized Structured

This structure is the same as the above one the main difference here is the PD controller is being integrated which results in PI controller, and fuzzy logic is also being attached between PD and integrator. The whole structure as a sum give origin to fuzzy PI controller for optimization, again genetic algorithm is being used with the help of inbuilt optimtool function.

3.4 Optimized Self-Tuned Fuzzy PID (STFPID) structure

Here, two fuzzy layer are utilized, the output of fuzzy layer 1 is being multiplied by the product of gains and output of fuzzy layer 2, then fed to the plant. The fuzzy logic self-tuning structure is being represented below is termed as a self-tuned fuzzy PID controller (STFPID). This controller gives the lesser error value after optimizing the genetic algorithm.
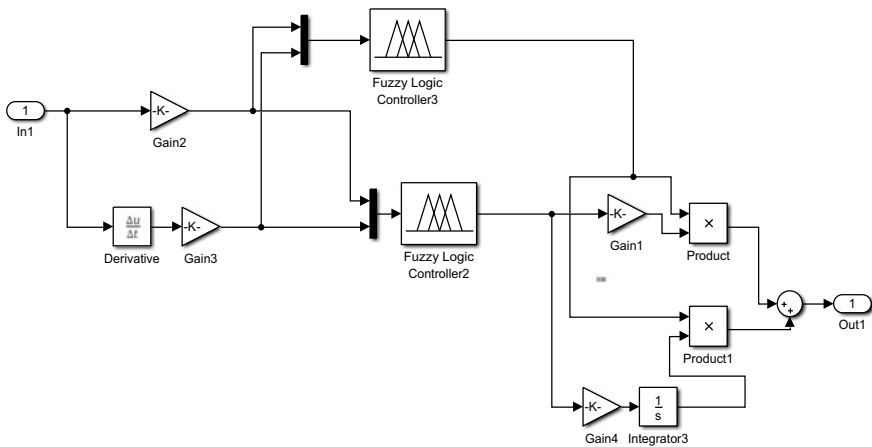

Fig. 8 Structure of self tuned fuzzy PID controller

4 Results and Discussions

For tracking the trajectory, the two-link robotic manipulator and the controller are optimized on its own with the help of a self-tuned fuzzy PID controller. The controller structure as discussed above, these controllers is working efficiently and fulfilling all their aims for which they have been designed. To check all this, a simulator is needed and MATLAB is used to compare each optimized controller. Figure 9 represents the tracking, the reference, and the output wave is perfectly matched. If it does not match, differentiate it between these two waves with the guidance of color-coding. The dynamic model is designed with the equations, and the controllers, which are being attached to it, are perfect and working precisely. Figure 10 describes a self-tuned fuzzy PID controller when operating through the genetic algorithm process as a simulation result, it shows the minimum error value. In comparison to all the other controllers which are being used, this controller contains many features. The self-tuned structure is allowing the controller to automatically tune itself depending upon the presence of non-linearities. The fuzzy has many advantages and can deal with multiple-links non-linearities, increased time, overshoot, and many more. The base
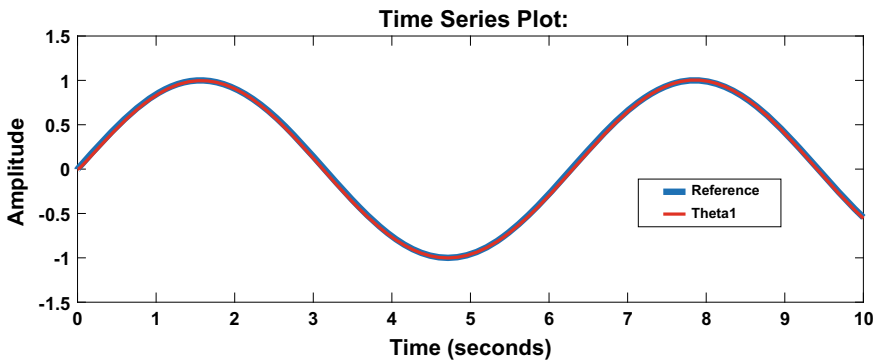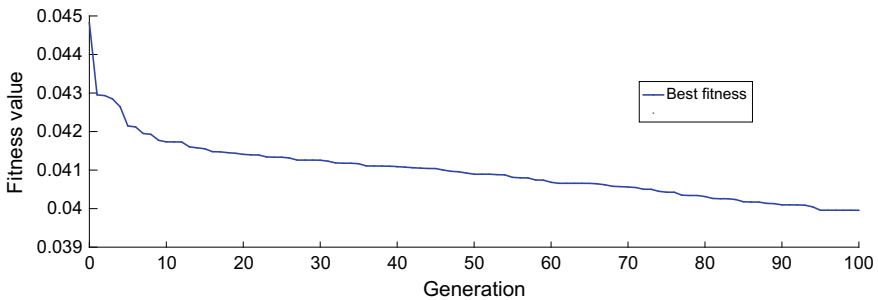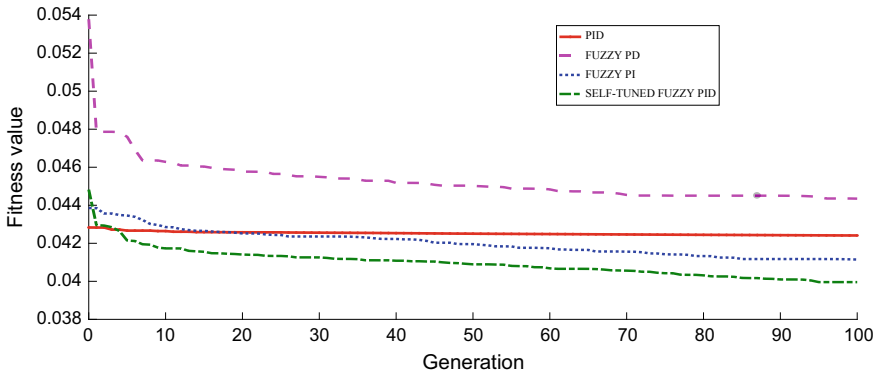
Fig. 9 Representing accurate tracking

Fig. 10 Self-tuned fuzzy PID after optimization

Fig. 11 IAE values of various controllers

for the fuzzy is PID in which all the three control parameters such as proportional, integral, and derivative are combined and serves in reducing error and providing stability to the system. Figure 11 showing the simulation result after working through a genetic algorithm all the five controllers are being individually simulated up to 100 generations. The self-tuned fuzzy PID controller has the least value of error as compared to any other. Table 2 represents the mathematical value of error for each controller in descending order and the self-tuned fuzzy logic has an error value of 0.03665 which is like next to perfection in the third graph all the controller's error. These are identified by separate graphs that are being differentiated with different color coding. This controller can work as an adaptable controller and can self-tuned and optimized itself according to the surrounding parameters and non-linearities.

5 Conclusion

In today's world of automation, the need for developing an adaptable controller is essential. This controller is working as a self-adaptable controller that can be tuned and adjusted on its own with the help of a self-tuned fuzzy PID controller and genetic algorithm. With the optimized simulation results, it is found that this controller has the least value of error when compared to other controllers that are also being designed and simulated. The proposed controller not only helps to control the two link manipulator but also has the application for controlling the underwater robot, to control an adaptable gripper, and many more.

Table 1 Parameters of the dynamic model

Parameters	Link 1	Link 2	Values
m_{11}	0.392924 kg	m_{22}	0.094403 kg
l_{c1}	0.104648 m	l_{c2}	0.081788 m
l_{11}	0.2032 m	l_{22}	0.1524 m
I_{1P}	0.0011411 kg m^2	I_{2P}	0.0020247 kg m^2
b_{1vp}	0.141231 N-m/radian/s	b_{2vp}	0.3530776 N-m/radian/s
m_{vp}	0.56699 kg		
g	/s^2		

Table 2 Tabulated form of IAE value in descending order

Various controller	Values of IAE
PID without optimization	0.2218
PID with optimization	0.0429
Fuzzy PD with optimization	0.04438
Fuzzy PI with optimization	0.04114
Self-tuned Fuzzy PID using GA	0.03665

References

1. Sharma R, Rana KPS, Kumar V (2014) Performance analysis of fractional order fuzzy PID controllers applied to a robotic Manipulator. Expert Syst Appl 41(9):725, 4274–4289:726
2. Kumar V, Rana KPS, Mishra JK, Nair SS (2016) A robust fractional order fuzzy P+ fuzzy I+ fuzzy D controller for nonlinear and uncertain system. Int J Autom Comput 14(4):474–488
3. Kumar V, Rana KPS (2017) Nonlinear adaptive fractional order fuzzy PID control of a 2-link planar rigid manipulator with payload. J Franklin Inst 354:993–1022
4. Kumar J, Kumar V, Rana KPS A fractional order Fuzzy PD+I controller for three link electrically driven rigid robotic manipulator system. J Intell Fuzzy Syst IOS Press, Netherlands (SCI Index, Impact factor-1.261)
5. Weile DS, Michielssen E (1997) Genetic Algorithm Optimization applied to electromagnetics, a review. IEEE Trans Antennas Propag 45(3):343–353
6. Kong Z, Jia W, Zhang G, Wang L (2015) Normal parameter reduction in soft set based on particle swarm optimization algorithm. Appl Math Model 39:4808–4820
7. Ghanbari A, Kazemi SMR, Mehmanpazir F, Nakhostin MM (2013) A cooperative ant colony optimization-genetic algorithm approach for construction of energy demand forecasting knowledge based expert systems. Knowl-Based Syst 39:194–206
8. Jagatheesan K, Anand B, Samanta S, Dey N, Ashour AS, Balas VE (2017) Design of a proportional-integral-derivative controller for an automatic generation control of multi-area power thermal systems using firefly algorithm, IEEE/CAA J Automatica Sinica, 1–14 (2017). https://doi.org/10.1109/JAS.2017.7510436
9. Yang XS, Deb (2009) Cuckoo search via Lévy Flights In: . Proceedings world congress on nature and biologically inspired computing, India, pp 210–214
10. Yang XS, Gandomi AH (2012) Bat-algorithm, a novel approach for global engineering optimization. Eng Comput 29(5):464–483
11. Ohtani Y, Yoshimura (1996) Fuzzy control of a manipulator using the concept of sliding mode. Int J Syst Sci 27(2):179–186

12. Hazzab A, Bousserhane IK, Zerbo M, Sicard P (2006) Real-time implementation of fuzzy gain scheduling of PI controller for induction motor machine control. Neural Process Lett 24:203–215
13. Sharma R, Bhasin S, Gaur P, Joshi D (2019) A switching-based collaborative fractional order fuzzy logic controllers for robotic manipulators. Appl Math Modell
14. Lin F (2007) Robust control design: an optimal control approach. John Wiley & Sons Ltd., England