Improving Redundant Manipulator Control with Genetic Fuzzy Systems

Sathya Karthikeyan, Anirudh Chhabra, and Donghoon Kim

Abstract Redundant manipulators play a significant role in industrial pick-and-place applications, offering enhanced capabilities and flexibility in various tasks. Efficient control strategies are crucial for optimizing the extra degrees of freedom in redundant manipulators to enhance their capabilities during such operations. Hence, to enhance the performance of the proportional-derivative (PD) control framework in closed-loop inverse kinematics (CLIK), this paper proposes augmenting the CLIK control logic with a genetic fuzzy system (GFS) that adaptively tunes the PD gains of the controller. The proposed GFS-aided CLIK controller is compared to the standard CLIK controller through simulation studies. The results demonstrate that the GFS-aided controller achieves a reduction in total control effort while maintaining tracking efficiency in terms of root-mean-square error.

1 Introduction

The advent of Industry 4.0 has brought about significant changes in the field of industrial automation, with the integration of robotics and artificial intelligence at the forefront of this transformation. This integration minimizes human involvement and finds applications across diverse sectors, including manufacturing, logistics, healthcare, and agriculture, with robots serving as integral components within automated systems. Among these, the most prevalent application of robots in manufacturing, warehouse management, and distribution lies in their pick-and-place (PAP) opera-

S. Karthikeyan (✉) · A. Chhabra · D. Kim
University of Cincinnati, Cincinnati, Ohio, USA
e-mail: karthist@mail.uc.edu

A. Chhabra
e-mail: chhabrad@mail.uc.edu

D. Kim
e-mail: Donghoon.Kim@uc.edu

tions. This involves activities such as picking up an object from one location and placing it at a different location for loading and unloading equipment, removing components from a belt, and sorting components.

A controller plays a vital role in improving productivity, reducing errors, and increasing the flexibility of these systems. Throughout history, various control techniques, such as optimal control [8], model-predictive control [2], and robust control [13], have been employed in the industry. Yet, the most widely used controllers for industrial robots are the proportional-integral-derivative (PID) controllers due to their simplistic nature and robust performance [16, 14]. However, the PID controller, being a linear controller, is not well-suited for strongly nonlinear systems like industrial robots, particularly robotic manipulators [18].

A closed-loop inverse kinematics (CLIK) controller, on the other hand, is known to perform better than a PID controller for manipulator control because it can handle the kinematic constraints of the robot. It can account for the nonlinearities and inter-joint dependencies in the robot, which can lead to increased accuracy and efficiency in control. However, in CLIK, the control gains are not adaptive to changing environments and therefore can perform poorly. The use of fuzzy logic to tune the controller gains in PAP operations can be seen in the literature. Khoury et al. [6] implemented a fuzzy-PID controller in a 5-degrees of freedom (DOF) industrial robot and concluded its higher efficiency in trajectory tracking. However, the study uses manual selection of rules to achieve acceptable performance.

Thus, this work proposes the implementation of a genetic fuzzy system (GFS) to enhance the tracking accuracy of a 3-DOF robotic arm as it follows a predefined two-dimensional (2-D) trajectory within the arm's workspace. The primary aim is to minimize the necessary control effort without compromising the tracking accuracy.

The paper is structured as follows. In Sect. 2, the kinematics and dynamics of the robot are obtained, and the CLIK control framework is defined. Further, the importance of GFSs is discussed. In Sect. 3, the GFS is modeled, and the proposed GFS-aided control technique is designed. In Sect. 4, a simulation setup is created, and the proposed controller is trained and tested. Further, the results are analyzed and discussed. Finally, concluding remarks are provided in Sect. 5.

2 Preliminaries

2.1 Problem Formulation

The robotic arm is assumed to have a fixed base and is constructed as a combination of thin and light rods representing the links and point masses at the joint locations. For simplicity, accurate tracking of a predefined trajectory from a starting point (the picking location) to an endpoint (the placing location) within the robotic arm's workspace is considered to be a PAP operation. It is also assumed that the target is a solid, rigid object, and its orientation is as desired when defining a reasonable trajectory.

2.2 System Modelling

The robotic arm is planar and consists of 3 links each possessing a single DOF as shown in Fig. 1. The arm is attached to a fixed base which is referred to as body 0. The three links are referred to as bodies 1, 2, and 3 respectively, and all bodies are represented in the inertial frame (N). Based on this, the position and velocity vectors

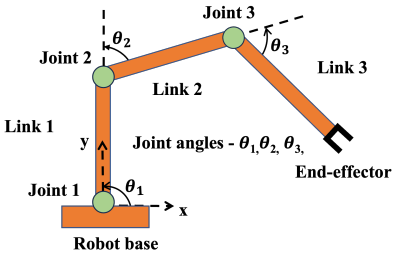


Fig. 1: A 3-DOF robot arm presenting a kinematic redundancy in a 2-D workspace

of the three bodies are defined in N-frame as follows:

$$\boldsymbol{r}_i = \boldsymbol{r}_{i-1} + \left[l_i \cos\left(\sum_{j=1}^{i} \boldsymbol{\theta}_j \right) \; l_i \sin\left(\sum_{j=1}^{i} \boldsymbol{\theta}_j \right) \right]^{\mathrm{T}}, \tag{1}$$

$$\boldsymbol{v}_i = \boldsymbol{v}_{i-1} + \left[-l_i \sin\left(\sum_{j=1}^{i} \theta_j \right) \dot{\theta}_i, \; l_i \cos\left(\sum_{j=1}^{i} \theta_j \right) \dot{\theta}_i \right]^{\mathrm{T}}, \tag{2}$$

where $\boldsymbol{r}_i$ and $\boldsymbol{v}_i$ are the position and velocity vectors of the i^{th} link, l_i is the length of the i^{th} link, and θ_j and $\dot{\theta}_i$ are the joint angle and velocity for the j^{th} joint. Here, i varies from 1 to 3, and j varies from 1 to i. From this, the kinetic and potential energies of the arm are defined as

$$T = \frac{1}{2} \sum_{i=1}^{3} \boldsymbol{v}_i^{\mathrm{T}} m_i \boldsymbol{v}_i, \qquad U = -\sum_{i=1}^{3} m_i \boldsymbol{g}^{\mathrm{T}} \boldsymbol{r}_i, \tag{3}$$

where m_i represents the mass of the i^{th} body, and $\boldsymbol{g} = \left[0, \, 0, \, g\right]^{\mathrm{T}}$ is the gravity vector in N-frame. The robot dynamics can then be obtained using the Lagrangian method as follows:

$$Q = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial L}{\partial \boldsymbol{q}} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial T}{\partial \boldsymbol{q}} + \frac{\partial U}{\partial \boldsymbol{q}}, \tag{4}$$

where Q is the torque applied on the joints, $L = T - U$ is the Lagrangian, and $\boldsymbol{q} = \left[\theta_1, \, \theta_2, \, \theta_3\right]^{\mathrm{T}}$ is the set of generalized coordinates. A relationship between the

joint velocity of the i^{th} body and the generalized coordinates can be expressed using the Jacobian matrix (J_i) as $\boldsymbol{v}_i = J_i\dot{\boldsymbol{q}}$. Based on this, the kinetic energy can be expressed using the generalized coordinates as

$$T = \frac{1}{2}\dot{\boldsymbol{q}}^T M(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{5}$$

where the inertia matrix of the arm is given by

$$M(\boldsymbol{q}) = \sum_{i=1}^{3} m_i {J_i}^{\mathrm{T}} J_i. \tag{6}$$

Extending the above methodology, one can write the generalized dynamics of the robotic arm as [3]

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q}) = Q, \tag{7}$$

where $C(\boldsymbol{q},\dot{\boldsymbol{q}})$ matrix represents the nonlinear centrifugal and Coriolis forces, and $G(\boldsymbol{q})$ is the gravitational force expressed as $G(\boldsymbol{q}) = \partial U/\partial \boldsymbol{q}$. Using the Christoffel symbol ($Z_{\alpha\beta\gamma}$), the Coriolis matrix is represented as [12]

$$C(\boldsymbol{q},\dot{\boldsymbol{q}})_{\alpha,\beta} = \sum_{\gamma=1}^{3} Z_{\alpha\beta\gamma}\dot{\boldsymbol{q}}_\gamma, \tag{8}$$

where $\alpha,\beta \in \{1,2,3\}$, and the Christoffel symbol is represented in terms of the partial derivative of $M(\boldsymbol{q})$ as [12]

$$Z_{\alpha\beta\gamma} = \frac{1}{2}\left(\frac{\partial M_{\alpha\beta}}{\partial q_\gamma} + \frac{\partial M_{\alpha\gamma}}{\partial q_\beta} - \frac{\partial M_{\beta\gamma}}{\partial q_\alpha}\right). \tag{9}$$

3 Proposed Methodology

3.1 Closed-Loop Inverse Kinematics (CLIK) Control

In the case of redundancy (i.e., when the robotic manipulator has redundant DOF), multiple solutions of inverse kinematics exist, and secondary tasks can be used to prioritize specific movements or behaviors of the robot. This modeling of the secondary task can help obtain a unique inverse kinematics solution that satisfies the constraints while being able to track the desired trajectory. In this study, the Jacobi pseudo-inverse method is used to derive the inverse kinematics of the system. This involves using the Jacobian to map the joint velocities of the arm to the velocities of the end-effector as follows:

$$\dot{\boldsymbol{x}}_e = J_e(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{10}$$

where $J_e(\boldsymbol{q}) \in \mathcal{R}^{3\times5}$ is the Jacobian matrix of the end effector, and $\dot{\boldsymbol{x}}_{\boldsymbol{e}}$ is the velocity of the end-effector. By differentiating Eq. (10), the desired joint accelerations can be obtained. Since the obtained Jacobian is not a square matrix, the Moore-Penrose inverse [9] is introduced to represent the joint accelerations as follows:

$$\ddot{\boldsymbol{q}} = J_e^{+}(\ddot{\boldsymbol{x}}_{\boldsymbol{e}} - \dot{J}_e\dot{\boldsymbol{q}}), \tag{11}$$

where ${J_e}^{\dagger}$ is the pseudo-inverse of the Jacobian matrix represented as $J_e^{\mathrm{T}}(J_eJ_e^{\mathrm{T}})^{-1}$. From Eq. (11), a PD control law for tracking a given trajectory, considered to be the primary task, can be written as [3, 5]

$$\ddot{\boldsymbol{q}}_P = {J_e}^{\dagger}\left[\ddot{\boldsymbol{x}}_d + K_d(\dot{\boldsymbol{x}}_d - \boldsymbol{v}_3) + K_p(\boldsymbol{x}_d - \boldsymbol{r}_3) - \dot{J}_e\dot{\boldsymbol{q}}\right], \tag{12}$$

where $\boldsymbol{x}_d$ is the desired end-effector trajectory, $\boldsymbol{r}_3$ is the actual end-effector trajectory, $\boldsymbol{v}_3$ is the end-effector velocity, and K_p and K_d are the positive definite gain matrices.

However, the control law obtained in Eq. (12) lacks efficiency in avoiding singularities and fails to utilize the benefits, such as joint-limit avoidance, that a redundant manipulator offers. Hence, to exploit the redundant space, joint limit avoidance is added as a secondary task. The resultant control law, based on CLIK control, is now expressed as [3, 15, 11]

$$\ddot{\boldsymbol{q}}_S = S(\dot{\Phi}_N + K_N\dot{\boldsymbol{e}}_N) - ({J_e}^{\dagger}\dot{J}_e{J_e}^{\dagger} + \dot{J_e}^{\dagger})J_e(\Phi_N - \dot{\boldsymbol{q}}), \tag{13}$$

where $e_N = S\,(\Phi_N - \dot{\boldsymbol{q}})$ is the null-space velocity error, and $S = (I - {J_e}^{\dagger}J_e)$ projects the velocity error onto the null-space of the Jacobian, ensuring the same end-effector position for a different set of joint angles. Furthermore, Φ_N denotes the joint velocity vector corresponding to the secondary task. This is obtained by defining the secondary velocity as a negative gradient of a cost function ϕ that defines the secondary task constraint. Therefore, one can write $\Phi_N = -\partial\phi/\partial\boldsymbol{q}$ for ϕ expressed as follows aimed at ensuring that the joints of the arm stay within the specified physical limits [15, 17]

$$\phi = -\frac{1}{3}\sum_{i=1}^{3}\left[\frac{q_i - \bar{q}_i}{\max\,(q_i) - \min\,(q_i)}\right]^2, \tag{14}$$

where $\bar{q}_i$ represents the center of the angular range of the i^{th} joint q_i.

The stability of the resultant control law can be proven by the Lyapunov stability theorem by choosing the Lyapunov function as $V = \frac{1}{2}||\dot{\boldsymbol{e}}_N||$. This ensures that the null-space velocity error ($\dot{\boldsymbol{e}}_N$) converges to zero, and the controller is asymptotically stable as long as it doesn't encounter singularity [5].

However, the control gains in the CLIK controller are constant and tuned by trial and error, which is tedious and time-consuming. Moreover, they lack adaptability to changes and disturbances in the environment, leading to sub-optimal performance in the control system. Therefore, to enhance performance, it is important to adaptively tune the control gains. In this study, fuzzy inference systems (FISs) are used to obtain the control gains based on the available environmental information.

3.2 Fuzzy Inference System (FIS)

A FIS is a nonlinear mapping of a given input to an output using fuzzy logic reasoning [4]. The main advantage of FIS is the *interpretability* offered by the linguistic rules and the *scalability* offered by the categorical nature of the fuzzified inputs. In this study, the use of FIS enables us to adaptively update the PD control gains, making the control technique more adaptive to changes in the environment and thereby improving performance [10]. For this purpose, two 2-input and 1-output FISs are designed to obtain the K_p and K_d gains. Both FISs utilize joint position error ($\boldsymbol{e}$) and joint velocity error ($\dot{\boldsymbol{e}}$) of the robotic arm as input parameters to derive the appropriate gains.

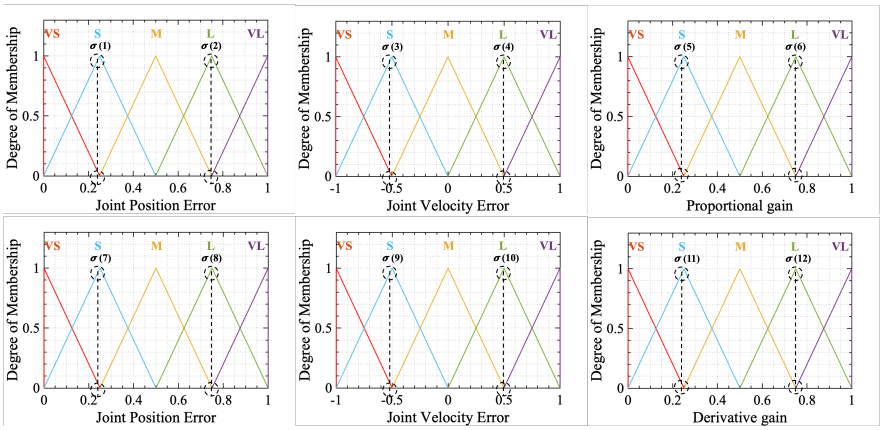


Fig. 2: MFs of the FISs. The top row depicts FIS_1 (Left: e, Center: $\dot{e}$, Right: K_p) and the bottom row depicts FIS_2 (Left: e, Center: $\dot{e}$, Right: K_d)

Five triangular membership functions (MFs) are defined for each input and output, as depicted in Fig. 2. The MFs are categorized as Very Small (**VS**), Small (**S**), Medium (**M**), Large (**L**), and Very Large (**VL**). The ranges for the joint position and velocity errors are normalized and defined within the ranges of [0, 1] and [-1, 1], respectively. Similarly, the ranges for the outputs K_p and K_d are normalized and defined within [0, 1]. Each FIS optimizes the two vertices labeled as $\boldsymbol{\sigma}$, while the remaining trailing and leading vertices are predetermined based on the input data range. Thus a total of 12 parameters are required to be optimized by the GA. Table 1 illustrates how the rules for two FISs are formulated and structured. Each FIS has 25 rules with five possible outputs corresponding to two inputs and one output. For instance, $\boldsymbol{\sigma}(13)$ indicates the fuzzy output resulting from the combination of "**VS**" joint velocity error and "**VS**" joint angle for the first FIS. Similarly, $\boldsymbol{\sigma}(39)$ represents the same for the second FIS. The output of the FIS system lies between 0 and 1 and is multiplied by a scaling factor. Through trial and error, the scaling factor was chosen as 2 from a range of 0 to 10, aiming to ensure that the resulting gains remain within the same

order of magnitude as the original gains. This scaled output is further multiplied by the hand-tuned gains used in the CLIK controller. This approach effectively reduces the control effort, thereby enhancing the overall performance of the controller.

Table 1: FIS rules formulation for two FISs

		Joint Angle				
		VS	**S**	**M**	**L**	**VL**
Joint velocity error	**VS**	$\sigma(13,39)$	$\sigma(18,43)$	$\sigma(23,48)$	$\sigma(28,53)$	$\sigma(33,58)$
	S	$\sigma(14,39)$	$\sigma(19,44)$	$\sigma(24,49)$	$\sigma(29,54)$	$\sigma(34,59)$
	M	$\sigma(15,40)$	$\sigma(20,45)$	$\sigma(25,50)$	$\sigma(30,55)$	$\sigma(35,60)$
	L	$\sigma(16,41)$	$\sigma(21,46)$	$\sigma(26,51)$	$\sigma(32,56)$	$\sigma(36,61)$
	VL	$\sigma(17,42)$	$\sigma(22,47)$	$\sigma(27,52)$	$\sigma(32,57)$	$\sigma(37,62)$

In complex cases where the relationship between the input and output is hard to infer and the number of variables is high, it is difficult to manually design the rules and MFs of a FIS. Genetic algorithm (GA) is a popular evolutionary algorithm used to tune the MFs and rules in a FIS [7]. GA improves FISs by optimizing the components used in the user-defined search spaces. In this study, the total number of parameters (MFs and rules) to be optimized by GA is 62. The integration of the FISs with the GA leads to the GFS framework that combines the ability of FISs to tolerate imprecision and the strength of fuzzy reasoning with the ability of GA to optimize its internal rules and MFs.

3.3 Genetic Algorithm (GA) Optimization

The vertices of the MFs and rules to be optimized are stored as $\boldsymbol{\sigma}$ and are passed on to the GA. GA creates a random population of potential solutions, which are then optimized through a sequence of operations including crossover, mutation, selection, and elitism [7]. To ensure the GA accurately evaluates each solution, it is essential to establish a suitable fitness function, which is defined as

$$\mathcal{J} = \int_{t=0}^{t_F} \boldsymbol{\tau}_{\mathrm{s}}^{\mathrm{T}} \boldsymbol{\tau}_{\mathrm{s}} \, \mathrm{d}t + \Delta_{\mathrm{JL}} + \Delta_{\mathrm{P}} + \Delta_{\mathrm{V}}, \tag{15}$$

where t_F is the total simulation time, $\boldsymbol{\tau}_{\mathrm{s}}$ is the control effort applied at each time step. In this study, a huge penalty is imposed on solutions that violate joint limit constraints (Δ_{JL}) and exceed predefined position (Δ_{P}) and velocity (Δ_{V}) tracking error limits. This penalty is aimed at minimizing control effort and preserving tracking accuracy by favoring desirable solutions for future generations.

4 Simulation Study

In this work, a 3-DOF planar manipulator is considered. Each body of the manipulator is assumed to be a thin rod of length 0.3 m, with a point mass located at the end of each rod (0.25 kg each). To represent realistic behavior, each body is constrained with joint limits of $[0,\ \pi]$, $[-\pi,\ \pi]$, and $[-\pi,\ \pi]$, respectively. Assuming that an object is being picked up from one end and placed at another, a curvilinear trajectory is designed based on the structural parameters of the robot. Spline-based smooth polynomial solutions are used to define a reasonable trajectory between the two points. This ensures the smooth movement of the robotic manipulator during the PAP process [1].

Table 2: GA parameters

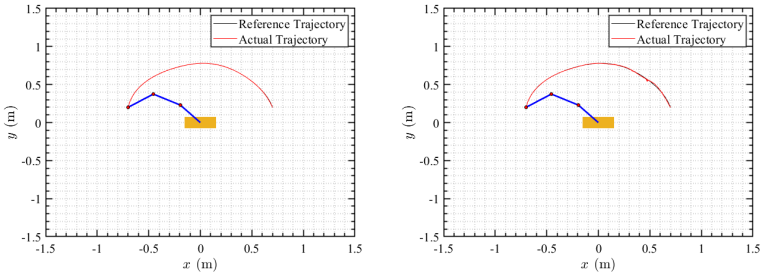
Description	Value
Probability of Crossover	0.80
Mutation Rate	0.30
Elitism Ratio	0.10
Population Size	20
Number of Generations	100
Selection Algorithm	Tournament selection
Tournament size	4
Crossover Algorithm	Scattered crossover
Δ_{JL}	10^{10}
Δ_{V}, Δ_{P}	10^{15}

This study examines two scenarios: a training scenario and a testing scenario. In the training scenario, the shape of the trajectory is defined as a linear trajectory with slight curvature, forming a rounded pattern. The initial manipulator pose of the arm is set as $[0.1311, -0.5174, 1.5273]^{\mathrm{T}}$. Table 2 outlines the training parameters for the GA, and Table 3 outlines the search region for the parameters to be optimized.

For testing the proposed controller's performance, the curvature of the trajectory is decreased, and the shape exhibits more pronounced curvature, with sharper turns and a more complex path. To accommodate this difference, the initial pose of the arm is changed to $[0.4951, 0.36862, 0.7772]^{\mathrm{T}}$ in the testing scenario.

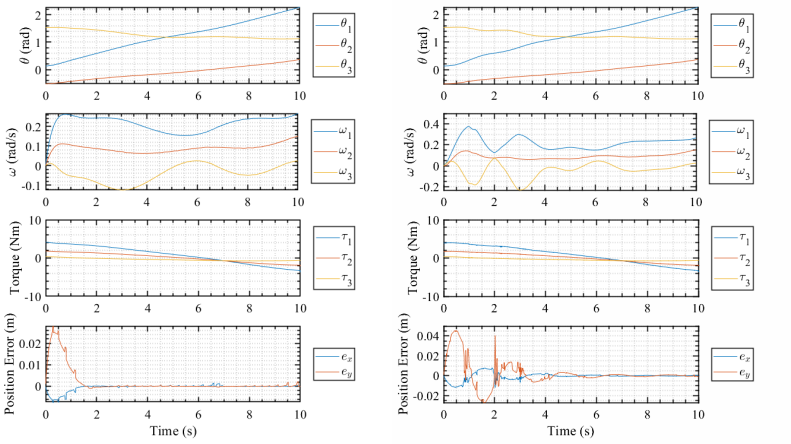
Table 3: Search space for FIS parameters

Description	Value
$\sigma(1), \sigma(5), \sigma(7), \sigma(11)$	[0, 0.5]
$\sigma(2), \sigma(6), \sigma(8), \sigma(12)$	[0.5, 1]
$\sigma(3), \sigma(9)$	[-1, 0]
$\sigma(4), \sigma(10)$	[0, 1]
$\sigma(13{:}62)$	{1,2,3,4,5}



(a) CLIK (b) GFS-aided CLIK

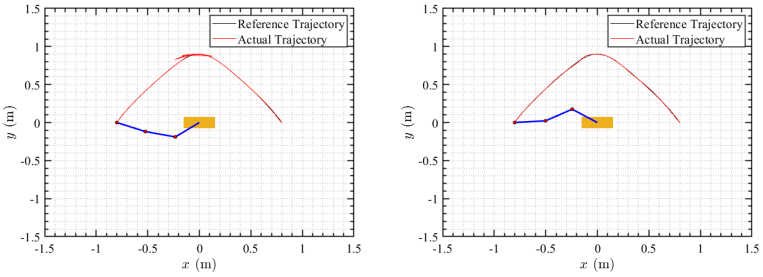
Fig. 3: Training scenario: trajectory tracking results



(a) CLIK (b) GFS-aided CLIK

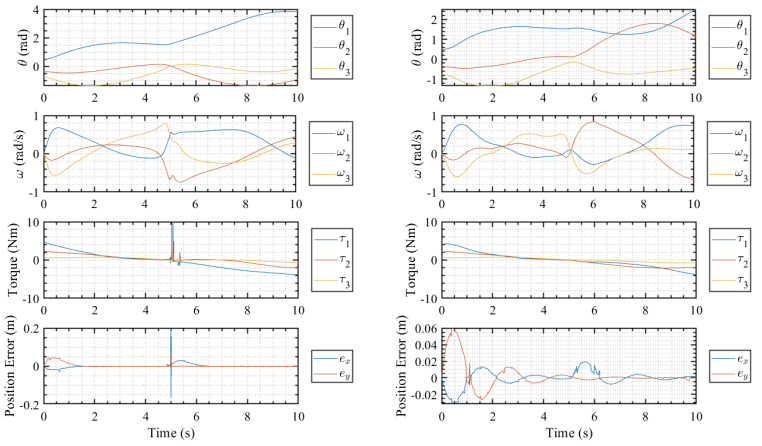
Fig. 4: Training scenario: state histories, torque, and tracking error

Fig. 3 illustrates the motion of the robot arm under both controllers during training. In both cases, the arm tracks the defined trajectory while respecting joint limits. For CLIK control, the tracking error (RMSE) is 0.0072 m, with a total control effort of 73.01. For the designed GFS-CLIK control, the tracking error is 0.0131 m, with a total control effort of 72.54. Fig. 4 shows smooth control torques for both techniques, further depicting that the designed control technique is as efficient as CLIK control. The difference in tracking errors is minimal, suggesting similar tracking accuracy. GFS-CLIK demonstrates a slight reduction in control effort. Although this improvement is not significant, it highlights the importance of finely hand-tuning the PD control gains in the CLIK control technique. The advantages of the GFS-aided CLIK control become apparent in the testing scenario when the environment is changed without altering the control gains.



(a) CLIK (b) GFS-aided CLIK

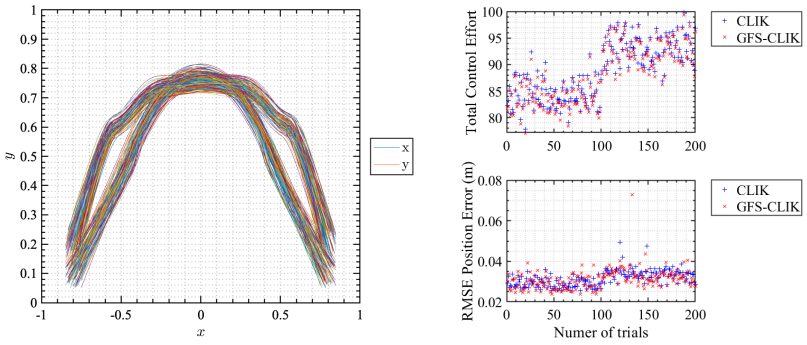
Fig. 5: Testing scenario: trajectory tracking results



(a) CLIK (b) GFS-aided CLIK

Fig. 6: Testing scenario: state histories, torque, and tracking error

In the testing scenario, similar to the training scenario, both controllers successfully track the defined trajectory. As shown in Fig. 5, in CLIK control, the joint angle θ_1 exceeds π, violating the defined joint limit constraints. Notably, the tracking error for CLIK control (RMSE: 0.0247 m) is higher than that for the designed GFS-CLIK control (RMSE: 0.0157 m). Moreover, GFS-CLIK exhibits lower total control effort (J : 60.65) compared to CLIK control (J : 67.69), as evident in Fig. 6. Additionally, sharp peaks can be observed in the torque response of CLIK control, whereas the torque generated by GFS-CLIK control is much smoother. Similarly, Fig. 6 illustrates large error peaks during CLIK control operation, while the proposed controller maintains better tracking accuracy with reduced control effort and no joint limit violations.



(a) Trajectories (b) Control efforts and error

Fig. 7: Monte Carlo simulation results

To verify the robustness of the proposed approach for varying trajectories, a Monte Carlo simulation is performed. For this, 200 different trajectories are chosen, as shown in Fig. 7. The results further demonstrate that the control effort is lower for GFS-CLIK than for CLIK in all 200 cases. Additionally, out of the 200 cases, GFS-CLIK demonstrated higher tracking accuracy over CLIK in 127 instances. This highlights the adaptability and scalability of the GFS to changing environments. The defined cost function prioritizes optimizing control effort, yet further reducing position error is possible with consideration. From the results, it can be seen that the integration of GFS with CLIK increases the efficiency of the controller by minimizing the required control effort while maintaining tracking accuracy. Furthermore, it can be observed from the results that when the environment changes under different conditions, the proposed GFS-CLIK controller outperforms the CLIK controller, highlighting the robustness of the controller to different environmental settings.

PAP operations occur in dynamic environments and require robots that can adapt to the changes. GFSs can enable a robot to adaptively tune its gains when the PAP operation changes from a static platform to a conveyor belt, and allowing it to perform efficiently without any additional training or manual tuning. Thus GFS improves the existing control techniques by adding adaptability and interpretability through fuzzy reasoning.

5 Conclusions

In this paper, a closed-loop inverse kinematics (CLIK) controller enhanced by genetic fuzzy systems (GFS) is developed for the pick-and-place operation of a robot arm. The designed controller uses fuzzy inference systems (FISs) to adaptively tune the proportional-derivative control gains. Additionally, a genetic algorithm is employed to optimize the membership functions and rules of the FISs to improve the

performance of the GFS-aided CLIK controller. The performance of the designed controller is validated through simulation studies in training and testing scenarios. The results show that the GFS-aided controller ensures lower control effort with high tracking accuracy while satisfying the joint limit constraints imposed on the system.

References

1. Angeles, J., Alivizatos, A., Zsombor-Murray, P.J.: The synthesis of smooth trajectories for pick-and-place operations. IEEE Transactions on Systems, Man, and Cybernetics **18**(1), 173–178 (1988)
2. Avanzini, G.B., Zanchettin, A.M., Rocco, P.: Constrained model predictive control for mobile robotic manipulators. Robotica **36**(1), 19–38 (2018)
3. Chhabra, A., Kim, D.: Ground robotic platform for simulation of on-orbit servicing missions. Journal of Aerospace Information Systems **19**(7), 480–493 (2022). DOI 10.2514/1.I011008. URL https://doi.org/10.2514/1.I011008
4. Cord, O., et al.: Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases, vol. 19. World Scientific (2001)
5. Hsu, P., Hauser, J., Sastry, S.: Dynamic control of redundant manipulators. In: 1988 American Control Conference, pp. 2135–2139 (1988). DOI 10.23919/ACC.1988.4790077
6. Khoury, G., Saad, M., Kanaan, H.Y., Asmar, C.: Fuzzy pid control of a five dof robot arm. Journal of Intelligent and Robotic systems **40**, 299–320 (2004)
7. Mirjalili, S.: Evolutionary algorithms and neural networks. In: Studies in computational intelligence, vol. 780. Springer (2019)
8. Nagurka, M.L., Yen, V.: Optimal design of robotic manipulator trajectories: a nonlinear programming approach. Carnegie-Mellon University. The Robotics Institute (1987)
9. Penrose, R.: A generalized inverse for matrices. Mathematical Proceedings of the Cambridge Philosophical Society **51**(3), 406–413 (1955). DOI 10.1017/S0305004100030401
10. Pokhrel, S., Sathyan, A., Eisa, S.A., Cohen, K.: Use of fuzzy pid controller for pitch control of a wind turbine. In: Applications of Fuzzy Techniques: Proceedings of the 2022 Annual Conference of the North American Fuzzy Information Processing Society NAFIPS 2022, pp. 205–216. Springer (2022)
11. Sciavicco, L., Siciliano, B.: A solution algorithm to the inverse kinematic problem for redundant manipulators. IEEE Journal on Robotics and Automation **4**(4), 403–410 (1988)
12. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics: Modelling, Planning and Control. Advanced Textbooks in Control and Signal Processing. Springer London (2008). URL https://books.google.com/books?id=VsTOQOnQjCAC
13. Soltanpour, M.R., Khalilpour, J., Soltani, M.: Robust nonlinear control of robot manipulator with uncertainties in kinematics, dynamics and actuator models. International Journal of Innovative Computing, Information and Control **8**(8), 5487–5498 (2012)
14. Su, Y., Duan, B., Zheng, C.: Nonlinear pid control of a six-dof parallel manipulator. IEE Proceedings-Control Theory and Applications **151**(1), 95–102 (2004)
15. Wang, J., Li, Y., Zhao, X.: Inverse kinematics and control of a 7-dof redundant manipulator based on the closed-loop algorithm. International Journal of Advanced Robotic Systems **7**(4), 37 (2010)
16. Wen, J.T., Murphy, S.H.: Pid control for robot manipulators (1990)
17. de Wit, C., Siciliano, B., Bastin, G.: Theory of Robot Control. Communications and Control Engineering. Springer London (2012). URL https://books.google.com/books?id=5NnUBwAAQBAJ
18. Yesil, E., Guzelkaya, M., Eksin, I.: Fuzzy pid controllers: An overview. In: The Third Triennial ETAI International Conference on Applied Automatic Systems, Skopje, Macedonia, pp. 105–112. ETAI Society of Macedonia (2003)