Improved Shuffled Frog Leaping Algorithm with Self-Adaptive Shuffling for Fuzzy Logic PD+G Controller Optimization in Robotic Manipulators

Duc Hoang Nguyen
Faculty of Electrical and Electronics Engineering
HCMC University of Technology
Vietnam National University Ho Chi Minh City

ABSTRACT
This paper introduces a novel self-adaptive shuffling mechanism within the Shuffled Frog Leaping Algorithm (SFLA) to improve its efficacy in the tuning of fuzzy logic Proportional-Derivative with Gravity Compensation (PD+G) controllers for trajectory tracking in the UP6 robotic manipulator. The proposed mechanism improves the balance between exploration and exploitation by continually modifying the frequency and intensity of scrambling in accordance with population diversity. This adaptive approach overcomes the constraints of the conventional SFLA, which implements a static scrambling process by facilitating more efficient global search and local refinement. The fuzzy controller parameters for a 6-DOF robotic manipulator are optimized using the enhanced SFLA to guarantee precise trajectory tracking. The self-adaptive shuffling mechanism results in enhanced tracking accuracy and faster convergence in comparison to the standard SFLA, as evidenced by the simulation results. The results of this study suggest that the proposed method is a plausible solution for real-time control applications that necessitate efficient parameter tuning in nonlinear systems.

General Terms
Algorithms.

Keywords
Optimization, SFLA, Fuzzy, PD+G controller, Manipulator.

1. INTRODUCTION
In robotics, the precise control of manipulators in environments affected by gravity is challenging. Gravitational forces can alter a manipulator's dynamics, leading to deviations from intended paths and compromising accuracy. Proportional-Derivative (PD) controllers have been commonly used for their straightforward implementation and effectiveness in various control tasks. However, gravity introduces complexities that PD controllers alone may not effectively handle. Adding Gravity Compensation (PD+G) to PD controllers marks a significant advancement in control, as it helps stabilize and improve manipulators' accuracy by offsetting the impact of gravitational forces. This makes PD+G controllers a well-regarded choice for achieving precise control in robotic systems [1], [2].

Even with the progress made, tuning PD+G controllers remains a challenging task. Effective tuning requires a solid understanding of the manipulator's dynamics and how gravity affects each joint. This challenge grows in systems with high degrees of freedom, where the controller must deal with constant dynamic changes and environmental uncertainties. Zhang et al. [3] took on this problem by introducing a PD controller combined with a Radial Basis Function (RBF) Neural Network designed for both gravity and inertia compensation. Their approach significantly enhanced trajectory tracking in manipulators with complex shapes and varying mass distributions. Experiments on a 3-DOF robotic arm demonstrated that this method performs better than traditional approaches by accurately estimating gravitational disturbances and adjusting the center of mass in real-time.

Building on these concepts, Fuzzy Logic Controllers (FLC) have been integrated with PD+G controllers to manage nonlinearities and uncertainties in robotic systems. Fuzzy logic adds adaptability to the control process, allowing controllers to adjust parameters in response to changing conditions dynamically. Yueyuan Zhang et al. [4] introduced a fuzzy-PD+G control scheme to improve trajectory tracking performance in a 3-DOF robotic manipulator by compensating for the nonlinear disturbances introduced by gravity. This approach has shown significant improvements in trajectory tracking accuracy and stability, making fuzzy-PD+G controllers a promising solution for complex robotic control applications.

While PD+G controllers with fuzzy logic have demonstrated improvements in control accuracy, the effectiveness of these controllers depends heavily on the optimal tuning of their parameters, such as membership functions and control rules. Manual tuning of fuzzy controllers is time-consuming and often suboptimal, especially in systems with high degrees of freedom and dynamic environments. To address these challenges, metaheuristic algorithms, such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA), have been applied to tune fuzzy logic controllers. These approaches have shown promise in navigating the complex, multidimensional search spaces involved in tuning fuzzy controllers, but they often involve high computational costs [5], [6].

The Shuffled Frog Leaping Algorithm (SFLA) is a relatively new metaheuristic optimization technique inspired by the foraging behavior of frogs. SFLA has been applied successfully in various optimization problems due to its simplicity and efficiency. However, the static nature of SFLA's shuffling mechanism—where frogs are shuffled at fixed intervals—limits its ability to adapt to real-time changes in population diversity. This limitation reduces the algorithm's effectiveness in optimizing fuzzy controllers for dynamic systems like robotic manipulators, where rapid adaptation is necessary for achieving optimal performance.

This paper proposes a novel self-adaptive shuffling mechanism for the Shuffled Frog Leaping Algorithm (SFLA) to improve its ability to optimize fuzzy logic PD+G controllers for robotic manipulators. The self-adaptive mechanism dynamically adjusts the shuffling process based on population diversity, enhancing the balance between exploration (global search) and exploitation (local search). By incorporating this adaptive shuffling mechanism, the improved SFLA is better suited for optimizing fuzzy controllers in real-time, where dynamic adjustments are critical for effective control.

The remaining paper is organized as follows: Section 2 reviews the existing literature on related work. Section 3 presents an overview of the Shuffled Frog Leaping Algorithm (SFLA) and the proposed self-adaptive shuffling mechanism. The process of tuning the fuzzy PD+G controller parameters using the enhanced SFLA is detailed in Section 4. Section 5 discusses the simulation results, and Section 6 concludes the paper.

2. RELATED WORK

2.1 Enhancing the Shuffled Frog Leaping Algorithm

The Shuffled Frog Leaping Algorithm (SFLA) has undergone considerable enhancements, focusing on its balance between global exploration and local exploitation in solving optimization problems. SFLA has been widely used in various fields, with researchers continuously seeking methods to improve its performance. One commonly used approach is hybridization, where SFLA is integrated with other algorithms such as Particle Swarm Optimization (PSO) or Genetic Algorithms (GA). Xi Hu et al. presented a hybrid SFLA-PSO model that demonstrated a marked improvement in global search efficiency and solution precision [7]. Similarly, Lin, M.J. proposed the SFLA-PSO hybrid algorithm, combining PSO’s fast searchability with SFLA’s global search strategy to address SFLA’s slow convergence and PSO’s tendency to get trapped in local optima [8]. Nguyen. introduced a hybrid of SFLA with the Bees Algorithm (BA) to further enhance convergence speed and solution accuracy in complex optimization problems [9].

However, a key limitation of the original SFLA is its static shuffling mechanism, which does not adapt to changes in population diversity during the optimization process. This can lead to premature convergence or inefficient exploration of the solution space. To overcome the shortcomings, Zhao, Z. et al. proposed a novel modified shuffled frog leaping algorithm (MSFLA) with inertia weight [10]. While these modifications have improved overall performance, the adaptive adjustment of the shuffling mechanism, particularly for real-time tuning applications like fuzzy logic controllers in robotic systems, remains underexplored.

2.2 Tuning Fuzzy Logic Controllers for Robotic Manipulators

Fuzzy logic controllers (FLC) have proven highly effective in handling nonlinearities and uncertainties in robotic systems. In particular, Proportional-Derivative with Gravity Compensation (PD+G) controllers have been widely used to enhance the stability and precision of robotic manipulators, especially in dynamic environments influenced by gravity [11]. The integration of fuzzy logic into PD+G control frameworks offers enhanced adaptability and control precision, enabling more responsive manipulation of complex systems. Zhang et al. [12] introduced a fuzzy-PD+G control scheme to improve trajectory tracking in robotic manipulators, specifically addressing the challenge of gravity compensation. By combining fuzzy logic with PD control, their approach dynamically adjusts to nonlinearities introduced by gravitational forces, demonstrating significant improvements in tracking accuracy.

The tuning of fuzzy logic controllers, however, remains a critical challenge. Traditional methods for tuning such as trial-and-error or manual adjustment are time-consuming and often suboptimal, especially when dealing with the intricacies introduced by the dynamic changes and uncertainties in robotic environments. Recent advancements have seen the adoption of metaheuristic algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) for tuning fuzzy logic controllers. Perrusquía, A. et al. introduce a new method for tuning Proportional-Derivative (PD) controllers with gravity compensation (PD+G) for robotic manipulators. Recognizing the limitations of PD controllers in compensating for gravitational forces, the authors propose a tuning method that relies on the gravitational torques vector's bound, simplifying the tuning process by not requiring full knowledge of the robot dynamics. This method is based on global asymptotic stability principles derived from La Salle’s theorem and robot dynamics properties. [13].

The precise control of robotic manipulators in environments influenced by gravitational forces remains a significant challenge. Gravity Compensation (PD+G) controllers enhance manipulator control by counteracting the effects of gravity, leading to improved trajectory tracking and positional accuracy. Zhang et al. introduced a PD controller integrated with a Radial Basis Function (RBF) Neural Network for both gravity and inertia compensation in robotic manipulators. Their method outperformed traditional least mean square approaches in terms of accurately estimating gravity disturbances and identifying the center of mass, resulting in improved control performance for a 3-DOF robotic arm [14].

Despite these advancements, the use of SFLA for tuning fuzzy controllers in PD+G frameworks is relatively new. SFLA, with its ability to handle nonlinear, multidimensional optimization tasks efficiently, provides a promising alternative to traditional algorithms like PSO and GA. The self-adaptive shuffling mechanism, as proposed in this research, aims to further improve SFLA’s performance by dynamically adjusting the shuffling process based on population diversity. This allows for a better balance between global exploration and local refinement, which is particularly beneficial in robotic control applications where real-time adaptation is crucial.

3. PROPOSED METHODOLOGY

3.1 Overview of the Shuffled Frog Leaping Algorithm (SFLA)

The Shuffled Frog Leaping Algorithm (SFLA) is a population-based optimization method inspired by the behavior of frogs searching for food. In SFLA, a population of frogs is divided into memeplexes (subgroups), where local searches are performed based on the best-performing frog within each memeplex. After several iterations, frogs are shuffled between memeplexes, enabling global information sharing across the entire population. This combination of local search (within memeplexes) and global exchange (shuffling between memeplexes) enables the algorithm to explore and exploit the search space effectively [15].

However, the standard SFLA uses a static shuffling process, where frogs are shuffled at fixed intervals, regardless of the population’s performance or diversity. This static approach can be inefficient, as it doesn’t adapt to changes in the search landscape, potentially leading to premature convergence or excessive exploration.

3.2 Self-Adaptive Shuffling Mechanism

To address the limitations of the static shuffling process in SFLA, this paper proposes a self-adaptive shuffling mechanism that dynamically adjusts the shuffling frequency based on the diversity of the population. The goal is to improve the balance between global exploration and local exploitation, enabling the algorithm to respond more effectively to changes during the optimization process.

3.2.1 Adaptation Based on Population Diversity

In this proposed mechanism, the diversity of the population is monitored throughout the optimization process. Diversity can be measured using the variance of fitness values of frogs in the search space (eq. 1). When the diversity within memeplexes decreases below a certain threshold, it indicates that the population may be converging prematurely to a suboptimal solution. In this case, the shuffling frequency (eq. 2) is increased to encourage global exploration and prevent stagnation. Conversely, if the diversity remains high, shuffling is delayed allowing for more focused local refinement.

Let $F_{best}(t)$ represent the fitness of the best frog at iteration t, and the change in fitness be defined as:

$$\Delta F = F_{best}(t) - F_{best}(t-1) \qquad (1)$$

The algorithm adjusts the shuffling frequency S_f according to the following rule:

$$S_f = S_f + \alpha . sign(\Delta F) \qquad (2)$$

where α is a control parameter that determines the rate of adaptation, and $sign(\Delta F)$ is the sign function that indicates whether the fitness is improving or stagnating.

3.2.2 Dynamic Shuffling Intervals

The self-adaptive mechanism also includes dynamic adjustments to the shuffling intervals. Rather than using a fixed shuffling interval, the algorithm adjusts the interval based on the progress of convergence. When convergence slows down, the algorithm shortens the interval to introduce more frequent shuffling and explore new regions of the search space. On the other hand, when the algorithm is rapidly approaching an optimal solution, the interval is lengthened to allow for more thorough local searches within memeplexes.

3.2.3 Pseudocode

The pseudocode for the self-adaptive shuffling mechanism is presented in Fig. 1.

```
Initialize frog population
Divide population into memeplexes
While termination condition not met:
  For each memeplex:
    Perform local search
   Calculate diversity of each memeplex
  If diversity < threshold:
    Shuffle frogs between memeplexes
    Increase shuffling frequency
  Else:
    Delay next shuffling
    Adjust shuffling interval dynamically
End while
```

Fig. 1: Pseudocode for the Self-Adaptive Shuffling Mechanism in SFLA

3.3 Application to Tuning Fuzzy PD+G Controllers

The proposed self-adaptive SFLA is applied to tune the parameters of a fuzzy logic Proportional-Derivative with Gravity Compensation (PD+G) controller. Fuzzy logic controllers (FLC) are well-suited for handling uncertainties and nonlinearities in control systems, but their performance depends heavily on the precise tuning of parameters such as membership functions and rule sets. Integrating a fuzzy logic controller with a PD+G framework allows the system to handle the gravitational forces acting on a robotic manipulator, ensuring precise control and trajectory tracking.

3.3.1 Modified SFLA-based fuzzy PD+G controller

The structure of the modified SFLA-based fuzzy PD+G controller is illustrated in Fig. 2, where the parameters of the fuzzy PD+G controller are optimized using the modified SFLA algorithm [16]. The fuzzy controller inputs are:

- The trajectory tracking error $\tilde{q}(t)$, defined as the difference between the desired position and the actual position of the manipulator.
- The rate of change of the tracking error $d\tilde{q}(t)/dt$, which represents the error's velocity.

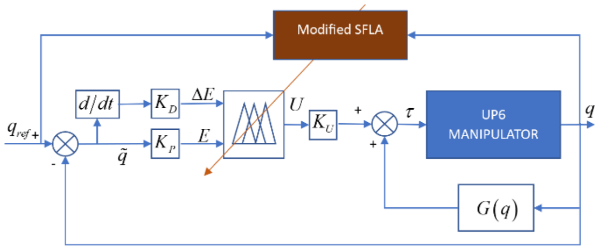

Fig. 2: Structure of Modified SFLA-based fuzzy PD+G controller

The membership functions for the inputs (E, ΔE) and outputs (U) of each axis controller have triangular shapes, as depicted in Fig. 3.

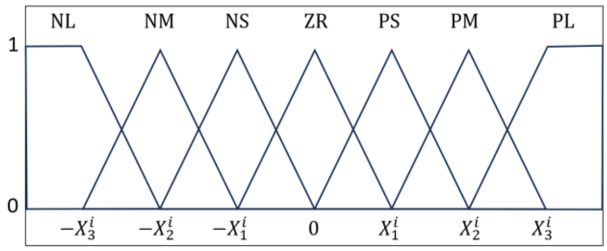

Fig. 3: Membership functions of inputs and output

The parameters of membership functions X_1^i, X_2^i, X_3^i and scaling factors K_D^i, K_P^i, K_U^i will be chosen by trial and error in normal fuzzy PD controller and tuned by SFLA algorithm.

3.3.2 Objective Function

The primary objective of this research is to optimize the fuzzy controller's parameters using the self-adaptive SFLA to improve the PD+G controller's performance. Specifically, the fuzzy controller's membership functions (input and output) and the scale values are tuned to minimize the tracking error in a 6-DOF robotic manipulator.

The fitness function employed to optimize the fuzzy PD controller parameters is expressed as a weighted sum of multiple performance criteria:

$$f(x) = \omega_1 E + \omega_2 S + \omega_3 O \quad (3)$$

where:

- $f(x)$ represents the fitness value of the individual x,
- ω_1, ω_2, and ω_3 are positive weighting factors,
- E, S, and O refer to the different performance criteria.

The primary objective is to minimize the fitness function by identifying the optimal values for the fuzzy PD controller parameters. The weighting factors ω_1, ω_2, and ω_3 are selected to ensure an appropriate balance among the different performance criteria, allowing for effective tuning of the controller. The details of the fitness function and its formulation are presented in [17].

4. SIMULATION RESULTS

4.1 Simulation parameters

- **Weighting factors of the fitness function**

The fitness function, as outlined in Section 3.3, is used to evaluate the performance of each solution. In the following simulation results, the primary focus is on improving the settling time. To prioritize this, the author assigns the weighting factors as follows: $\omega_1 = 0.3$, $\omega_2 = 0.5$, and $\omega_3 = 0.2$. These weights reflect the relative importance of each performance criterion, with ω_2 (corresponding to settling time) being the most significant factor. This allocation of weights ensures that the optimization process targets faster settling times while maintaining a balanced focus on other relevant performance metrics.

- **Modified SFLA parameters**

The parameter settings for the modified Shuffled Frog Leaping Algorithm (SFLA) are provided in Table 1. This table outlines the key algorithmic parameters, including population size, number of memeplexes, maximum number of iterations, and other relevant values used to guide the optimization process. These settings have been carefully chosen to ensure an optimal balance between exploration and exploitation, enhancing the algorithm's ability to search the solution space effectively and improve the performance of the fuzzy PD+G controller.

Table 1. The Modified SFLA parameter settings

G	n	c	m
500	100	2	10
D_{max}	$thresh$	α	
∞	10	0.02	

- **Termination Criteria**

The termination criteria for the optimization process in this study are predicated on two critical factors: the stability of the fitness value and the maximum number of generations. Initially, a limit is established on the number of generations to prevent the optimization process from running indefinitely, thereby conserving computational resources. Despite the complexity of the problem space, this upper constraint on the number of iterations guarantees that the search is managed and that there is no excessive runtime.

The stability of the fitness value is the primary focus of the second criterion. The optimization procedure is terminated if the fitness value remains constant for a predetermined number of consecutive iterations. This absence of development implies that the algorithm has either reached an optimal or near-optimal solution, or that further progress is improbable. The optimization process is dynamically halted when it is evident that additional iterations would not result in substantial improvements by monitoring the change in fitness value.

This optimization concludes under two conditions: when a satisfactory solution has been found, as indicated by fitness stability, or when the maximum number of generations is reached, thereby preventing superfluous computational effort. These termination criteria operate in tandem. This method maintains efficiency during the optimization process while simultaneously obtaining high-quality solutions.

4.2 Simulink model

To validate the effectiveness of the proposed method, simulation experiments are conducted on a 6-DOF robotic manipulator (UP6 Motoman). The fuzzy PD+G controller is applied to control the manipulator’s movements, while the self-adaptive SFLA is used to tune the fuzzy controller’s parameters. The performance of the system is evaluated in terms of tracking accuracy and convergence speed.

Fig. 4 illustrates the closed-loop control diagram where a fuzzy PD+G controller is employed to regulate the angular position of six axes. The system consists of six individual fuzzy PD controllers. Each of these controllers follows the same configuration, ensuring consistent control logic across all six axes [17].

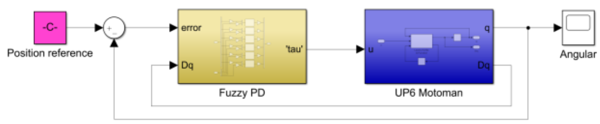

Fig. 4: Simulink schematic controlling the UP6 robot

In the simulation results presented below, the controller is responsible for regulating the angular position of the robot's axes, moving them from an initial position of 0 to a target final position of $q_{ref} = \left[\frac{\pi}{5}, \frac{\pi}{6}, \frac{\pi}{7}, \frac{\pi}{2}, \frac{2\pi}{3}, \pi\right]^T$. The goal is to ensure precise and smooth transitions between these points while minimizing settling time, overshoot, and error.

Figures 5 through 10 display the response results of the UP6 robot axes using three different controllers: the modified SFLA-based fuzzy PD+G (FPDGMSFLA), the SFLA-based fuzzy PD+G (FPDGSFLA), and the standard fuzzy PD+G (FPDG). These results compare the performance of each controller in terms of key metrics such as settling time, overshoot, and overall control precision, highlighting the improvements achieved by incorporating the SFLA and its modified version into the fuzzy PD+G controller framework.

Table 2 provides a summary of the performance metrics for the three controllers—modified SFLA-based fuzzy PD+G (FPDGMSFLA), SFLA-based fuzzy PD+G (FPDGSFLA), and fuzzy PD+G (FPDG). The table highlights key metrics including 2% settling time, overshoot, and error.

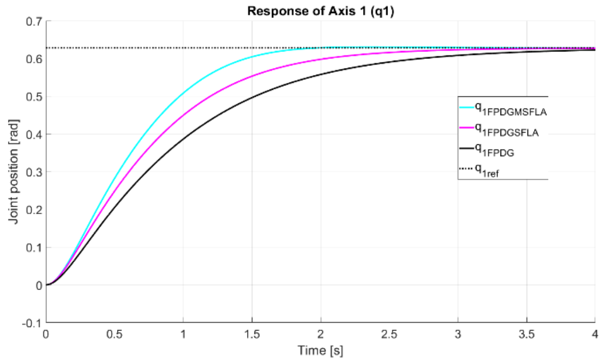

Fig. 5: Position response of the joint 1

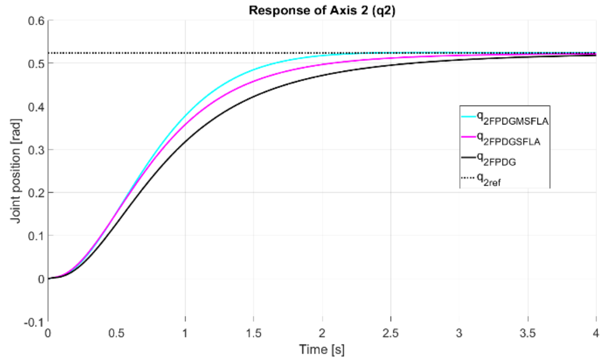

Fig. 6: Position response of the joint 2

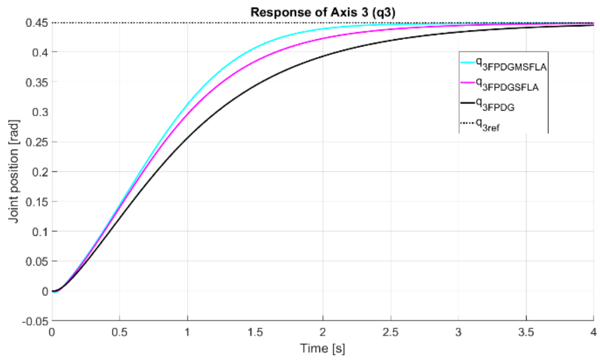

Fig. 7: Position response of the joint 3

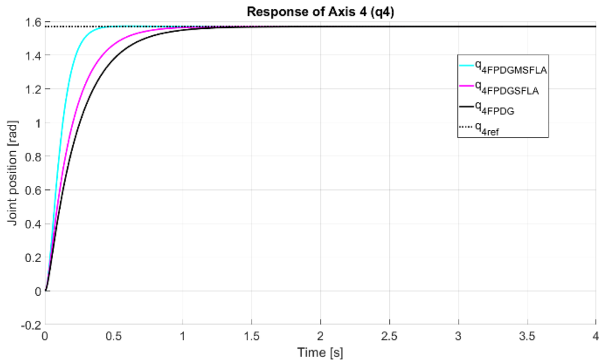

Fig. 8: Position response of the joint 4

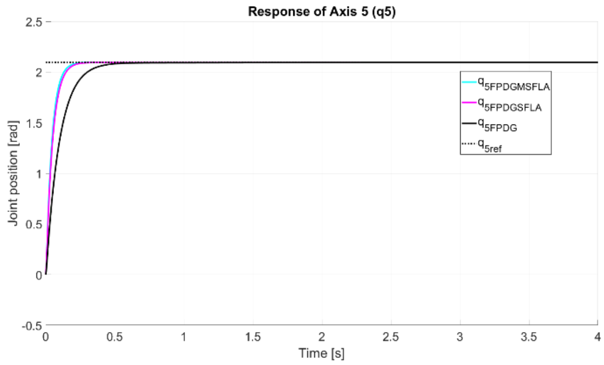

Fig. 9: Position response of the joint 5

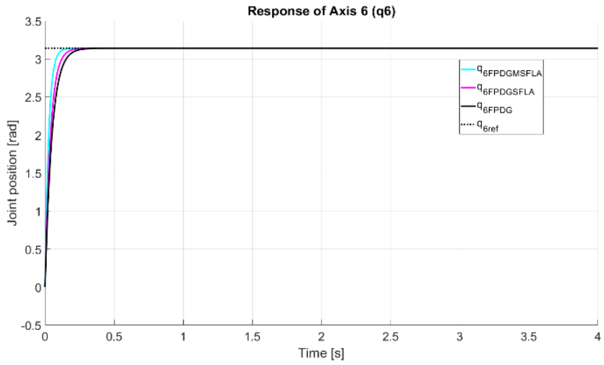

Fig. 10: Position response of the joint 6

Table 2. Comparing the response results of the controllers

		FPDGMS FLA	FPDGSF LA	FPDG
Axis 1	Settling time (sec)	1.7	2.7	3.6
	Overshoot (%)	0.02	0	0
	Error	0	0	0.01
Axis 2	Settling time (sec)	1.8	2.4	3.2
	Overshoot (%)	0	0	0
	Error	0	0	0.02
Axis 3	Settling time (sec)	2.1	2.6	3.4
	Overshoot (%)	0.0	0.0	0
	Error	0	0	0
Axis 4	Settling time (sec)	0.34	0.72	0.92
	Overshoot (%)	0	0	0
	Error	0	0	0
Axis 5	Settling time (sec)	0.15	0.16	0.38
	Overshoot (%)	0	0	0
	Error	0	0	0
Axis 6	Settling time (sec)	0.1	0.15	0.19
	Overshoot (%)	0	0	0
	Error	0	0	0

The results presented indicate that both the modified SFLA and the SFLA-based fuzzy PD+G controllers outperform the standard fuzzy PD+G controller, particularly in terms of reducing the settling time. The comparison reveals that the modified SFLA-based fuzzy PD+G controller achieves marginal improvements over the standard SFLA-based fuzzy PD+G controller, specifically in the regulation of axes 1, 2, 3, 4 and 5. The differences in performance for the remaining axes are negligible, with both controllers yielding nearly identical outcomes. This suggests that, for these cases, the parameters are already close to optimal, leaving limited room for further enhancement.

Furthermore, the slight edge provided by the modified SFLA-based fuzzy PD+G controller in certain axes demonstrates its potential for fine-tuning performance in more complex systems, where even minor improvements can be critical. This enhancement can be attributed to the adaptive nature of the modified SFLA, which allows for a more refined search in the parameter space, thus achieving better control precision. Overall, the findings support the effectiveness of the modified SFLA in optimizing the controller's performance, particularly in scenarios where high precision and fast response times are essential.

5. CONCLUSION

The paper proposed a self-adaptive shuffling mechanism to enhance the performance of the Shuffled Frog Leaping Algorithm (SFLA) for tuning fuzzy logic Proportional-Derivative with Gravity Compensation (PD+G) controllers in robotic manipulators. The self-adaptive mechanism dynamically adjusts the shuffling frequency and interval based on population diversity, improving the balance between exploration and exploitation throughout the optimization process. By incorporating these dynamic adjustments, the enhanced SFLA demonstrated superior convergence speed and accuracy compared to the standard SFLA, especially in complex and nonlinear optimization tasks like trajectory tracking in a 6-DOF robotic manipulator.

The results of simulation studies showed that the proposed method not only improves the precision of trajectory tracking but also provides a more robust solution against environmental disturbances and uncertainties. The dynamic tuning of fuzzy controllers via the self-adaptive SFLA offers a promising approach for real-time control applications where system parameters need to be efficiently adjusted in changing environments.

Future work could explore the application of the self-adaptive SFLA to other control systems and optimization tasks and further refine the adaptation criteria for more complex robotic configurations.

6. ACKNOWLEDGMENT

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

7. REFERENCES

[1] Spong, M., Vidyasagar, M. 2008. Robot Dynamics and Control, 2nd ed., Wiley.

[2] Kelly, R., Santibáñez, V., Loria, A. 2005. Control of Robot Manipulators in Joint Space, Springer.

[3] Zhang, Y., Kim, D., Zhao, Y., Lee, J. 2021. PD Control of a Manipulator with Gravity and Inertia Compensation Using an RBF Neural Network. In International Journal of Control, Automation and Systems, vol. 19, no. X, pp. 1-10.

[4] Soriano, L., Zamora, E., Vazquez-Nicolas, J., Hernández, G., Barraza Madrigal, J., Balderas, D. 2020. PD Control Compensation Based on a Cascade Neural Network Applied to a Robot Manipulator. Frontiers in Neurorobotics, vol. 14, article 577749.

[5] Zhang, Y., Zhao, Y., Zhang, S., Lee, J. 2019. Fuzzy-PD Controller with Online Gravity Compensation for 3-DOF Manipulator. In Proceedings of the 2019 16th International Conference on Ubiquitous Robots (UR), Jeju, Korea.

[6] Perrusquía, A., Flores-Campos, J., Torres-San-Miguel, C. 2020. A Novel Tuning Method of PD With Gravity Compensation Controller for Robot Manipulators. IEEE Access, vol. 8, pp. 114773-114783.

[7] Xi Hu et al. 2023. A Hybrid Clustered SFLA-PSO algorithm for optimizing the timely and real-time rumor refutations in Online Social Networks. Expert Systems with Applications, Volume 212.

[8] Lin, M.J., Luo, F., Xu, Y.G., Luo, L., 2013. A Novel Hybrid Optimization Method of Shuffled Frog Leaping Algorithm and Particle Swarm Optimization. AMR 717, 433–438.

[9] Duc Hoang Nguyen. 2015. A Hybrid SFL-Bees Algorithm. International Journal of Computer Applications. 128, 5, 13-18.

[10] Zhao, Z., Wang, M., Liu, Y., Chen, Y., He, K., & Liu, Z. 2024. A modified shuffled frog leaping algorithm with inertia weight. *Scientific Reports*, *14*(1), 1-14.

[11] Soriano, L. A., Zamora, E., M., J., Hernández, G., Barraza Madrigal, J. A., & Balderas, D. 2020. PD Control Compensation Based on a Cascade Neural Network Applied to a Robot Manipulator. Frontiers in Neurorobotics, 14, 577749.

[12] Zhang, Y., Zhao, Y., Zhang, S., Lee, J. 2019. Fuzzy-PD Controller with Online Gravity Compensation for 3-DOF Manipulator. In Proceedings of the 2019 16th International Conference on Ubiquitous Robots (UR), Jeju, Korea.

[13] Perrusquía, A., Flores-Campos, J., Torres-San-Miguel, C. 2020. A Novel Tuning Method of PD With Gravity Compensation Controller for Robot Manipulators. IEEE Access, vol. 8, pp. 114773-114783.

[14] Zhang, Y., Kim, D., Zhao, Y. et al. 2020. PD Control of a Manipulator with Gravity and Inertia Compensation Using an RBF Neural Network. Int. J. Control Autom. Syst. 18, 3083–3092.

[15] Nguyen, DH., Ngo, MD. 2016. Comparing Convergence of PSO and SFLA Optimization Algorithms in Tuning Parameters of Fuzzy Logic Controller. In: Duy, V., Dao, T., Zelinka, I., Choi, HS., Chadli, M. (eds) AETA 2015: Recent Advances in Electrical Engineering and Related Sciences. Lecture Notes in Electrical Engineering, vol 371. Springer, Cham.

[16] Candan, F., Dik, O. F., Kumbasar, T., Mahfouf, M., & Mihaylova, L. 2023. Real-Time Interval Type-2 Fuzzy Control of an Unmanned Aerial Vehicle with Flexible Cable-Connected Payload. Algorithms, 16(6), 273. https://doi.org/10.3390/a16060273

[17] Duc Hoang Nguyen. 2023. Modified DE-based Fuzzy PD Controller for UP6 Manipulator. International Journal of Computer Applications. 185, 38, 36-40.