Modified DE-based Fuzzy PD Controller for UP6 Manipulator

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

ABSTRACT
The paper proposes a method using modified differential evolution to tune the parameters of the fuzzy PD controller for regulating the MOTOMAN UP6 robot manipulator. In this paper, the modified differential evolution is based on a combination of DE mutation strategies: "DE/rand/1" and "DE/best/1". The proposed method is used to find the optimal parameters of a fuzzy PD controller. Simulated results show that the modified DE-based fuzzy PD controller improves settling time response significantly.

General Terms
Algorithms.

Keywords
Optimization, DE, Fuzzy, PD controller, Manipulator.

1. INTRODUCTION
Industrial manipulators are robotic arms that can perform various tasks in manufacturing, such as welding, painting, assembly, or pick-and-place. They consist of links and joints that can move in different directions and orientations. A typical industrial manipulator has six degrees of freedom (DOF), meaning it can move in three translations and rotations. A control system is required to generate appropriate joint torques based on the desired and actual joint angles to control the motion and position of the manipulator's end-effector.

Designing and implementing a control system for industrial manipulators is extremely challenging. Industrial manipulators are complex and unpredictable systems subject to numerous factors, including parameter variations, friction, backlash, and external disturbances. These factors can cause errors and instability in the system performance and reduce the accuracy and robustness of the control system. Therefore, conventional control methods, such as proportional-integral-derivative (PID) or adaptive controllers, may not be able to cope with these challenges and achieve satisfactory results [1-2].

To overcome these limitations, fuzzy logic and evolutionary algorithms have been proposed as alternative methods to design and optimize control systems for industrial manipulators. Fuzzy logic is a mathematical framework that can handle imprecision and uncertainty using linguistic terms and fuzzy rules instead of crisp values and equations. Fuzzy logic can capture human knowledge and experience in the form of fuzzy rules and incorporate them into the control system [3]. Evolutionary algorithms are population-based optimization techniques that can search for optimal solutions in complex and multimodal problem spaces by mimicking natural evolution.

This paper uses a fuzzy PD controller based on differential evolution (DE) to control the position and orientation of the UP6 manipulator. A fuzzy PD controller is a type of fuzzy logic controller that uses two inputs: the error and the change of error, and one output: the control action. The error is the difference between the desired and actual values of the joint angle of the manipulator, and the change of error is its derivative. The control action is a function of the error and the change of error, which are fuzzified by membership functions and then processed by a rule base. The output of the rule base is then defuzzified to obtain the final control action. A DE algorithm is a population-based optimization technique that uses a set of candidate solutions, called individuals, to search for the optimal solution in a given problem space. Each individual is represented by a vector of real numbers, called genes, that correspond to the parameters of the fuzzy PD controller, such as the scaling factors and quantization levels.

The remaining paper is organized as follows: Section 2 introduces the UP6 industrial manipulator and its dynamics, fuzzy PD controller structure. Section 3 presents an overview of the DE algorithm and modified DE. The description of how to tune the parameters of the fuzzy PD controller is given in section 4. Section 5 shows the obtained results, and Section 6 concludes this paper.

2. SYSTEM MODEL AND FUZZY PD CONTROLLER STRUCTURE
This section presents the mathematical model of the UP6 industrial manipulator, and the fuzzy PD controller structure is used to control its motion and position.

2.1 UP6 Manipulator Model
The industrial manipulator considered in this paper is the UP6 Motoman manipulator. This six-axis articulated robot arm can perform various tasks in manufacturing, such as welding, painting, assembly, or pick-and-place. The UP6 Motoman manipulator has six revolute joints, denoted by $q_1, q_2, q_3, q_4, q_5, q_6$ that can rotate around their respective axes [4].

The dynamic equations of UP6 robot can be derived from the Newton–Euler formulation, which is a method that applies the principles of Newton's second law of motion and Euler's equations of motion to each link of the robot. To obtain the dynamic equations of UP6 robot, these equations are applied to each of the six links of the robot, and then eliminate the internal forces and moments by using the principle of action and reaction. The resulting equations will relate the joint torques to the joint accelerations, and can be expressed in a matrix form as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \qquad (1)$$

where $M(q)$ is inertial matrix, $C(q,\dot{q})$ is Coriolis and centrifugal matrix, $G(q)$ is gravity vector, and τ is applied torque. These matrices and vectors depend on the mass, inertia, and geometry of each link of the manipulator [5].

2.2 Fuzzy PD Controller Structure

The fuzzy PD controller used to control the UP6 Motoman manipulator is a type of fuzzy logic controller that uses two inputs: the error and the change of error, and one output: the control action. The error is the difference between the desired and actual joint angles, and the change of error is its derivative. The control action is a function of the error and the change of error, which are fuzzified by membership functions and then processed by a rule base. The output of the rule base is then defuzzified to obtain the final control action as in Fig. 1.

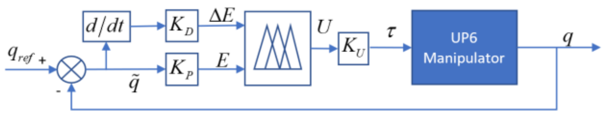

Fig. 1: Structure of fuzzy PD controller

The membership functions have triangular shapes and are chosen for inputs (E, DE) and outputs (U) of each axis controller, as in Fig. 2.

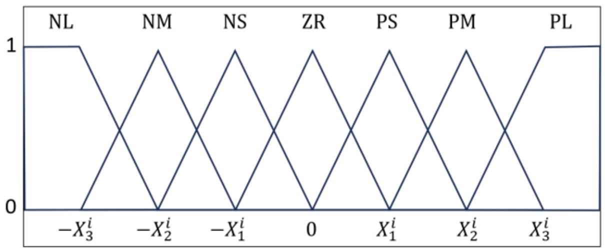

Fig. 2: Membership functions of inputs and output

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 DE algorithm. Details will be presented in section 4.

3. MODIFIED DIFFERENTIAL EVOLUTION

3.1 Overview of Differential Evolution

Differential evolution (DE) is an evolutionary algorithm, which mimics the natural process of evolution by maintaining a population of candidate solutions (called agents) and improving them over time. Each agent represents a possible solution to the optimization problem and has a fitness value that measures how good it is. The goal of DE is to find the agent with the highest fitness value, or the global optimum [6, 7].

The DE first generates a random initial population within the solution scope, then uses differential mutation, crossover, and selection operation to produce a new population generation [8]. The main steps of DE are as follows:

- ***Initialization:*** A random population of agents is created, each with a vector of parameters that define the solution.
- ***Mutation:*** For each agent, a new agent (called mutant) is created by adding a scaled difference between two randomly selected agents from the population as illustrated in Fig. 3.

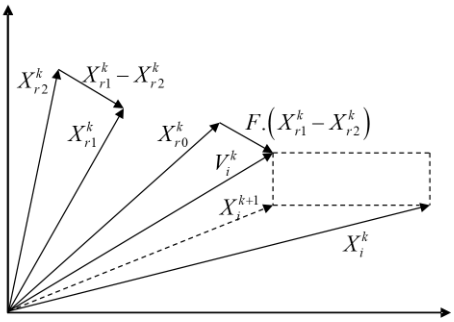

Fig. 3: Mutation process of DE

If X_{r0}^k is chosen randomly, it is mutation strategy “DE/rand/1”. If X_{r0}^k is selected best, it is mutation strategy “DE/best/1” [7]. These mutation strategies will introduce diversity and exploration in the search space.

- ***Crossover:*** For each agent, a new agent (called trial) is created by mixing some parameters from the original agent and some from the mutant. This allows recombination and exploitation of good features.

$$U_i^k = \langle U_i^k(j) \rangle = \begin{cases} \langle V_i^k(j) \rangle \ if \ rand(0,1) \le p_c \ or \ j = j_{rand} \\ \langle X_i^k(j) \rangle \ otherwise \end{cases} \quad (2)$$

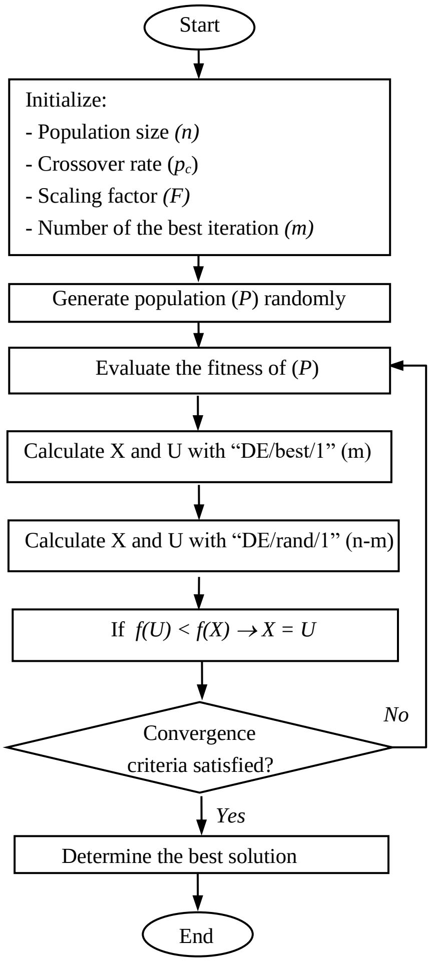

Fig. 4. Flowchart of the modified DE

3.2 Modified Differential Evolution

The idea of the modified DE is to combine two mutation strategies, "DE/rand/1" and "DE/best/1". The "DE/best/1" strategy creates new agents based on the best agent, considered the most promising agent in the solution space. Therefore, to increase the search accuracy, the mutant agents are created by the "DE/best/1" strategy with *m* iterations.

In addition, to provide the opportunity for the random generation of improved agent, random agents are generated based on "DE/rand/1" strategy. The flowchart of the modified DE is illustrated in Fig. 4.

4. TUNING PARAMETERS OF FUZZY PD CONTROLLER

This section presents the procedures for tuning the parameters of the Fuzzy PD controller using modified DE.

4.1 Modified DE-based Fuzzy PD Controller Structure

The structure of the modified DE-based fuzzy PD controller is shown as Fig. 5. The parameters of the fuzzy PD controller will be tuned by the modified DE algorithm.

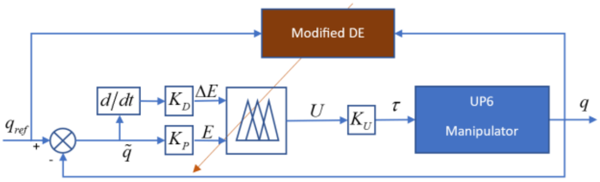

Fig. 5: Structure of Modified DE-based fuzzy PD controller

4.2 Tuning Parameters

Each fuzzy PD controller will have six parameters to be tuned, including three member function parameters X_1^i, X_2^i, X_3^i and three gains K_D^i, K_P^i, K_U^i (Fig. 2). Therefore, in total, there will be 36 parameters to be tuned. The modified DE optimization algorithm is used to determine these parameters to minimize the fitness function as follows.

- **Fitness Function**

The fitness function is used to evaluate the performance of each individual (fuzzy PD controller) based on some performance criteria that measure how well the fuzzy PD controller can control the UP6 Motoman manipulator. These performance criteria are:

- ***Tracking error:*** This criterion measures the difference between the desired and actual joint angles of the manipulator. It can be calculated as:

$$E = \sum_{i=1}^{6} \sqrt{\frac{1}{T} \int_0^T \left(q_{ref}^i - q^i\right)^2 dt} \tag{3}$$

where E is the tracking error, q_{ref}^i and q^i are the desired and actual joint angles for joint i, and T is the final time of simulation. The tracking error reflects how accurately and precisely the fuzzy PD controller can follow a given trajectory.

- ***Settling time:*** This criterion measures how fast the fuzzy PD controller can reach a steady state after a step change in input. It can be calculated as:

$$S = max_{i=1}^{6} t_i \tag{4}$$

where S is the settling time, and t_i is the time when joint i reaches within 5% of its final value. The settling time reflects how quickly and smoothly the fuzzy PD controller can respond to a change in input.

- ***Overshoot:*** This criterion measures how much the fuzzy PD controller exceeds its final value after a step change in input. It can be calculated as:

$$O = max_{i=1}^{6} \frac{q_{max}^i - q_{final}^i}{q_{final}^i} \times 100\% \tag{5}$$

where O is the overshoot, q_{max}^i and q_{final}^i are the maximum and final values of joint angle i. The overshoot reflects how stable and robust the fuzzy PD controller is to disturbances or uncertainties.

The fitness function that we use to optimize the fuzzy PD controller parameters is defined as a weighted sum of these performance criteria:

$$f(x) = \omega_1 E + \omega_2 S + \omega_3 O \tag{6}$$

where $f(x)$ is the fitness value of an individual x, and ω_1, ω_2, and ω_3 are positive weighting factors that balance the trade-off between different performance criteria. The goal is to minimize this fitness function by finding the optimal values of the fuzzy PD controller parameters.

- **Constraints**

To ensure the linguistic meaning and polarity of the parameters, the following constraints must be satisfied.

$$0 < X_1^i < X_2^i < X_3^i \leq 1 \tag{7}$$

$$0 < K_D^i, K_P^i, K_U^i \tag{8}$$

5. SIMULATION RESULTS

5.1 Simulation parameters

- **Weighting factors of the fitness function**

The fitness function is presented in section 4.2 determine how the performance of each solution is evaluated. In the simulation results below, the author focuses on improving the settling time, so choosing $\omega_1 = 0.3$, $\omega_2 = 0.5$, and $\omega_3 = 0.2$.

- **Modified DE parameters**

The parameter settings of modified DE are shown in Table 1.

Table 1. The Modified DE parameter settings

G	n	p_c	m	F
500	100	0.5	40	0.8

- **Termination Criteria**

The termination criteria used to stop the optimization process are based on two factors: the number of generations and the fitness value. In this paper, the author sets a maximum number of generations as a limit for the search process, which prevents it from running indefinitely or wasting computational resources. The number of iterations that fitness value does not change, which indicates that an acceptable solution has been found or that further improvement is unlikely. The termination criteria ensure that the optimization process ends when either a satisfactory solution is obtained, or enough iterations are performed.

5.2 Simulink model

Fig. 6 presents the closed loop controlling diagram using a fuzzy PD controller to regulate the angular position of six axes. The fuzzy PD controller includes six controllers, as in Fig. 7. Each controller has the configuration, as in Fig. 8.

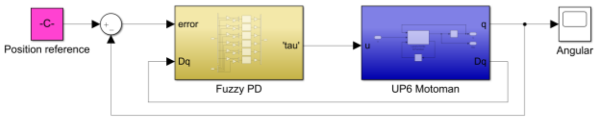

Fig. 6: Simulink schematic controlling the UP6 robot

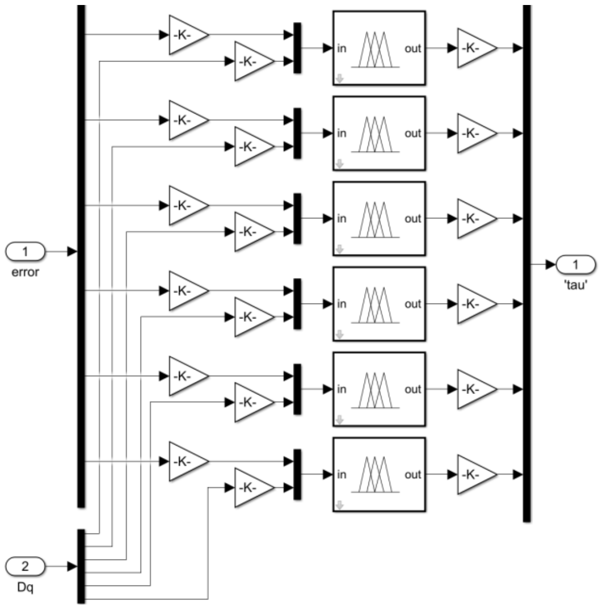

Fig. 7: Simulink model of 6 Fuzzy PD controllers

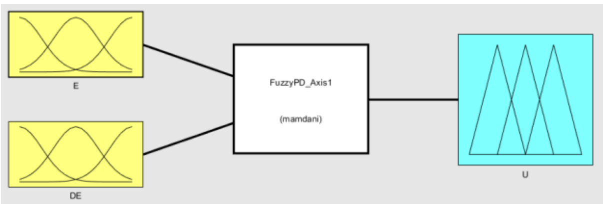

Fig. 8: Configuration of fuzzy PD controller

In the below simulation results, the controller will regulate the angular position of robot axes from the initial point of 0 to the final point of $q_{ref} = \left[\frac{\pi}{3}, \frac{\pi}{4}, \frac{\pi}{5}, \frac{\pi}{2}, \frac{2\pi}{3}, \pi\right]^T$.

Fig. 9 to Fig. 14 are the response results of UP6 axes with respect to controllers: modified DE-based fuzzy PD (FPDMDE), DE-based fuzzy PD (FPDDE) and fuzzy PD (FPD).

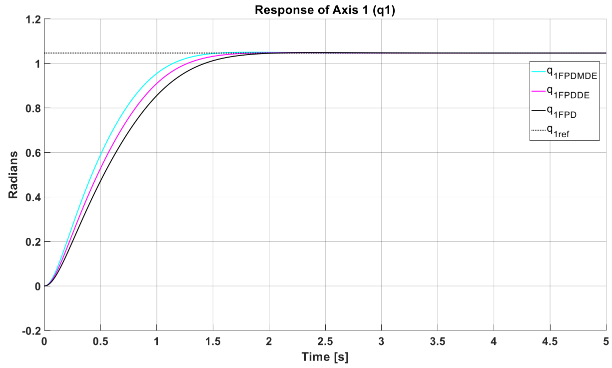

Fig. 9: Position response of the joint 1

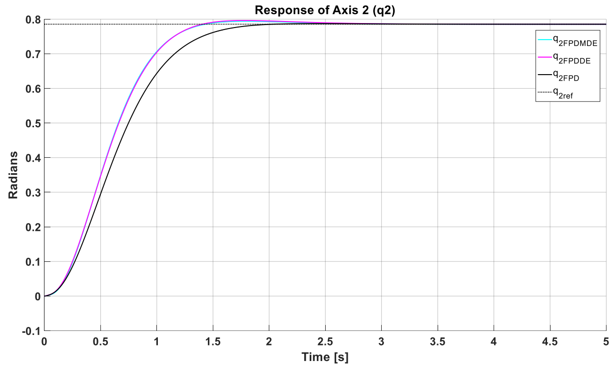

Fig. 10: Position response of the joint 2

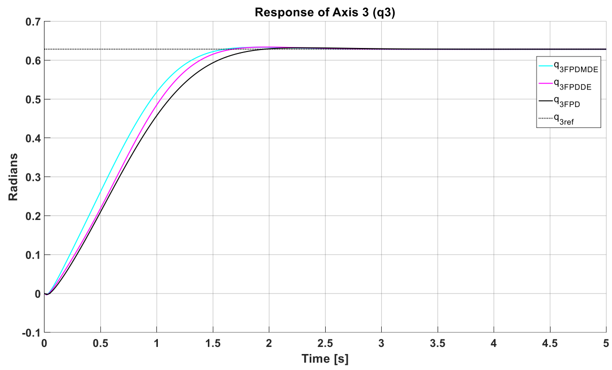

Fig. 11: Position response of the joint 3

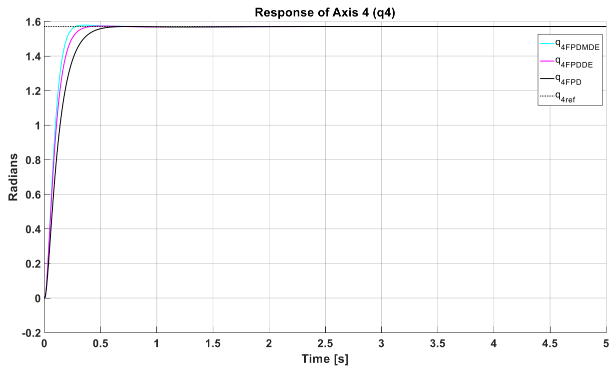

Fig. 12: Position response of the joint 4

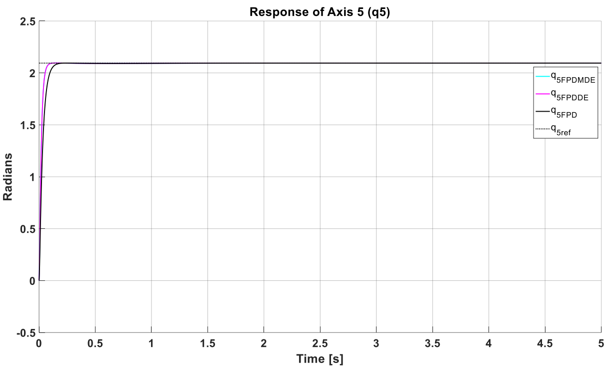

Fig. 13: Position response of the joint 5

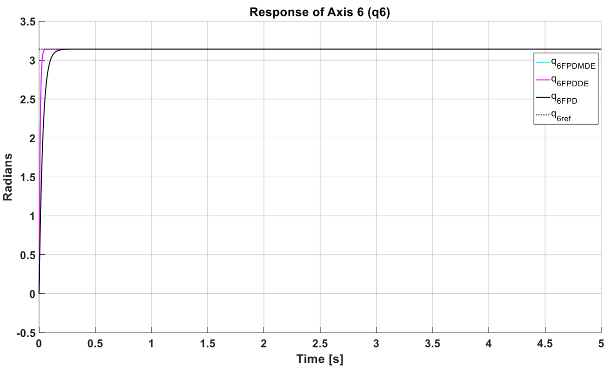

Fig. 14: Position response of the joint 6

Table 2. summarizes the results of setting time (2%), overshoot, and error to these controllers.

Table 2. Comparing the response results of the controllers

		FPDMDE	FPDDE	FPD
Axis 1	Settling time (sec)	1.29	1.44	1.62
	Overshoot (%)	0	0	0
	Error	0	0	0
Axis 2	Settling time (sec)	1.26	1.26	1.59
	Overshoot (%)	1.1	1.2	0
	Error	0	0	0
Axis 3	Settling time (sec)	1.4	1.5	1.7
	Overshoot (%)	0.7	0.7	0
	Error	0	0	0
Axis 4	Settling time (sec)	0.23	0.29	0.42
	Overshoot (%)	0.5	0.2	0
	Error	0	0	0
Axis 5	Settling time (sec)	0.07	0.07	0.12
	Overshoot (%)	0	0	0
	Error	0	0	0
Axis 6	Settling time (sec)	0.03	0.03	0.13
	Overshoot (%)	0	0	0
	Error	0	0	0

The above results show that the modified DE and DE-based fuzzy PD controllers give responses better than fuzzy PD controller, especially setting time. The modified DE-based fuzzy PD controller slightly improves results over that the DE-based fuzzy PD controller when regulating axes 1, 3, and 4. The remaining axes' results are almost the same. In these cases, the parameters are nearly optimal.

6. CONCLUSION

In this paper, the modified DE and DE algorithms are applied to tune the parameters of the fuzzy PD controller. This controller is then used to regulate the position of the UP6 robot manipulator. The simulated results show that the modified DE and DE-based fuzzy PD controllers outperform trial and error-based fuzzy PD controller regarding settling time. In addition, the modified DE-based fuzzy PD controller improves response results compared to DE-based fuzzy PD controller. In the future, the author will use benchmark functions to evaluate this modified DE based on the ability to find the optimal solution, convergence rate, etc..[9].

7. ACKNOWLEDGMENT

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

8. REFERENCES

[1] Mai, T.A., Dang, T.S. 2021. Optimal fuzzy PD control for a two-link robot manipulator based on stochastic fractal search. Eur. Phys. J. Spec. Top. 230, 3935–3945.

[2] 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.

[3] Urrea, Claudio, John Kern, and Johanna Alvarado. 2020. Design and Evaluation of a New Fuzzy Control Algorithm Applied to a Manipulator Robot. Applied Sciences 10, 7482.

[4] Industrial Robot MOTOMAN-UP6, MOTOMAN ROBOTICS EUROPE AB.

[5] Duc Hoang Nguyen. 2018. Dynamic Parameter Identification of UP6 Robot Manipulator using SFLA. International Journal of Computer Applications 182.

[6] Storn R, Price K. 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359

[7] S. Das and P. N. Suganthan. 2011. Differential Evolution: A Survey of the State-of-the-Art. IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 4-31.

[8] Zhu, L., Ma, Y. & Bai, Y. 2020. A self-adaptive multi-population differential evolution algorithm. Nat Comput 19, 211–235.

[9] A. Baştuğ and C. Karakuzu. 2017. Comparison of heuristic algorithms with performance metrics on multimodal benchmark functions. International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, pp. 1078-1083.