SEMARAK ILMU
PUBLISHING
202103268166(003316878-P)

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

PID Controller Parameter Tuning Based on a Modified Differential Evolutionary Optimization Algorithm for the Intelligent Active Vibration Control of a Combined Single Link Robotics Flexible Manipulator

Abbas Moloody[1], Azizan As'arry[1,*], Tang Sai Hong[1], Raja Kamil[2], Ali Zolfagharian[3]

[1] Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia
[2] Department of Electrical and Electronics Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia
[3] School of Engineering, Faculty of Science Engineering and Built Env, Deakin University (DU), Geelong, Victoria 3220, Australia

ARTICLE INFO

Article history:
Received 28 March 2024
Received in revised form 16 July 2024
Accepted 29 August 2024
Available online 1 October 2024

Keywords:
Modified differential evolutionary optimization algorithm (MDEOA); Intelligent active vibration control (IAVC); Combined single link robotics flexible manipulator (CSLRFM)

ABSTRACT

This paper introduces some of the various techniques of vibration control and optimization for the purpose of vibration reduction and balancing. Here, in this research by comprising three of the most effective variational techniques now, a Modified Differential Evolutionary Optimization Algorithm (MDEOA) method is suggested to handle the challenge of adjusting the PID controller parameters for the Intelligent Active Vibration Control (IAVC) of a Combined Single Link Robotics Flexible Manipulator (CSLRFM) in order to reduce the undesired effects of vibration. The Crossover Probability Factor (CPF) as the Certain Ratio (CR) and the Mutation Factor (MF) of the algorithm are gradually altered during algorithm iteration to enhance the method's performance during optimization. On this foundation, the PID controller parameter tuning and the issue of CSLRFM mechanical vibrations are addressed using the MDEOA method. This research suggests an evolutionary algorithm that incorporates the variational techniques mentioned above, which will be combined by a certain ratio, and the specific computational procedure. In this strategy, a Strictly Bearish Distributed Exponential Function (SBDEF) has been used as the main target and the criteria and indicators for evaluating and measuring the optimal performance of differential evolution are the Integral Absolute Error (IAE) rate and the PID controller parameter values. According to simulation findings, the technique can be used to optimize the PID controller parameters settings for the IAVC of the CSLRFM and a reduction in the mechanical vibrations. Simulation results illustrate the effectiveness of the proposed MDEOA strategy which is significantly and quite satisfactory about 25 to 30 (%) better than comparing to the other algorithms in improvement stabilization and vibration control of CSLRFM.

* *Corresponding author.*
E-mail address: zizan@upm.edu.my

https://doi.org/10.37934/araset.52.1.234258

1. Introduction

For many years, attempts have been made to preserve mechanical vibration factors and regulate mechanical vibration, as the structures may be damaged or system performance may be harmed as a result of the undesired vibration [1]. Vibration reduction is a serious concern in the usage of flexible structures, especially in the robotics and aerospace sectors, which typically utilize lightweight, low-absorption flexible structures for their vital and fundamental models. A manipulator as a flexible structure is a mechanical component made up of many segments that may be applied for a variety of tasks [2]. Robotics Flexible Manipulators (RFMs) are in high demand to replace humans in tough vocations, routine tasks, and dangerous situations in order to gain and achieve faster, more cost-effective, and more precise operations. The accurate positioning of the manipulator's tip is challenging due to the oscillations. In the 1970s, work on the dynamics, mathematical modelling, analysis, and control of the flexible mechanism really got going. Only a few of the feedback control techniques that have been studied for accurate positioning and vibration control of single link robotics flexible manipulators include Pole Placement (PP) [3], Lyapunov-Based Control (LBC) [4] and Integral Resonance Control (IRC) [5]. Since its origin, Artificial Intelligence (AI) techniques including Fuzzy Logic (FL), Neural Networks (NN), and process identification and control have been shown useful in a wide range of fields and applications [6]. PID controllers are extensively employed in process control for industrial processes because of their benefits of simplicity, ease of installation, and resilience [7]. This leads to the development of the equivalency issue between a Fuzzy Logic PID (FL-PID) controller and a traditional (PID) controller. Flexible structures have been utilized with effective vibration control methods and approaches, such as Differential Evolution Optimization (DEO), to achieve the required vibration suppression for precise accuracy. The MDE method is used in this study effort to maximize the gains of a PID controller for vibration suppression in the flexible beam. The PID controller parameter tuning is based on a Modified Differential Evolutionary Optimization Algorithm (MDEOA) for the Intelligent Active Vibration Control (IAVC) of a Combined Single Link Robotics Flexible Manipulator (CSLRFM). The PID controller is utilized to provide the control signal that is delivered to the flexible beam in order to dampen vibrations. The flexible beam is treated as a distributed parameter system. In this case, the major purpose is to combine two of the most efficient variational approaches now available to address the issue of altering and optimizing the PID controller settings of a CSLRFM as the nonlinear system. The goal of the optimization problem is to reduce the beam's vibration levels while decreasing the energy used by the active vibration control system. The MDE algorithm is used to search for the optimal PID gains that achieve these objectives, while considering the constraints and limitations of the system, such as the maximum control signal amplitude, the frequency range of the vibrations, and the physical properties of the beam. The main target is addressed for the issue of mechanical vibration control for eliminating and minimizing the deflection and oscillation angle of the End Effector of the beam for better performance in the vibration control and suppression, and keeping the rotation angle at a desired and acceptable positional accuracy in a horizontal plane motion. Classical vibration control is the process of designing controllers that may lessen or completely get rid of undesirable vibrations in mechanical systems using approaches from classical control theory, Figure 1 [8].

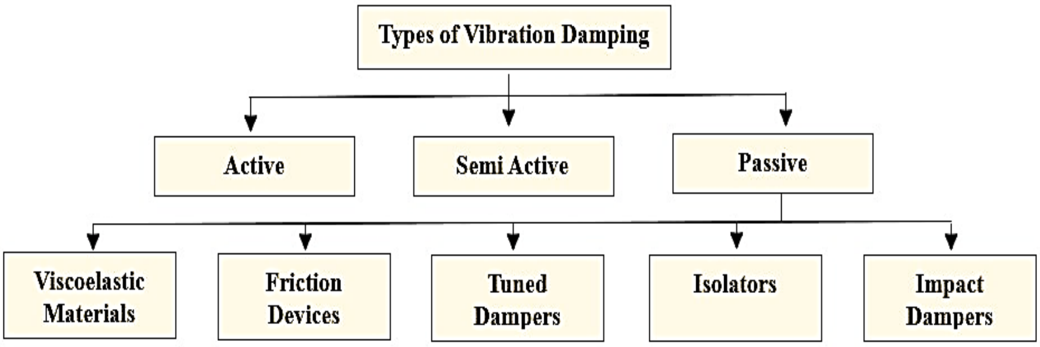
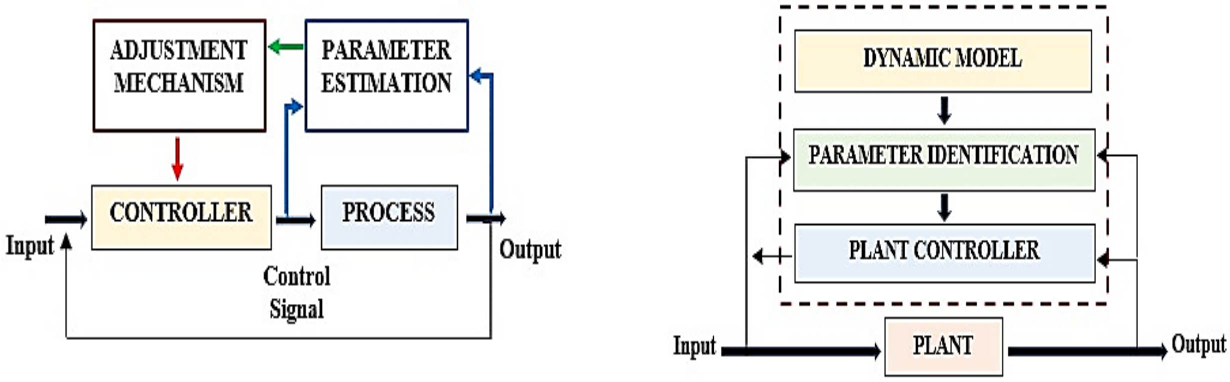


Fig. 1. Flow chart for types of classical vibration control [8]

Automatic and self-tuning vibration control are advanced vibration control techniques that use modern control theory and signal processing techniques to design controllers that can automatically adapt to changes in the system dynamics and disturbances, Figure 2 [9].



(a) Self-tuning regulator (b) Self-tuning controller

Fig. 2. Block diagram of a self-tuning regulator (a) and a self-tuning controller (b) [9]

Active Force Control (AFC) loop corrects for the disturbance force discovered by comparing the ideal and real force vectors incorrectly [10]. The following schematic loop, Figure 3, may be used to create the AFC controller of the suspension system. where D, F, and derivatives are the estimated disturbance Force, measured force, and body acceleration respectively.

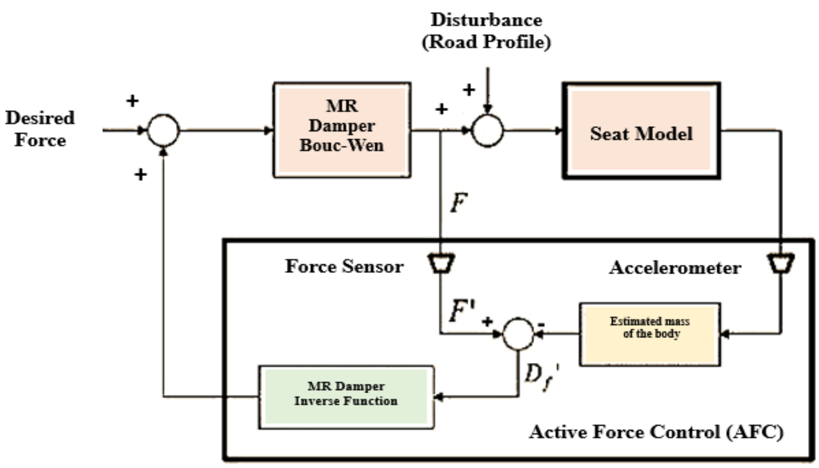


Fig. 3. AFC scheme loop [12]

Active Vibration Control (AVC) consist a set of techniques used to reduce or eliminate unwanted vibrations in mechanical systems by applying a control force that is in opposition to the vibration, Figure 4 and Figure 5 [11].

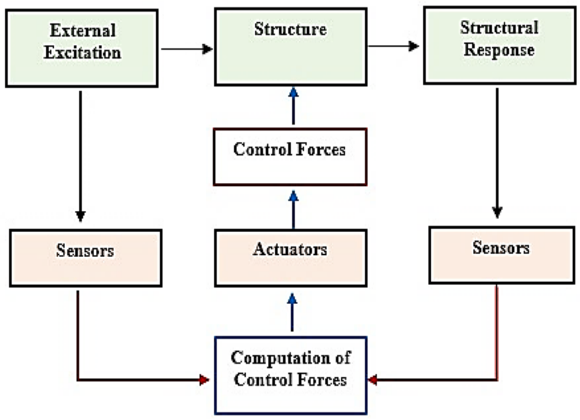


Fig. 4. General idea of AVC system (a) and principles of signal operation (b) [11]

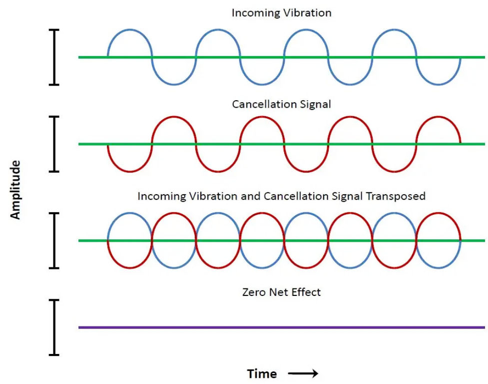


Fig. 5. Principles of signal operation (b) [11]

The principles of natural evolution serve as the foundation for the class of optimization algorithms known as Evolutionary Algorithms (EA). Complex optimization issues that are challenging to solve using conventional techniques are solved utilizing these algorithms [13]. Differential Evolution (DE) is one kind of evolutionary algorithm used to address optimization issues and effectiveness in resolving challenging optimization issues [14]. Five variation strategies commonly used by DE [18]. Summary of studies related to modifying DE mutation strategy shown in Table 1.

Table 1
Summary of studies related to modifying DE mutation strategy

No	Name	Subject	Problem
1	Kaelo & Ali	Electromagnetism concept in mutation factor	Using a complex method of generating mutation factor
2	H.-Y. Fan & Lampinen	Trigonometric mutation scheme	Using random numbers to generate a mutation strategy
3	Lilla *et al.,*	DE/rand/1/either-or	algorithm generates the mutant vectors with a probability
4	Q. Fan & Yan	DE/current-to-gr_best/1 scheme	Complex method combines the mutant vector with the crossover
5	F. Zhao *et al.,*	A hybrid algorithm	-
6	Das *et al.,*	Neighbourhood-based mutation strategy comparing the effect of	Complex methods combine two strategies: neighbourhood based mutation
7	Gokul *et al.,*	Combination of the two categories of DE modification	the self-adaptation strategy and modifying DE mutation strategy
8	Yu *et al.,*	Faster convergence mutation strategy that uses	The greedy logic based is a random vector strategy
9	Xiang *et al.,*	New DE mutation strategy that combines DE/current/1	combines two mutation strategies and also uses randomization to find the mutant factor

2. Methodology

2.1 Modified Differential Evolution Optimization Algorithm (MDEOA)

The Modified Differential Evolution (MDE) method is a development of the traditional Differential Evolution (DE) algorithm with the goal of enhancing its performance in handling challenging optimization issues. Das and Suganthan first presented MDE in 2011, and it has subsequently been used to solve several optimization issues in other disciplines. The mutation, crossover, and selection processes of the MDE algorithm are used to create a population of candidate solutions. In order to better adapt to the task at hand, MDE makes several improvements to the DE algorithm, starting with a dynamic mutation strategy that adjusts the mutation strategy based on the success of prior mutations. Second, self-adaptive control parameters. The crossover operation in MDE adjusts the probability distribution using self-adaptive control parameters, which can result in better convergence. Third, adaptive mutation rate, which allows for improved exploration and utilization of the solution space by adjusting the mutation rate based on the success of prior mutations. In several benchmark optimization issues, MDE has been demonstrated to outperform the traditional DE technique and other cutting-edge optimization algorithms. Compared to the conventional DE technique, MDE may be more difficult to develop and call for more processing resources. The difficulty of modifying the PID controller settings for the Intelligent Active Vibration Control (IAVC) of a Combined Single Link Robotics Flexible Manipulator (CSLRFM) in order to mitigate the unfavourable impacts of vibration is addressed in this research study by combining two of the most effective variational approaches currently available. In order to improve the algorithm's performance during optimization, the crossover probability factor (CR) and mutation factor are gradually changed during algorithm iteration. As the stages of the optimization process, in this article. The definition of the optimization issue, including the search space, restrictions, and objective function, comes first. In most cases, the objective function is a multi-objective cost function that considers both the effectiveness of vibration suppression and the energy usage of the active vibration control system. The second stage is to use partial differential equations to represent the flexible beam as a distributed parameter system. This model is employed to predict how the beam will behave under various operating scenarios and to assess how well the PID controller will manage vibrations. The third stage involves setting the control gains to some initial values and implementing the PID controller in hardware or software. The MDE technique is then used to conduct an optimization process to find the best PID gains that minimize the objective function while meeting the restrictions. By performing mutation and crossover operations on the current solutions and choosing the best ones for the following generation, the algorithm iteratively creates new candidate solutions. The final step is to evaluate the outcomes by updating the control signal applied to the beam using the optimal PID gains acquired from the optimization process. By comparing the system's energy use and vibration levels before and after the optimization, the performance of the improved PID controller is assessed. The optimization procedure can be repeated if necessary to boost the PID controller's performance even further.

2.2 Mutation Scheme Modification of MDEOA

Numerous methods exist for enhancing differential evolution algorithms, and almost all of them try to compromise between local exploitation and global search capabilities. The DE/rand/1 variation strategy is the most widely used and beneficial for sustaining population variety, according to a detailed assessment of the most recent literature [16], which also includes all variation strategies for the DE algorithm. However, DE/best/2 has the best solution [17,18] which is more useful for addressing specific algorithmic technical concerns and accelerating the algorithm's convergence rate. Due to the ease with which the variational strategy with the most information might easily fall into the local optimum, they all construct a modified differential evolutionary algorithm with two variational strategies. Based on the above, this research proposes an evolutionary algorithm that combines the two variational approaches stated above in a certain ratio and goes as follows in terms of computing process:

$$M_i^1(time+1) = [N(K)* [(x_{r2}(time)-x_{r3}(time))]] + [x_{r1}(time)] \tag{1}$$

$$M_i^2(time+1) = [N(K)*[(x_{r1}(time)-x_{r2}(time))]] + [x_{best}(time)] \tag{2}$$

$$M_i(time+1) = [[(1- \zeta)] * [M_i^2(time+1)]] + [[\zeta]* [M_i^1(time+1)]] \tag{3}$$

$$M_1(time+1) = [N(K)*[(x_2(time)-x_3(time))]] + [x_1(time)] \tag{4}$$

$$M_2(time+1) = [N(K)*[(x_1(time)-x_2(time))]] + [x_{best}(time)] \tag{5}$$

$$M(time+1) = [[(1- [\zeta])] *[M_2(time+1)]] + [[\zeta]*[M_1(time+1)]] \tag{6}$$

Where the amount of DE/rand/1 in the variation strategy is represented by Gamma in Eq. (3) and Eq. (6) the proportion of the two variant strategies in the final variance strategy is altered to better balance the global search capability and convergence speed for various issues by adjusting the value of [ζ] as Zita.

2.3 Crossover Scheme Modification (CR) of MDEOA

The crossover probability factor (CR) of MDEOA, which also balances the efficacy of local and global search, may be used to control each randomly selected mutation vector's involvement in the crossover. The crossover probability factor Cr is frequently selected to have a range of [0,1]. The pace of convergence will be too slow and the effect of individual disturbances will be amplified if the selection is too big. If it is too minimal, the population variety will be reduced and early convergence will be simple. In conventional differential evolution, the crossover probability factor CR will pick a fixed value that disregards population changes during recurrent development. The population's rate of convergence would be accelerated if, as the number of repeats increased, the crossover probability factor (CR) gradually increased and the variation factor (VF) gradually decreased. The following formula can be used to alter the crossover probability Cr:

$$CR = C_r = ŋ =1/ [ʎ_{max}/ [ŋ_{min} [ʎ_{max} - ʎ] + [ŋ_{max} * [ʎ]]]] \tag{7}$$

2.4 Variance Factor (F) of MDEOA

Variation Factor (VF), which also affects the population's diversity and convergence, is the main factor controlling the search phase of the differential evolution process. Population variety increases during population evolution when the Mutation Factor (MF) falls, increasing the likelihood that the algorithm may depart from the extreme value while slowing convergence. In contrast, population diversity rises as Mutation Factor (MF) increases, which makes it more probable for the algorithm to "leap out" of the extreme value. The conventional differential evolution method cannot fully leverage the characteristics of each stage of algorithm evolution since the variation factor f normally assumes a constant value between [0,2]. As a result, the number of rounds determines how the variation factor is calculated in this study.

$$R(K) = e^{[[K-1]/\,[K-Km-1]]} \tag{8}$$

$$N(K) = [[[R(K)] * [(2N_{ave})]] + N_{min}] \tag{9}$$

In Eq. (8), K stands for the current iteration count while K_m for the maximum iteration count. in Eq. (9), N(K) stands for the value of the current iteration count's Variation Factor (VF) while N_{ave} and N_{min} stand for the factor's average and minimum values, respectively.

2.5 Optimization Methodology of MDEOA

The DE method is introduced to address the genetic algorithm's primary drawback, namely the absence of local search in this approach. The Selection Operators are where the genetic algorithm and the DE algorithm diverge most. However, the DE algorithm gives each response an equal chance of being chosen. That is, once a new response is created using a mutation and crossover operator, the new answer is compared with the prior value and replaced if it is superior. As a result, the likelihood of being chosen does not depend on their merit value. Unlike other algorithms, the differential evolutionary algorithm first performs the crossover operator and then the mutation operator in such a manner that the mutation operator is applied first and the crossover operator is used later to produce a new generation. The mutation operator is used without the need of a specific distribution, and the duration of the mutation step is equal to the separation between the present members. As a result, the population is first generated using a uniform distribution, whose members are scattered throughout the space evenly. As the DE algorithm advances, these members get closer to one another, and this convergence eventually produces the best result. It should be noted that a large population can aid in the discovery of the ideal solution, which is why the initial population in this case is generated as shown in Figure 6, Figure 7 and Figure 8.

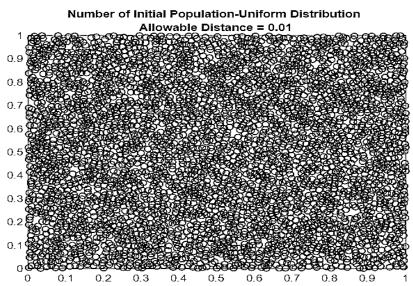


Fig. 6. Number of initial populations as a uniform distribution with allowable distance = 0.01. in MDEOA

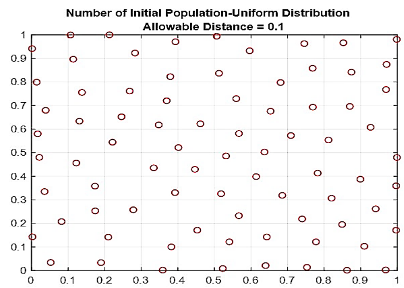


Fig. 7. Number of initial populations as a uniform distribution with allowable distance = 0.1. in MDEOA

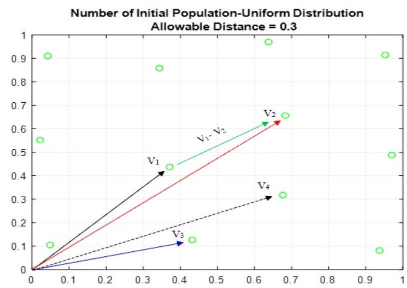


Fig. 8. Number of initial populations as a uniform distribution with allowable distance = 0.3. in MDEOA

The scale factor's proper value is one of the key considerations in this method; if it is set to be small, the jump operator's step sizes will be shorter and more time will be required for searching. Additionally, the differential evolution method fails to evaluate appropriate solutions when the scale factor is big. Therefore, it is crucial to calculate this coefficient with considerable care. After the mutation, crossover is carried out in a fashion that generates a random number between 0 and 1; if the produced number is less than the crossover rate, the desired element in that member of the population is removed from the mutation part; otherwise, the desired element is removed from the member's starting value. This procedure is continued until all of a member's members are either selected from the modified part or from their original value. The newly created matrix is then compared to the old matrix, and if the new matrix is more affordable, it takes the place of the original matrix. Every person in the population receives this treatment. The error (IAE) rate and the PID controller parameter values in this method serve as the criteria and indicators for assessing and monitoring the optimal performance of differential evolution. As a result, Figure 9 shows that a distributed exponential function has been employed to get the best performance in this technique, which is a continuous distribution, has the following probability density function,

$$f(x; \Lambda) = \Lambda\, e^{(-\Lambda x)}{}_{,\ x \geq 0} \quad (10)$$

$$f(x; \Lambda) = 0_{,\ x < 0} \quad (11)$$

where the parameter (Λ) is the inverse of the mean (mathematical expectation) of the distribution and this distributed exponential function is a strictly descending function so that it can perform the objective function which is the optimization and minimization of the error rate with the optimal performance of the algorithm, Figure 10.

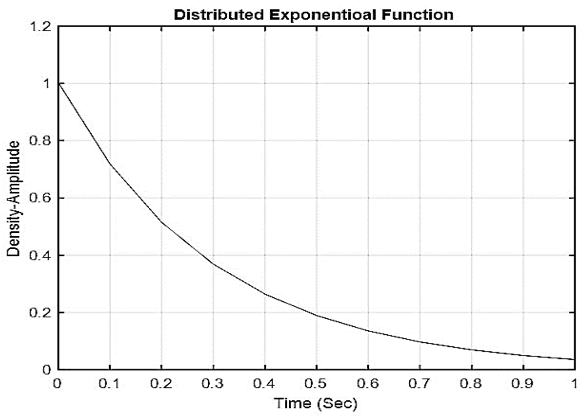


Fig. 9. Distribution exponential function of MDEOA.

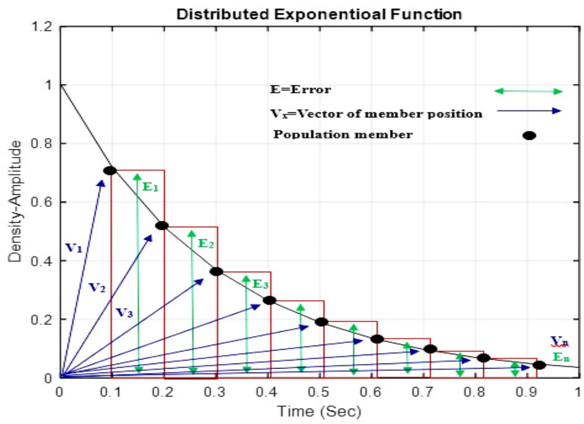


Fig. 10. Distribution exponential function and mutation scheme of MDEOA.

If the created population's members can be positioned on this exponential distribution function and the concentration of population production is concentrated at the lower end of the distribution function, then the error rate as the goal function will be as low as feasible in the differential evolution strategy and algorithm provided in this research. is feasible, bringing PID control parameter values closer to their ideal values and ultimately resulting in proper performance. The accuracy of the topic is demonstrated by Figure 11, Figure 12, Figure 13, Figure 14, Figure 15 and Figure 16.

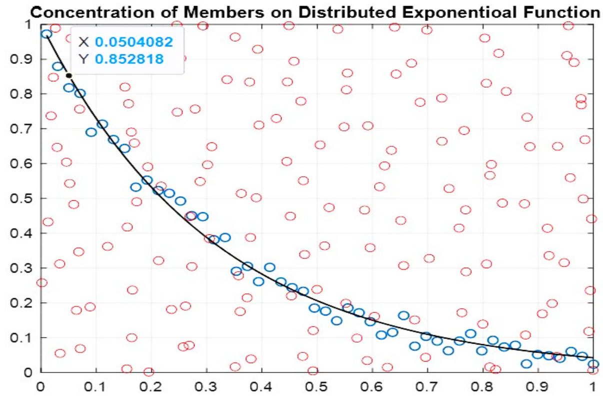


Fig. 11. Concentration of members on distribution exponential function of MDEOA

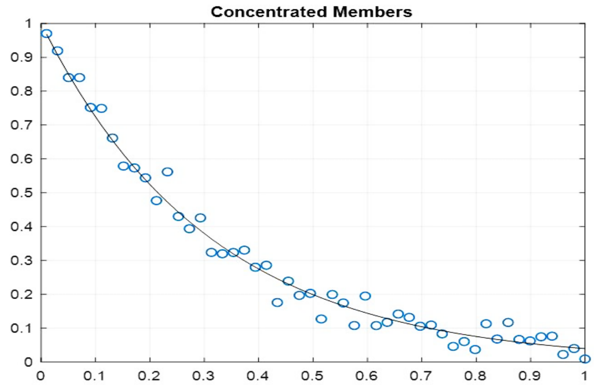


Fig. 12. Concentrated of members on distribution exponential function of MDEOA

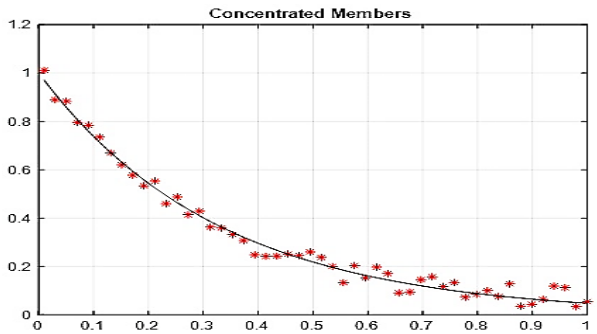


Fig. 13. Concentrated of members of MDEOA

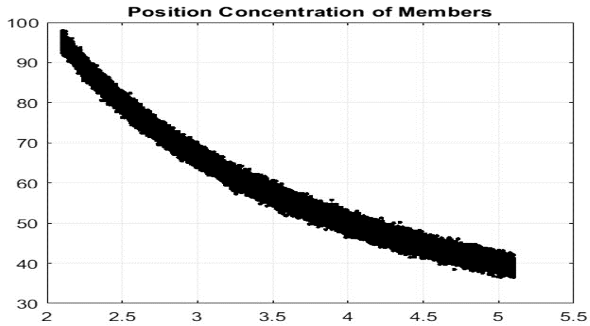


Fig. 14. Position concentrated of members of MDEOA

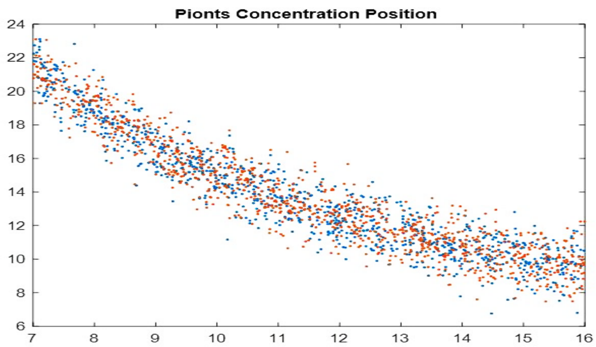


Fig. 15. Points concentrated of members of MDEOA

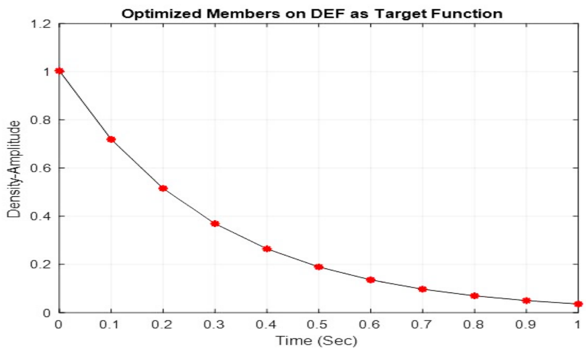


Fig. 16. Optimized and concentrated members on DEF as target function

2.6 Flowchart of MDEOA

The flow chart of MDEOA algorithm is shown as Figure 17.

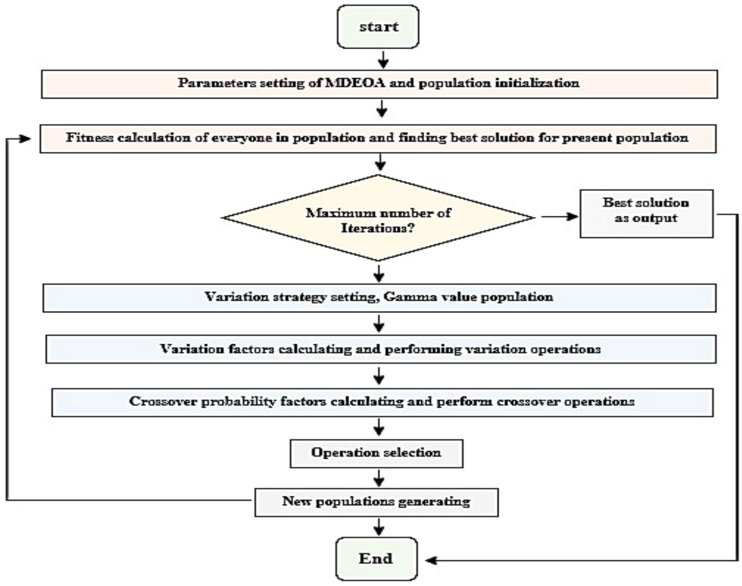


Fig. 17. Flowchart of MDEOA

2.7 Evaluating of MDEOA

According to the findings in Figure 18 to Figure 26, the modified DE (MDEOA) was evaluated using MATLAB in two different ways for this study. First, Liang investigated the effectiveness of the suggested method using three mathematics issues from CEC'14. The total number of iterations for all experiments was 30, with a 200-iteration cap. The settings for each experiment are shown in Table 2 to Table 4. The MDEOA and Classic DE methods' MATLAB code has been altered, and both algorithms utilized the same initialized population. The results were then shown as graphs using the MATLAB plot function with the best fitness value, best location, and iteration. The identical mathematical problem parameters were then provided to the algorithms. PlatEMO, a MATLAB evolutionary optimization platform developed by Cuate for testing optimization algorithms, was utilized to evaluate the improved DE. PlatEMO is a well-known practical program that allows users to test 345 optimization problems, choose an algorithm from 176 methods, change the parameter values, and obtain static and graphical data to assess the effectiveness of the algorithms in solving the preset various mathematical problems by Tian.

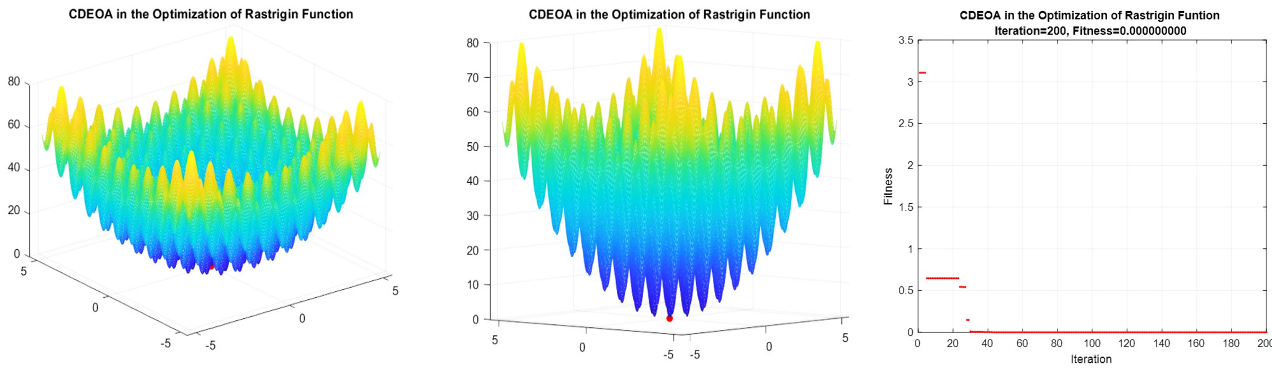


Fig. 18. The CDEOA in the optimization of Rastrigin function with the parameters as shown in Table 2

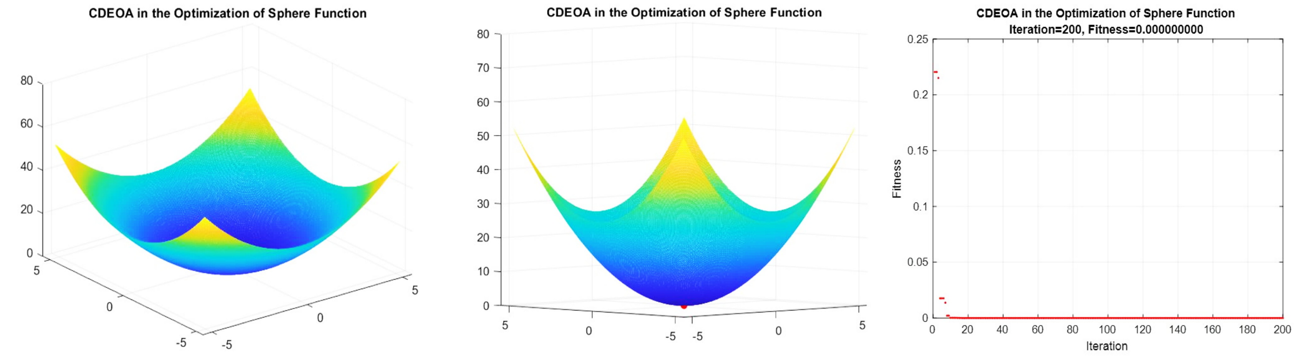


Fig. 19. The CDEOA in the optimization of Sphere function with the parameters as shown in Table 2

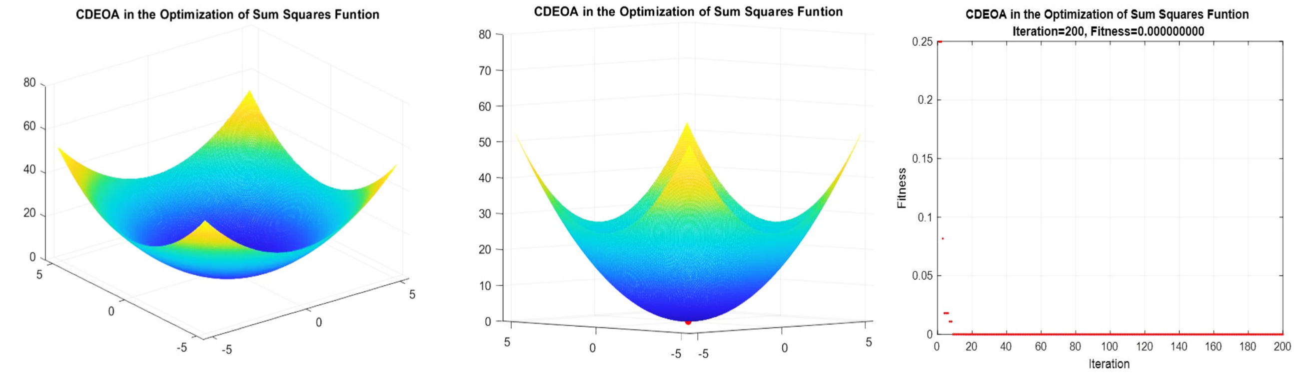


Fig. 20. The CDEOA in the optimization of Sum Squares function with the parameters as shown in Table 2

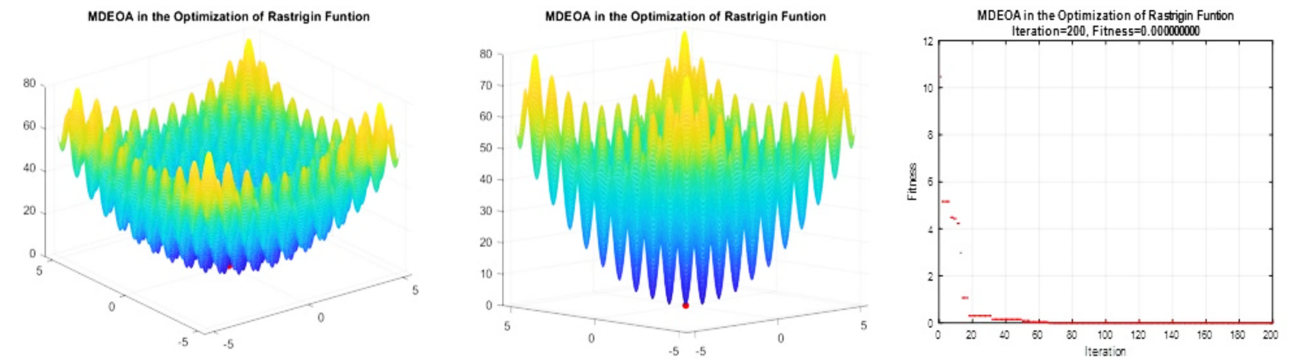


Fig. 21. The MDEOA in the optimization of Rastrigin function with the parameters as shown in Table 2

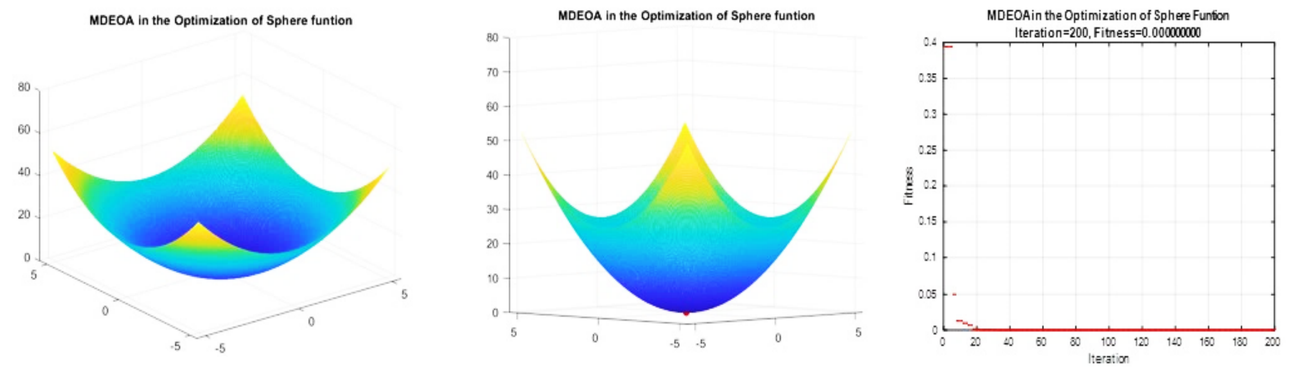


Fig. 22. The MDEOA in the optimization of Sphere function with the parameters as shown in Table 2

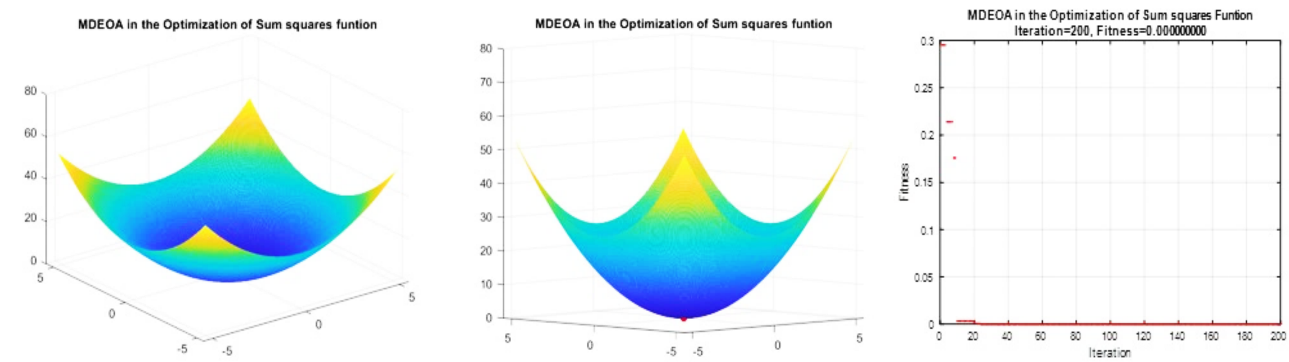


Fig. 23. The MDEOA in the optimization of Sum Squares function with the parameters as shown in Table 2

Table 2
Results comparison of 3 mathematical problems using the CDEOA and MDEOA

Fixed Parameters Set	CDEOA			MDEOA		
F=0.5, C=0.9, D=2, NP=50,	Rastrigin Function	Sphere Function	Sum square's Function	Rastrigin Function	Sphere Function	Sum square's Function
Maximum Iteration	200	200	200	200	200	200
Minimum Iteration	30	10	10	15	5	5
Maximum Fitness	3.1	0.23	0.25	10.5	0.4	0.3
Minimum Fitness	0	0	0	0	0	0

Table 3
Results percentage of the CDEOA improvement over MDEOA in minimum iteration

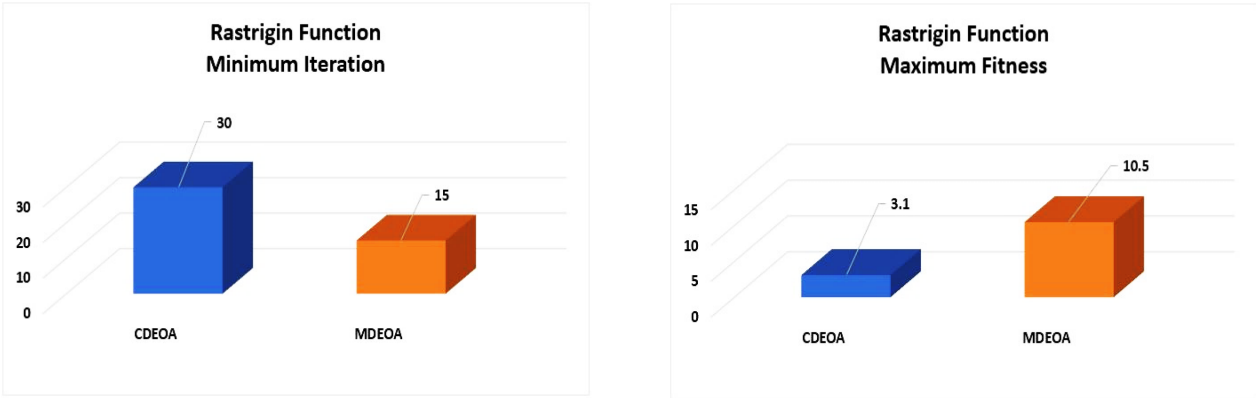
Function	Minimum Iteration Improvement in CDEOA
Rastrigin	50%
Sphere	50%
Sum square's	50%

Table 4
Results percentage of the CDEOA improvement over MDEOA in maximum fitness

Function	Maximum Fitness Improvement in CDEOA
Rastrigin	29.50%
Sphere	57.50%
Sum square's	83%

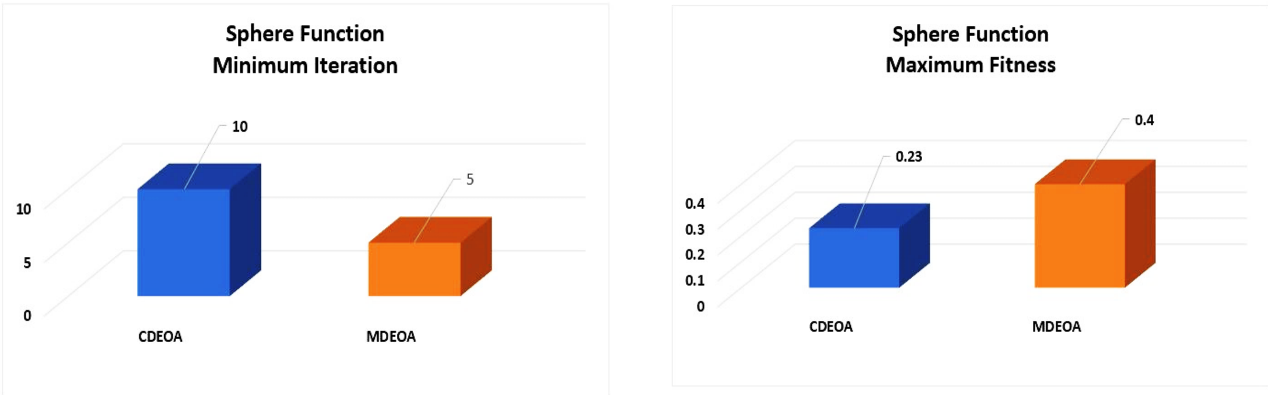
The results from Table 2 to Table 4 will be evaluated using the improvement calculation to understand how much the difference between the modified DE and the other algorithms. The improvement percentage is calculated using the following formula where (I) is the improvement, (V_a) is the value before, (V_b) is the value after.

$$I = (V_a/V_b) \times 100 \tag{12}$$



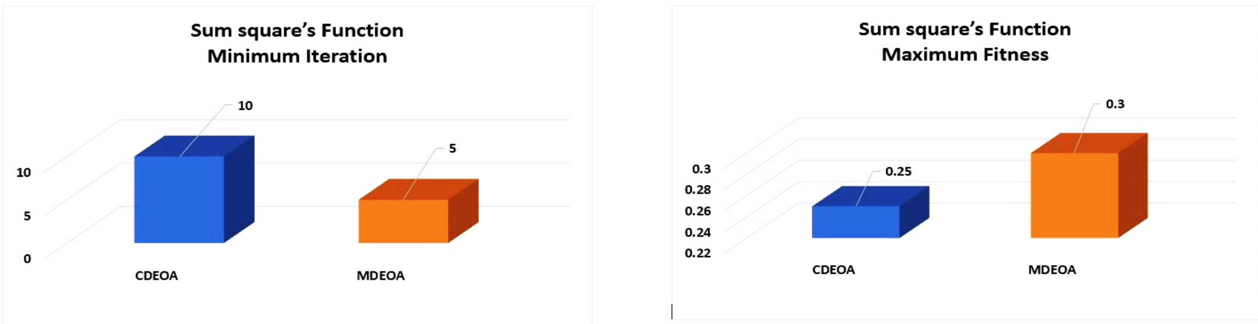
(a) Minimum Iteration (b) Maximum Fitness

Fig. 24. Results comparison of the CDEOA and MDEOA in the optimization of Rastrigin function



(a) Minimum Iteration (b) Maximum Fitness

Fig. 25. Results comparison of the CDEOA and MDEOA in the optimization of Sphere function



(a) Minimum Iteration (b) Maximum Fitness

Fig. 26. Results comparison of the CDEOA and MDEOA in the optimization of Sum square's function

2.8 Control Block Diagram Framework of MTRS

The combination of the Combined Plant (CP) and the NGCAM as the Combined Actuator (CA) has been considered as the Mechatronics Test Rig System (MTRS) with three degrees of freedoms (θ_1), (θ_t) and (θ_g) which the input and output is the angular position with (θ_t) as the main output of the system. The suggested (AVC-DEO-PID) control scheme is employed in this research study to assess system performance in vibration control when the RFM is to be acted with the disturbance. Here, the MATLAB simulations, control strategies, results, and data analysis for the MTRS has been investigated and carried out for the general circuit situation and performance assessment of the MTRS respectively in different models and situations based on mathematical and dynamical modelling, equations of motion. Figures 27 and 28, and the overall block control structure of MTRS are all examples of this. A Proportional, Integral and Derivative (PID) is used as the primary controller to regulate the gyroscope. After the controller has analysed the command data and sent the command set to the flywheel motor driver through communication, the traditional and intelligent control framework communicates the flywheel speed command to the controller wirelessly. The flywheel motor receives its energy from the motor driver. the MTRS's traditional and intelligent control scheme, which employs a gyroscopic effect brought on by a combination of a flywheel's angular momentum and a gimbal's rate of tilt. The control block can be attached as a NGCAM for the system's vibration control and employed in a new intelligent control framework block. The flywheel's tilt motion regulates the gyroscopic force acting in the yaw direction [19-23].

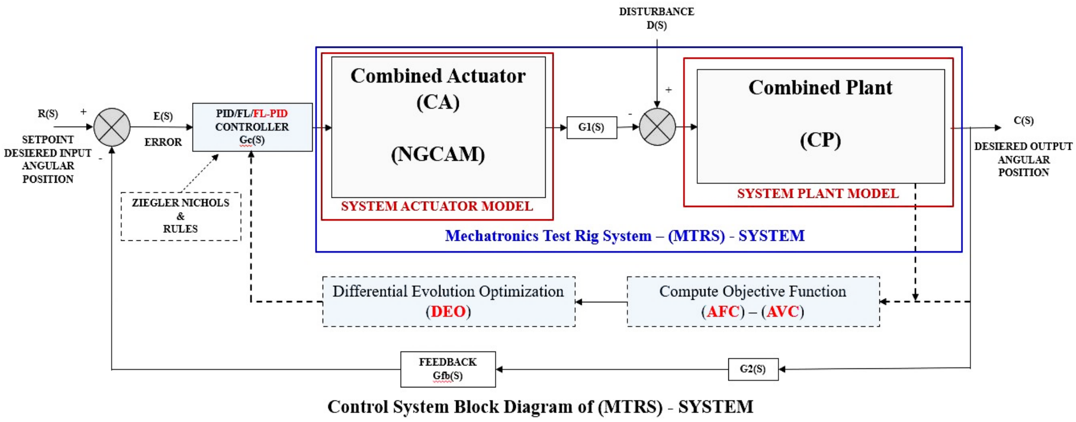


Fig. 27. AVC-DEO-PID control block diagram framework of MTRS

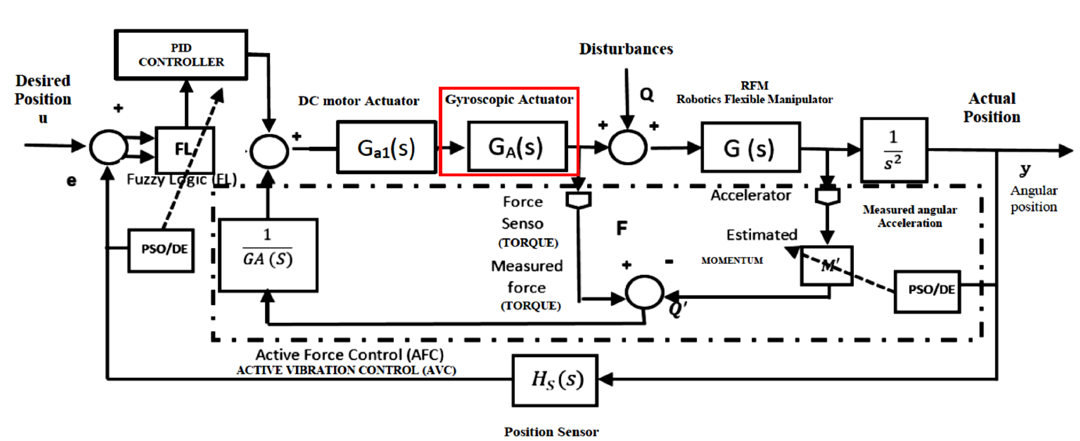


Fig. 28. Closed loop control system block diagram with AVC and DE optimization of MTRS

3. Results

3.1 MATLAB Simulation, Control, Results and Data Analysis

Create the simulation model in accordance with the equation; the parameters for the differential evolution simulation are: the dimension of population number is D; the population size is NP; the maximum and minimum values of the variance factor are N_{max} and N_{min}, respectively, with the average value N_{ave}. The crossover probability factor has a maximum value of η_{max} and a minimum value of η_{min} utilizing an average value of η_{ave}. Following 100 generations of evolution, the corrected results are shown in Table 10.

3.2 PID Control Design and Simulation of MTRS

The simulations of the closed loop system for a Single Link Robotics Flexible Manipulator Model (SLRFMM) consist of a Flexible Manipulator Model (FMM) with a Servo Direct Current Motor Model (SDCMM) both as a system Combined Plant (CP) and a Gyroscope Model (GM) with a Direct Current Motor Model (DCMM) both as a Combined Actuator (CA) have been performed and obtained for the performance assessment of the system model with PID controller tuning by Ziegler Nichols method where the input signal to the system is a step function. In this simulation the system input (I_2) is a step function as a voltage (u = v) and the real output is in three ($O_1 = \theta_t$) as an angular position, ($O_2 = \dot{\theta}_t = \omega$) as an angular velocity and ($O_3 = \ddot{\theta}t = \alpha$) as an angular acceleration known as vibration. The system input (I_1) is a step function as an angular position, the real output as angular position ($O_1 = \theta_t$), their derivatives as angular velocity ($O_2 = \dot{\theta}_t = \omega$) and angular acceleration known as vibration ($O_3 = \ddot{\theta}t = \alpha$). The simulation block diagram of the system model is represented in continuous time domain model in MATLAB & SIMULINK with the input/output signal graphs which has been shown in Figure 29, Figure 30, Table 5 and Table 6.

Table 5
Tuned parameters coefficients of PID controller of MTRS [24]

MTRS Parameters	Unit	Description	PID	Reference 1, [24]
K_P	Cte.	Proportional	0.6744	0.2033
K_I	Cte.	Integral	0.6051	0.9333
K_D	Cte.	Derivative	3.3624	0.0538
N	Cte.	Cte.	75	-

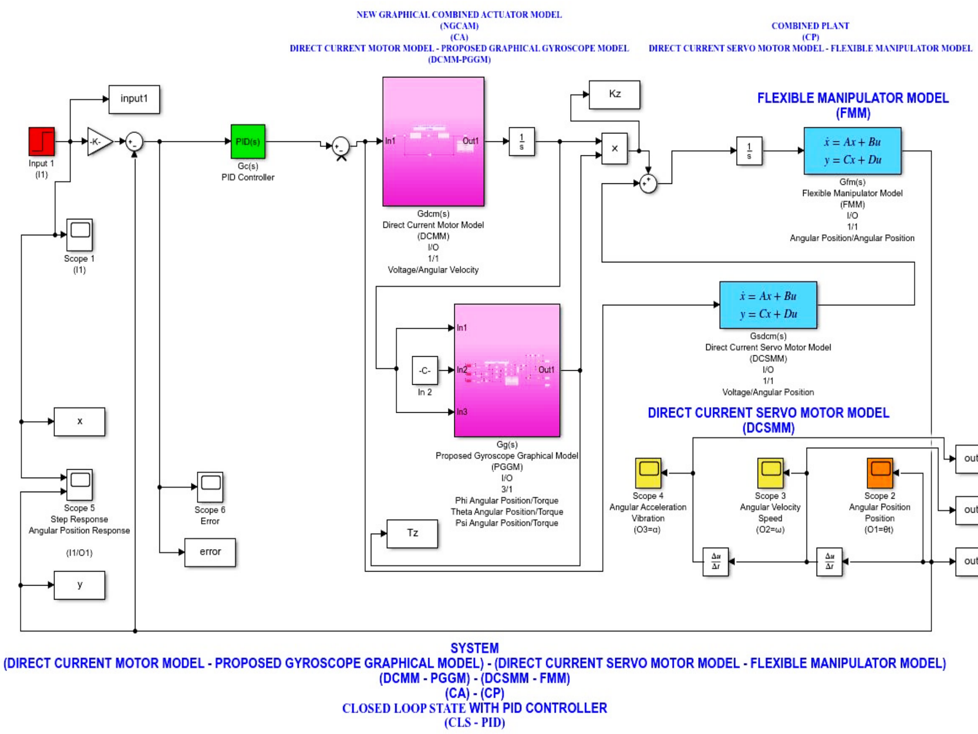
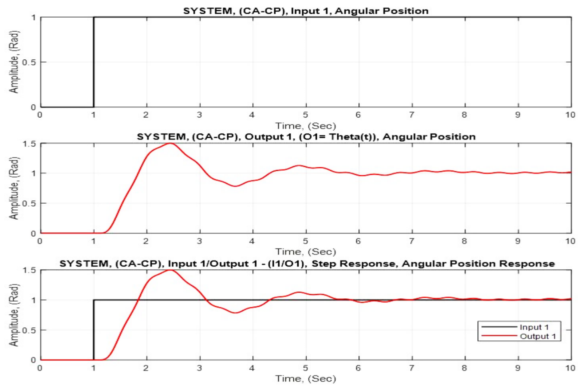
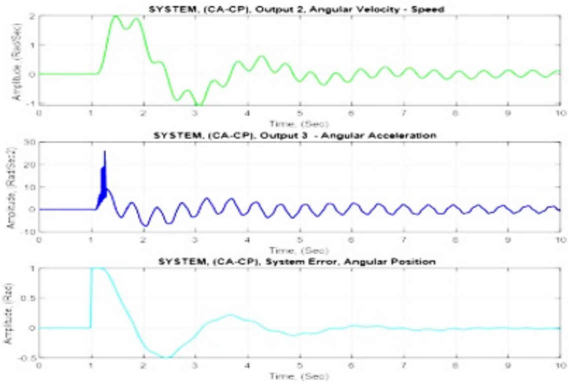


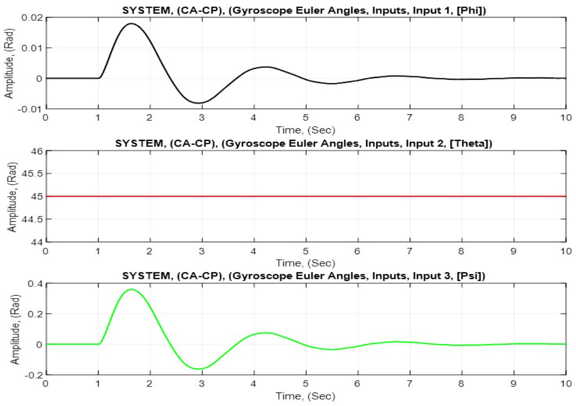
Fig. 29. The PID control block diagram simulation of MTRS



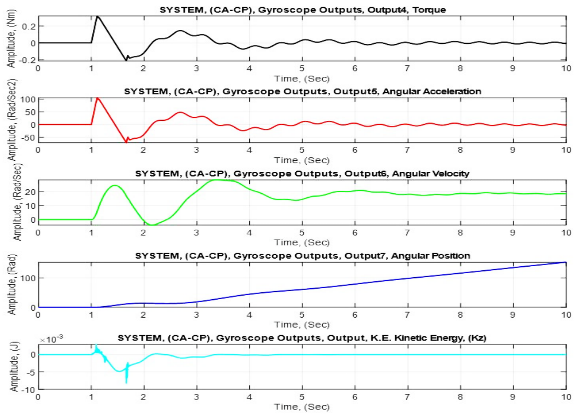
(a) Angular position as input (1) of MTRS



(b) Angular velocity as output (2) of MTRS



(c) Gyroscope Euler angles as inputs



(d) Gyroscope outputs

Fig. 30. The PID control signal graph simulation results of MTRS

Table 6
Performance criteria of PID controller of MTRS [25,26]

TRS Parameters	Unit	Description	PID	Reference 2, [25]	Reference 3, [26]
T_R	(ms)	Rise Time	1.755	1.157	4.07
T_P	(s)	Peak Time	2.532	1.5	5.272
T_S	(s)	Settling Time	5.851	2.185	10.7
M_P	(%)	Overshoot	0.532	1.5	0.819
E_{SS}	(/s)	Steady State Error	0.02	0.0298	0.0982

3.3 AVC-PID Control Design and Simulation of MTRS

Here, the input/output signal graphs for AVC-PID control simulation results of MTRS has been evaluated and shown in Table 7.

Table 7
Tuned parameters coefficients of AVC-PID controller of MTRS [19,20]

MTRS Parameters	Unit	Description	AVC-PID	Reference 4, [19]	Reference 5, [20]
K_P	Cte.	Proportional	0.6796	7.1	4
K_I	Cte.	Integral	0.6052	0.0061	0.03
K_D	Cte.	Derivative	3.3804	70.46	0.006
N	Cte.	Cte.	75	-	-

3.4 AVC-DEO-PID Control Design and Simulation of MTRS

For the performance evaluation of a Single Link Robotics Flexible Manipulator Model (SLRFMM), Differential Evolution Optimization (DEO) simulations of a closed loop system model with PID controller have been carried out and created in continuous time.

The optimization run test results are displayed in Figure 31, Figure 32, Figure 33, Figure 34 and Figure 35 run tests comparison in Figure 36 according to the system optimization parameters as Table 8, Table 9 and Table 10.

DE strategies are essential for stochastic global optimization, which is highly dependent on control parameters, and have a significant and significant influence on DE performance.

Deterministic optimization (DE) is a population-based metaheuristic method for producing numerical solutions to optimization problems.

3.4.1 Run test 1

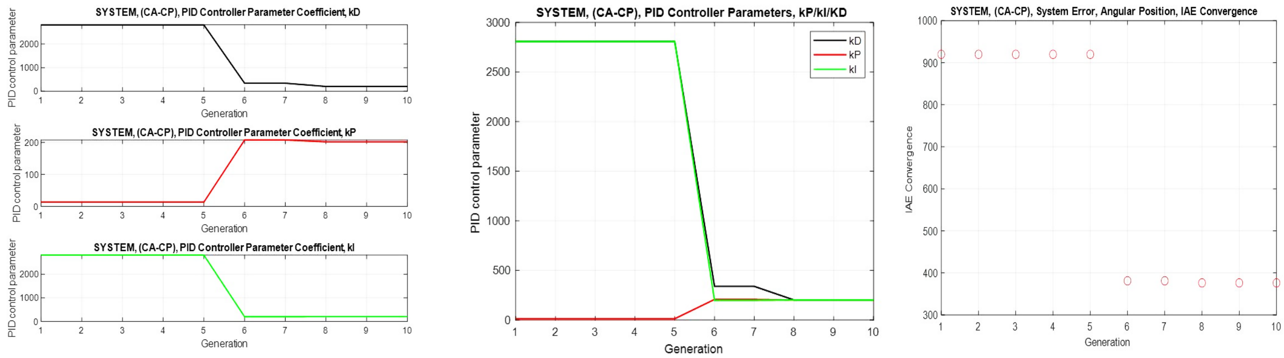


Fig. 31. The optimization results of MTRS for run test 1, with kP=kI=kD=44.0332

3.4.2 Run test 2

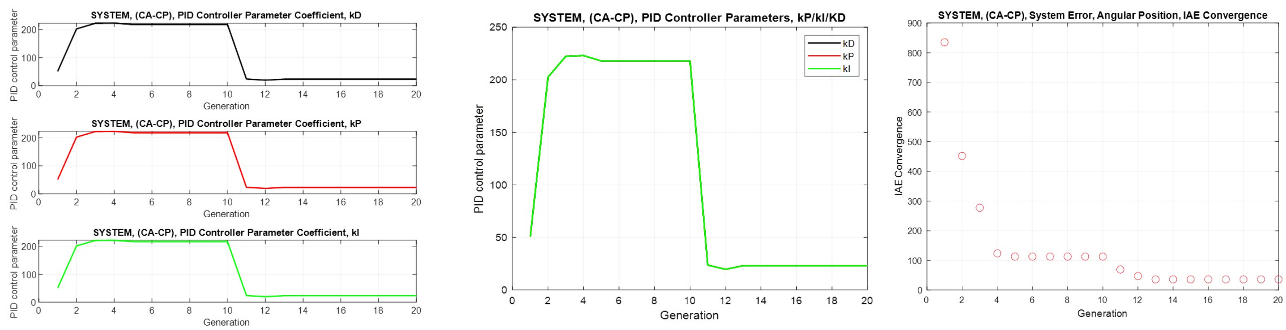


Fig. 32. The optimization results of MTRS for run test 2, with kP=kI=kD=23.0551

3.4.3 Run test 3

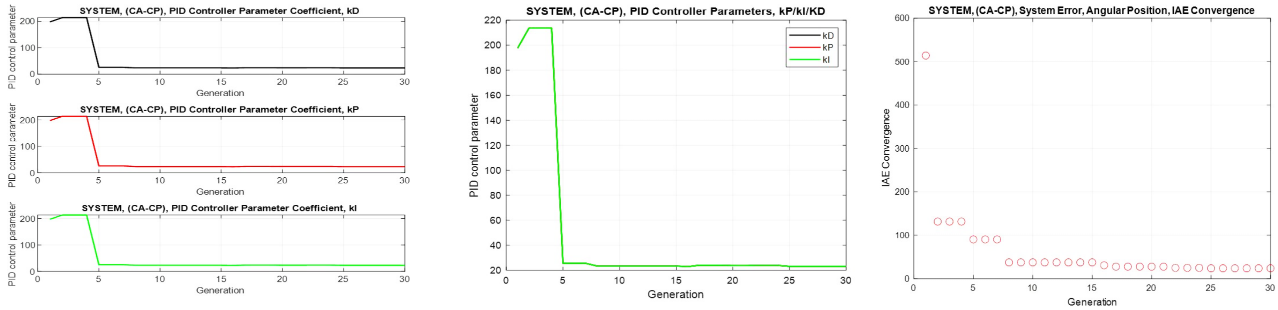


Fig. 33. The optimization results of MTRS for run test 3, with kP=kI=kD=23.0284

3.4.4 Run test 4

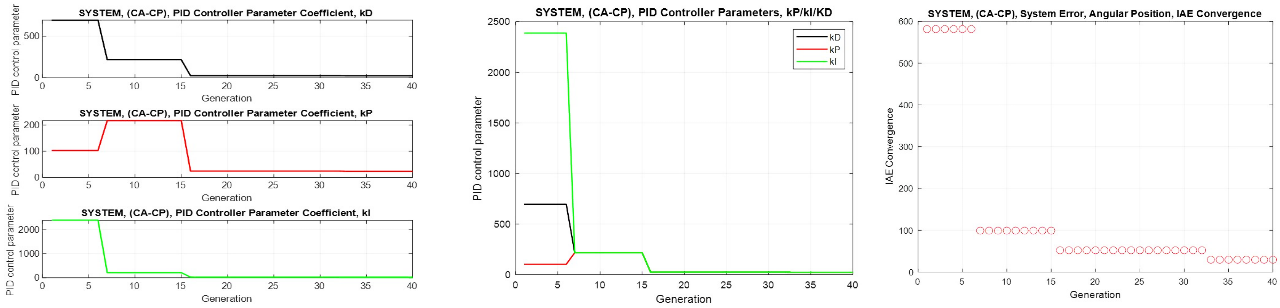


Fig. 34. The optimization results of MTRS for run test 4, with kP=kI=kD=22.7323

3.4.5 Run test 5

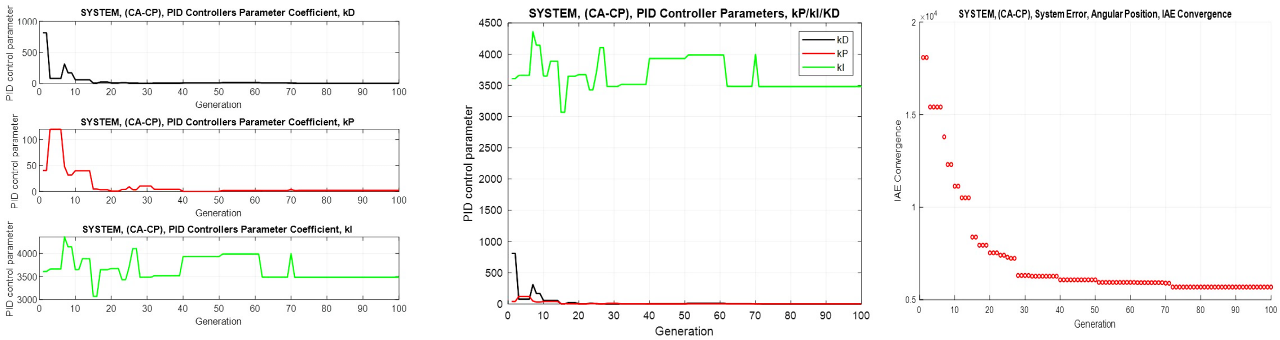
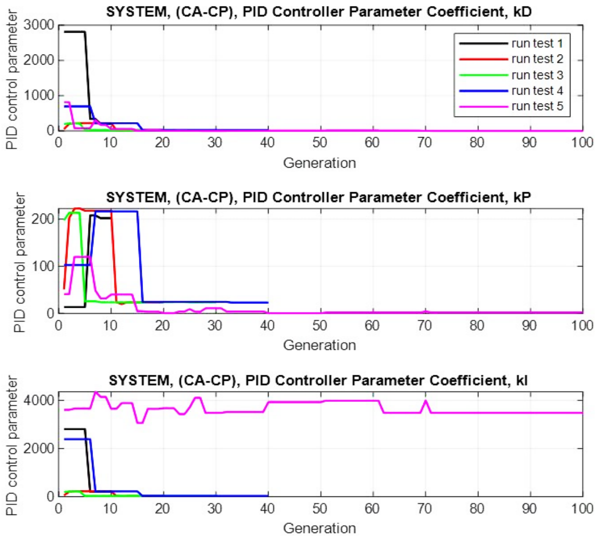
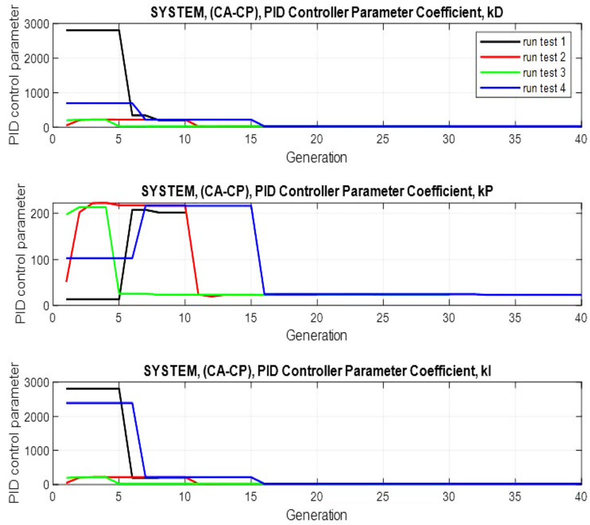


Fig. 35. The optimization results of MTRS for run test 5, with kP=0.4539, kI=1.8310, kD=3481.2

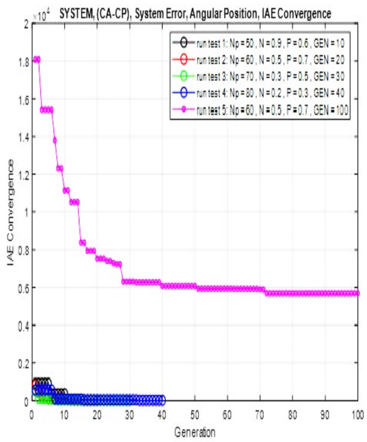
3.4.6 Run tests comparison



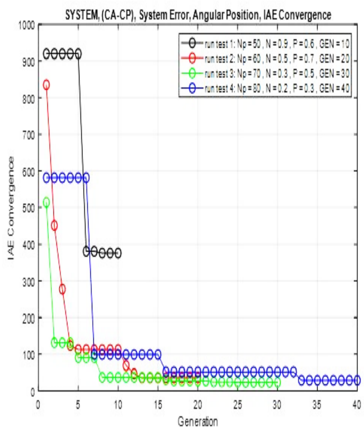
(a) Run tests 1 to 5 comparison of MTRS



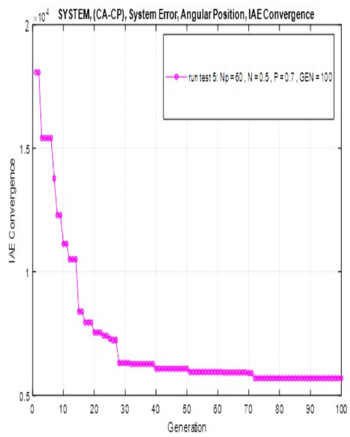
(b) Run tests 1 to 4 comparison of MTRS



(c) IAE convergence comparison of run tests 1 to 5



(d) IAE convergence comparison of run tests 1 to 4



(e) IAE convergence and error of run test 5 of MTRS

Fig. 36. The optimization run tests comparison results (a), (b), (c), (d) and (e) of MTRS

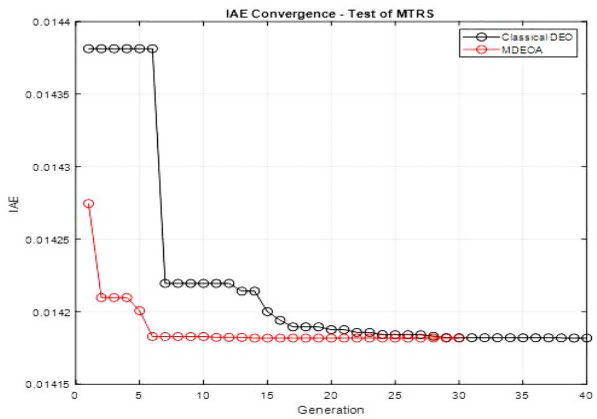
Table 9
DEO coefficients of MTRS

Run Test	I_{best}	F_{best}	Time
Run Test 1-30 Times	19	1152	475.226
Run Test 2-30 Times	39	-	1063.8
Run Test 3-30 Times	25	-	1818.4
Run Test 4-30 Times	69	-	2756.5
Run Test 5-30 Times	36	-	777.6

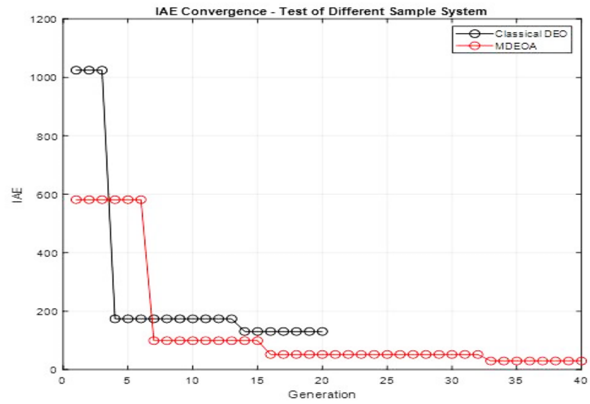
The preceding trials/tests were carried out to demonstrate the constancy of the best parameters. The proposed algorithm's performance on the system has been assessed. The result of the IAE criteria is used to determine good performance in this case. In the current study, the algorithm was run (30) times on the system independently for each run-in order to compare the results of different runs and determine the average output performance of (DE).

All combinations of control parameter values are evaluated to identify the influence of population size (NP), differential weight (N) and crossover (ŋ). The problem dimension (D) has been considered in this case. The differential weight range is set to (0.1) to (3), the crossover range is set to (0.1) to (3), the population size parameters are set to (50) to (100) and the generation number is set to (10) to (100). As a result, like other evolutionary algorithms, the output is based on a random chance to get better results and as indicated in Figure 37, Table 8, Table 9, Table 10 and Figure 38 to Figure 41 and Table 11 to Table 13, the output of (DE) is always unclear.

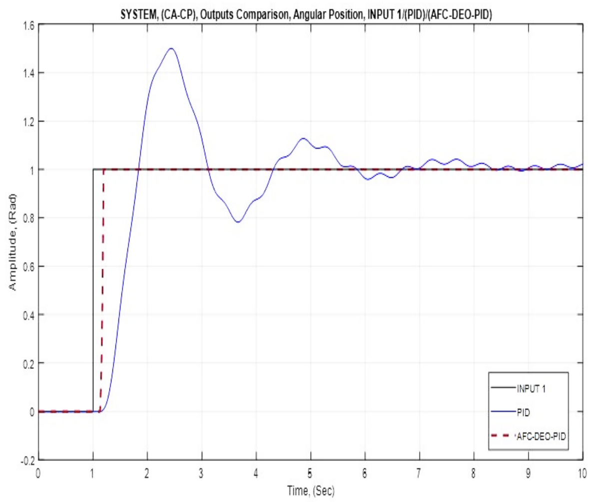
3.5 Simulation Performance Evaluation of MTRS



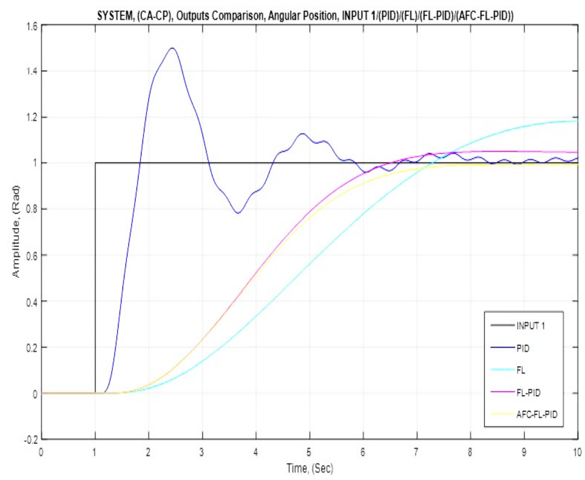
(a) Test of MTRS with CDEOA and MDEOA



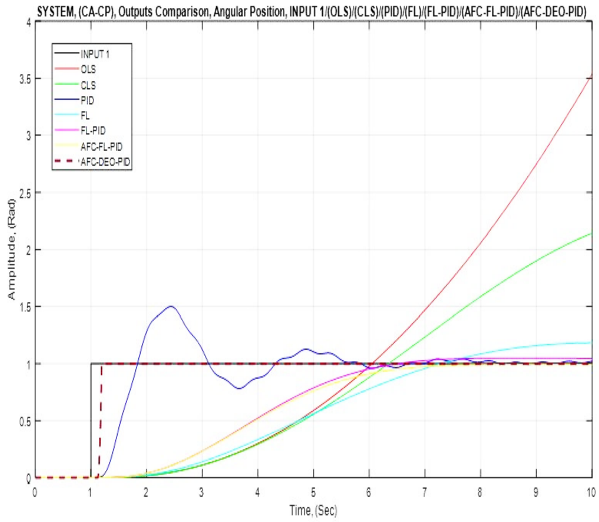
(b) Test of other system with CDEOA and MDEOA



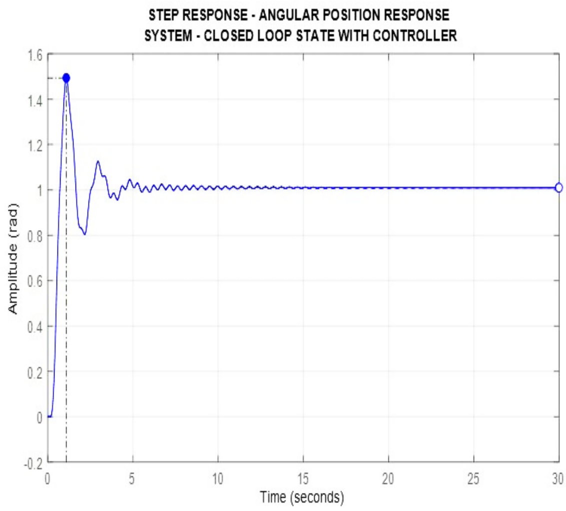
(c) Controller comparison of Input, PID and AFC-DEO-PID



(d) Controller comparison of Input, PID, FL, FL-PID



(e) Controller comparison of Input, PID, FL, FL-PID, AFC-FL-PID and AFC-DEO-PID



(f) Angular position response of closed loop system with controller

Fig. 37. The simulation performance evaluation results (a), (b), (c), (d), (e) and (f) of MTRS

Table 11
Tuned parameters coefficients of PID, AVC-PID and AVC-DEO-PID controller of MTRS

MTRS Parameters	Unit	Description	PID	AVC-PID	AVC-DEO-PID	Reference 4 [19]	Reference 5 [20]
K_P	Cte.	Proportional	0.674	0.6796	0.4539	7.1	4
K_I	Cte.	Integral	0.605	0.6052	1.831	0.0061	0.03
K_D	Cte.	Derivative	3.362	3.3804	3481.2	70.46	0.006
N	Cte.	Cte.	75	75	75	-	-

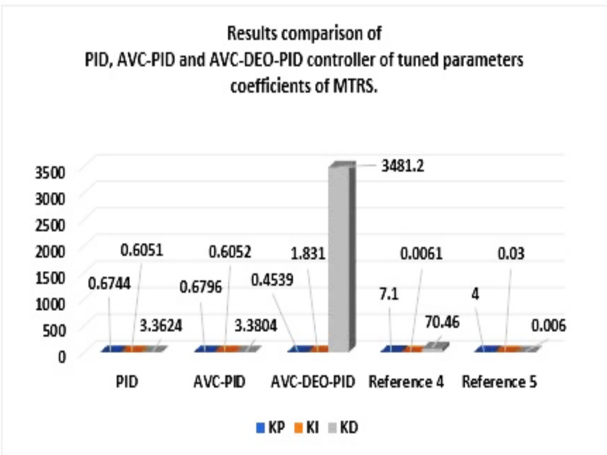


Fig. 38. Results comparison of PID, AVC-PID and AVC-DEO-PID controller of tuned parameters coefficients of MTRS

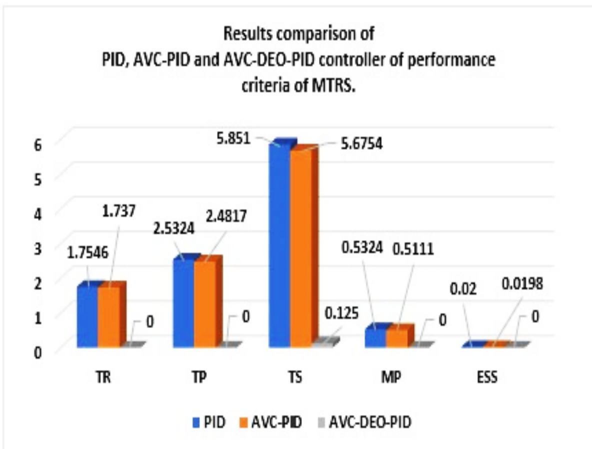


Fig. 39. Results comparison of PID, AVC-PID and AVC-DEO-PID controller of performance criteria of MTRS

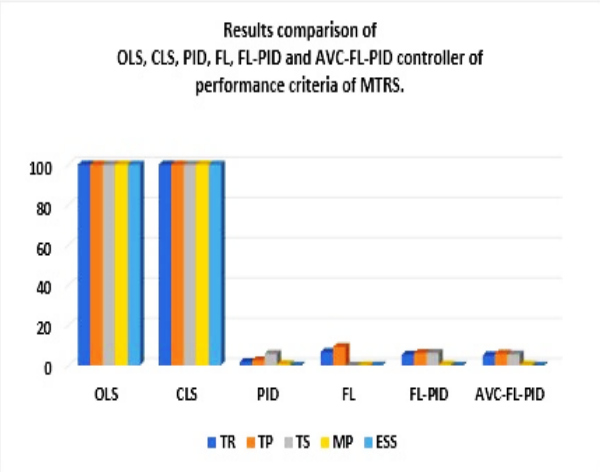


Fig. 40. Results comparison of OLS, CLS, PID, FL, FL-PID and AVC-FL-PID controller of performance criteria of MTRS

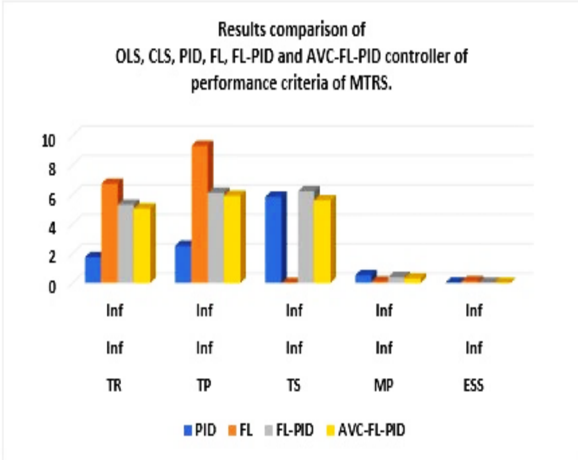


Fig. 41. Results comparison of OLS, CLS, PID, FL, FL-PID and AVC-FL-PID controller of performance criteria of MTRS

Table 12
Performance criteria of PID, AVC-PID and AVC-DEO-PID controller of MTRS

MTRS Parameters	Unit	Description	PID	AVC-PID	AVC-DEO-PID
T_R	(ms)	Rise Time	1.755	1.737	0
T_P	(s)	Peak Time	2.532	2.4817	0
T_S	(s)	Settling Time	5.851	5.6754	0.125
M_P	(%)	Overshoot	0.532	0.5111	0
E_{SS}	(/s)	Steady State Error	0.02	0.0198	0

Table 13
Performance criteria of OLS, CLS, PID, FL, FL-PID and AVC-FL-PID controller of MTRS

MTRS Parameters	Unit	Description	OLS	CLS	PID	FL	FL-PID	AVC-FL-PID
T_R	(ms)	Rise Time	Inf	Inf	1.755	6.745	5.321	5.0549
T_P	(s)	Peak Time	Inf	Inf	2.532	9.321	6.132	5.9254
T_S	(s)	Settling Time	Inf	Inf	5.851	Inf	6.231	5.6079
M_P	(%)	Overshoot	Inf	Inf	0.532	0.109	0.399	0.3074
E_{SS}	(/s)	Steady State Error	Inf	Inf	0.02	0.127	0.014	0.0093

Computational Thinking (CT) and Data Science (DS) are methods of problem-solving that use a variety of approaches and abilities to comprehend and approach complicated problems methodically. It covers ideas like as abstraction, pattern recognition, decomposition, and algorithm design. It entails taking organised and unstructured data and applying scientific procedures, systems, algorithms, and methodologies to extract information and insights. Together, (CT) and (DS) provide strong approaches and instruments to tackle challenging issues, particularly in the field of optimisation. When creating and utilising optimisation strategies in a variety of sectors, (CT) and (DS) are essential. They make it possible to see patterns, break down difficult issues into smaller, more manageable chunks, and develop effective algorithms for locating the best answers. These techniques are crucial in sectors where optimisation may result in major gains in efficacy and efficiency, including supply chain management, healthcare, finance, and transportation, summary as Table 14 and Table 15 [27].

Machine learning (ML) in Algorithm Optimization has revolutionized various fields by enabling systems to learn from data and make informed decisions. One significant application of ML is in the optimization of algorithms themselves. This involves improving the performance, efficiency, and effectiveness of algorithms by leveraging ML techniques such as Hyperparameter Tuning, Algorithm Selection, Parameter Optimization, Adaptive Control, Surrogate Modelling, which significantly enhances algorithm optimization by providing methods for them. These techniques lead to more efficient, effective, and adaptive algorithms across various fields, from deep learning to scientific simulations [28].

Table 14
Computational thinking activities of ISTE in practical [27]

No	Title	Ref.
1	Formulating, organizing, analysing, modelling, abstractions, algorithmic thinking, automating, efficiency, generalizing, transferring	ISTE (2011)
2	Creativity, algorithmic thinking, critical thinking, problem solving, cooperation	ISTE (2015; Oden *et al.,*)
3	Data analysis, abstract thinking, algorithmic thinking, modelling, representing data, breaking problems into components, automation	ISTE (2016) (Computational Thinker Definition)

Table 15
Computational Thinking items [27]

No	Title	Ref.
1	Abstraction, Algorithms, Automation, Problem Decomposition, Parallelization, Simulation	Barr & Stephenson (2011)
2	Abstraction, Algorithms, Automation, Problem Decomposition, Generalization	Wing (2006, 2008, 2011)
3	Abstraction, Algorithmic Thinking, Decomposition, Evaluation, Generalization	Selby & Woollard (2013)
4	Abstraction, Automation, Analysis	Lee *et al.,* (2011)
5	Abstraction, Algorithms, Decomposition, Debugging, Generalization	Angeli *et al.,* (2016)

4. Conclusions

In this study, a superior differential evolution approach is proposed for configuring the CSLRFM mechanical vibrations control parameters. The method combines two excellent mutation algorithms and may adjust the combination's balance to better meet the demands of problem-solving as needed. The typical differential evolution algorithm has a constant value for the mutation factor, but the algorithm also uses crossover probability factors that change with the number of iterations to make up for this shortcoming. The proposed MDEOA is utilised to modify the CSLRFM mechanical vibrations' control parameters. According to the simulation results, it performs better dynamically and steadily than the classic PID algorithm and standard differential evolution algorithm due to its benefits of reduced overshoot, rapid reaction, and quick adjustment times. It is compared to other algorithms to demonstrate that the parameters optimised by the MDEOA method have greater control performance in PID. In this research, As the objective, to investigate a new tuning method using a New Graphical Combined Actuator Model (NGCAM) and a Modified Differential Evolution Optimization Algorithm (MDEOA) both as a new approach to compute gains and optimizing the Intelligent Active Vibration Controller (IAVC) parameters in order to control the mechanical vibration of the CSLRFM and eliminating and minimizing the deflection and oscillation angle of the End Effector of the beam for better performance in the vibration control and suppression and keeping the rotation angle at a certain time, the flexible arm has a constant rotational angular position without vibration and in a desired and acceptable positional accuracy in a horizontal plane motion. The feasibility of the proposed NGCAM with MDEOA and the proposed controller is validated by simulations using MATLAB & SIMULINK program. The DE optimization simulation results demonstrate that with the optimised controller parameters coefficients, (Kp), (Ki), and (Kd), the system will provide a suitable response, and the research aims and performance are acceptable with the parameters. As a result, DE's output is always unpredictable. It is obvious from the optimised system's tables and comparison graphs, as well as the observed reduction in IAE, that the recommended DE optimization was successful in increasing the system's vibration control performance. The angular position is variable in this case, but at a specific moment, it is constant with a specific angular velocity, and the angular acceleration is zero. The system results based on the intelligent active vibration controller with the proposed NGCAM with MDEOA for minimizing the error function according to the performance criteria and simulation findings, the technique can be used to optimize the PID controller parameters settings for the IAVC of the CSLRFM and a reduction in the mechanical vibrations. Simulation results illustrate the effectiveness of the proposed MDEOA strategy which is significantly and quite satisfactory about 25 to 30 (%) better than comparing to the other algorithms in improvement stabilization and vibration control of CSLRFM.

Acknowledgments
This work supported by Research Program supported by the Department of Mechanical and Manufacturing Engineering. The authors would like to thank Universiti Putra Malaysia (UPM) and the Malaysian Ministry of Higher Education (MOHE) for their continuous support in the research work.

References
[1] Kelly. *Mechanical vibrations: theory and applications*. Cengage learning, 2012.
[2] Tokhi, M. Osman, Z. Mohamed, and M. Hasan Shaheed. "Dynamic characterisation of a flexible manipulator system." *Robotica* 19, no. 5 (2001): 571-580. https://doi.org/10.1017/S0263574700003209
[3] Kotnik, Paul T., Stephen Yurkovich, and Ümit Özgüner. "Acceleration feedback for control of a flexible manipulator arm." *Journal of Robotic Systems* 5, no. 3 (1988): 181-196. https://doi.org/10.1002/rob.4620050302
[4] Ge, Shuzhi Sam, Tong Heng Lee, and G. Zhu. "Genetic algorithm tuning of Lyapunov-based controllers: An application to a single-link flexible robot system." *IEEE Transactions on Industrial Electronics* 43, no. 5 (1996): 567-574. https://doi.org/10.1109/41.538614
[5] Pereira, Emiliano, Sumeet S. Aphale, Vicente Feliu, and SO Reza Moheimani. "Integral resonant control for vibration damping and precise tip-positioning of a single-link flexible manipulator." *IEEE/ASME Transactions on Mechatronics* 16, no. 2 (2010): 232-240. https://doi.org/10.1109/TMECH.2009.2039713
[6] Li, Hong-Xing, and CL Philip Chen. "The equivalence between fuzzy logic systems and feedforward neural networks." *IEEE Transactions on Neural Networks* 11, no. 2 (2000): 356-365. https://doi.org/10.1109/72.839006
[7] Abdulameer, Ashwaq, Marizan Sulaiman, Mohd Shahrieel Mohd Aras, and Dawood Saleem. "Tuning methods of PID controller for DC motor speed control." *Indonesian Journal of Electrical Engineering and Computer Science* 3, no. 2 (2016): 343-349. https://doi.org/10.11591/ijeecs.v3.i2.pp343-349
[8] Bankar, Vishal K., and A. S. Aradhye. "A review on active, semi-active and passive vibration damping." *Int. J. Curr. Eng. Technol* 6, no. 10.1155 (2016): 2019.
[9] Saad, Mohd Sazli, Hishamuddin Jamaluddin, and Intan Zaurah Mat Darus. "Online monitoring and self-tuning control using pole placement method for active vibration control of a flexible beam." *Journal of Vibration and Control* 21, no. 3 (2015): 449-460. https://doi.org/10.1177/1077546313485105
[10] Rajeswari, K., and P. Lakshmi. "Simulation of suspension system with intelligent active force control." In *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 271-277. IEEE, 2010. https://doi.org/10.1109/ARTCom.2010.21
[11] Miljković, Dubravko. "Review of active vibration control." *32nd Information and Communication Technology, Electronics and Microelectronics* (2009): 103-108.
[12] Rosli, R., Z. Mohamed, and G. Priyandoko. "Simulation of active force control using MR damper in semi active seat suspension system." In *IOP Conference Series: Materials Science and Engineering*, vol. 1062, no. 1, p. 012005. IOP Publishing, 2021. https://doi.org/10.1088/1757-899X/1062/1/012005

[13] Sloss, Andrew N., and Steven Gustafson. "2019 evolutionary algorithms review." *Genetic programming theory and practice XVII* (2020): 307-344. https://doi.org/10.1007/978-3-030-39958-0_16
[14] Storn, Rainer, and Kenneth Price. "Minimizing the real functions of the ICEC'96 contest by differential evolution." In *Proceedings of IEEE international conference on evolutionary computation*, pp. 842-844. IEEE, 1996.
[15] Qin, A. Kai, Vicky Ling Huang, and Ponnuthurai N. Suganthan. "Differential evolution algorithm with strategy adaptation for global numerical optimization." *IEEE transactions on Evolutionary Computation* 13, no. 2 (2008): 398-417. https://doi.org/10.1109/TEVC.2008.927706
[16] Das, Swagatam, and Ponnuthurai Nagaratnam Suganthan. "Differential evolution: A survey of the state-of-the-art." *IEEE transactions on evolutionary computation* 15, no. 1 (2010): 4-31. https://doi.org/10.1109/TEVC.2010.2059031
[17] Zhang, Jingqiao, and Arthur C. Sanderson. "JADE: adaptive differential evolution with optional external archive." *IEEE Transactions on evolutionary computation* 13, no. 5 (2009): 945-958. https://doi.org/10.1109/TEVC.2009.2014613
[18] Pahner, Uwe, and Kay Hameyer. "Adaptive coupling of differential evolution and multiquadrics approximation for the tuning of the optimization process." *IEEE Transactions on magnetics* 36, no. 4 (2000): 1047-1051. https://doi.org/10.1109/20.877621
[19] Jovanović, Miroslav M., Aleksandar M. Simonović, Nemanja D. Zorić, Nebojša S. Lukić, Slobodan N. Stupar, and Slobodan S. Ilić. "Experimental studies on active vibration control of a smart composite beam using a PID controller." *Smart materials and structures* 22, no. 11 (2013): 115038. https://doi.org/10.1088/0964-1726/22/11/115038
[20] Hadi, M. Sukri, Intan Z. Mat Darus, M. Osman Tokhi, and Mohd Fairus Jamid. "Active vibration control of a horizontal flexible plate structure using intelligent proportional–integral–derivative controller tuned by fuzzy logic and artificial bee colony algorithm." *Journal of Low Frequency Noise, Vibration and Active Control* 39, no. 4 (2020): 1159-1171. https://doi.org/10.1177/1461348419852454
[21] Mohamed, Ali Wagdy, and Abdulaziz S. Almazyad. "Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems." *Applied Computational Intelligence and Soft Computing* 2017, no. 1 (2017): 7974218. https://doi.org/10.1155/2017/7974218
[22] Rahman, TA Zahidi, and IZ Mat Darus. "Active vibration control of a flexible plate via active force control strategy." In *2011 4th International Conference on Mechatronics (ICOM)*, pp. 1-6. IEEE, 2011. https://doi.org/10.1109/ICOM.2011.5937194
[23] Bonanno, Francesco, Giacomo Capizzi, Grazia Lo Sciuto, Christian Napoli, Giuseppe Pappalardo, and Emiliano Tramontana. "A cascade neural network architecture investigating surface plasmon polaritons propagation for thin metals in openmp." In *Artificial Intelligence and Soft Computing: 13th International Conference, ICAISC 2014, Zakopane, Poland, June 1-5, 2014, Proceedings, Part I 13*, pp. 22-33. Springer International Publishing, 2014. https://doi.org/10.1007/978-3-319-07173-2_3
[24] Deif, Sarah, Mohammad Tawfik, and Hanan A. Kamal. "Vibration and position control of a flexible manipulator using a PD-tuned controller with Modified Genetic Algorithm." In *Proceedings of the ICCTA 2011 Conference*, pp. 15-17. 2011.
[25] John, Subash, Abdul Imran Rasheed, and Viswanath K. Reddy. "ASIC implementation of fuzzy-PID controller for aircraft roll control." In *2013 International Conference on Circuits, Controls and Communications (CCUBE)*, pp. 1-6. IEEE, 2013. https://doi.org/10.1109/CCUBE.2013.6718551
[26] Lakshmi, K. Vijaya, and P. Srinivas. "Fuzzy adaptive PID control of pitch system in variable speed wind turbines." In *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 52-57. IEEE, 2014. https://doi.org/10.1109/ICICICT.2014.6781252
[27] Hashim, Mohd Ekram Alhafis, Nur Safinas Albakry, Wan Azani Mustafa, Banung Grahita, Miharaini Md Ghani, Hafizul Fahri Hanafi, Suraya Md Nasir, and Catherina ana Ugap. "Understanding the Impact of Animation Technology in Virtual Reality: A Systematic Literature Review." International Journal of Computational Thinking and Data Science 1, no. 1 (2024): 53-65.
[28] Ong, Siew Har, Sai Xin Ni, and Ho Li Vern. "Dimensions Affecting Consumer Acceptance towards Artificial Intelligence (AI) Service in the Food and Beverage Industry in Klang Valley." Semarak International Journal of Machine Learning 1, no. 1 (2024): 20-30.