PSO-Optimized Model Reference Adaptive PID Controller for Precise DC Motor Speed Control

J. E. Oche[1*], H. A. Bashir[2], T. J. Shima[3]

[1]Operations Unit, Netcom Africa Limited, Lagos, Nigeria.
[2]Department of Electrical Engineering, Bayero University Kano, Nigeria.
[3]Department of Electrical and Electronic Engineering, University of Ibadan, Ibadan, Nigeria.

ABSTRACT: Whenever DC motor operates at varying speeds and is expected to rotate in two directions with high precision, it becomes essential that uncertainties in the form of external disturbances and load variations in the system be considered. In such systems, the use of fixed-gain Proportional Integral Derivative (PID) controller alone is suboptimal as the controller lacks the ability to handle the uncertainties which causes the output response of the motor to have excessive overshoot, poor disturbance rejection and poor command-following. This paper presents a particle swarm-optimized model reference adaptive PID controller **(**PSO MRAC-PID) for precise DC motor speed control. The proposed controller can set the speed of a DC motor to track a predefined reference model and adapt to the effect of uncertainties and external disturbances, using a combination of two basic techniques namely, the Massachusetts Institute of Technology (MIT) rule and particle swarm optimization (PSO). The MIT rule was utilised for the design of the adaptation mechanism and PSO was implemented for optimization of the adaptation gains of the MRAC-PID controller to enhance its overall performance. Comparative analyses with fixed gain PSO-PID and conventional MRAC-PID controllers indicate that the proposed PSO MRAC-PID achieves a 66.7% improvement in settling time, 0.4% improvement in overshoot, 35.3% reduction in control effort and a general improvement in command tracking and disturbance rejection. This outcome suggests that the proposed POS MRAC-PID controller can better suit real-world applications where precision is critical such as motor drives in conveyor belts, crane systems, among others. It also achieves control with greater energy savings as compared to the traditional methods – a crucial feature for greener and cleaner operations.

KEYWORDS: DC motor control, model reference adaptive control, PID control, particle swarm optimization, MIT rule.

[Received Mar. 23, 2024; Revised Aug. 15, 2024; Accepted Dec. 2, 2024] Print ISSN: 0189-9546 | Online ISSN: 2437-2110

I. INTRODUCTION

One of the major causes of non-linearity in control system is the impact of uncertainties. Other causes are sudden changes in environmental conditions such as change in wind load on an operational DC motor and change in environmental temperature that may alter the expected performance of components of control systems (Khalil, 2002). Nonlinear systems are systems in which the relationship between input and output is not a straight line (nonlinear), meaning that the system's behaviour cannot be described using linear equations alone, and the system's response to inputs can vary significantly depending on the current state and conditions. Recently, many fixed-gain nonlinear control schemes such as Model Predictive Control (MPC) (Ruchika *et al.*, 2013), Integral Sliding Mode Control (ISMC) (Khan and Jalani, 2016)) and Nonlinear PI controller (Ren *et al.*, 2016) have been used by researchers to solve real-world problems in various domains like electrical, mechanical, medical and in avionics. However, such fixed-gain controllers are mostly suitable for systems with stable and well-understood dynamics, where the control parameters can remain constant over time. Thus, they are not suitable for systems where significant uncertainties are expected. In systems where significant uncertainties are expected there is the need for an adaptive control scheme that can provide more flexibility and robustness (Hazlina and Rubia, 2014).

Furthermore, the adaptation gain is a pivotal element in adaptive control systems, determining the speed and effectiveness of the system's response to changes; inadequate adaptation gain can result to slow rise time, poor command-following and poor disturbance rejection. Empirical method

*Corresponding author: ocheupthrust2@gmail.com
doi: http://dx.doi.org/10.4314/njtd.v21i4.2473

has been used by many researchers such as Narendra and Annaswamy (1987), Harris and Yu (2007), Åström and Wittenmark (2008), Landau et al. (2011) for the selection of the adaptation gains but the use of empirical method does not guarantee optimal gains as such system experienced a too-high or too-low adaptation gains which resulted to slow rise time, poor command-following and poor disturbance rejection or excessive overshoot and abnormally abrupt control response as also reported by Nguyen (2018) and Akbar *et al*. (2016). To overcome the challenges that come with fixed-gain controllers as well as the use of empirical methods in selection of adaptation gains, this paper proposed PSO- Optimized Model Reference Adaptive PID Controller for Precise DC Motor Speed Control (PSO MRAC-PID Controller), using MIT rule and particle swarm optimization (PSO).

The MIT Rule refers to a method for determining the parameter adjustments needed to minimize the error between the desired and actual output of a system. This rule is particularly used in adaptive control systems where the system parameters are adjusted dynamically to maintain optimal performance by providing a systematic way to update these parameters based on error feedback (Pankaj *et al*., 2011). Particle Swarm Optimization (PSO) on the other hand is a computational method used for finding optimal solutions in complex problems by simulating the social behaviour of birds flocking, by iteratively improving candidate solutions regarding a given measure of quality (Kennedy and Eberhart, 1997). Although other optimization techniques such as genetic algorithm and artificial bee colony exist, PSO was selected for this research due to its few parameters' requirement, fast convergence, and low computational cost (Poli, 2007). Since its introduction in 1995 by Kennedy and Eberheart many variations of PSO have emerged over the years. Cheng and Cheng (2011) presented Bare-Bones Particle Swarm Optimization with Crossed Memory for Constrained Optimization (BPSO-CM), integrating a crossed memory mechanism that increases the likelihood of escaping local optima and enhancing the overall search capability. Kirange and Nema (2024) also presented the Hybrid Butterfly Particle Swarm Optimization (HBPSO) that integrates the concepts of the Butterfly Optimization Algorithm (BOA) to enhance performance. The hybridization aims to combine the strengths of both algorithms to achieve better optimization results, particularly in terms of exploration and exploitation balance. Other variations are the Craziness Particle Swarm Optimization (CPSO) (Yen and He, 2013), Multi-Objective Particle Swarm Optimization (MOPSO) (Coello and Pulido, 2002)) and New Stable Particle Swarm Optimization (NSPSO) (Xia and Zhang, 2006) which focus on improvement on various parameters of the traditional PSO. Although this research still uses the traditional PSO, it combines it with the Model Reference Adaptive PID control method for precise DC motor control.

The PSO MRAC-PID controller is a model reference adaptive PID controller with optimized adaptation gains which continuously adjust the gains of the PID controller to maintain a desired output performance of a DC motor speed irrespective of sudden parameter variations, loads and uncertainties.

The rest of the paper is organized as follows: Section II presents the model of the DC motor and the design of the control technique under study. Section III discusses the results obtained via simulations while Section IV presents a conclusion on the paper.

II. BRIEF OVERVIEW OF MODEL REFRENCE ADAPTIVE PID CONTROLLER AND PARTICLE SWARM OPTIMIZATION

A. Model Reference Adaptive Control (MRAC)

The basic MRAC system consists of four main components.

The uncertain plant to be controlled: Adaptive control can deal with either linear or nonlinear plants with various types of uncertainty which could be structured uncertainty, unstructured uncertainty, or un-modelled dynamics.

Reference model: A reference model is used to specify a desired response of an adaptive control system to a command input. It is essentially a command shaping filter to achieve a desired command following. Since adaptive control is formulated as a command following or tracking control, the adaptation is operated on the tracking error between the reference model and the system output. A reference model must be designed properly for an adaptive control system to be able to follow. Typically, a reference model is formulated as a linear time invariant (LTI) model, but a nonlinear reference model can be used. An LTI reference model should capture all important performance specifications such as rise time and settling time, as well as robustness specifications such as phase and gain stability margins (Nguyen, 2018).

Controller: A controller must be designed to provide overall system performance and stability for a nominal plant without uncertainty. Thus, it can be thought of as a baseline or nominal controller. The type of controllers is dictated by the objective of a control design. A controller can be linear or nonlinear, but as always nonlinear controllers are much more difficult to design, analyse, and ultimately certify for operation in real systems. The controller can be a nominal controller augmented with an adaptive controller or a fully adaptive controller. The adaptive augmentation control design is more prevalent and generally should be more robust than a fully adaptive control design. In this research the PID controller which is a liner controller is used.

Adaptive mechanism: An adaptive law is a mathematical relationship that expresses explicitly how adaptive parameters should be adjusted to keep the tracking error as small as possible. Ultimately, designing an adaptive control system comes down to a trade-off between performance and robustness. This trade-off can be made by a suitable selection of an adaptive law and a set of tuning parameters known as adaptation gain that are built into an adaptive law.

Regardless of the actual process parameters, adaptation in MRAC takes the form of adjustment of some or all the controller coefficients to force the response of the resulting closed-loop control system to that of the reference model. Therefore, the actual parameter values of the controlled system do not really matter.

Lyapunov method, the method of augmented error and the gradient method also known as the MIT rule developed in the Massachusetts Institute of Technology (MIT) are the three main methods of designing MRAC. The MIT rule which is the original method for the design of MRAC is the most widely used due to its simplicity and the ease to the choice of desired reference model (Hazlina and Rubiah, 2014).

B. Particle swarm optimization (PSO)

Optimization methods are widely used in various fields, including engineering, economics, management, physical and social sciences. Several optimization algorithms have also been applied in optimal control.

The task is to choose the best or a satisfactory one from amongst the feasible solutions to an optimization problem, providing the scientific basis of decision-making for decision-makers. The process of using optimization methods to solve a practical problem mainly involves these two steps; first, formulation of the optimization problem which involves determining the decision variables, objective function, and constraints, and possibly an analysis of the optimization problem, secondly, selection of an appropriate numerical method, solving the optimization problem, testing the optimal solution, and deciding accordingly.

Optimal tuning of PID controllers has many advantages; when a PID controller is tuned optimally, the controller minimizes deviation from set point and responds to disturbances quickly with minimal overshoot (Yonghong et al, 2011). For PID controller gains to be selected optimally, the gains must be tuned jointly (Ai and Yongkum, 2007). This can easily be done through an optimization algorithm. Particle swarm optimization (PSO) algorithm is one of the optimization algorithms that can be used to jointly tune the adaptive gains of the controller to select the optimal adaptation gains for which overshot is minimized and disturbance rejection and command-following is improved.

In Deacha (2018), optimal PID controller design for DC Motor Speed Control with tracking and regulating constrained optimization via cuckoo search (CS) was presented. In the research, PID gains obtained through Zigler-Nichol PID tunning method were compared with those obtained through cuckoo search optimization algorithm. As simulation results were compared with the Ziegler Nichols (Z-N) tuning rules, it was found that Z-N provided very high gains of PID controller which cannot be realized and implemented, but the CS presented an optimal PID controller for DC motor speed control system satisfying both tracking and regulating constraints. Although the overall results were satisfying, it was pointed out that the CS algorithm has slow convergence rate.

Sanitosh and Vinod (2016) also used genetic algorithm (GA-PID) to obtain optimal PID gains for a DC motor speed control. The response of GA-PID was also compared with Z-N PID after which it was found that the GA-PID has a better time response, but GA has a short coming over PSO in that GA has complex algorithm with many variables to tune. It also has slow convergences time (Guangon et al, 2017).

PSO algorithm is an important member of swarm intelligence algorithms originally developed by Kennedy and Eberhart (1995). It was motivated by social behavior of bird flock or fish schooling and the technique shares many similarities with evolutionary computation techniques such as genetic algorithms. As in other population-based intelligence systems, PSO requires an initial population of random solutions. The search for optima is obtained by updating generations without evolution operators such as crossover and mutation. The potential solutions are usually called particles in PSO. These particles fly through the solution space by following their own experiences and the current optimum particles.

Various researchers have considered PSO as a superior technique based on its high computational efficiency (Banu and Uma, 2008). Unlike genetic-algorithm (GA) and ant colony optimization (ACO), PSO approach provides quicker solutions, higher convergence rate and fewer parameters. It can be considered as a powerful optimization approach in system parameter for identifying the PID controller gains (Alfi and Modars, 2017). This technique starting with an artificial swarm group such as a bird begins by initiating a random position and velocities, and randomly dispersed inside the search space. Based on the objective function, supervising their own flying experience, and flying experience of their companions, every swarm particle in the group adjusts its flight position and its velocity dynamically with every particle remembering its best position and obtaining the global best position datum which is achieved by any particle in the group during the optimization process.

The key steps in PSO algorithm are presented in the following:

i. Initialization of particle population.
ii. Evaluation of fitness of the particles using an objective function

iii. Recognition and selection of best fit for objective function
iv. Computing the new modified velocity for each particle.
v. Updating of particle position

III. METHODOLOGY

A. Modelling of the DC Motor

The electrical equivalent circuit and the free-body diagram of the DC motor is shown in Figure 1.

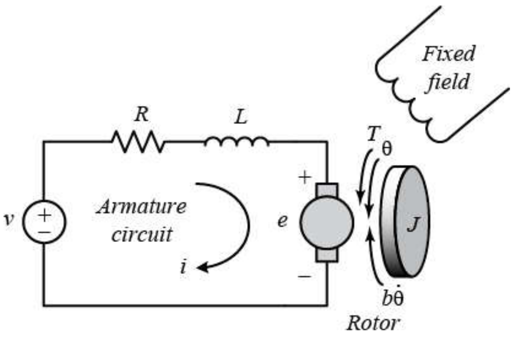

Figure 1: DC Motor free-body diagram (Messner *et al.*, 2017).

where
J = Moment of inertia of the rotor
b = Motor viscous friction constant
R = Electric resistance of the armature
L = Electric inductance of the armature
θ = Shaft angular position
v =Armature input voltage
e = Rotor back emf
T = Rotor torque
i = Armature current
Based on Newton's second law and Kirchhoff's Voltage law, the governing equations for the DC motor are given in (1) and (2).

$$J\ddot{\theta} + b\dot{\theta} = Ki \quad (1)$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta} \quad (2)$$

Where K represents the torque and back emf constants.

From Equations (1) and (2), the general speed-voltage transfer function of the DC motor in Laplace domain is obtained as shown in Equation (3).

$$\frac{\theta(s)}{V(s)} = \frac{K}{(sL+R)(Js+b)+K^2} \quad (3)$$

The parameters of the DC motor used for this design are as presented in Table 1.

Table 1: Parameters of the DC motor (Messner *et al*., 2017).

Parameters	**Values**
Torque and Back emf constant, K	0.01 (N.m/A)
Armature resistance, R	1.0 Ohms
Motor inertia, J	0.01 (kg.m^2)
Motor viscous friction coefficient, b	0.1 (N.m.s)
Electrical inductance, L	0.5 H

Substituting the parameters of Table 1 in Equation (3) yields the transfer function of Equation (4).

$$\frac{\theta(s)}{V(s)} = \frac{2}{(s+10)(s+2)} \quad (4)$$

Since the pole at $s = -10$ is five times more negative than the pole at $s = -2$, the slower of the two poles will dominate the dynamics; the system can be approximated as a first order system as shown in Equation (5) (Katsuhiko, 2009)

$$\frac{\theta(s)}{V(s)} = \frac{0.2}{s+2} \quad (5)$$

B. Traditional PID Controller Design using PSO

The transfer function for the PID controller is presented in Equation (6)

$$\frac{Y(s)}{U(s)} = K_p + K_i\frac{1}{s} + K_d s \quad (6)$$

Where $K_p, K_i,$ and K_d are proportional gain, integral gain and derivative gain respectively. To achieve the optimal PID gains an integral time absolute error (ITAE) cost function, expressed as $\int_0^T t|e(t)|dt$, where the error $e(t)$ is the difference between the plant output speed and the set-point input, was selected. The PSO algorithm was run in a MATLAB/Simulink environment to obtain the optimal PID gains. Other parameters that were used in the optimization are as shown in Table 2.

Table 2: Parameters of the PSO for PID controller tunning.

Parameter	**Value**
Number of swarm particles	50
Number of iterations	50, 100, 150, 200, 250
Lower PID gains bound	[100, 1000, 1]
Upper PID gains bound	[200, 2000, 2]
Cognitive acceleration coefficient, c1	1.2
Social acceleration coefficient, c2	0.12
Inertial weight, w	0.9

Despite a larger swarm size potentially leading to better solutions through search space exploration, a 50-particle swarm was used due to device computational limits. Five iterations with varying iteration numbers were performed to avoid local minima convergence. Initial optimization and sensitivity analysis using MATLAB and Simulink was done to determine boundary constraints and effective variable ranges. The cognitive component (c1) was set higher than the social component (c2) to prioritize individual particle experience over collective knowledge, ensuring comprehensive search. A high inertia weight of 0.9 was chosen to support sustained exploration, aligning with the high cognitive and low social components for better solution space coverage.

B. The MRAC PID Controller Design

The block diagram of the improved PSO MRAC-PID structure is as shown in Figure 2.

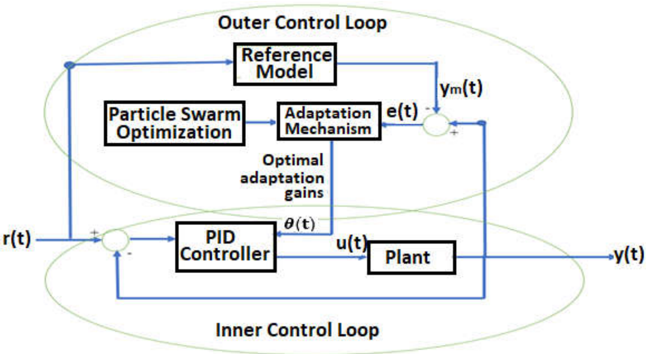

Figure 2: Improved PSO MRAC-PID Structure

As presented in the figure the controller can be thought of as consisting of two loops; the parameters of the controller are adjusted by the adaptation gains from the outer control loop based on feedback from the difference between the process output y and the model output y_m. The parameter adjustment mechanism used in this research is designed based on the MIT rule.

1. Designing of the parameter adjustment mechanism

To present the MIT rule, the closed loop system in Figure 2 in which the controller has adjustable parameters is considered. The desired closed loop response is specified by a model output y_m. The error $e(t)$, is the difference between the output of the system y and the output of the reference model y_m. The modeling error e is given by Equation (7)

$$e(t)=y(t)-y_m(t) \tag{7}$$

The cost function or loss function to be minimized by the MIT rule is defined as

$$F(\theta)=\frac{1}{2}e^2 \tag{8}$$

where θ is the adjustment parameter. Note that the use of θ here differs from that used as angular velocity in the DC motor modelling.

To make F minimal, it is reasonable to change the parameters in the direction of the negative gradient of F as shown in Equation (9)

$$\frac{d\theta}{dt}=-\gamma\frac{\partial F}{\partial\theta}=-\gamma e\frac{\partial e}{\partial\theta} \tag{9}$$

Equation (9) is called the MIT rule (Jain and Nigam, 2013)

The partial derivative $\frac{\partial e}{\partial\theta}$ is called the sensitivity derivative of the system which tells how the error is influenced by the adjustment parameter θ. γ is known as the adaptation gain.

From Figure 2 the general PID control law in time domain is presented as

$$u(t)=K_p\big(r(t)-y(t)\big)+K_i\int_0^t\big(r(t)-y(t)\big)dt+K_d\frac{d\big(r(t)-y(t)\big)}{dt} \tag{10}$$

where K_p, K_i, and K_d are the PID gains.

For the inner-loop, a variant to the standard PID structure is adopted, which uses the process variable instead of the error signal, for the derivative term as presented in Laplace form in (11) (Abdullahi, 2018).

$$U(s)=K_p\big(R(s)-Y(s)\big)+K_i\frac{\big(R(s)-Y(s)\big)}{s}+sK_dY(s) \tag{11}$$

The model in Figure 3 can be represented by a black box transfer function as shown in (12)

$$G(s)=\frac{Y(s)}{U(s)} \tag{12}$$

Substituting (11) into (12) and simplifying, yields (13)

$$Y(s)=\frac{\big(sG(s)K_p+G(s)K_i\big)R(s)}{G(s)K_ds^2+\big(1+G(s)K_p\big)s+G(s)K_i} \tag{13}$$

Substituting (13) into (7) and simplifying, yields the general error equation in (14)

$$E(s)=\frac{\big(sK_p+K_i\big)R(s)-Y_m(s)\left\{K_ds^2+\left[\frac{1}{G(s)+K_p}\right]s+K_i\right\}}{K_ds^2+\left[\frac{1}{G(s)+K_p}\right]s+K_i} \tag{14}$$

Because the major consideration is how the controller can handle uncertainties, specific values are not substituted into (14), rather the denominator of (14) is mapped with the denominator of the reference model in (15), to which the MIT rule can then be applied. Based on *apriori* knowledge of the plant, a second order model is selected as the reference model as shown in Equation (15).

$$G_m(s)=\frac{\omega_n s+\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} \tag{15}$$

where ω_n is the natural frequency of the reference model and ζ is the damping ratio.

Mapping the denominator of (15) to that of (14) yields (16).

$$E(s)=\frac{\left(sK_p+K_i\right)R(s)-Y_m(s)\left\{K_d s^2+\left[\frac{1}{G(s)+K_p}\right]s+K_i\right\}}{s^2+2\zeta\omega_n s+\omega_n^2} \quad (16)$$

The MIT rule in Equation (9) can now be applied to Equation (16) to obtain the three-dimensional adjustment mechanism, $\theta = \{K_p, K_i, K_d\}$, as presented in Equations (17), (18) and (19) .

$$K_p = -\frac{\gamma_p}{s}e\frac{\{R(s)-Y_m(s)\}s}{s^2+2\zeta\omega_n s+\omega_n^2} \quad (17)$$

$$K_i = -\frac{\gamma_i}{s}e\frac{R(s)-Y_m(s)}{s^2+2\zeta\omega_n s+\omega_n^2} \quad (18)$$

$$K_d = -\frac{\gamma_d}{s}e\frac{Y_m(s)s^2}{s^2+2\zeta\omega_n s+\omega_n^2} \quad (19)$$

(17) to (19) are the main equations to implement the outer loop of the model reference adaptive PID controller.

In a cascade control system, it is desirable that the outer loop leads the inner loop (Abdullahi, 2018), therefore a damping ratio of 1.48 was empirically selected by careful observation of the systems performance on Simulink with the aim to maximize stability, minimize oscillations, and ensure the system's robustness and reliability. The damping ratio, ζ has a value of 1.48. Using 2% settling time criterion, the desired frequency can be calculated according to Equation (20) .

$$t_s = \frac{4}{\zeta\omega_n} \quad (20)$$

The 2% settling time criterion is used due to its emphasis on precision, stability, and responsiveness as it provides a tighter and more accurate measure of system performance as compared with 4% criterion (Franklin et al, 2019).

In this research a settling time of 0.02 seconds is used in designing the reference model to ensure a rapid response and high precision. This settling time requirement is also used for the PSO-PID as well as the MRAC-PID controller design for a comparative platform. From the settling time requirement in (21) the undamped natural frequency ω_n is found to be 135.1 rad/s. Substituting the values for ζ and ω_n into the reference model in (15) and the adaptation laws in (17) to (19), yields the reference model in (21) and the adaption laws in (22) to (24)

$$G_m(s) = \frac{135s+18252}{s^2+400s+18252} \quad (21)$$

$$K_p = -\frac{\gamma_p}{s}e\frac{\left[R(s)-Y_m(s)\right]s}{s^2+400s+18252} \quad (22)$$

$$K_i = -\frac{\gamma_i}{s}e\frac{R(s)-Y_m(s)}{s^2+400s+18252} \quad (23)$$

$$K_d = -\frac{\gamma_d}{s}e\frac{s^2Y_m(s)}{s^2+400s+18252} \quad (24)$$

2. *PSO tuning of the MRAC-PID*

To attain the most favourable adaptive gains, a choice was made to employ an integral time-weighted absolute error (ITAE) objective function as earlier used for the PID tunning. The flow-chart algorithm for the traditional PSO is also as presented in Figure 3.

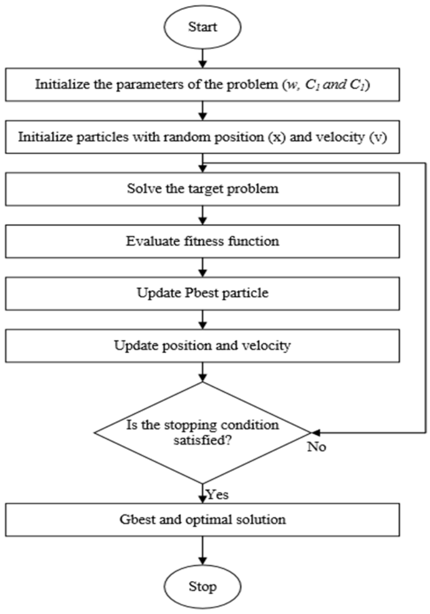

Figure 3: Flow chart for traditional PSO algorithm

Other parameters used in the optimization of the PSO MRAC-PID are as shown in Table 3.

Table 5: Set Parameters for the implementation of PSO MRAC-PID

Parameter	**Value**
Number of swarm particles	50
Number of iterations	50, 100, 150, 200, 250
Lower PID gains bound	[-110000 -35000 -0.05]
Upper PID gains bound	[-100000 -30000 -0.01]
Cognitive acceleration coefficient, c1	2
Social acceleration coefficient, c2	2
Inertial weight, w	1.0

The number of swarm particles selected in this section is 50 due the computational resources limitation. The boundary constraint was also selected by initial exploration and sensitivity analysis using MATLAB and Simulink. The inertia weight, the cognitive and social acceleration coefficients were adopted from **(**Clerc and Kennedy (2002)**)** as the coefficient have been proven to have balanced influence between the particle's personal best position and the global best position without significantly affecting stability.

IV. RESULTS AND DISCUSSION

This section presents performance evaluation of the PSO optimal model reference adaptive PID controller on the speed control of the DC motor. The performance of the PSO MRAC-PID is compared with those of the traditional MRAC-PID (Abdullahi, 2018) and the PSO-PID controller. This comparison is on the grounds of transient performance, steady state performance, disturbance rejection, load variation and command-following. For this comparison, the adaptation gains $\gamma_p, \gamma_i, \gamma_d$ of MRAC-PID are set to -1000, -1000 and -0.1 (Abdullahi, 2018). Also, the gains of the PID-PSO used (K_p, K_i, K_d) and the PSO MRAC-PID adaptation gains used $(\gamma_p, \gamma_i, \gamma_d)$ are the optimal results obtained from Table 4 and Table 5 respectively.

A. *Evaluation of the PSO-PID*

The parameters used for the PSO-PID were already presented in Table 2. To minimize the effect of stochasticity and improve optimality, the PSO was run five times with different number of iterations. The results from the optimizations are as shown in Table 4.

Table 4: PID gains obtained with different iteration numbers

S/No	Iterations	PID Gains			Convergence
		K_p	K_i	K_d	
1	50	200	2000	1	0.0027768
2	100	200	2000	1.001	0.0027723
3	150	168	1680	1	0.0033905
4	200	168	1680	1	0.0033905
5	250	168.29	1687.9	1	0.0033827

The step response of the plant with different values of iteration numbers is as shown in Figure 4

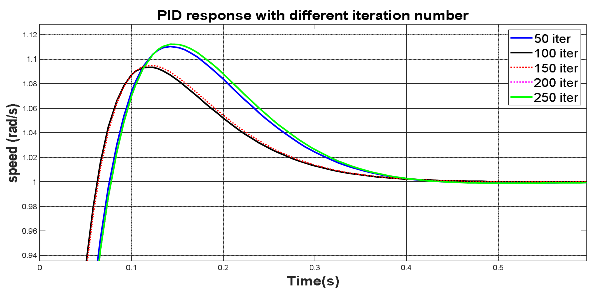

Figure 4: PID gains with different iteration numbers

Although the convergence characteristics are in close margin with slightly different optimal values due to random initialization of parameters, in DC motor control overshoot is undesirable and so from Figure 4, the optimization with 100 iterations yielded the best result as it has the minimum overshoot of 9% and a settling time of 0.4 seconds hence the values 200, 200, 1.0013 are used as the PID gains.

B. *Evaluation of the PSO-MRAC-PID*

The parameters used for the optimization of the adaptation gains have been presented in Table 3. after running the optimization for 5 different iteration numbers the result is as presented in Table 5 and the step response of the system from the iterations is as shown in Figure 5.

Table 5: Result of PSO for the MRAC with different iteration numbers

S/No	Iterations	Adaptation gains			Convergence
		γ_P	γ_i	γ_d	
1	50	-9000	-3.01E+4	-0.0297	0.00093568
2	100	-10,0000	-3.01E+4	-0.0500	0.00093568
3	150	-109770	-30000	-0.0500	0.00093926
4	200	-110000	-3.01E+4	-0.0500	0.00093568
5	250	-110000	-3.01E+4	-0.0500	0.00093568

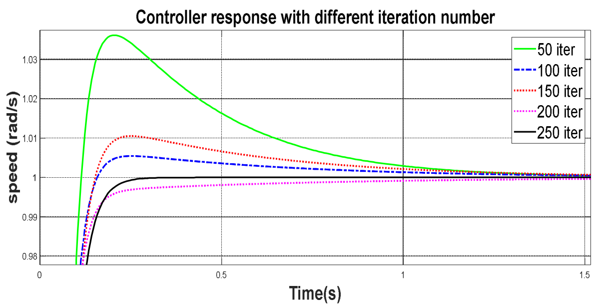

Figure 5: Step Responses for 5 different iterations

Here the convergence characteristics are also in close margin like that of the PSO-PID, but Figure 5 indicates that the response from 250 iterations has the most minimal overshoot and a settling time of 0.1 seconds hence this will be chosen as the adaptation gains for the PSO MRAC-PID.

C. *Comparison of Transient and Steady State Performance*

To compare the transient and the steady state response the unit step response of the PSO PID, MRAC-PID and the PSO MRAC-PID were plotted in Simulink environment. Figure 6 compares the transient and steady state response of the controllers.

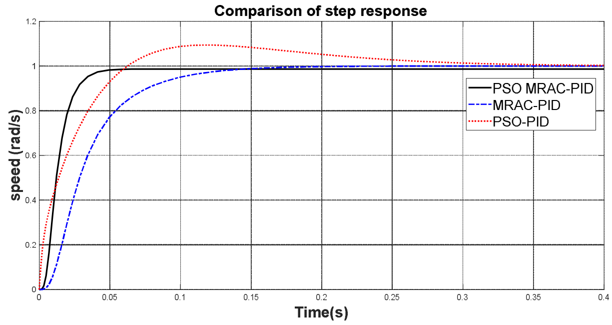

Figure 6: Step response comparison

It can be seen from the plot in Figure 6 that the PSO MRAC-PID and the MRAC-PID has no overshoot whereas the PSO-PID gave a response with 9% overshoot (1.09 rad/s). The PSO MRAC-PID has a better settling time of around 0.1 seconds, the MRAC-PID having 0.15 seconds and the PSO-PID 0.25 seconds. the PSO-PID, MRAC-PID and the PSO MRAC-PID all eliminated steady state error to zero.

D. Comparison of Regular Disturbance Rejection

To compare the ability of the PSO-PID, the MRAC-PID and the PSO MRAC-PID controllers in terms of disturbance rejection, a square wave signal of unit amplitude and a period of 1 second with a pulse width of 50% was used to simulate regular disturbance. This disturbance was injected to the controllers simultaneously and run for 1000 seconds. The comparative plot is as shown Figure 7.

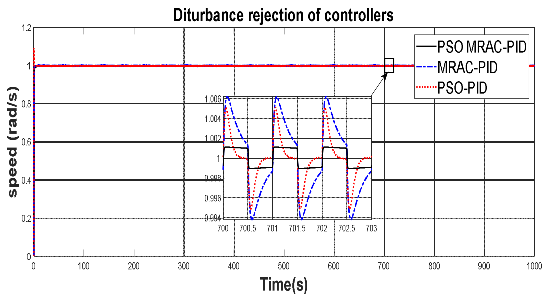

Figure 7: Comparison of Disturbance Rejection

From Figure 7, the PSO MRAC-PID was able to reject the disturbance within 0.1% overshoot, representing 99.9% disturbance rejection. The PSO-PID and the MRAC-PID rejected the disturbance to 0.6% and 0.5%, representing 99.4% and 99.5% rejection respectively. Again, this has placed the PSO MRAC-PID controller ahead in terms of disturbance rejection.

E. Comparison of Load Variation

To compare the load variation response of the DC motor, a unit step input is used as the desired speed command while a uniform random number signal is superimposed directly on the output load of the model in Simulink environment. The variation is limited to 50% (representing a magnitude of 0.5) The comparative plot of the response to the load variation is as shown in Figure 8.

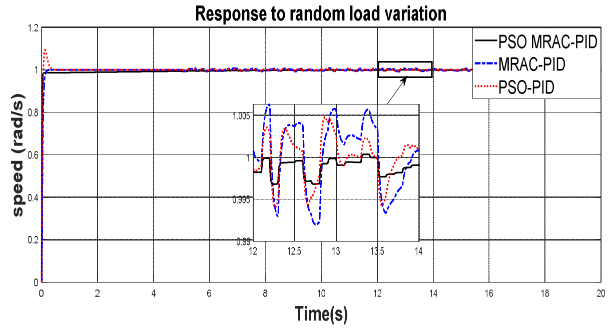

Figure 8: Performance Comparison of Load Variation

As can be seen from Figure 8, the PSO MRAC-PID has a better response to output load variation since the output speed has very minimal overshoot because of the load variation as compared with the PSO-PID and the MRAC-PID. This demonstrates that the PSO MRAC-PID has reduced speed fluctuations.

F. Comparison of Command Tracking

The command tracking ability of the controllers were compared in the Simulink environment using a unit square wave signal as the desired command. This signal was injected into the control systems simultaneously. The comparative plot is as shown in Figure 9.

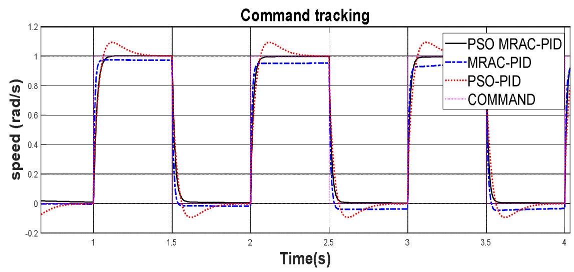

Figure 9: Performance Comparison of Command Tracking

As seen in Figure 9, the PSO MRAC-PID outperforms the PSO-PID and the MRAC-PID, tracking the square wave speed command more perfectly without overshoot. The MRAC-PID can be seen to have steady state error of 1% as it is unable to track the square wave accurately

G. Comparison of Control Effort

Similarly, the comparative control effort between the PSO-PID, MRAC-PID and the PSO MRAC-PID was done in the Simulink environment by injecting a step response as input and measuring the control signal into the plant (control effort) to maintain the desired control. Control effort here refers to the amount of DC voltage compensation required by the controller to maintain the best output speed. The comparative plot is as shown in Figure 10.

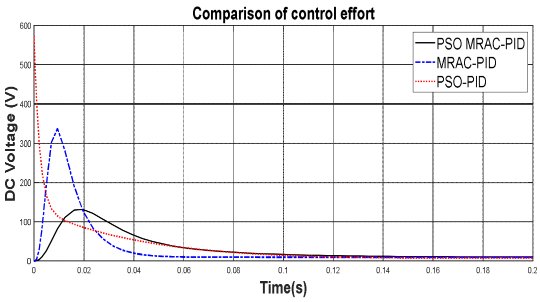

Figure 10: Performance Comparison of Control Effort

It can be seen from the plot in Figure 10 that all the controllers exhibit higher demand of voltage during the transient period that lasted for 0.1 s, settling at a steady state value of 10 V. However, the PSO MRAC-PID pulled a voltage of 120 V during the transient period whereas the MRAC-PID and the PSO-PID pulled 340V and 580 V respectively. This makes the PSO MRAC-PID better in terms of energy efficiency and robustness.

V. CONCLUSION

Performance comparisons for the PSO MRAC-PID, the PSO-PID and the MRAC-PID have been presented. With zero overshoot the PSO MRAC-PID has improved accuracy and reduced stress on the mechanical components of the DC motor system as compared with PSO-PID which has overshoot of 9%. These zero overshoots also result in minimal DC voltage requirements, leading to a more energy efficient system. This would minimize the overall capital and operational cost of the DC motor system. The improved command tracking of the proposed PSO MRAC-PID controller as shown in Figure 8 also makes it a better choice in complex and safety-critical applications where minimal error margins are highly desired, such as in surgical robots and alike. This improved tracking also helps to improve predictability and reduced wear and tear in the entire system where the controller is used.

Overall, the PSO MRAC-PID is a preferred controller when greater precision, stability, energy efficiency, productivity, reliability, safety, and adaptability are needed for operations of DC motors.

AUTHOR CONTRIBUTIONS

J. E. Oche: Conceptualization; Formal analysis; Methodology; Software; Writing - original draft; **H. A. Bashir:** Formal analysis; Methodology; Resources; Supervision; Validation; Visualization; Writing - review & editing**. T. J. Shima**: Software; Visualization.

REFERENCES

Abdullahi, U. I. (2018). Design of Adaptive PID Controller for DC Motor System. *Unpublished M.Eng. Thesis, Department of Electrical Engineering, Bayero University Kano, Nigeria.*

Ahmed, M. I., Hazlina, M. Y., & Rashid, M. M. (2016). Mathematical modeling and control of active suspension system for a quarter car railway vehicle. *Malaysian Journal of Mathematical Sciences, 10(5), 227-241.*

Akbar, M. A., Naniwa, T., & Taniai, Y. (2016). Model reference adaptive control for DC motor based on Simulink. *2016 6th International Annual Engineering Seminar (InAES) (pp. 101-106). IEEE.*

Alfi, A., & Modarres, M. (2017). A particle swarm optimization approach for PID controller design in system parameter identification. *International Journal of Optimization and Control: Theories & Applications*, 7(2), 97–104. https://doi.org/10.11121/ijocta.2017.01.02

Annaswamy, A. M. (2023). Adaptive Control and Intersections with Reinforcement Learning. *Annual Review of Control, Robotics, and Autonomous Systems, 6*, 65-93.

Åström, K. J., & Wittenmark, B. (2008). *Adaptive Control*. Dover Publications.

Banu, R. S. D., & Uma, G. (2008). PSO-based tuning of a PID controller for a high-performance drilling machine. *International Journal of Advanced Manufacturing Technology*, 37(9–10), 948–957. https://doi.org/10.1007/s00170-007-1036-5

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, vol. 6, no. 1*, pp. 58-73, Feb. 2002,

Coello Coello, C. A., & Toscano Pulido, G. (2002). A Micro-Particle Swarm Optimizer for Multi-objective Optimization. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)* (pp. 851-857). Morgan Kaufmann Publishers.

Deacha, W. (2018). Optimal PID controller design for DC motor speed control with tracking and regulating constrained optimization via cuckoo search (CS). *Engineering Journal*, 22(1), 1–15

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2019). *Feedback Control of Dynamic Systems (8th ed.). Pearson. ISBN: 978-0134685717.*

Harris, C. J., & Yu, W. (2007). Optimal adaptive control using empirical tuning rules. *International Journal of Adaptive Control and Signal Processing*, 21(8-9), 773-794. doi:10.1002/acs.982

Jain, P., & Nigam, M. J. (2013, December). Real time control of ball and beam system with model reference adaptive control strategy using MIT rule. *2013 IEEE International Conference on Computational Intelligence and Computing Research* (pp. 1-4). IEEE.

Katsuhiko, O. (2009). *Modern control engineering* (5th ed.). Editorial Félix Varela. ISBN 978-0136156734, p. 123.

Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation* (Vol. 5, pp. 4104-4108). IEEE.

Khalil, H. K. (2002). *Nonlinear systems* (3rd ed.). Prentice Hall. ISBN 978-0130673893, p. 245.

Khan, S. G., & Jalani, J. (2016). Realization of model reference compliance control of a manipulator using a disturbance observer in task space. *Robotics and Computer-Integrated Manufacturing, 41,* 33–45.

Kirange, Y. K., & Nema, P. (2024). Optimization of PID controller parameters for an SMIB system using a hybrid butterfly particle swarm optimization approach. *International Journal of Advanced Technology and Engineering Exploration, 11*(113), 529.

Landau, I. D., Lozano, R., & M'Saad, M. (2011). *Adaptive control: Algorithms, analysis and applications*. Springer. ISBN 978-0857296629, p. 310**.**

Messner B., Hill R., & Taylor J. D. (2017). Control Tutorial for MATLAB and Simulink. Available online at: Control Tutorials for MATLAB and Simulink - Motor Speed: System Modeling (umich.edu).

Narendra, K. S., & Annaswamy, K. A. (1987). *Stable adaptive systems*. Prentice-Hall. ISBN 978-0138398392, p. 197.

Nguyen, N. T (2018). Model-reference adaptive control (pp. 83-123). Springer International Publishing.

Pankaj, S., Kumar, J. S., & Nema, R. K. (2011). Comparative analysis of MIT rule and Lyapunov rule in model reference adaptive control scheme. *Innovative Systems Design and Engineering, 2*(4), 154-162.

Poli, R. (2007). An Analysis of Publications on Particle Swarm Optimisation Applications. *Department of Computer Science, University of Essex. Technical Report* CSM-469. ISSN: 1744-8050.

Ren, Y., Li, L., Brindley, J., & Jiang, L. (2016). Nonlinear PI control for variable pitch wind turbine. *Control Engineering Practice, 50*, 84-94.

Ruchika, & Kumar, N. (2013). Force/position control of constrained reconfigurable manipulators with sliding mode control based on adaptive neural network. *International Journal of Modelling, Identification and Control, 42*(3), 259-269.

Selamat, H., & Yusof, R. (2014). Introduction to Adaptive and Self-Tuning Control. *Transactions of the Institute of Measurement and Control, 41*(4), 1019-1031.

Yen, G. G., & He, X. B. (2013). Craziness-based Particle Swarm Optimization for Multi-Modal Optimization Problems. *International Journal of Swarm Intelligence Research (IJSIR), 4*(1), 1-23.

Xia, Y., & Zhang, Q. (2006). New Stable Particle Swarm Optimization. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2006)* (pp. 5156-5163). EEE.