PD, PI, PID CONTROLLER TUNING ANALYSIS FOR SPEED CONTROL OF DC MOTOR USING DIFFERENT OPTIMIZATION TECHNIQUE

DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
CONTROL AND INSTRUMENTATION

Submitted by:

SHUBHAM PATNAIK
Roll No. 2K21/C&I/10

Under the supervision
Of

SH. ROHAN PILLAI
DR. RAJESH KUMAR

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

MAY 2023

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering) Bawana Road, Delhi-110042)

CANDIDATE'S DECLARATION

I, **SHUBHAM PATNAIK**, Roll No – 2k21/C&I/10 student of M.Tech (Control & Instrumentation), hereby declare that the project Dissertation titled "**PD, PI, PID CONTROLLER TUNING ANALYSIS FOR SPEED CONTROL OF DC MOTOR USING DIFFERENT OPTIMIZATION TECHNIQUE**" which is submitted by me to the Department of Electrical Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Date: **SHUBHAM PATNAIK**

ELECTRICAL ENGINEERING DEPARTMENT
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering) Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**PD, PI, PID CONTROLLER TUNING ANALYSIS FOR SPEED CONTROL OF DC MOTOR USING DIFFERENT OPTIMIZATION TECHNIQUE**" which is submitted by Shubham Patnaik, whose Roll No. is 2K21/C&I/10, Electrical Engineering Department, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date: **SH. ROHAN PILLAI**

(SUPERVISOR)

ABSTRACT

Speed control of DC motors plays a very significant role in numerous industrial applications such as steel mills, rolling mills, locomotives etc. This research work aims to derive robust and efficient speed control technique based on Proportional-Derivative (PD), Proportional-Integral (PI) and Proportional-Integral-Derivative (PID) controller. Moreover, the optimization of all the mentioned controller parameters is carried out using PSO, GA, BWO and Coot techniques. PID controller is used for control applications to regulate speed and other process variables. It uses a control closed loop feedback to keep the practical output as close as the target output. However, selecting suitable parameters for the controller is challenging and therefore, this study uses PSO, GA, BWO and Coot optimization techniques to derive appropriate controller parameters. Different methodologies are proposed with the combination of PD, PI and PID controller with PSO, GA, BWO and Coot optimization technique through MATLAB simulations. The simulation demonstrates a comparative analysis of all the controllers with different optimization techniques and hence we can derive the best controller-optimization technique for speed control of DC motor.

Keywords: DC motor, Speed Control, PD controller, PI controller, PID controller, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Beluga Whale Optimization (BWO), Coot Algorithm, Parameter Optimization.

Place: Delhi

Date: **SHUBHAM PATNAIK**

ACKNOWLEDGEMENT

I would like to thank to all people who have helped and inspired me during my dissertation work throughout the year. I sincerely acknowledge the earnestness and patronage of my supervisor Sh. Rohan Pillai Sir and Dr. Rajesh Kumar Sir, Department of Electrical Engineering, Delhi Technological University, New Delhi, for their valuable guidance, support and motivation throughout this project work. The valuable hours of discussion and suggestion that I had with them have undoubtedly helped in supplementing my thoughts in the right direction for attaining the desired objective.

I wish express my gratitude to my beloved parents and friends for their understanding and support. Above all, thanks to Almighty for blessing and guiding me throughout my life.

Shubham Patnaik

(2k21/C&I/10)

TABLE OF CONTENTS

PARTICULARS	PAGE NO.
CANDIDATE’S DECLARATION	ii
CERTIFICATE	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
“TABLE OF CONTENTS”	vi
LIST OF FIGURES	viii
LIST OF TABLES	ix
ACRONYMS	x
CHAPTER 1	1
INTRODUCTION	1
1.1 DC MOTOR	1
1.2 WORKING PRINCIPLE OF DC MOTOR	2
1.3 SELECTION OF DC MOTOR	3
1.4 SPEED CONTROL OF DC MOTOR	4
1.5 MOTIVATION	9
1.6 OBJECTIVE	9
CHAPTER 2	10
LITERATURE REVIEW	10
CHAPTER 3	13
MODELLING OF DC MOTOR	13
3.1 PHYSICAL MODEL OF DC MOTOR	13
3.2 DC MOTOR MODELLING USING STATE SPACE ANALYSIS	14
3.3 DC MOTOR SIMULATION USING MATLAB	15
CHAPTER 4	19
CONTROLLER DESIGN FOR SPEED CONTROL OF DC MOTOR	19

4.1 PD CONTROLLER 19
4.2 PI CONTROLLER 20
4.3 PID CONTROLLER 21
4.4 OPTIMIZATION METHODS 22

CHAPTER 5 33
SIMULATION RESULTS AND DISCUSSION 33
5.1 PD CONTROLLER 33
5.2 PI CONTROLLER 34
5.3 PID CONTROLLER 35

CHAPTER 6 37
CONCLUSION AND FUTURE SCOPE 37

REFERENCES 39

LIST OF FIGURES

FIGURES	PAGE NO.
Fig 1.1: Block diagram of electromagnetic energy conversion.	1
Fig 1.2: (a) Circuit Diagram (b) N-T Curve	5
Fig 1.3: Circuit diagram for shunted Armature control method	6
Fig 1.4: Circuit diagram for field divertor method	7
Fig 1.5: Circuit Diagram for Armature Diverter	7
Fig 1.6: Circuit Diagram for tapped field control	8
Fig 1.7: (a)Two motor connected in series (b)Two motor connected in parallel	8
Fig 3.1: (a) Main Field (b) Field due to current carrying coil (c) Resultant Field	13
Fig 3.2: Position of main field F_m and rotor field F_r	14
Fig 3.3: Motor Action	14
Fig 3.4: Discrete PD controller for speed control of DC motor	16
Fig 3.5: Discrete PI controller for speed control of DC motor	16
Fig 3.6: Discrete PID controller for speed control of DC motor	16
Fig 3.7: DC Motor Parameters	17
Fig 3.8: DC motor State Space simulation model	18
Fig 3.9: State Space Matrix Simulation Mask for DC motor	18
Fig 4.1 PD Controller	19
Fig 4.2 PI Controller	20
Fig 4.3 PID Controller	21
Fig 4.4 GA Flowchart	23
Fig 4.5 PSO Flowchart	25
Fig 4.6 WBO Flowchart	28
Fig 4.7 COOT Flowchart	32
Fig 5.1: Output response of PD controller for speed control of DC Motor	33
Fig 5.2: Output response of PI controller for speed control of DC Motor	34
Fig 5.3: Output response of PI controller for speed control of DC Motor	36

LIST OF TABLES

TABLES	PAGE NO.
TABLE 5.1: PD GAIN VALUES	33
TABLE 5.2: PD STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITHOUT LOAD	33
TABLE 5.3: PD STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITH LOAD	34
TABLE 5.4: PI GAIN VALUES	34
TABLE 5.5: PI STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITHOUT LOAD	35
TABLE 5.6: PI STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITH LOAD	35
TABLE 5.7: PID GAIN VALUES	35
TABLE 5.8: PID STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITHOUT LOAD	36
TABLE 5.9: PID STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITH LOAD	36

ACRONYMS

Emf: Electro motive Force

PD: Proportional Derivative

PI: Proportional Integral

PID: Proportional Integral Derivative

GA: Genetic Algorithm

BWO: Beluga Whale Optimization

FLC: Fuzzy Logic Control

PLC: programmable logic controller

MOPSO: Multi-Objective Particle Swarm Optimization

PSO: Particle Swarm Optimization

DNC: Direct Neural Controller

PWM: Pulse Width Modulation

TLBO: Teacher Learning Based Optimization

GWO: Grey Wolf Optimizer

HHO: Harris Hawks Optimization

ASO: Atom Search Optimization

FPGA: Field Programmable Gate Arrays

ANFIS: Adaptive Neuro-Fuzzy Inference System

CHAPTER 1
INTRODUCTION

1.1 DC MOTOR

A DC generator is a device that transforms mechanical energy into DC electrical energy. The same device is known as a DC motor when it is used to transform DC electrical power into mechanical power. A DC generator and a motor are identical from a construction standpoint. In applications like electric traction, where a wide range of speeds and accurate speed regulation are necessary, DC motors are highly helpful.

A DC motor is an electromechanical energy conversion device (electrical machine) that transforms electrical energy or power (EI) from a DC source into mechanical energy or power (ωT).

Industrial devices like hammers, rollers in the paper, drilling machines, lathes and steel industries, blowers for furnaces, etc. are driven by electric motors. Domestic appliances like refrigerators, fans, water pumps, toys, mixers, etc. are also driven by electric motors. Figure 1.1 depicts the block diagram of energy conversion when an electro-mechanical device functions as a motor.

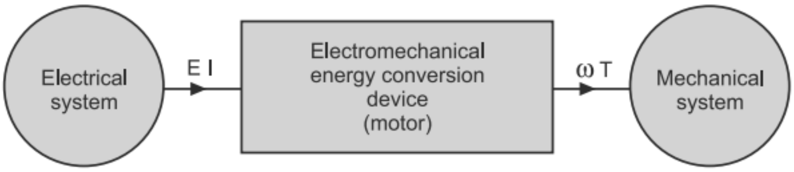


Fig 1.1: Block diagram of electromagnetic energy conversion.

1.1.1 WORKING PRINCIPLE OF DC MOTOR

The idea behind how an electric motor works is that a coil rotates when it is put in a magnetic field because electricity flows through it.

Align the poles of two bar magnets with a thin spacing between them. With a short piece of conducting wire, create a loop. So that it stays within the magnetic field of the magnets, keep this loop in the gap between the magnets. The last step is to connect the ends of the loop.

You'll notice that your loop moves when energy is applied to your simple circuit. Why does this happen? The magnetic field created by the electric current passing through the conductor is disrupted by the magnetic field created by the magnets. One side of the loop will be pulled to the north pole and the other to the south pole because the loop has transformed into a magnet. The loop thereafter rotates endlessly as a result.

1.2 APPLICATIONS OF DC MOTOR

Different types of DC motors are used for various applications according to the features of DC motors, as mentioned below:

1.2.1 SEPARATELY EXCITED DC MOTOR

These motors may produce very precise speeds. Additionally, these motors work best in situations where a need for speed variation between very low and high values exists.

These motors are employed in paper mills, diesel-electric ship propulsion systems, steel rolling mills, and other facilities.

1.2.2 SHUNT MOTOR

Based on their properties, we can infer that shunt motors operate at nearly constant speeds. Therefore, it is utilised in the following situations:

(i) When a nearly constant speed must be maintained from no load to full load.

(ii) When it is necessary to drive the load at several RPMs (different speeds are acquired using speed control methods), and any one of the speeds must be kept almost constant for a sizable amount of time.

1.2.3 SERIES MOTOR

A series motor's features show that it is a variable-speed motor, meaning that its speed is low at larger torques and vice versa. Additionally, the motor reaches dangerously high speeds at low or no loads. Therefore, it is used:

(i) In situations where a lot of torque is needed when large loads are first starting to accelerate quickly.

(ii) In situations where there are significant load fluctuations and speed adjustments are necessary automatically.

As a result, series motors are best suited for electric traction, air compressors, hoover cleaners, hair dryers, sewing machines, cranes, lifts and other similar devices.

1.2.4 COMPOUND MOTOR

This motor's key feature is that, unlike a series motor, its maximum speed is safely constrained at light loads while the speed reduces noticeably at heavy loads. As a result, it is used:

(i) Where strong torque is needed upon starting and where the load may be rapidly thrown off.

(ii) In locations where there are significant load swings.

As a result, cumulative compound motors work best for lifts, mine hoists, rolling mills, punching and shearing machines, etc.

1.3 SELECTION OF DC MOTOR

The following factors should be taken into account when choosing a DC motor for a certain task:

1.3.1 CHOOSING POWER RATING

A motor will operate at a lighter load than its rating if its size (HP or kW capacity) is greater than the load it must lift. As a result, there will be increased losses (i.e., the motor would work inefficiently) and unnecessary power wasting. Therefore, operating costs will increase. Its initial cost will be higher at the same time.

On the other hand, if the motor's size is smaller than the load it must lift, it will be overloaded and heated above its safe operating temperature. Consequently, if the motor is used continuously, it may become damaged. Otherwise, the protection mechanisms will also disconnect the motor when it is overloaded, which would have an impact on production.

Therefore, a motor's size should be determined by how much power it can use. It should be warmed up during use within the permitted temperature ranges, and it should never be overheated. However, it should also be able to temporarily handle any excess loads.

1.3.2 CHARACTERISTICS OF MOTOR

Choosing a motor with the right power rating is only one factor in ensuring smooth operation; one must also be aware of a motor's characteristics, or how it behaves under various load conditions, in order to complete a task quickly and effectively without experiencing any problems. Therefore, one should be aware of the following task specifics before choosing a motor.

The amount of torque needed for starting and running under various loads.

(i) The need for braking and accelerating torque
(ii) Switching frequency
(iii) Workplace temperature
(iv) Environmental circumstances, etc.

Following knowledge of the aforementioned specifics at the workplace, a suitable motor (with appropriate enclosures) to satisfy the criteria is chosen.

1.4 SPEED CONTROL OF DC MOTOR

The speed of DC series motors can be controlled by any one of the following methods:

(i) Armature control method
(ii) Field control method
(iii) Series – parallel control method

ARMATURE CONTROL METHOD

A DC series motor's speed can be altered by making different changes to the armature circuit.

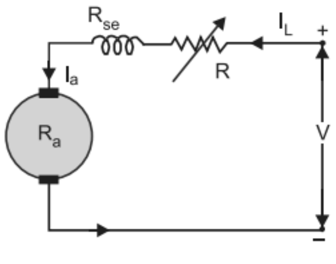
These techniques have been modified and are known as:

Armature series resistance control method: As shown in Fig. 1.2(a), this method involves connecting a variable resistance in series with the armature or motor. The speed of the motor depends on back emf, or $N \propto E_b$, if the load and torque produced by the machine are constant. when the armature is not connected to any additional resistance in series. Consequently, $E_{b1} = V - I_a (R_a + R_{se})$. The back emf, however, is equal to $V - I_a (R_a + R_{se} + R)$ when an additional resistance R is connected in series with the armature. Naturally, $E_{b2} > E_{b1}$, and $N_2 > N_1$. Therefore, we can get speeds below normal by adding an extra resistance in series with the armature.

Figure 1.2(b) depicts the speed-torque characteristics of a DC series motor with and without additional series resistance. Depending on the load, a maximum speed control range of around 3:1 will be possible.

This is the approach used the most frequently to regulate the speed of series motors. Since the control is frequently used in these applications to reduce speed under light loads, the power loss in the control resistance is not too significant for many DC series motor applications.

Due to the intermittent nature of such drives, the main uses of this form of speed control are for moving cranes, hoists, trains, etc.



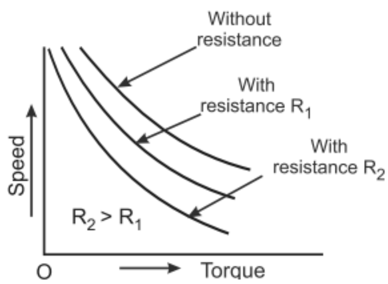


Fig 1.2: (a) Circuit Diagram (b) N-T Curve

Shunt Armature Control Method: This approach uses a rheostat R_1 in series with the armature and a rheostat R_2 shunting the armature, as shown in Fig. 1.3. At low loads, it offers modest speeds. With this setup, the speed may be controlled by adjusting the flux and reducing the voltage applied to the armature. In reality, the exciting current is controlled by adjusting the shunting rheostat R_2, while the voltage delivered to the armature terminals is controlled by varying the series rheostat R_1, keeping the armature current constant.

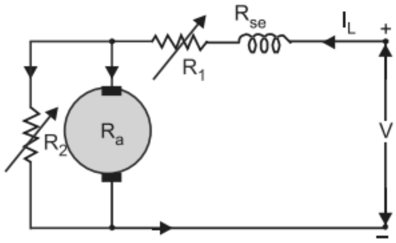


Fig 1.3: Circuit diagram for shunted Armature control method

As armature current decreases in this manner due to the addition of armature diverter R_2, flux I must increase in order to compensate for the armature's constant torque development ($T \propto I_a$). As a result, more current is pulled from the supply mains, increasing the field and slowing down the speed.

Thus, speeds below average can be attained using this strategy. By using various R_1/R_2 ratios, it is possible to reach a wide variety of speeds below the standard.

Because this technology is not very cost-effective due to significant power losses in the speed controlling resistance, it is only used in situations where speed control for short internals is necessary.

Armature Terminal voltage control method: This approach alters the supply voltage, which alters the speed of a DC series motor, using a variable voltage power source. However, because such equipment is so expensive, this approach is rarely used.

FIELD CONTROL METHOD

By altering the flux that the series field winding produces, series motors can have their speed changed.

Any of the following methods may be used to cause the flow to vary:

Field Diverters: As seen in Fig. 1.4, this method involves connecting a variable resistance R in parallel with the series field winding. It has the result of changing the direction in which the motor's I_L current flows. The flux Φ is decreased when a portion of the current I_D passes through the diverter and less current flows through the series field winding. As a result, the motor's speed increases (N ∝ I/ Φ). Thus, only speeds over average may be attained using this strategy.

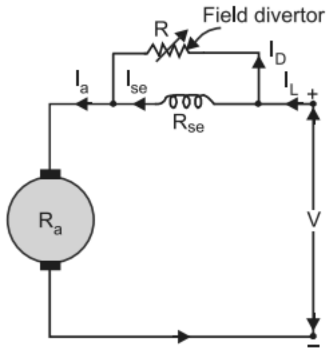


Fig 1.4: Circuit diagram for field divertor method

Armature Diverter: As seen in Fig. 1.5, this method involves connecting the variable resistance R in parallel with the armature. It has the result of changing the course taken by the line current I_L. The diverter allows some of the current I_D to pass through, which lowers the armature current I_a. If I_a is decreased for a given constant load torque, Φ must increase (T ∝ Φ I_a). The motor's current draw increases as a result, and speed decreases (N∝1/Φ). This approach allows for the acquisition of any speed below normal by varying the diverter resistance value.

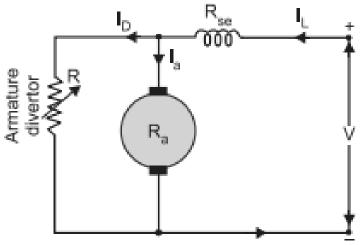


Fig 1.5: Circuit Diagram for Armature Diverter

Tapped Field Control: According to this technique, a section of the series field winding can be short-circuited to alter the number of turns it has, as seen in Fig.1.6 . We are aware that the ampere-turns (Φ ∝ I_{se} x No. of turns) determine how much flux the winding produces. The speed of the motor increases as the number of rotations is decreased (N ∝ 1/Φ). Thus, using this strategy, only speeds above the norm may be attained.

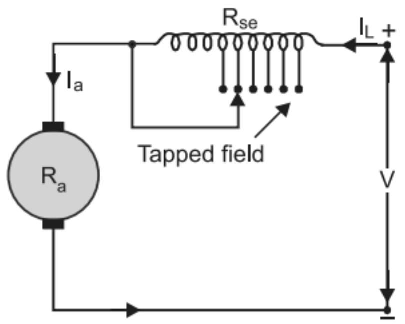


Fig 1.6: Circuit Diagram for tapped field control

VOLTAGE CONTROL METHOD:

By connecting the series motors in series, parallel, or a combination of the two, the voltage between them can be altered using this technique. The usage of this in electric traction is widespread.

Let's use just two comparable series motors with mechanically coupled shafts in order to simplify the explanation of this technology. As may be seen in Figs. 1.7 (a) and (b), they are first joined in series and then in parallel. Fig. 1.7 displays the voltage across each motor as well as the current flowing through it.

When in series: speed, $N_1 \propto \frac{E_{b1}}{\emptyset_1} \propto \frac{V/2 - I_a(R_a - R_{se})}{I_a}$ (1.1)

Neglecting drops, $N_1 \propto \frac{V/2}{I_a} \propto \frac{V}{2I_a}$ (1.2)

When in parallel, $N_2 \propto \frac{E_{b2}}{\emptyset_2} \propto \frac{V - \frac{I_a}{2}(R_a + R_{se})}{\frac{I_a}{2}} \propto \frac{2V}{I_a}$ (1.3)

From equation 1.1 and 1.3, $\frac{N_2}{N_1} = \frac{2V}{I_a} X \frac{V}{2I_a} = 4 \; or \; N_2 = 4N_1$ (1.4)

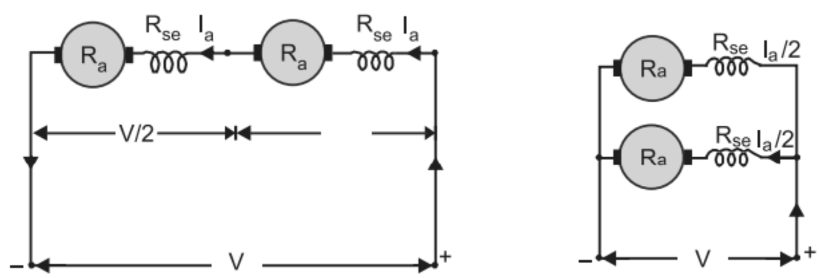


Fig 1.7: (a)Two motor connected in series (b)Two motor connected in parallel

As a result, modest speeds are obtained when the motors are connected in series, and high speeds (almost four times that of the first case) are obtained when they are connected in parallel. By using more motors and connecting them in series, parallel, or a combination of the two, one can achieve a variety of speeds.

This technique is typically used in electric locomotives to regulate train speed. In this situation, a series-parallel and resistance control technique combination is used.

1.5 MOTIVATION

A popular controller is the PID (Proportional-Integral-Derivative) because of its straightforward and effective operation under various operating circumstances. The creation of a control of this nature that necessitates the specification of three parameters for a short period of time, a short period of time, and a period of time from the other. Tuning is the process of setting the PID parameter. PID control adjustment via various techniques. In Genetic Algorithm (GA) are too slow in determining the controller parameters. In Particle Swarm Optimization (PSO) the quality of the solution is very low. Therefore, in this study a new optimization technique is used for tuning of controller parameters.

1.6 OBJECTIVE

1. State Space modelling for speed control of DC motor.
2. PD controller tuning for speed control of DC motor using Genetic Algorithm (GA) , Particle Swarm Optimization (PSO) and Coot Algorithm technique.
3. PI controller tuning for speed control of DC motor using Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Coot Algorithm technique.
4. PID controller tuning for speed control of DC motor using Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Coot Algorithm, Beluga Whale Optimization (BWO) Technique.
5. Analysing all the controller techniques and determining which controller is more reliable and efficient.

CHAPTER 2

LITERATURE REVIEW

A lot of work has been put into controlling the speed of DC motors using various methods, including PID, fractional-order dynamics, and sliding mode control (SMC).
In the year 1976, a speed controller for motor that measured, compared, and adjusted motor speed was given. The prototype model's speed control range was from 200 rpm to 3000 rpm. By using a digital technique, measurement nonlinearity was eliminated and the speed setting was precisely repeatable [1]. Implementation of a controller for neural network trajectory for a DC motor in a laboratory was done in 1993 where the speed of the rotor was controlled and it was made to follow trajectory multiple times [2].

In 1995, (MNN) Multilayer Neural Network was developed for the purpose of identification in real-time control of the DC motors [3]. In 1998, controller with variable structure and feedback gains and also, a switching vector for a separately stimulated DC motor were chosen using a novel evolutionary algorithm application [4]. In the research done in year 1999, a slightly different approach of developing a fuzzy logic speed controller and implementing for a DC motor while utilizing microcontroller with fuzzy logic was proposed. That controller could easily be upgraded and become an adaptive fuzzy controller by employing minimum number of components for implementation [5].

In the year 2000, Design for the parameters of the controller for separately stimulated DC motor was illustrated a study using an application of GA. The design criterion was transformed into an optimization problem and subjected to the general GA laws. The suggested GA adopted a new selection technique that combined elitism with roulette wheel selection to accelerate convergence to the ideal solution [6]. The speed controlling of a DC motor was simulated in the work using FLC in a MATLAB in 2004 [7].

In the year 2006, with the use of TMS320LF2407A digital signal processor which was low cost, real-time DC motor speed management was achieved. A set of coefficients for a PID controller that had the desired controller properties were created using MATLAB tools [8]. Another study presented a simpler method for individually excited DC motor speed regulation utilizing a PLC. This method was based upon feeding DC motor's armature circuit with a variable DC voltage from a fixed DC voltage using PLC that served as a DC/DC chopper [9].

For finding out the set for trade off solutions, which were also known as Pareto-set optimization solution for objective functions, which were conflicting for DC motor system, a controlling strategy based on MOPSO was proposed [10]. One study applied a DNC to control the speed of the DC motor, and examined how the sampling frequency affected the DNC's performance [11]. In the year 2012, the use of PSO to the design of an ideal fuzzy logic DC motor controller was demonstrated [12]. In year 2014, infrared remote control and the ATmega16 microcontroller were used to control DC motor's speed using the PWM technique [13]. A MATLAB-based PI fuzzy rule-based controller for dc motor speed control was designed [14]. Using an Arduino Due, a real-time fuzzy logic speed tracking controller for a DC motor was created [15]. The decrease of input chattering through the sliding mode control of a brushless DC motor speed was observed using an analogue switching function [16]. The utilization of a cascade architecture with a proportional-integral (PI) controller was done [17]. By utilizing the Taguchi approach, a study proposed an ideal PID controller design for a DC motor [18]. Another study described an embedded Linux and FPGA-based remote DC motor control. The proposed approach was based on a co-design architecture in which an FPGA was connected to an embedded Linux operating system which was running on a Beagle-bone via the master-slave SPI protocol. An experimental PWM DC motor drive had been implemented in this study using an arm-based platform, a Linux compatible operating system, and an FPGA [19]. Another work suggested using speed to regulate a direct current (DC) motor with an input PWM signal and it was possible to model an actual system using tuning and motor data [20]. Various nature inspired algorithms have been used to tune PID gain values like Genetic Algorithm has been used to find out values of PID gains [21]. PSO had been used in boost converter [22]. Differential Evolution was used for controlling heated steam temperature [23]. Fuzzy PID controller was optimized in [24] using the teacher learning based optimization (TLBO) technique. PSO was used in to adjust the PID parameters for a DC motor [25]. GWO was used in [26] for speed control and for fine-tuning PID parameters. Jaya Optimization Algorithm was also implemented in finding out the gain values [27]. HHO algorithm was used for fine-tuning a PID controller for DC motor speed regulation [28]. GWO was used to optimize the controller [29]. In order to assess the effectiveness of controllers which were in command following control and also, disturbance rejection control, both PID and PI controllers design which were using the modulus hugging approach was provided in the research study [30]. The fractional order proportional integral derivative (FOPID) controllers' ideal parameters were proposed to be determined using the ASO algorithm and a brand-new chaotic version of it, called chaotic ASO [31].

PID controller employing fuzzy logic is a significant area of current study. The work of Sant and Rajagopal used a switching function between a FL controller and a traditional PID controller to manage the speed of the motor. Each controller type could perform under many operations thanks to the switching algorithm[32]. For generation of delta change in the control signal for each time step, a fuzzy PID controller was designed by Hohan and Sinha, based on the errors, difference in errors, and the difference in difference [33]. For better control of a motor drive, Rubaai et al.'s work used GA to evolve the ideal scale factors of the fuzzy-based PID gains. The genetic scale factor increased the sensitivity of the PID control to mistake when used with FL gain selection [34]. The creation of an online learning-capable fuzzy neural networks PI/PD-like controllers for brushless drive system speed trajectory monitoring was implemented [35].

Digital PID controllers were implemented using Microprocessors [36] and Microcontrollers too [37]. Digital control systems could also be realised using FPGAs [38]. FPGA based controllers included benefits like quick computation, intricate functionalities, low power usage and real-time processing. With the use of a GA and an (ANFIS), a speed controller to be used for DC motor was constructed [39]. The method of a single neuron PID adaptive control for BLDCM was presented which was based on wavelet neural network online identification. The approach had the advantages of being simple to manufacture and adaptable, used a single neuron PID to build an adaptive controller [40].

Thus, the speed of the DC motor has been tried to be controlled using various controllers namely PD, PI, PID or be it related to FPGAs or even neural networks.

CHAPTER 3

MODELLING OF DC MOTOR

3.1 PHYSICAL MODEL OF DC MOTOR

A DC motor works on the premise that a current-carrying conductor experiences a mechanical force when it is exposed to a magnetic field. Fleming's Left Hand Rule determines the force's direction, and the relation between its magnitude and direction states:

$$F = Bil \text{ newton}$$

Let's assume that a bipolar machine's magnetic field only affects one coil of the armature [see Fig. 3.1(a)]. Current flows through the coil when the DC supply is connected, creating the field as depicted in Fig. 3.1(b). A resultant field, as shown in Fig. 3.1(c), is created by the interaction of the two fields (i.e., field produced by the main poles and field produced by the coil). This has a propensity to return to its initial position, which is in a straight line, which causes force to be applied to the two sides of the coil and torque to form, which causes the coil to rotate.

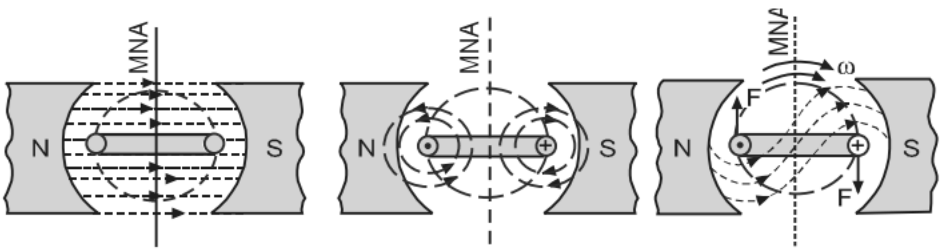


Fig 3.1: (a) Main Field (b) Field due to current carrying coil (c) Resultant Field

Another way to put it is that the principal poles generate the field F_m. Fig. 3.2 indicates its direction. The coil's (armature conductors') own field, designated as F_r, is produced when current is applied to it. As this field seeks to align itself with the main field, a clockwise electromagnetic torque forms as seen in Fig. 3.2.

On the armature of the actual machine, there are many conductors attached. The current flows in one direction (outward) through all conductors when they are all placed under the influence of one pole, as the North Pole.

The current flows in the opposite direction through the other conductors that are under the influence of the south pole, as shown in Fig. 3.3. A rotor field is created as a result. The arrowhead F_r indicates its direction. Torque (T_e) emerges as this rotor field F_r attempts to align with the main field F_m. Rotor thus turns.

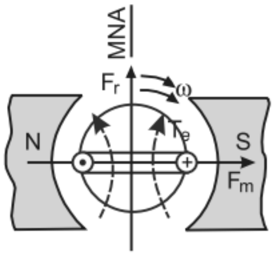


Fig 3.2: Main field F_m and rotor field F_r positions

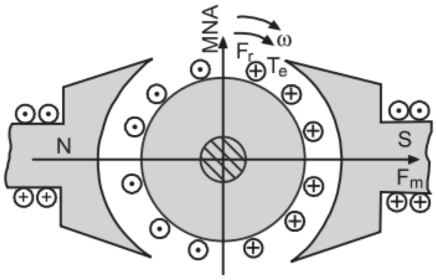


Fig 3.3: Motor Action

As can be observed, in order to produce a continuous torque, each conductor or coil side's direction of current flow must be reversed when it passes through the magnetic neutral axis (MNA). Using a commutator, this is accomplished.

3.2 STATE SPACE ANALYSIS OF DC MOTOR

The different equations related to DC motor are given below:

$$e_m(t) = K_m \frac{d\theta(t)}{dt} \tag{3.1}$$

$$e_a(t) = L_m \frac{di_a(t)}{dt} + R_m i_a(t) + e_m(t) \tag{3.2}$$

$$T(t) = K_t i_a(t) \tag{3.3}$$

$$J \frac{d\theta^2(t)}{dt^2} + B \frac{d\theta(t)}{dt} = T(t) \tag{3.4}$$

Where $i_a(t)$ is armature current, $e_m(t)$ is back emf, and $e_a(t)$ is armature voltage, T(t) = torque developed, θ(t) is the motor shaft angle, and $\frac{d\theta(t)}{dt} = \omega(t)$, θ is the Shaft Speed, J is the rotor's moment of inertia, B is the viscous frictional constant, L is the armature windings inductance, R_m is their resistance, K_t is the motor torque constant, and K_m is the motor constant.

Here, changing the armature voltage e_a(t) regulates the motor speed ω(t). As a result, the input variable is e_a(t), and the output variable is ω(t).

The state variable is calculated as follows:

$$x_1(t) = \omega(t) = \frac{d\theta(t)}{dt} \tag{3.5}$$

$$x_2(t) = i_a(t) \tag{3.6}$$

The state equations are derived using above equation:

$$\frac{dx_1(t)}{dt} = -\frac{B}{J}x_1(t) + \frac{K_t}{J}x_2(t) \tag{3.7}$$

$$\frac{dx_2(t)}{dt} = -\frac{K_m}{L_m}x_1(t) - \frac{R_m}{L_m}x_2(t) + \frac{1}{L_m}e_a \tag{3.8}$$

$$y(t) = \frac{d\theta(t)}{dt} = \omega(t) = x_1(t) \tag{3.9}$$

Now the state model of the DC motor is derived from the equations (3.7), (3.8), (3.9)

$$\begin{bmatrix} \frac{dx_1(t)}{dt} \\ \frac{dx_2(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{B}{J} & \frac{K_t}{J} \\ -\frac{K_m}{L_m} & -\frac{R_m}{L_m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_m} \end{bmatrix} u(t)$$
(3.10)

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{3.11}$$

3.3 DC MOTOR MODEL SIMULATION USING MATLAB

As shown in figure 3.4, 3.5, 3.6 the DC motor measured speed is fed to A-D converter and is subtracted from the desired speed to calculate the error. The error is then given to controller which further gives the voltage input to DC motor. The DC motor taken as two inputs, one is voltage from the controller and another is the torque disturbance. Simulation circuit of discrete PD, PI and PID controller for speed control of DC motor are shown below:

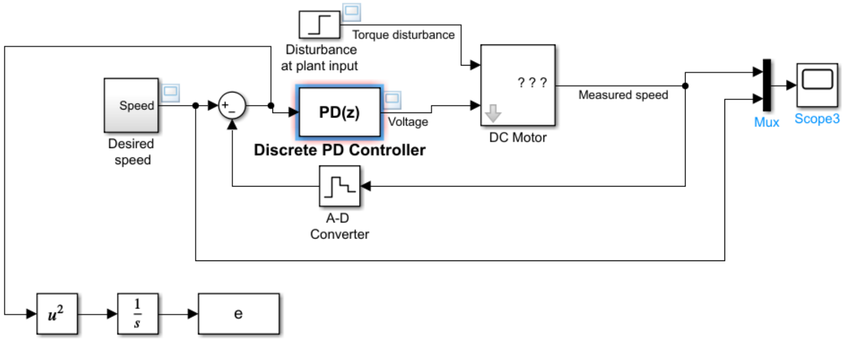


Fig 3.4: Discrete PD controller for DC motor

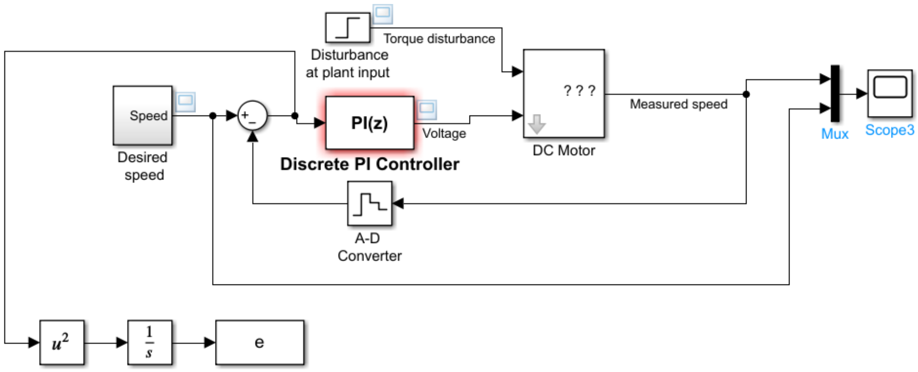


Fig 3.5: Discrete PI controller for DC motor

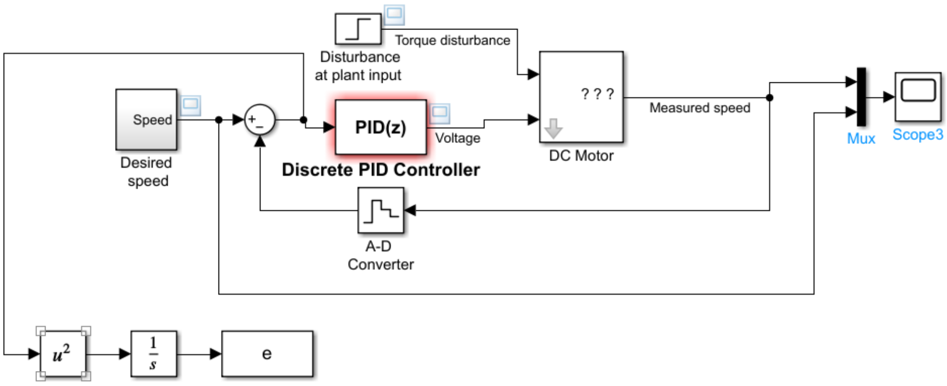


Fig 3.6: Discrete PID controller for DC motor

3.3.1 DC MOTOR PARAMETER

Figure 3.7 shows the mask of DC motor taken in the simulink.

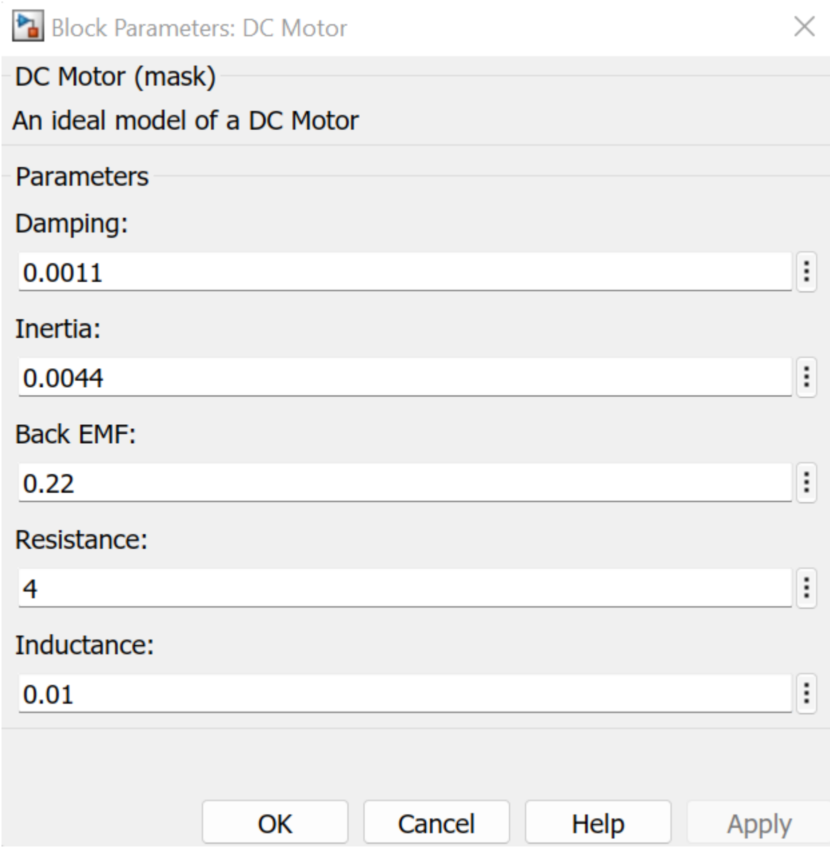


Fig 3.7: DC Motor Parameters

DC motor used in the simulation has the following parameter:

Back EMF (b)	:	0.22 V
Motor Torque Constant (K_t)	:	0.0011 N-m/A
Motor Constant (K_m)	:	0.0011 V-sec/rad
Armature Winding Resistance (R_m)	:	4 Ohm
Inductance of Armature Winding (L_m)	:	0.01H
Moment of Inertia of Rotor (J)	:	0.0044 Kg-m^2

3.3.2 STATE SPACE MODEL OF DC MOTOR USING MATLAB

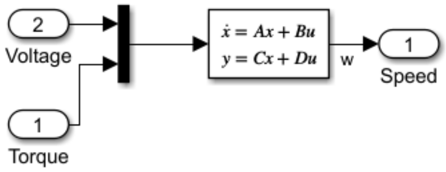


Fig 3.8: DC motor State Space simulation model

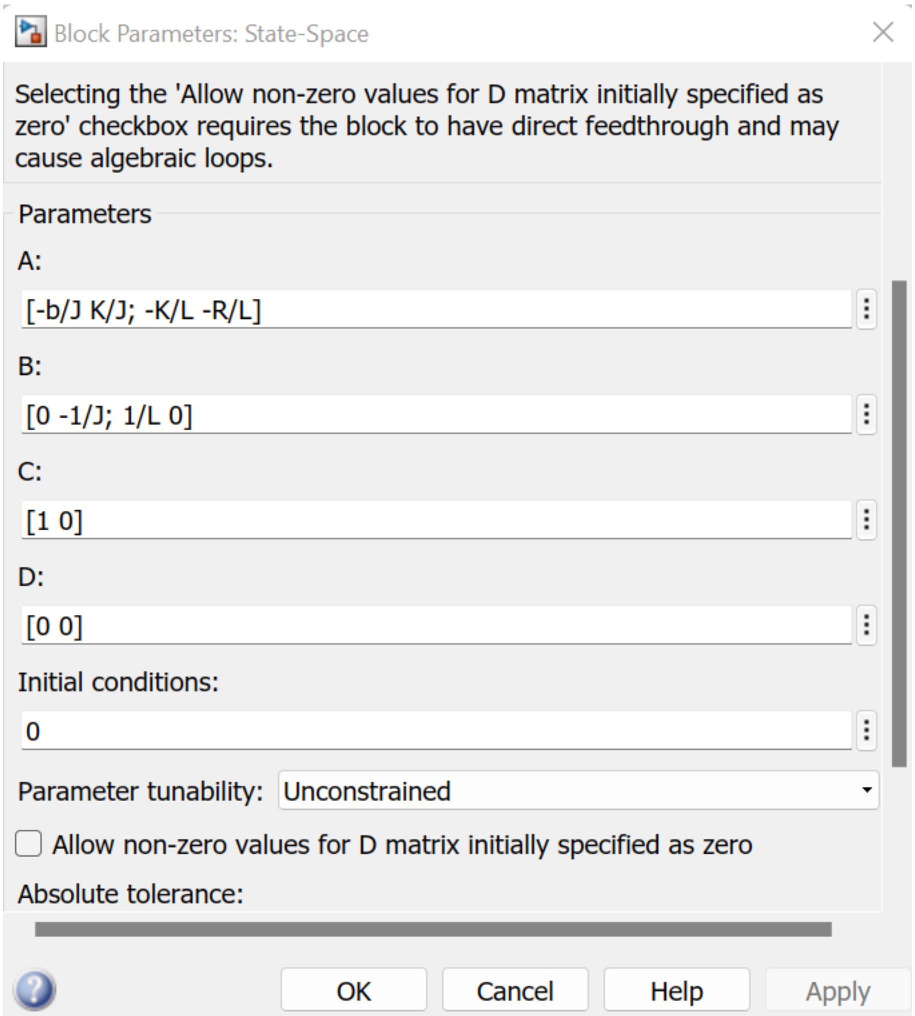


Fig 3.9: State Space Matrix Simulation Mask for DC motor

Figure 3.8 and 3.9 shows the state space model block and mask of DC motor for speed control applications. State parameter taken in the simulation are shown in figure 3.9.

CHAPTER 4

CONTROLLER DESIGN FOR SPEED CONTROL OF DC MOTOR

4.1 PD CONTROLLER

A proportional derivative controller is one where output varies proportionally to both error signal and its derivative in a control system. This particular kind of controller combines the effects of both proportional and derivative control actions.

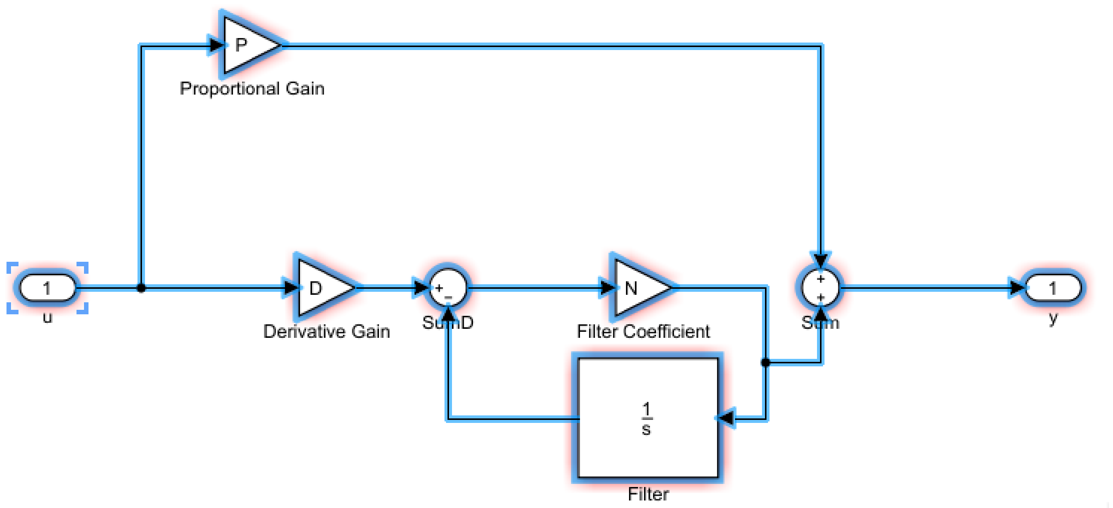


Fig 4.1 PD Controller

In this type of controller, both the error signal and its integral are proportional to the control signal. Equation (4.1), which describes the mathematical formulations for proportional and integral controllers, defines e(t) as difference between the reference input and achieved output.

$$c(t) = K_p e(t) + K_d \frac{d}{dt} e(t) \tag{4.1}$$

The differential controller's control effect was used individually in the control system. A more effective system is produced when the proportional and derivative controllers are combined. In this case, proportional controllers get rid of the problems with differential controllers.

It turns out that these controller's main purpose is to tune the output when the error signal changes. For a constant error signal, it stays the same. This is due to the fact that if the error signal's value stays constant, the rate of change over time is 0. In order to take into consideration even constant error signals, we therefore combine a differential controller and a proportional controller.

Sensitivity is raised when a proportional controller has a derivative control action. This contributes to greater system stability by generating an early corrective response even with low error signal quantities. However, we are also aware that the derivative controller increases steady-state inaccuracy. Conversely, proportional controllers minimize steady-state errors.

4.2 PI CONTROLLER

A proportional-integral controller (often referred to as a PI controller). a particular kind of controller that combines the effect of both the integral as well as proportional control actions. Hence, it is termed a PI controller.

Controllers that combine proportional and integral control strategies are known as proportional-integral controllers. Combining two distinct regulators yields a more effective regulator that does away with the shortcomings of each one.

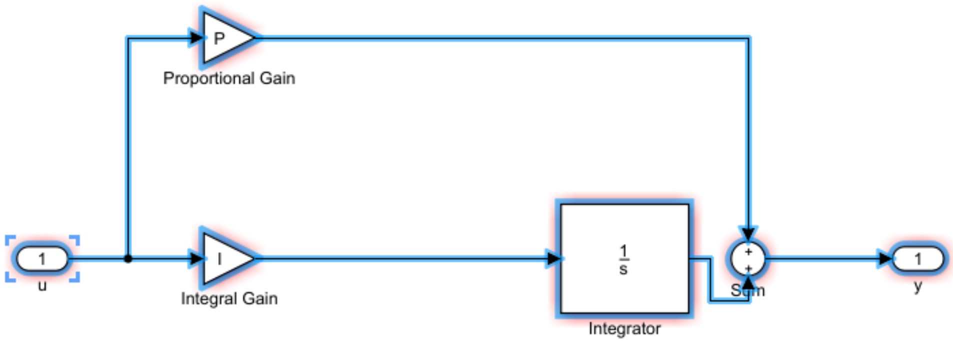


Fig 4.2 PI Controller

The error signal and its integral are both proportional to the control signal in this situation. The mathematical equations for proportional and integral controllers are provided by

Equation (4.2) where e(t) represents difference between the reference input and achieved output.

$$c(t) = K_p e(t) + K_i \int e(t) \tag{4.2}$$

Integral governors have a significant instability issue as its main flaw. Due to their significantly delayed response time to created errors, monolithic controllers are the cause of this. The fact that steady-state errors are greatly reduced by proportional controllers, leading to a more stable system, is a significant advantage of these devices.
This is the main reason why combining the two results in a controller that produces consistent outcomes.

4.3 PID CONTROLLER

A Proportional Integral Derivative controller is a form of controller whose output fluctuates in proportion to the error signal, its integral, and its derivative. PID is the abbreviation for this kind of controller.

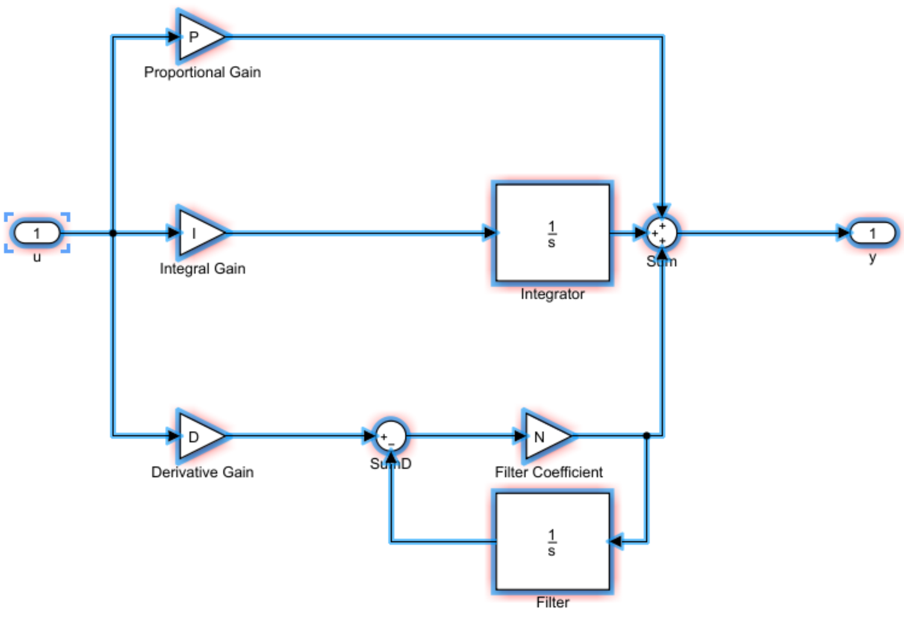


Fig 4.3 PID Controller

Combining all three of these control action types enhances the performance of the system and produces the intended result. Equation (4.3), where e(t) represents difference between the reference input and achieved output.

$$c(t) = K_p e(t) + K_i \int e(t) + K_d \frac{d}{dt} e(t) \tag{4.3}$$

As we already know, the combined action of proportional and integral controllers in PI controllers minimises the steady-state error, which benefits the entire control system. As a result, the system as a whole operates more steadily. However, since no improvement is shown in this instance, the system's stability is unchanged.

We also acknowledge that the PD controller makes the system more sensitive. This is so that the controller output in this instance changes in proportion to the error signal and its derivative. Therefore, you will observe significant changes in the output even if the rate of change of inaccuracy is minor.

By doing this, the system generates an early corrective response, increasing its overall stability. The steady-state inaccuracy is unaffected by PD controllers, which is a noteworthy characteristic. A derivative controller causes a steady-state error, to put it another way. With integrated controllers, stability issues happen. To get around each of the weaknesses of each type of controller, PID controllers are employed. PID controllers produce systems with more stability and less steady-state error as a result.

4.4 OPTIMIZATION METHODS

4.4.1 GENETIC ALGORITHM

Genetic Algorithm, which is the mechanism of propelling of biological evolution, is said to be the foundation, a technique used for tackling limited optimization issues which are unconstrained. The population of a single solution is kept to be modified iteratively by GA. GA chooses members from the present population for serving as parents at every stage and then, employs them for producing the offsprings that will make up the next generation.

The population evolves towards the ideal outcome over generations. Numerous optimization issues which are not well suited for conventional optimization techniques can be solved using genetic algorithms. This includes issues where the objective function is stochastic, severely nonlinear, nondifferentiable, or discontinuous. Mixed integer programming issues with certain components restricted to integers can be handled by genetic algorithms.

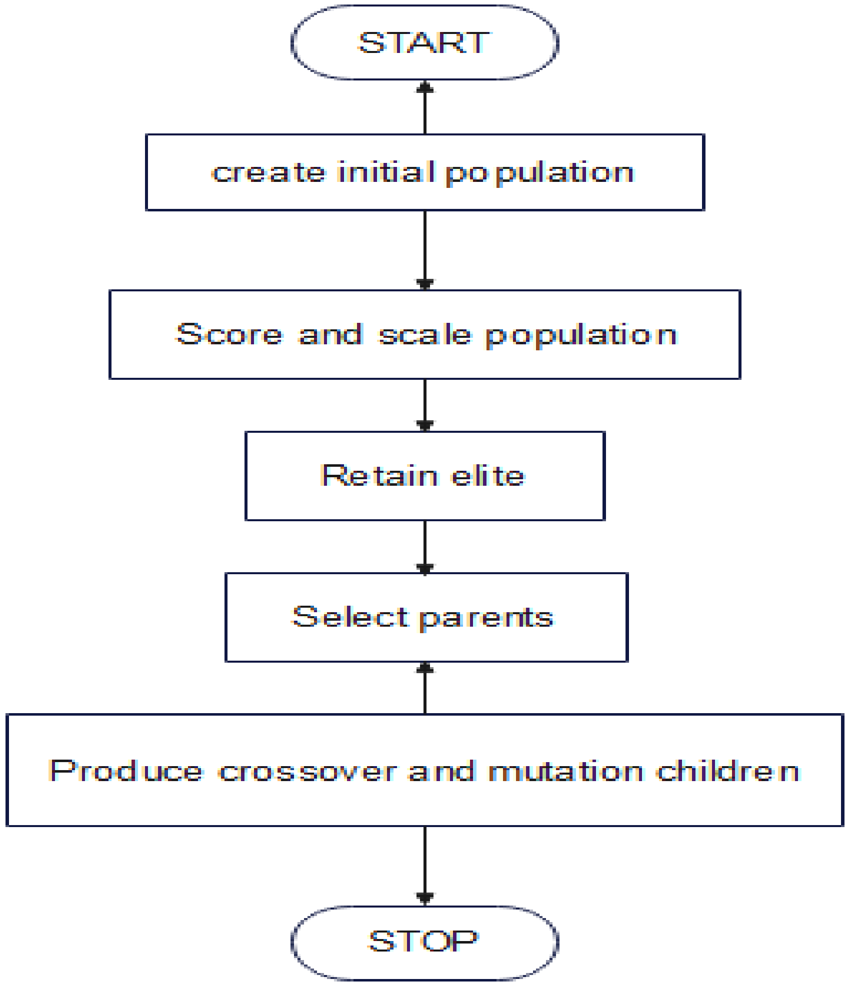


Fig 4.4 GA Flowchart

The operation of the genetic algorithm is described in the overview that follows.

1. The method generates an initial population at random.
2. The algorithm develops a set of fresh populations. The programme builds the subsequent population step by step using the individuals in the current generation. To produce a new population, the algorithm carries out the subsequent actions.
   - 2.1 Give each person in the present population a score based on their level of fitness. Raw fitness values are what these numbers stand for.
   - 2.2 Reduce the raw fitness values to a more practical range. Expected values are these scaled values.

2.3 Based on your expectations, select a member called Parent.

2.4 Some of the population's less fit members are chosen to be elites. The following population inherits these outstanding individuals.

2.5 Give birth to a kid from a parent. Children are produced by randomly changing a single parent (mutation) or by mixing the vector entries of two parents (crossover).

2.6 In order to create the next generation, replace the present population with children.

3. The algorithm terminates once one of the halting criteria is satisfied.
4. The algorithm performs modified steps for linear and integer constraints.
5. Nonlinear constraints are added to this algorithm.

The given GA toolbox had been utilized for determination of the gain values of the controllers. With 10 generations, an objective function has been established where the error is made to be minimized.

4.4.2 PARTICLE SWARM OPTIMIZATION

One of the biologically inspired algorithms that is adept at finding the optimum answer in the available solution space is particle swarm optimization (PSO). It differs from other optimization methods in that it simply needs an objective function and is independent of the objective function's gradient or differential form. There aren't many hyper parameters either.

Kennedy and Eberhart introduced the concept of particle swarm optimization in 1995. According to the original publication, sociobiologists think that a school of fish or a flock of birds that moves together "can benefit from the experience of all other members."

We can mimic a flock of birds flying around, but each one is made to discover the best answer in a high-dimensional answer space, and the flock's best answer is the one you can also assume is the greatest answer. A heuristic approach would be this. Because it is usually not possible to demonstrate the existence of a true global optimum. The PSO solution is frequently extremely near to the overall optimum.

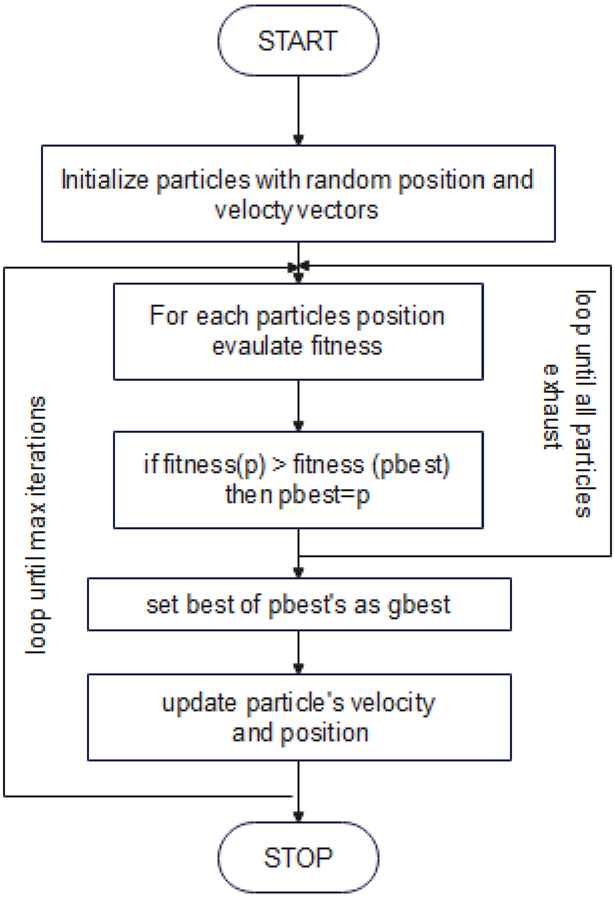


Fig 4.5 PSO Flowchart

The given algorithm essentially describes all PSO algorithms. The PSO was configured to run with a fixed number of iterations in the aforementioned case. Setting the amount of iterations to execute based on progress is simple. As soon as you cease seeing updates to Global Best Solutions, for instance, you can quit. Repeat a number of times. The determination of hyper parameters has been the primary focus of PSO research. c1, c2, and w. Or alter their values while the algorithm is running. For instance, it has been suggested to linearly reduce inertial weight. A recommendation to increase the cognitive coefficient is also present as the social coefficient declines. gradually increases the amount of exploration at the start and the amount of exploitation at the conclusion.

In particle swarm optimization:

$$v_i^{t+1} = v_i^t + \emptyset_1 U_1^t (pb_i^t - x_i^t) + \emptyset_2 U_2^t (gbt^t - x_i^t) \tag{4.4}$$

In Modified particle swarm optimization:

$$v_i^{t+1} = wv_i^t + \emptyset_1 U_1^t (pb_i^t - x_i^t) + \emptyset_2 U_2^t (gbt^t - x_i^t) \tag{4.5}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{4.6}$$

While going through iteration process the value of inertia weight updating by

$$w = w_{max} + \left[\frac{(w_{max} - w_{min})}{iter_{max}}\right] * iter \tag{4.7}$$

4.4.3 BELUGA WHALE OPTIMIZATION

BWO draws its inspiration from the beluga whales' swimming, hunting, and whale-falling behaviors. The beluga whale's scientific name is Delphinapterus leucas and it is a species of whale which lives in the sea. It is renowned for its adults' snow-white color and has also earned the nickname "canary of the sea" due to the variety of sounds it makes. A beluga whale has a medium-sized, spherical, stocky body that ranges in length from 3.5 to 5.5 m and weighs roughly 1500 kg. The ability to hear and see clearly allows belugas to maneuver and hunt by sound. Beluga whales are primarily found in the Arctic and subarctic oceans, including northwest Canada, Alaska, and the waters off Ellesmere Island.

The belugas are omnivorous, eating a variety of foods like prawns, worms, codfish, salmon and trout. When summer arrives, a large number of animals congregate in some estuaries, causing whales to congregate and feed. Due to their blunt teeth, beluga whales typically suction their prey into their mouths. Beluga whales, sometimes, with coordinating groups attack and then feed on the fishes by steering the fishes into shallow water. In addition, because of the high population density of beluga whales in estuaries during the summer, they are threatened by killer whales, polar bears, and humans. During migration, a few whales may perish and fall into the sea with great depth, a phenomenon known as "whale fall" , providing plenty of food for numerous creatures that lack sunlight and oxygen.

4.4.3.1 MATHEMATICAL MODEL OF BWO

The BWO algorithm imitates beluga whale behaviours like swimming, hunting, and whale falls. BWO has an exploration phase and an exploitation phase, just like other metaheuristics. The random selectivity of the beluga during the exploration phase ensures the ability to search whales globally in the design space and the local search within the design space is controlled by exploitation phase. Additionally, BWO takes into account the possibility of a whale falling, which alters the beluga whales' positions.

Beluga whales are thought of as the search agents because BWO process is population based, where each beluga whale behaves as a candidate solution that is being updated during optimization. The search agent positions matrix is modelled as:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \tag{4.8}$$

Where n : population size

d: dimension of beluga whales

The corresponding fitness values for every beluga whale are kept as mentioned:

$$F_X = \begin{bmatrix} f\ (x_{1,1}\ ,\ \ x_{1,2}, \dots x_{1,d}) \\ f\ (x_{2,1}\ ,\ \ x_{2,2}, \dots x_{2,d}) \\ \vdots \\ f\ (x_{n,1}\ ,\ \ x_{n,2}, \dots x_{n,d}) \end{bmatrix} \tag{4.9}$$

The balancing factor Bf is modelled as

$$B_f \ = \ B_0\ (1 - T/2T_{max}) \tag{4.10}$$

Where T: current iteration

T_{max}: Max Iterative number

B_0: Random number between 0 and 1

It determines if the BWO algorithm shifts from exploration phase to exploitation phase. B_0 fluctuates arbitrarily between the values of (0,1) during each iteration.

When Bf > 0.5, the exploration phase begins, and when Bf 0.5, the exploitation phase begins. The fluctuation range of Bf decreases from (0, 1) to (0, 0.5) as iteration T rises, showing the significant shift in the probabilities of the exploitation and the exploration phases where the exploitation phase probability rises as iteration T rises.

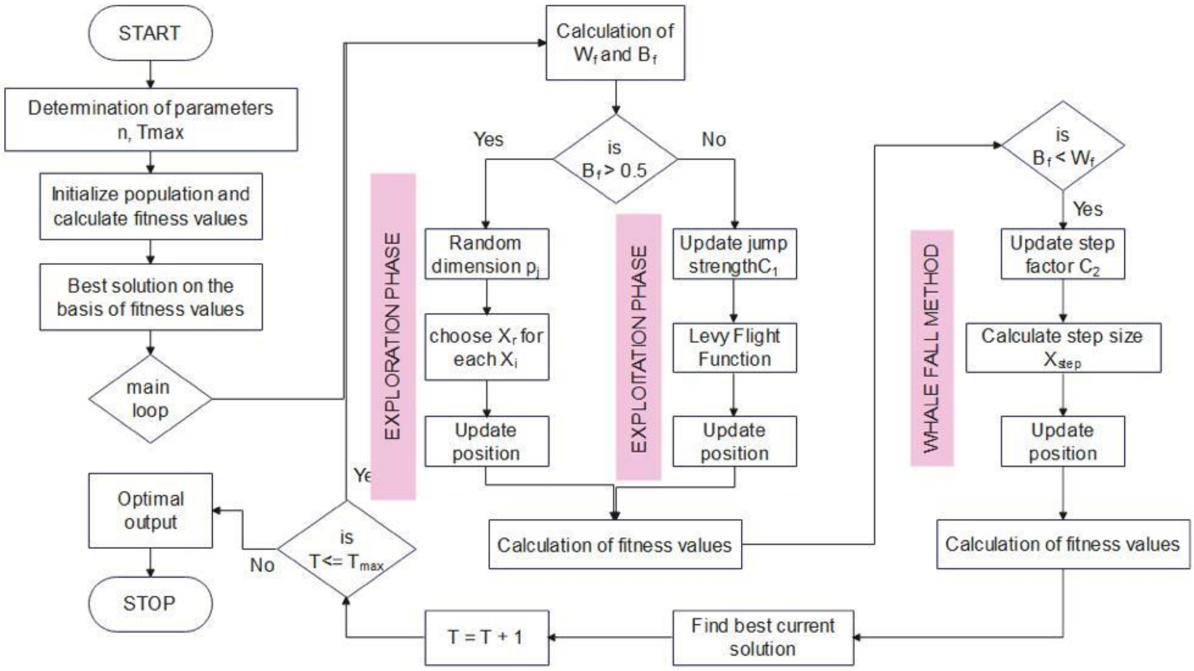


Fig 4.6 WBO Flowchart

4.4.3.2 EXPLORATION PHASE

Beluga whale's swim behavior is taken into consideration during establishment of the BWO exploration phase. The Beluga whales can perform social behaviours in a variety of postures, pair of two beluga whales swimming closely together in a synchronized or mirrored manner. This behaviour has been observed in beluga whales which were kept in human care. The beluga whale pair swim serves as a guide for search agents' positions, and the positions for beluga whales need to be updated as follows

$$\begin{cases} X_{ij}^{T+1} = X_{i,}^{T} + \left(X_{r,p_1}^{T} - X_{i,p_j}^{T} \right) (1 + r_1)\, sin(2\pi r_2)\,, j = even \\ X_{ij}^{T+1} = X_{i,p_j}^{T} + \left(X_{r,p_1}^{T} - X_{i,p_j}^{T} \right) (1 + r_1)\, cos(2\pi r_2)\,, j = odd \end{cases} \quad (4.11)$$

Where T: current iteration

X_{ij}^{T+1}: new position of ith beluga whale on the jth dimension

p_j: Random number from d dimension

X_{i,p_j}^{T}: position for i^{th} beluga whale on p_j dimension

r_1 *and* r_2: Randomly selected numbers between 0 and 1

$sin(2\pi r_2)$ *and* $cos(2\pi r_2)$: fins of mirrored beluga whales which are toward the surface

4.4.3.3 EXPLOITATION PHASE

The beluga whale's hunting strategy served as inspiration for the BWO exploitation phase. Beluga whales can forage cooperatively and move in accordance with the location of neighbouring beluga whales.

Thus, beluga whales hunt after exchanging information about job openings among themselves and selecting the best applicant among them. For improving the convergence, the Levy flight technique is added during exploitative stage of BWO. We believed that they could catch the prey with the Levy flight method. The mathematical model is given below:

$$X_i^{T+1} = r_3 X_{best}^T - r_4 X_i^T + C_1 C_F(X_r^T - X_i^T) \tag{4.12}$$

Where T: current iteration

X_r^T *and* X_i^T: current position of i[th] beluga whale and one random beluga whale

X_i^{T+1}: new position of ith beluga whale

X_{best}^T: best position amongst the beluga whales,

C_1: random jump strength

r_3 *and* r_4: Randomly selected numbers between 0 and 1

L_f is Levy flight function given as

$$L_F = 0.05 * \frac{u * \sigma}{|v|^{\frac{1}{\beta}}} \tag{4.13}$$

And σ is given by

$$\sigma = \left(\frac{\Gamma(1+\beta) * \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma((1+\beta)/2) * \beta * 2^{\frac{\beta-1}{2}}} \right)^{1/\beta} \tag{4.14}$$

Where u and v are normally distributed random numbers and β is 1.5

4.4.3.4 WHALE FALL

These beluga whales face threats from the killer whales and polar bears and also, the people while migrating and foraging. Since most beluga whales are intelligent, they can avoid danger by exchanging information with one another. However, a small percentage of beluga whales perished and fell to the deep sea floor. Many different creatures are fed by the phenomenon known as "whale fall." The carcass of a dead whale attracts a large group of sharks and invertebrates, and exposed bones and the bodies of the deceased whales also draw a large group of hair crustaceans.
The skeleton then undergoes decades of decomposition or is inhabited by bacteria and corals.

The expression for the mathematical model is:

$$X_i^{T+1} = r_5 X_i^T - r_6 X_r^T + r_7 X_{step} \tag{4.15}$$

Where r_5 , r_6 and r_7: Randomly selected numbers between 0 and 1

Step size of whale fall is given by:

$$X_{step} = (u_b - l_b) \exp(-C_2 T / T_{max}) \tag{4.16}$$

Where, C_2: step factor related with probability of whale fall
u_b and l_b: Upper and lower boundary variables

In WBO, probability for whale fall w_f, is a linear function

$$w_f = 0.1 - \frac{0.05T}{T_{max}} \tag{4.17}$$

4.4.4 COOT ALGORITHM

A brand-new heuristic optimization technique called the Coot algorithm (CA) was developed by Naruei and Keynia initially. The optimization process is put into practice by accounting for the four movements of the coots on the surface of the water.

Random movement

To carry out this movement, a random coot location is originally produced in the search space using equation 4.18.

$$Q = rand(1, n).(ub - lb) + lb \quad (4.18)$$

Where, n is dimension’s number, ub and lb are upper and lower limits

In the area of the search, the coot moves around. This movement helps the algorithm avoid being trapped in the local optima. The new position of the coot is given by equation 4.19.

$$CootPos(i) = CootPos(i) + A * r2 * \left(Q - CootPos(i)\right) \quad (4.19)$$

Where A is calculated using

$$A = 1 - L * \left(\frac{1}{Iter}\right) \quad (4.20)$$

Where, iter is current iteration

Chain movement

By calculating the average position of two coots, the chain movement is determined. The new location of the coot is given by equation 4.21.

$$CootPos(i) = 0.5 * \left(CootPos(i-1) + CootPos(i)\right) \quad (4.21)$$

Position adjustment based on the group leaders

The remaining coots modify their positions proportionately to that of the group leaders once they have determined their average position. The leader is chosen with equation 4.22.

$$K = 1 + (iMOD\ NL) \qquad (4.22)$$

Where, k is index number of current coot, L is no. of leaders

The next location of the coot is given by equation 4.23.

$$CootPos(i) = LeaderPos(k) + 2 * R1 * \cos(2\pi r) * LeaderPos(k) - CootPos(i)) \qquad (4.23)$$

Where R is random numbers between [-1, 1]

Leader Movement

For finding the optimal location, the leader's position is updated by equation 4.24.

$$LeaderPos(i) = \begin{cases} B * R3 * \cos(2\pi r) * \left(gBest - LeaderPos(i)\right) + gBest & R4 < 0.5 \\ B * R3 * \cos(2\pi r) * \left(gBest - LeaderPos(i)\right) - gBest & R4 \geq 0.5 \end{cases} \qquad (4.24)$$

Where B is best global best solution and B is given by equation 4.25

$$B = 2 - L * \left(\frac{1}{Iter}\right) \qquad (4.25)$$

The flowchart for the coot algorithm is given below

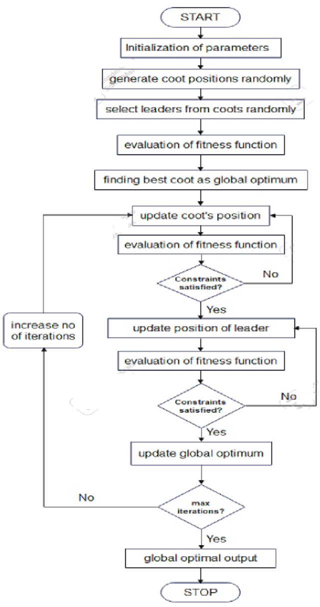


Fig 4.7 COOT Flowchart

CHAPTER 5

SIMULATION RESULT AND DISCUSSION

5.1 PD GRAPH SIMULATION

Figure 5.1 shows PD controller with Genetic Algorithm (GA) Partical swarm optimization (PSO) and COOT Algorithm as tuning methods for K_p and K_d.

TABLE 5.1: PD GAIN VALUES

Controller Type	K_P	K_D
PSO	30	0.484515
GA	29.98645	0.494397
COOT	30	0.525

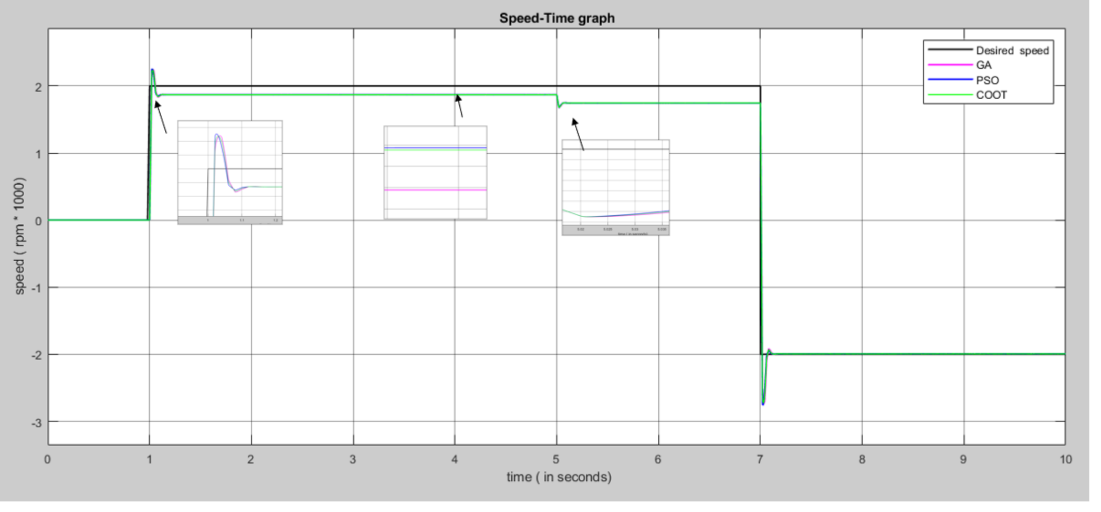


Fig 5.1: Output response of PD controller for speed control of DC Motor

TABLE 5.2: PD STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITHOUT LOAD

	Max Overshoot	rise time	settling time	Steady state error
PSO	11.9%	1.016s	-----	6.55%
GA	12.9%	1.017s	-----	6.55%
COOT	12.8%	1.012s	-----	6.55%

TABLE 5.3: PD STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITH LOAD

	Max Overshoot	settling time	Steady state error
PSO	16.2%	----	12.96%
GA	16.15%	----	12.8%
COOT	15.8%	----	12.75%

Table 5.1 shows PD controller gain values. Table 5.5 and 5.6 shows input step response parameter response without and with load respectively. From the table we observe that COOT algorithm for tuning of controller gives superior results as compared to PSO and GA with minimum overshoot and faster rise.

5.2 PI GRAPH SIMULATION

Figure 5.2 shows PI controller with Genetic Algorithm (GA) Partical swarm optimization (PSO) and COOT Algorithm as tuning methods for K_p and K_I.

TABLE 5.4: PI GAIN VALUES

Controller Type	K_P	K_I
PSO	21.16896	10
GA	20.37548	9.972355
COOT	21.1232	10

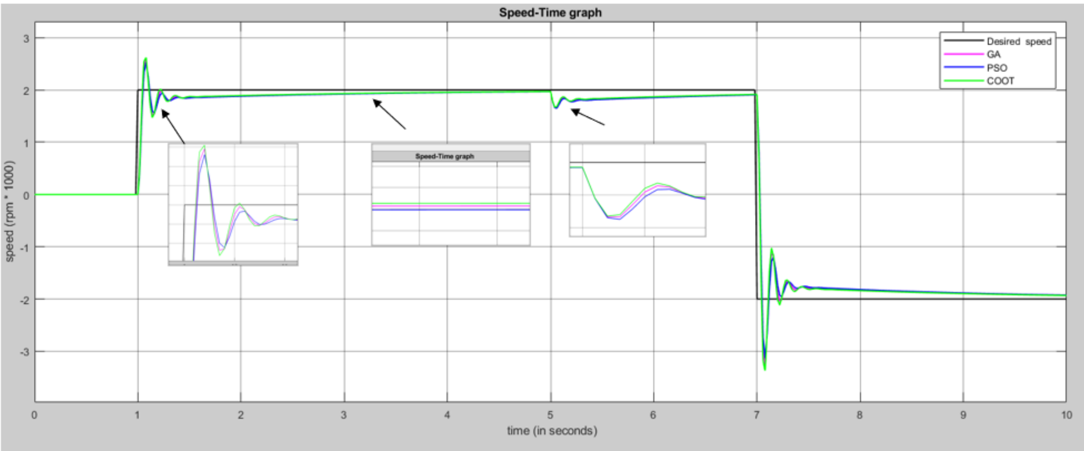


Fig 5.2: Output response of PI controller for speed control of DC Motor

TABLE 5.5: PI STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITHOUT LOAD

	Max Overshoot	rise time	settling time (5%)	Steady state error
PSO	30.25%	1.041s	2.212	1.5%
GA	28.7%	1.041s	2.186s	1.45%
COOT	31.26%	1.04s	2.174s	1.425%

TABLE 5.6: PI STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITH LOAD

	Max Overshoot	settling time (5%)	Steady state error
PSO	16.2%	1.62s	4.4%
GA	16.45%	1.599s	4.35%
COOT	16.15%	1.57s	4.25%

Table 5.4 shows PI controller gain values. Table 5.5 and 5.6 shows input step response parameter response without and with load respectively. From the table we observe that COOT algorithm for tuning of controller gives superior results as compared to PSO and GA with maximum overshoot and faster rise and settling time.

5.3 PID GRAPH SIMULATION

Figure 5.2 shows PI controller with Genetic Algorithm (GA) Partical swarm optimization (PSO), Beluga Whale Optimization (BWO) and COOT Algorithm as tuning methods for K_p, K_I and K_D.

TABLE 5.7: PID GAIN VALUES

Controller Type	K_P	K_I	K_D
GA	28.46334	19.7116	0.49541
PSO	28.2655	20	0.481305
COOT	30	20	0.4943
WBA	27.5883	19.3413	0.519167

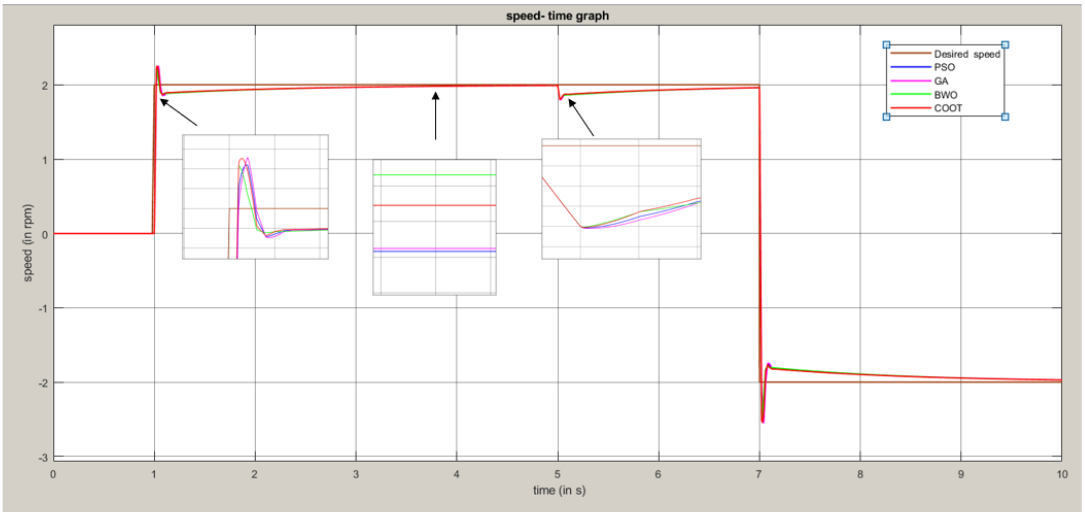


Fig 5.3: Output response of PI controller for speed control of DC Motor

TABLE 5.8: PID STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITHOUT LOAD

	Max Overshoot	rise time	settling time (5%)	Steady state error
PSO	9.35%	1.0189s	1.355s	0.5%
GA	9.8%	1.0178s	1.355s	0.45%
COOT	9.74%	1.0128s	1.240s	0.45%
WBO	8.7%	1.0175s	1.345s	0.45%

TABLE 5.9: PID STEP-INPUT RESPONSE PARAMETER OF DC MOTOR WITH LOAD

	Max Overshoot	settling time (5%)	Steady state error
PSO	10.1%	0.56s	2%
GA	10.05%	0.566s	2%
COOT	10.05%	0.47s	1.95%
WBO	10%	0.55s	1.955%

Table 5.7 shows PID controller gain values. Table 5.8 and 5.9 shows input step response parameter response without and with load respectively. From the table we observe that COOT algorithm for tuning of controller gives good results as compared to PSO and GA with maximum overshoot and faster rise and settling time but WBO gives maximum overshoot. The steady state error is almost similar in COOT and BWO algorithms. Settling time is faster in COOT as compared to BWO.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

From the above simulation results we can confirm that if we use PID with COOT algorithm for tuning of controller parameters for speed control of DC motor we get better better rise time (1.0128s) and settling time (1.240s) as compared to PID with BWO rise time (1.0175s), settling time (1.345s).

The Maximum Overshoot for PID controller with BWO technique (8.7% with load and with load 10%) is the best we got from all the controllers with different optimization technique.

When PID controller is tunes with COOT or BWO algorithms we get the minimum steady state error.

Overall we can say that if we want better rise time and settling time we can use PID controller with COOT algorithm for tuning its parameter for speed control of DC motor but if we want minimum overshoot from the controller then PID controller with BWO optimization technique is the best option.

6.2 FUTURE SCOPE

The future scope for tuning techniques of the controllers are bright with the ongoing developments in control theory and technology improvement. In order to maximise the effectiveness and performance especially in speed control of DC Motor, tuning techniques are necessary.

One way to improve the proposed tuning algorithms for more robustness and effectiveness of the controller will be combining with Model Predictive Control algorithms. As Metaheuristic Algorithms are very good with global optimization problems with complex solution spaces and MPC on the other hand gives accurate and real time control by taking system dynamics and constraints into solution. Combination of both will be very beneficial for future controllers.

Moreover, we can use AI algorithm with Metaheuristic Algorithms for tuning of controllers as we know AI learns from data rich environment and can learn from historical data and initial parameter settings of controller and the metaheuristic algorithms can be used to fine tune the parameters and get the controller parameters more efficiently to make the controller more stable and accurate

Also, the proposed algorithms can be combined with advance sensing and feedback techniques as metaheuristic algorithms has the capabilities of global searching the solution space and handling complex optimization problems and advance sensing and feedback techniques can offer real time monitoring and accurate system understanding.

In summary, combination of all these techniques can make the controller more robust, precise and the controller can achieve desired control performance as required.

REFERENCES

[1] T. J. Maloney and F. L. Alvarado, "A Digital Method for DC Motor Speed Control," in IEEE Transactions on Industrial Electronics and Control Instrumentation, vol. IECI-23, no. 1, pp. 44-46, Feb. 1976, doi: 10.1109/TIECI.1976.351346.

[2] S. Weerasooriya and M. A. El-Sharkawi, "Laboratory implementation of a neural network trajectory controller for a DC motor," in IEEE Transactions on Energy Conversion, vol. 8, no. 1, pp. 107-113, March 1993, doi: 10.1109/60.207413.

[3] R. S. Ahmed, K. S. Rattan and I. H. Khalifa, "Real-time tracking control of a DC motor using a neural network," Proceedings of the IEEE 1995 National Aerospace and Electronics Conference. NAECON 1995, Dayton, OH, USA, 1995, pp. 593-600 vol.2, doi: 10.1109/NAECON.1995.521998.

[4] Z. M. Al-Hamouz and H. N. Al-Duwaish, "A new variable structure DC motor controller using genetic algorithms," Conference Record of 1998 IEEE Industry Applications Conference. Thirty-Third IAS Annual Meeting (Cat. No.98CH36242), St. Louis, MO, USA, 1998, pp. 1669-1673 vol.3, doi: 10.1109/IAS.1998.729769.

[5] Y. Tipsuwan and Mo-Yuen Chow, "Fuzzy logic microcontroller implementation for DC motor speed control," IECON'99. Conference Proceedings. 25th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.99CH37029), San Jose, CA, USA, 1999, pp. 1271-1276 vol.3, doi: 10.1109/IECON.1999.819394.

[6] K. Sundareswaran and M. Vasu, "Genetic tuning of PI controller for speed control of DC motor drive," Proceedings of IEEE International Conference on Industrial Technology 2000 (IEEE Cat. No.00TH8482), Goa, India, 2000, pp. 521-525 vol.2, doi: 10.1109/ICIT.2000.854212.

[7] S. Aydemir, S. Sezen and H. M. Ertunc, "Fuzzy logic speed control of a DC motor," *The 4th International Power Electronics and Motion Control Conference, 2004. IPEMC 2004.*, Xi'an, China, 2004, pp. 766-771 Vol.2.

[8] S. g. Kadwane, S. P. Vepa, B. M. Karan and T. Ghose, "Converter Based DC Motor Speed Control Using TMS320LF2407A DSK," 2006 1ST IEEE Conference on Industrial Electronics and Applications, Singapore, 2006, pp. 1-5, doi: 10.1109/ICIEA.2006.257181.

[9] A. S. El Din Zein El Din, "PLC-Based Speed Control of DC Motor," 2006 CES/IEEE 5th International Power Electronics and Motion Control Conference, Shanghai, China, 2006, pp. 1-6, doi: 10.1109/IPEMC.2006.4778111.

[10] A. A. A. El-Gammal and A. A. El-Samahy, "Adaptive Tuning of a PID Speed Controller for DC Motor Drives Using Multi-objective Particle Swarm Optimization," 2009 11th International Conference on Computer Modelling and Simulation, Cambridge, UK, 2009, pp. 398-404, doi: 10.1109/UKSIM.2009.60.

[11] M. -H. Chu, Y. -W. Chen, Y. Kang and Z. -W. Chen, "The research of sampling frequency for A DC servo motor speed control system based on neural networks," 2010 Sixth International Conference on Natural Computation, Yantai, China, 2010, pp. 1401-1405, doi: 10.1109/ICNC.2010.5582887.

[12] F. Valdez, P. Melin and O. Castillo, "Particle Swarm Optimization for designing an optimal fuzzy logic controller of a DC motor," 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), Berkeley, CA, USA, 2012, pp. 1-6, doi: 10.1109/NAFIPS.2012.6291051.

[13] I. G. A. P. R. Agung, S. Huda and I. W. A. Wijaya, "Speed control for DC motor with pulse width modulation (PWM) method using infrared remote control based on ATmega16 microcontroller," 2014 International Conference on Smart Green Technology in Electrical and Information Systems (ICSGTEIS), Kuta, Bali, Indonesia, 2014, pp. 108-112, doi: 10.1109/ICSGTEIS.2014.7038740.

[14] V. Dutta, S. Borkakati, T. O'Donnell and D. Bora, "PI-Fuzzy rule based controller for Analysis and performance evaluation of dc motor speed control," The 2nd IEEE Conference on Power Engineering and Renewable Energy (ICPERE) 2014, Bali, Indonesia, 2014, pp. 175-180, doi: 10.1109/ICPERE.2014.7067207.

[15] H. R. Jayetileke, W. R. de Mei and H. U. W. Ratnayake, "Real-time fuzzy logic speed tracking controller for a DC motor using Arduino Due," 7th International Conference on Information and Automation for Sustainability, Colombo, Sri Lanka, 2014, pp. 1-6, doi: 10.1109/ICIAFS.2014.7069560.

[16] J. R. B. A. Monteiro, C. M. R. Oliveira and M. L. Aguiar, "Sliding mode control of brushless DC motor speed with chattering reduction," 2015 IEEE 24th International Symposium on Industrial Electronics (ISIE), Buzios, Brazil, 2015, pp. 542-547, doi: 10.1109/ISIE.2015.7281525.

[17] C. A. Ganzaroli et al., "Heuristic and deterministic strategies applied on cascade PI controller tuning for speed control of a DC motor," 2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Santiago, Chile, 2015, pp. 101-106, doi: 10.1109/Chilecon.2015.7400360.

[18] G. Dewantoro, "Robust fine-tuned PID controller using Taguchi method for regulating DC motor speed," 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE), Chiang Mai, Thailand, 2015, pp. 173-178, doi: 10.1109/ICITEED.2015.7408936.

[19] I. Jaziri, L. Chaarabi and K. Jelassi, "A remote DC motor control using Embedded Linux and FPGA," 2015 7th International Conference on Modelling, Identification and Control (ICMIC), Sousse, Tunisia, 2015, pp. 1-5, doi: 10.1109/ICMIC.2015.7409332.

[20] M. R. C. Reis et al., "Speed control for direct current motor using optimization tuning for PID controller," 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 2016, pp. 1-4, doi: 10.1109/EEEIC.2016.7555596.

[21] X. Qiao, F. Luo and Y. Xu, "Robust PID controller design using genetic algorithm for wastewater treatment process," 2016 IEEE Advanced Information Management, Communicates, Electronic and AutomationControl Conference (IMCEC), Xi'an, pp. 1081-1086, 2016.

[22] E. Mirzaei and H. Mojallali, "Auto tuning PID controller using chaotic PSO algorithm for a Boost converter," 2013 13th Iranian Conference on Fuzzy Systems (IFSC), Qazvin, pp. 1-6, 2013.

[23] Y. Liu, C. Zhou and W. Xiang, "DE Algorithm Fuzzy Control of Super-Heated Steam Temperature," 2016 9th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, pp. 282-285, 2016.

[24] T. K. Pati, J. R. Nayak, B. K. Sahu and S. K. Kar, "Automatic generation control of multi-area thermal power system using TLBO algorithm optimized fuzzy-PID controller," International Conference on Circuits, Power and Computing Technologies (ICCPCT-2015), Nagercoil, pp. 1-6, 2015.

[25] R. V. Jain, M. V. Aware and A. S. Junghare, "Tuning of Fractional Order PID controller using particle swarm optimization technique for DC motor speed control," 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, pp. 1-4, 2016.

[26] K. R. Das, D. Das and J. Das, "Optimal tuning of PID controller using GWO algorithm for speed control in DC motor," 2015 International Conference on Soft Computing Techniques and Implementations
(ICSCTI), Faridabad, pp. 108-112, 2015.

[27] R. K. Achanta and V. K. Pamula, "DC motor speed control using PID controller tuned by jaya optimization algorithm," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India, 2017, pp. 983-987, doi: 10.1109/ICPCSI.2017.8391856.

[28] S. Ekinci, D. Izci and B. Hekimoğlu, "PID Speed Control of DC Motor Using Harris Hawks Optimization Algorithm," 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Istanbul, Turkey, 2020, pp. 1-6, doi: 10.1109/ICECCE49384.2020.9179308.

[29] K. R. Das, D. Das and J. Das, "Optimal tuning of PID controller using GWO algorithm for speed control in DC motor," 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), Faridabad, India, 2015, pp. 108-112, doi: 10.1109/ICSCTI.2015.7489575.

[30] S. K. Sahoo, R. Sultana and M. Rout, "Speed control of DC motor using Modulus Hugging Approach," International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2011), Chennai, 2011, pp. 523-528, doi: 10.1049/cp.2011.0418.

[31] B. Hekimoğlu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," in IEEE Access, vol. 7, pp. 38100-38114, 2019, doi: 10.1109/ACCESS.2019.2905961.

[32] A. Sant and K. R. Rajagopal, “PM synchronous motor speed control using hybrid fuzzy-PI with novel switching functions,” IEEE Trans. Magn., vol. 45, no. 10, pp. 4672–4675, Oct. 2009.

[33] B. M. Hohan and A. Sinha, “Analytical structure and stability analysis of a fuzzy PID controller,” Appl. Soft Comput., vol. 8, no. 1, pp. 749–758, Jan. 2008.

[34] A. Rubaai, M. J. Castro-Sitiriche, and A. R. Ofoli, “DSP-based laboratory implementation of hybrid fuzzy-PID controller using genetic optimization for high performance motor drives,” IEEE Trans. Ind. Appl., vol. 44, no. 6, pp. 1977–1986, Nov./Dec. 2008.

[35] A. Rubaai and P. Young, "EKF-Based PI-/PD-Like Fuzzy-Neural-Network Controller for Brushless Drives," in IEEE Transactions on Industry Applications, vol. 47, no. 6, pp. 2391-2401, Nov.-Dec. 2011, doi: 10.1109/TIA.2011.2168799.

[36] M. Meenakshi, "Microprocessor based digital pid controller for speed control of d.c. motor," in Emerging Trends in Engineering and Technolnogy, 2008. ICET ET '08. First International Conference on, 2008, pp.
960-965.

[37] Y. S. Ettomi. S. Moor. S. Bashi. and M. K. Hassan. "Micro controller based adjustable closed-loop dc motor speed controller," in Research and Development, 2003. SCORED 2003. Proceedings. Student Conference on. 2003. pp. 59-63.

[38] K. V.Subasri and B. Umamaheswari, "Implementation of digital pid controller in field programmable gate array (fpga)." Proceedings of India International Conference on Power Electronics. 2006.

[39] W. M. Elsrogy, M. A. Fkirin and M. A. M. Hassan, "Speed control of DC motor using PID controller based on artificial intelligence techniques," 2013 International Conference on Control, Decision and Information Technologies (CoDIT), Hammamet, Tunisia, 2013, pp. 196-201, doi: 10.1109/CoDIT.2013.6689543.

[40] M. Zhang, C. Xia, Y. Tian, D. Liu and Z. Li, "Speed Control of Brushless DC Motor Based on Single Neuron PID and Wavelet Neural Network," 2007 IEEE International Conference on Control and Automation, Guangzhou, China, 2007, pp. 617-620, doi: 10.1109/ICCA.2007.4376429.

LIST OF PUBLICATIONS

S. No	PAPER TITLE	PUBLICATION	INDEX	AUTHORS	STATUS
1	PID Controller Tuning to Control Speed of DC Motor Using COOT Algorithm.	19th Asia Pacific Conference on Circuits and Systems, 2023.	SCOPUS	Shubham Patnaik	Communicated