

COMPARATIVE ANALYSIS OF DC MOTOR SPEED CONTROL USING VARIOUS CONTROLLER TUNING METHODS

Thesis submitted in partial fulfillment of the

Requirement for the award of degree of

Master of Engineering

In

Electronics Instrumentation and Control



Submitted By

Mani Kant

Roll no: 801251012

Under the Guidance of

Dr. Gagandeep Kaur

Assistant Professor

Electrical and Instrumentation Engineering Department

Thapar University

(Established under the section 3 of UGC act, 1956)

Patiala, 147001, Punjab, India

July 2014

DECLARATION

I hereby declare that the thesis submitted on “**Comparative Analysis of DC Motor Speed Control using various Controller Tuning Methods**”, is an authentic record of my own work carried out as the requirement for the award of degree of Master of Engineering in EIED at Thapar University, Patiala, under the guidance of Dr. Gagandeep Kaur, Assistant Professor, EIED, Thapar University, Patiala.

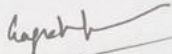
Date: 18/07/2014


Mani Kant


Roll No: 801251012


This is to certify that the above statement made by the student is correct and to the best of my knowledge and belief.

Date: 18/07/2014


Dr. Gagandeep Kaur
Assistant Professor, EIED
Thapar University, Patiala

Countersigned By


Dr. Ravinder Aggarwal
Head, EIED
Thapar University, Patiala
Punjab, India


Dr. S.K. Mohapatra
Dean of Academic Affairs
Thapar University, Patiala
Punjab, India

ACKNOWLEDGEMENT

First of all, I am very thankful to **Dr. Prakash Gopalan**, Director of Thapar University, Patiala for providing the facilities for the completion of Master of Engineering.

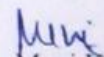
I express my deep sense of gratitude towards **Dr. Ravinder Aggarwal**, Head of the Department of Electrical & Instrumentation Engineering, Thapar University, Patiala who has been a constant source of inspiration for me throughout this work.

I am also thankful to our PG coordinator, **Dr. Mandeep Singh**, Associate Professor EIED, Thapar University for their full cooperation and help.

I would like to express my gratitude to **Dr. Gagandeep Kaur**, Assistant Professor EIED for her patient guidance and support throughout this report. I am truly very fortunate to have the opportunity to work with her. I found her guidance to be extremely valuable.

I would like thank to entire faculty and staff of EIED, Thapar University and then friends and parents who devoted their valuable time and help me in all possible ways towards Successful completion of this work. I thank all those who have contributed directly or indirectly to this work.

Date: 18/07/2014


Mani Kant

ABSTRACT

Motors have speed dynamics when mechanical parameters like inertia and damping and electrical parameters like resistance and inductance are taken into account. The objective of the controllers is to maintain the speed of rotation of motor with a particular step response with minimizing the properties. The performance of a dc motor controlled by a proportional-integral-derivative (PID) controller is analyzed. Overshoot appears with large settling time thereby confirming the behavior of a PID controller. So it is necessary to tune the PID controller in order to obtain the desired performance. Fuzzy logic based controller is used along with dc motor. Using appropriate expert rules, overshoot is reduced to minimum in this particular case on the cost of increased settling time and rise time. Manual tuning is eliminated and intelligent tuning take over the control with satisfactory performance using fuzzy logic controller. Technique based on PSO, particle swarm optimization is analyzed and simulation is done to control the PID controller and better results are obtained. The PID controller is designed using PSO algorithm to generate set of coefficients associated with controller. Program is used to activate the controller, calculate and plot the time response of control system. An improvement in PID controller is seen by using particle swarm optimization over conventional PID and fuzzy controllers.

CONTENTS

| | | |
|--|---|-------|
| I. | DECLARATION | |
| II. | ACKNOWLEDGMENT | |
| III. | ABSTRACT | |
| IV. | LIST OF FIGURES | |
| V. | LIST OF TABLES | |
| VI. | LIST OF ABBREVIATIONS | |
| CHAPTER 1 INTRODUCTION | | 1-3 |
| 1.1 | Overview | 1 |
| 1.2 | Organization of Thesis..... | 3 |
| CHAPTER 2 LITERATURE REVIEW | | 4-7 |
| CHAPTER 3 DC MOTOR MODELLING..... | | 8-13 |
| 3.1 | DC Motor..... | 8 |
| 3.2 | Principle of Operation..... | 9 |
| 3.3 | DC Motor Equation..... | 9 |
| 3.4 | DC Motor Transfer Function..... | 12 |
| CHAPTER 4 INTRODUCTION TO CONTROLLERS..... | | 14-18 |
| 4.1 | PID Controller..... | 14 |
| 4.2 | Fuzzy Controller..... | 15 |
| 4.3 | Particle Swarm Optimization based controller..... | 16 |
| 4.3.1 | <i>Algorithm</i> | 16 |
| 4.3.2 | <i>Parameter selection</i> | 18 |
| 4.3.3 | <i>Convergence</i> | 18 |

| | |
|---|-------|
| CHAPTER 5 DC Motor Speed Control: Case Study..... | 19-29 |
| 5.1 DC Motor Model..... | 19 |
| 5.2 PID tuning..... | 22 |
| 5.3 Fuzzy Tuning..... | 23 |
| 5.4 PSO Algorithm | 26 |
| 5.5 Comparison | 29 |
| CHAPTER 6 RESULTS AND DSCUSSION..... | 30-33 |
| 6.1 PID and Fuzzy Simulation Results..... | 30 |
| 6.2 PSO Simulation Results..... | 32 |
| CHAPTER 7 CONCLUSION AND FUTURE SCOPE..... | 34 |
| CHAPTER 8 CHECK FOR ORIGINALITY..... | 35 |
| REFERENCES..... | 36-40 |

LIST OF FIGURES

| | | |
|------------|---|----|
| FIGURE 3.1 | Schematic representation of DC motor..... | 9 |
| FIGURE 3.2 | (a) Dependence of torque on electric current..... | 10 |
| | (b) Dependence of back emf on angular speed..... | 11 |
| FIGURE 3.3 | Armature controlled DC motor with load..... | 12 |
| FIGURE 3.4 | Block diagram of DC motor system..... | 13 |
| FIGURE 4.1 | Block diagram of generalized PID controller..... | 14 |
| FIGURE 4.2 | Block diagram of fuzzy controller..... | 15 |
| FIGURE 5.1 | DC motor model..... | 19 |
| FIGURE 5.2 | Matlab/simulink model for PID controller..... | 22 |
| FIGURE 5.3 | 3D behavior of controller..... | 25 |
| FIGURE 5.4 | Matlab/simulink model for fuzzy controller..... | 25 |
| FIGURE 5.5 | Flow chart of PSO algorithm..... | 27 |
| FIGURE 5.6 | Matlab/simulink presentation of PSO..... | 28 |
| FIGURE 5.7 | Matlab/simulink model for PID, Fuzzy and PSO..... | 29 |
| FIGURE 6.1 | Step response of PID and Fuzzy controller..... | 31 |

LIST OF TABLES

| | | |
|-----------|--|----|
| TABLE 5.1 | Parameters and value of DC motor..... | 19 |
| TABLE 5.2 | Rules for Fuzzy Controller..... | 24 |
| TABLE 6.1 | Parameters of step response of PID and Fuzzy Controller..... | 30 |
| TABLE 6.2 | Values of parameters after simulation of PSO algorithm..... | 31 |
| TABLE 6.3 | Parameters of step response of PSO..... | 32 |

LIST OF ABBREVIATIONS

| | |
|---------------|--|
| ACO | Ant Colony Optimization |
| DMOPSO | Discrete Multi-Objective Particle Swarm Optimization |
| FIS | Fuzzy Inference System |
| FLC | Fuzzy Logic Controller |
| GA | Genetical Algorithm |
| MF | Membership Functions |
| PID | Proportional integral derivative |
| PMDC | Permanent Magnet Direct Current |
| PSO | Particle Swarm Optimization |

CHAPTER 1

INTRODUCTION

1.1 Introduction

The control systems are the integral and major part of the modern society. They consist of subsystems and processes which are assembled for controlling purposes of many processes in modern industry. For domestic or industrial works motion control is required. The systems that are employed for such controls are called drives. In electric drives use of various sensors and control algorithm is done to control the speed of motor. The open loop systems consist of input transducers that convert the signal in desired electrical form used by controller. The controller further drives a plant or a process. The input is also called reference and output is a control variable. The disadvantages of open loop system are sensitive to disturbances and inability to correct for those disturbances. This may be removed by using closed loop systems. The closed loop systems overcome the problem for disturbances by measuring output response and feeding that through feedback system and comparing them at summing junctions. If there is any difference in the two responses, the system drives the plant to make a correction. If no such difference is found, it does not drive the plant since plant response is already a desired response. The close loop systems have major advantage of accurate control than open loop system. The effect of disturbances, noise and environmental factors can be made ineffective. Transient and steady state response can be improved with great flexibility by redesigning the controller. [10]

The DC motors are expensive due to brushes and commutators. DC motors have low torque to volume and torque to inertia ratios. On other hand the characteristics are quite linear and are easy to control. DC motors are generally used for high power application like in machine tools and robotics. It is interesting to know that almost half of industries use PID or modified PID schemes. Because PID can be adjusted on site, many types of tuning rules have been proposed. The mathematical model of plant suggests that PID control can give best results. In the field of control systems, it is well known that basic and modified PID systems have proves their usefulness in providing satisfactory response although in many situations they may not provide optimal control.

To achieve that optimal control intelligent fuzzy controllers can be used. Fuzzy logic control (FLC) is one of the successful applications of fuzzy set theory introduced by L.A. Zadeh in 1973. Since then FLC has been an extremely active and fruitful research area with many industrial applications. FLC has evolved as an alternative or complementary to various control strategies in area of engineering. Fuzzy logic provides non linear controllers which are capable of performing different complex non linear control actions, even for uncertain nonlinear systems. Unlike conventional controllers, for designing a FLC it is not necessary to have a precise knowledge of system model such as poles and zeros of the system transfer function. Imitating the human way of learning, the tracking error and rate of change of error of inputs are required for the design of fuzzy inference system. [56]

Modern heuristic optimization techniques such as genetical algorithm and neural networks annealing are much into the work by many researchers due to their ability to find an optimal accepted solution. One of the modern heuristic optimization paradigms is the particle swarm optimization. PSO is a kind of evolutionary algorithm based on a population of individuals and motivated by the simulation of social behavior instead of the survival of the fittest individual. It is a population-based evolutionary algorithm. Similar to the other population-based evolutionary algorithms PSO is initialized with a population of random solutions. Unlike the most of the evolutionary algorithm solution PSO also works with a randomized velocity, and the potential solutions called particles which are flown through the problem space. The difference between PSO and the other evolutionary algorithms is that PSO opts the path of cooperation over competition. On other hand algorithms commonly use some form of decimation, survival of the fittest. In contrast, the PSO population is stable and individuals are not destroyed or created. Individuals are influenced by the best performance of their neighbors. Individuals eventually converge on optimal points in the problem domain. [54] PSO does not have genetic operators like between individuals and mutation, and other individuals never substitute particles during the process of implementation. The PSO refines its search by attracting the particles to positions with good solutions. While compared with genetic algorithms, the information sharing mechanism in PSO is totally different. In genetical algorithm chromosomes share information with each other. The whole population moves like a one group towards an optimal area. In PSO only the G_{best} i.e. global best or P_{best} i.e. fitness value gives the information. In PSO, all the particles tend to converge to the best solution quickly as compared with genetical algorithm. [8]

1.2 ORGANIZATION OF THESIS

The thesis carried out has been summarized in six chapters.

Chapter 1 highlights an overview to different methods that are opted for the analysis and a familiarization about those methods.

Chapter 2 contains the literature review.

Chapter 3 explains the modeling of DC motor.

Chapter 4 gives the introduction about the various controller methods.

Chapter 5 explains case study of DC motor speed control.

Chapter 6 provides with the results and discussions of various methods.

Chapter 7 gives the conclusion and future scope of the work.

CHAPTER 2

LITERATURE REVIEW

J.lygeros et.al. introduced a general formalism for generating a fuzzy plant model. It provides a description for Fuzzification, inference rules and defuzzification in terms of mapping. It demonstrated use of information by developing a general fuzzy logic, rule base plant model. It provides that fuzzy plant model exactly duplicates the behavior of discrete-time linear system.[1]

Pierre Guillemin et.al. implemented a fuzzy logic to regulate speed of universal motor by real time adjustment of motor current. This paper shows how a fuzzy logic approach can be applied to build a closed speed regulation loop form a very low cost tacho-generator. This paper also gives the practical procedure to define the input parameters and to build fuzzy logic rules when using fuzzy logic development tool. [2]

M. Perales et.al. proposed a system used to evaluate a variable speed, pitch angle of wind turbine by fuzzy logic techniques is described. A fuzzy logic control has been proposed based on the speed wind estimation in order to get maximum power and stability of the system. [3]

W.Tan et.al. introduced criteria based on disturbance rejection and system robustness are proposed to assess the performance of PID controllers. The robustness is measured by a two-block structured singular value, and the disturbance rejection is measured by the minimum singular value of the integral gain matrix. Examples show that the criteria can be applied to a variety of processes, whether they are stable, integrating or unstable; single-loop or multi-loop.[4]

QI Peng presented an improved particle swarm optimization (PSO) method for speed control of the brushless DC (BLDC) motor. With introducing a shrinkage factor into PSO algorithm, the speed control ability of the BLDC motor can be improved. The brushless DC motor is modeled in Simulink and the PSO algorithm is implemented in MATLAB. Comparing with fuzzy control method, the proposed method is more efficient in improving the step response characteristics, such as, reducing the steady-states error; rising time, settling time and maximum overshoot in speed control of a linear brushless DC motor. [5]

M.Ruderman et.al. gave a comparison of the state feedback control over standard cascade control architecture of DC motors. It gives advantages in terms of design complication, hardware realization and adaptivity. This paper presents a methodic approach to state space control of a DC motor. The state space model identified from experimental data provides the basis for a linear quadratic regulator (LQR) design. The state feedback linear control is augmented with a feed forward control for compensation of Coulomb friction. The controller is successfully applied and the closed loop behavior is evaluated on the experimental test bed under various reference signals. [6]

Z.Xuan et.al. introduced to the problem of time varying nonlinear and time delay in ventilation systems. This paper addresses variable- frequency fuzzy PID control for highway tunnel ventilator, which has disadvantage both dynamic performance of fuzzy control and steady state response of PID algorithm. Fuzzy PID algorithm depends upon on experts experience and fuzzy reasoning and it can carry out energy saving for enhancing the parameters and decreasing start up times. [7]

A.A. El Gammal et.al. presented the application of new particle swarm optimization technique for adjusting the gains of PID controller adaptively to give the minimum integral absolute error between speed demand and output response. This new technique converts all objective functions into a single objective function using specified weighted factors. Since the optimal PID control parameters are dependent on selected weighted factors, the weighted factors was also treated as dynamic optimizing parameters with PSO as a dual optimization and global selection of PID control parameters. [8]

M. Sharaf et.al. presented the novel application of PSO for optimal tuning of PID controller for high performance permanent magnet PMDC industrial motor drives. This will smooth the starting torque, enhance acceleration, dynamic tracking of reference speed. The dynamic error driven controller regulates the firing angle using PSO tuning PID block. The proposed dynamic controller utilizes the motor speed error, the armature current deviation, and dynamic current ripple error as input to PSO gain search algorithm. The proposed tuned PID is based on minimization of absolute of total error. [9]

L.Xing et.al. introduced to conventional Fuzzy PID controller. This paper proposes a FPID controller with variable integral parameter for temperature control in air conditioning system, which increases a fuzzy rule base to control the integral parameter of conventional PI controller. Through adjustment of integral coefficients, fuzzy control rule of conventional controller is indirectly adjusted to improve its function. [10]

Hui H.U. et.al. compared one of energy efficient mathematical models which use G.A. algorithm to solve with a new algorithm based on PSO algorithm. Through calculation of a cited application, simulation results such as speed position curve are obtained. Compared with other methods, it demonstrates this new PSO based algorithm has a better performance and can be considered to put into practical use. [11]

R.Rahmani et.al. presented an optimized fuzzy logic controller using particle swarm optimization for DC motor speed control. The controller model is simulated using MATLAB software and also experimentally tested on a laboratory DC motor. A contrast of the performance of different controllers such as PID controller, fuzzy logic controller and optimized fuzzy logic controller is presented as well. With reference to the results of digital simulations and experiment, the designed FLC-PSO speed controller obtains much better dynamic behavior compared to PID and the normal FLC designed. Moreover, it can acquire superior performance of the DC motor, and also perfect speed tracking with no overshoot. The optimized membership functions (MFs) are obviously proved to be able to provide a better performance and higher robustness in comparison with a regular fuzzy model, when the MFs were heuristically defined. Besides, experimental results verify the ability of proposed FLC under sudden change of the load torque which leads to speed variances. [12]

A.Visioli et.al. analyzed the most significant issues that have been recently been addressed by researchers in the field of Proportional-Integral-Derivative (PID) controllers. In particular, the most recent techniques proposed for tuning and designing PID-based control structures are briefly reviewed, together with methods for assessing their performance. Finally, fractional-order and event-based PID controllers are presented among the most significant developments in the field. [13]

K.Venu et.al. presented the state space approach in modeling of DC Motor. DC motors, because of their simplicity, ease of application, reliability and favorable cost have been a backbone of industrial applications. In these applications, the motor should be precisely controlled to give the desired performance. Modeling of any system is an important task in control applications because the electrical and mechanical components should be represented in mathematical form. Although several methods are available for representing DC Motor model, the state space approach is unique in the sense, accurate and precise control is possible. Mathematical models are developed for simple closed loop control of DC Motor and closed loop control with proportional Integral and Derivative (PID) controller. This approach of modeling is useful for designing intelligent control techniques like Fuzzy control, neural networks etc. [14]

H.E.A Ibrahim et.al. presented a PSO technique and bacterial foraging (BF) technique for determining the optimal parameters of PPID controller for speed control of brushless DC motor. The proposed technique was more efficient in improving the step response of motor. [15]

F.Valdez et.al. described the use of Particle Swarm Optimization (PSO) for designing an optimal fuzzy logic controller of a DC Motor. The speed of the DC motor can be adjusted to a great extent so as to provide easy control and high performance. There are several conventional and numeric types of controllers intended for controlling the DC motor speed and executing various tasks: PID Controller, Fuzzy Logic Controller; or the combination between them: Fuzzy-Swarm, Fuzzy-Neural Networks, Fuzzy-Genetic Algorithm, Fuzzy-Ants Colony. In this case the approach will optimize the membership functions of a fuzzy logic controller (FLC) using PSO and the obtained results were simulated on Simulink of Matlab. [16]

CHAPTER 3

DC MOTOR MODELLING

3.1 DC Motor

DC motors are motors which run on direct current either from a DC supply or a battery. Direct current defines electricity at a constant voltage. When a battery or DC supply is connected to DC motor leads. The motor converts electrical energy into mechanical energy. A DC motor relies on the fact that magnet poles repels and unlike magnetic poles attracts each other. A coil of wire with a current running throughout it generates an electromagnetic field associated with the center of the coil. By switching the current on or off in a coil its magnetic field can be switched on or off or by switching the direction of the current in the coil the direction of the generated magnetic field can be switched 180°. DC motor typically has a stationary set of magnets in the stator and an armature with a series of two or more windings of wire wrap in insulated stack slots just about iron pole pieces with the ends of the wires terminating on a commutator. The armature includes the escalating bearings that keep it in the center of the motor and the power shaft of the motor and the commutator connections. The winding in the armature continue to loop all the way around the armature and uses either single or parallel conductors (wires), and can circle numerous times around the stack teeth. The total amount of current sent to the coil, the coil's size and what it's wrapped around state the strength of the electromagnetic field created. The string of turning a particular coil on or off dictate what direction the effective electromagnetic fields are pointed. By turning on and off coils in sequence a rotating magnetic field can be created. These rotating magnetic fields intermingle with the magnetic fields of the magnets (permanent or electromagnets) in the stationary part of the motor (stator) to create a force on the armature which causes it to rotate. In some DC motor designs the stator fields use electromagnets to create their magnetic fields which allow greater control over the motor. At high power levels DC motors are cooled using forced air. The commutator allows each armature coil to be activated in turn. The current in the coil is typically supplied via two brushes that make moving contact with the commutator. Now, some brushless DC motors have electronics that switch the DC current to each coil on and off and have no brushes to wear out or create sparks. [11]

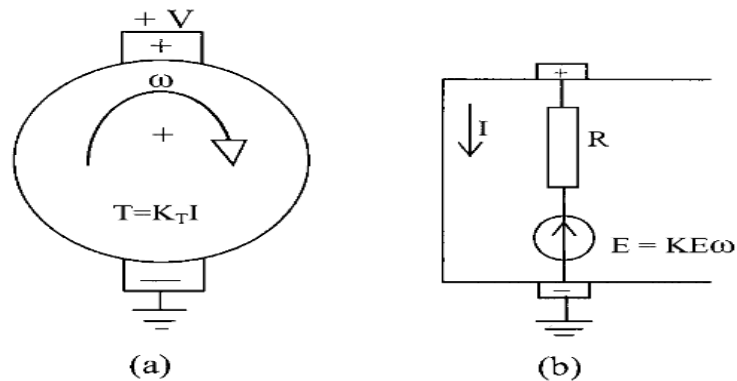


Figure 3.1 Schematic representation of DC motor [3]

3.2 Principles of Operation

DC motor consists of stator and rotor. Voltage is applied across the motor terminals and current flows from positive terminal to negative terminal. The system consists of brushless and collector transfers current from the stationary terminals of the coils of rotor. The current flowing through coils interacts with permanent magnetic field of rotor and generator and electromagnetic force according to ampere's law. Electromagnetic force turns the rotor and sets motor into working condition. In order to keep motor running in one direction, collector switches the rotor coils as rotor turns. The rotation of rotor causes its coil to interact with magnetic field lines. Thus an emf (electromotive force) is introduced in coils according to faraday's law. The induced emf is referred to as counter emf (also called back emf) because it opposes the applied voltage. Figure 3.1 shows back emf, E and internal resistance R of motor. This represents the resistance of rotor windings, brushes, collectors etc. The internal resistance is causes power loss in electrical motors. As current flows through motor, energy is converted into heat through joules effect. The resulting power must be dissipated otherwise it will result in short circuiting or burning of insulation. [11]

3.3 DC Motor Equation

The torque developed by the DC motor develops in electromagnetic force applied to rotor windings. The electromagnetic force on conductor is proportional to current flowing through the conductor, hence torque generated by DC motor is proportional with current flowing through

windings of rotor. The constant of proportionality K_t

$$T = K_t I \dots \dots \dots (3.1)$$

A representation of equation 3.1 is given in figure 3.2 (a, b). [3] The amplitude of back emf is in proportional to angular speed of rotor. The constant of proportionality K_e

$$E = K_e \omega \dots \dots \dots (3.2)$$

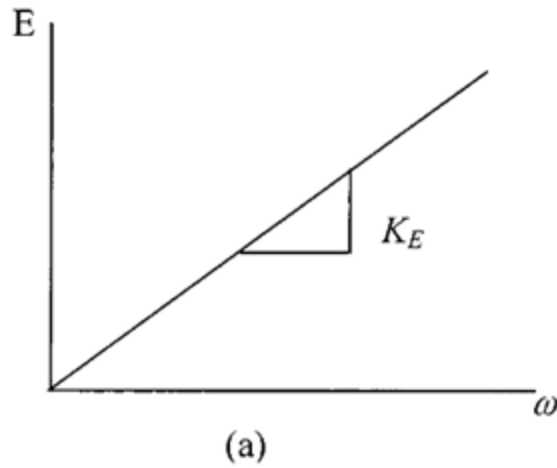


Figure 3.2 (a) Dependence of torque on electric current (E vs ω)

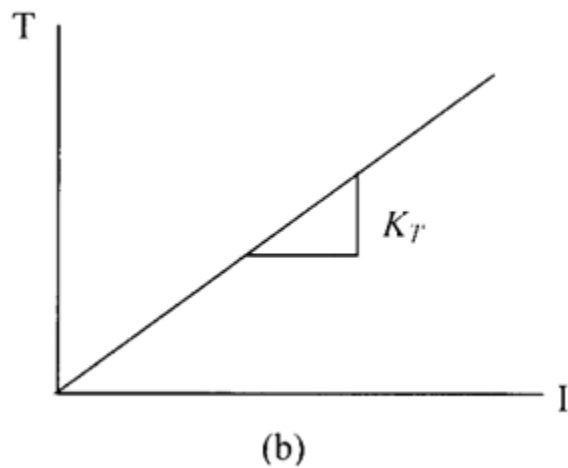


Figure 3.2 (b) Dependence of back emf on angular speed (T vs I)

The electrical model of DC motor is based on ohm's law and Kirchhoff's law

$$V = IR + E \dots\dots\dots (3.3)$$

In SI units it can be shown that the constants K_t and K_e are equal and can also be represented as a one constant K , which is motor constant.

The proof of the contention starts with principle of power conservation during the electromechanical conversion process. The electrical input power to DC motor is

$$P_{in} = VI + (RI + E)I = RI^2 + EI \dots\dots\dots (3.4)$$

From this input power, a part is lost as heat (RI^2) and rest is converted into mechanical power. The electrical power that is converted into mechanical power is total input power. According to equation (3.4) electric power transferred is given by

$$P_{electr} = EI \dots\dots\dots (3.5)$$

The mechanical power transferred to DC motor

$$P_{mechanical} = T\omega \dots\dots\dots (3.6)$$

Equating electric power and mechanical power we get

$$EI = T\omega \dots\dots\dots (3.7)$$

By substituting equation (3.1) and equation (3.2) in equation (3.7)

$$K_t I \omega = K_e I \omega \dots\dots\dots (3.8)$$

Cancelling $I\omega$ on both sides

$$K_t = K_e \dots\dots\dots (3.9)$$

It has been verified that both are equivalent so we can replace two constants with only one constant i.e. K and equation (3.1) and equation (3.2) can be rewritten as

$$T = KI \qquad E = K\omega$$

After substituting these values into equation (3.3) we get

$$V = \left(\frac{R}{T}\right) T + K\omega \dots\dots\dots (3.10)$$

Equation (3.10) represents electromechanical model of DC motor.

3.4 Transfer function of DC motor

Figure 3.3 shows the schematic representation of an armature controlled DC motor

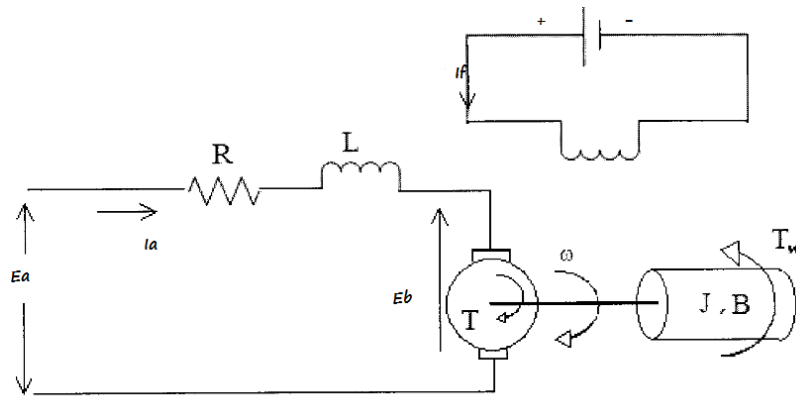


Figure 3.3 Armature controlled DC motor with load [2]

The variables and parameters of above diagram of a dc motor model are given below

R = winding resistance (Ohm)

L = winding inductance (Henry)

I_a = armature current (A)

I_f = field current (A)

E_a = armature applied voltage (V)

E_b = back emf (V)

ω = angular velocity of motor (rad/ sec)

T = torque developed by motor (Newton-m)

J = moment of inertia (kg m^2)

B = friction coefficient (N-m/ (rad/sec))

T_w = disturbance torque (N-m)

As the field current is maintained at constant level so torque become

$$T = K_t I_a \dots\dots\dots (3.11)$$

$$E_b = K_b \omega \dots\dots\dots (3.12)$$

Where K_t = torque constant, K_b = back emf constant

The differential equation of circuit is

$$L \frac{di_a}{dt} + RI_a + E_b = E_a \dots\dots\dots (3.13)$$

The torque equation is given by

$$J \frac{d\omega}{dt} + B\omega + T_w = T \dots\dots\dots (3.14)$$

Taking Laplace of equations (3.11) – (3.14) we get

$$T(s) = K_t I_a(s) \dots\dots\dots (3.15)$$

$$E_b(s) = K_b \omega(s) \dots\dots\dots (3.16)$$

$$(Ls + R) I_a(s) = E_a(s) - E_b(s) \dots\dots\dots (3.17)$$

$$(Js + B) \omega(s) = T(s) - T_w(s) \dots\dots\dots (3.18)$$

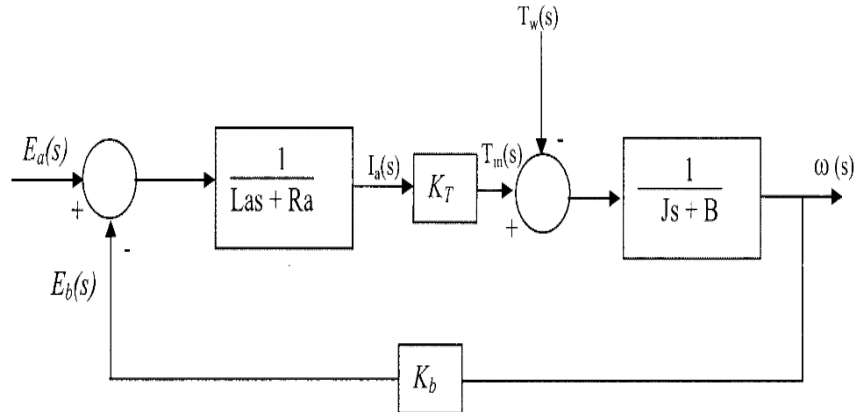


Figure 3.4 Block diagram of DC motor system [45]

Figure 3.4 shows the clear representation of block diagram of DC motor. The net voltage applied to circuit is $(E_a - E_b)$ having a transfer function $1 / (sL+R)$. The result is armature current $I_a(s)$. The torque rotates the load at $\omega(s)$ against the disturbance torque T_w ; load having moment of inertia J and friction coefficient B and has transfer function $1 / (Js + b)$. The back emf is feed backed to summing point. [45]

$$\frac{\omega(s)}{E_a(s)} = \frac{K_t}{(Ls+R)(Js+B) + K_t K_b} \dots\dots\dots (3.19)$$

Equation (3.19) gives the transfer function between angular velocity and input voltage.

CHAPTER 4

INTRODUCTION TO CONTROLLERS

4.1 PID CONTROLLER

Proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism which is used in industrial control systems. A PID controller calculates error value as the difference between measured process variable and desired set point. The controller tries to reduce the error by adjusting the process through the use of a manipulated variable.

The PID controller algorithm has three different parameters, i.e. the proportional, integral and derivative values, denoted as P, I, and D. These values can be interpreted in terms of time. P depends on the present error, I on accumulation of past errors and D is prediction of future errors, based on current rate of change. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, a damper, or the power supplied to a heating element. In the absence of understanding of the fundamental process, a PID controller has been considered as best controller. By tuning the three parameters of the PID controller using algorithm, the controller can give control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point, and the degree of system oscillation. [40]

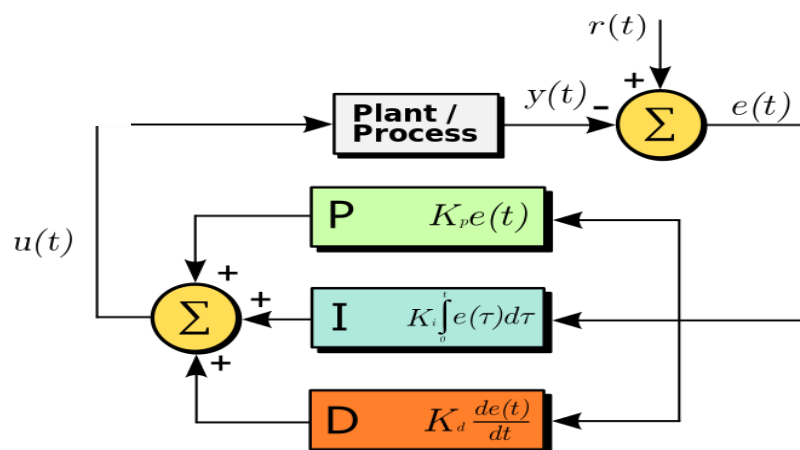


Figure 4.1 Block diagram of generalized PID controller [40]

4.2 FUZZY CONTROLLER

Fuzzy controllers are very simple practically. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs like switches, thumbwheels, and so on, to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. The output stage converts the combined result back into a specific control output value. Rules can be solved in parallel in hardware, or sequentially in software. The results of all the rules that have fired are "defuzzified" to a crisp value by one of several methods. Defuzzification is the process of producing a quantifiable result in fuzzy logic, given fuzzy sets and corresponding membership degrees. It is typically needed in fuzzy control systems. These will have a number of rules that transform a number of variables into a fuzzy result, that is, the result is described in terms of membership in fuzzy sets. Defuzzification is interpreting the membership degrees of the fuzzy sets into a specific decision or real value. [13]

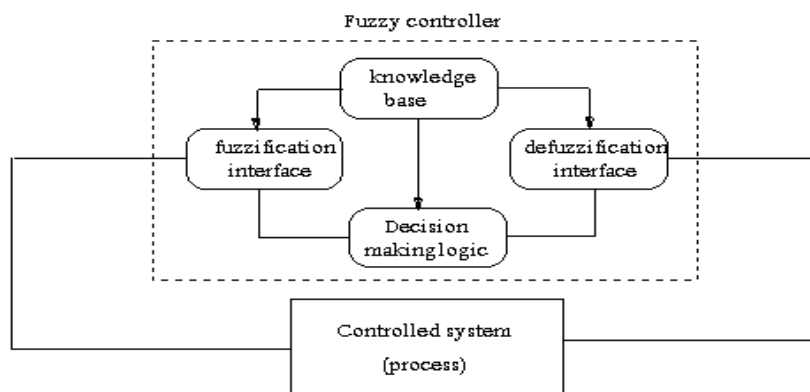


Figure 4.2 Block diagram representation for Fuzzy controller. [20]

The general process is as follows

- Document the system's operational specifications and inputs and outputs.
- Document the fuzzy sets for the inputs.
- Document the rule set.
- Determine the defuzzification method.
- Run through test suite to validate system, adjust details as required.
- Complete document and release to production.

4.3 PARTICLE SWARM OPTIMIZATIONS

Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position but, is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. PSO is originally attributed to Kennedy, Eberhart and Shi. It was first intended for simulating social behavior, as a stylized representation of the movement of organisms in a bird flock or fish school. The algorithm was simplified and it was observed to be performing optimization. The book by Kennedy and Eberhart describes many philosophical aspects of PSO and swarm intelligence. An extensive survey of PSO applications is made by Poli. PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and Quasi-Newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc. [41]

4.3.1 Algorithm

A basic variant of the PSO algorithm works by having a population called a swarm of candidate solutions called particles. These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated until a satisfactory solution will eventually be discovered.

Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be the cost function which must be minimized. The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the objective function value of the given candidate solution.

The gradient of f is not known. Let S be the number of particles in the swarm, each having a position $x_i \in \mathbb{R}^n$ in the search-space and a velocity $V_i \in \mathbb{R}^n$. Let p_i be the best known position of particle i and let g be the best known position of the entire swarm. A basic PSO algorithm is represented by

- For each particle $i = 1, \dots, S$ do:
 - Initialize the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$, where b_{lo} and b_{up} are the lower and upper boundaries of the search-space.
 - Initialize the particle's best known position to its initial position: $p_i \leftarrow x_i$
 - If $(f(p_i) < f(g))$ update the swarm's best known position: $g \leftarrow p_i$
 - Initialize the particle's velocity: $v_i \sim U(-|b_{up}-b_{lo}|, |b_{up}-b_{lo}|)$
- Until a termination criterion is met (e.g. number of iterations performed, or a solution with adequate objective function value is found), repeat:
 - For each particle $i = 1, \dots, S$ do:
 - Pick random numbers: $r_p, r_g \sim U(0,1)$
 - For each dimension $d = 1, \dots, n$ do:
 - Update the particle's velocity: $v_{i,d} \leftarrow \omega v_{i,d} + \phi_p r_p (p_{i,d}-x_{i,d}) + \phi_g r_g (g_d-x_{i,d})$
 - Update the particle's position: $x_i \leftarrow x_i + v_i$
 - If $(f(x_i) < f(p_i))$ do:
 - Update the particle's best known position: $p_i \leftarrow x_i$
 - If $(f(p_i) < f(g))$ update the swarm's best known position: $g \leftarrow p_i$
 - Now g holds the best found solution.

The parameters ω , ϕ_p , and ϕ_g are selected by the practitioner and control the behavior and efficiency of the PSO method.

4.3.2 Parameter selection

The choice of PSO parameters can have a large impact on optimization performance. Selecting PSO parameters that yield good performance has therefore been the subject of much research. The PSO parameters can also be tuned by using another overlaying optimizer, a concept known as meta-optimization. Parameters have also been tuned for various optimization scenarios. [41]

4.3.3 Convergence

Convergence refers to the swarm's best known position “g” converging to the optimum of the problem, regardless of how the swarm behaves. Convergence may also refer to a swarm collapse in which all particles have converged to a point in the search-space, which may or may not be the optimum.

Several attempts at mathematically analyzing PSO convergence exist in the literature. These analyses have resulted in guidelines for selecting PSO parameters that are believed to cause convergence, divergence or oscillation of the swarm's particles, and the analyses have also given rise to several PSO variants. The analyses were criticized by Pedersen for being oversimplified as they assume the swarm has only one particle, that it does not use stochastic variables and that the points of attraction, that is, the particle's best known position p and the swarm's best known position “g”, remain constant throughout the optimization process. Furthermore, some analyses allow for an infinite number of optimization iterations which is not possible in reality. This means that determining convergence capabilities of different PSO algorithms and parameters therefore still depends on empirical results. One attempt at addressing this issue is the development of an "orthogonal learning" strategy for an improved use of the information already existing in the relationship between “p” and “g”, so as to form a leading converging exemplar and to be effective with any PSO topology. The aims are to improve the performance of PSO overall, including faster global convergence, higher solution quality, and stronger robustness[41].

5.1 DC Motor Model:

The dc motor model used for simulation is shown in figure 5.1

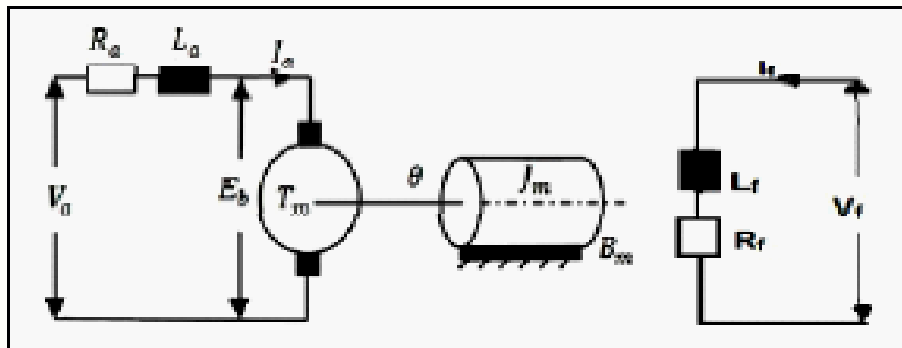


Figure 5.1 DC motor model [2]

The parameters and value used for simulation are shown in table 5.1

Table 5.1 Parameters and value of DC motor

| PARAMETERS | VALUES |
|-------------------------------------|------------------------|
| Armature circuit resistance (R) | 7.56 ohm |
| Armature circuit inductance (L) | 0.055H |
| Moment of inertia (I) | 0.068 kgm ² |
| Friction coefficient (B) | 0.03475 N.m.rad/sec |
| Torque constant (K _t) | 3.475 V.sec/rad |
| Back emf constant (K _b) | 3.475 V.sec/rad |

The mathematical model of DC motor can be constructed using following equations

$$\frac{d}{dt} \begin{bmatrix} \theta \\ i \end{bmatrix} = \begin{bmatrix} \frac{-B}{J} & \frac{K}{J} \\ \frac{-K}{L} & \frac{R}{L} \end{bmatrix} \begin{bmatrix} \theta \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} U \dots\dots\dots (5.1)$$

$$= [1 \quad 0] \begin{bmatrix} \theta \\ i \end{bmatrix} \dots\dots\dots (5.2)$$

These equations are in form of

$$\dot{\theta} = A \theta + Bu \dots\dots\dots (5.3)$$

$$Y = c \theta + Du \dots\dots\dots (5.4)$$

This is converted to a transfer function in order to make the dc motor model similar in terms of transfer function to that of PID in expression.

$$\frac{\theta}{V} = \frac{k}{(Js+B)(Ls+R)+K^2} \dots\dots\dots (5.5)$$

Now simulation is started using Matlab. Here is the result of simple transfer function obtained from the motor parameters.

```

Command Window
>> prgm1

P_motor =

          3.475
-----
0.00374 s^2 + 0.516 s + 12.34

Continuous-time transfer function.

fx >> |
    
```

Now state space analysis is applied on the equations of simple transfer function and following values of the parameters:

```

motor_ss =

  a =
        x1      x2
x1 -0.511      51.1
x2 -63.18     -137.5

  b =
        u1
x1      0
x2 18.18

  c =
        x1  x2
y1      1   0

  d =
        u1
y1      0

Continuous-time state-space model.

```

After the reduced transfer function is obtained

```

num =

      0      0  929.3616

den =

  1.0e+03 *

      0.0010      0.1380      3.3000

g =

      929.4
-----
s^2 + 138 s + 3300

Continuous-time transfer function.

```

5.2 PID TUNING

For improvement in the performance of the dc motor, a PID controller is introduced and applied. This PID controller for the control of the dc motor is set up in Matlab/Simulink as shown in figure 5.2

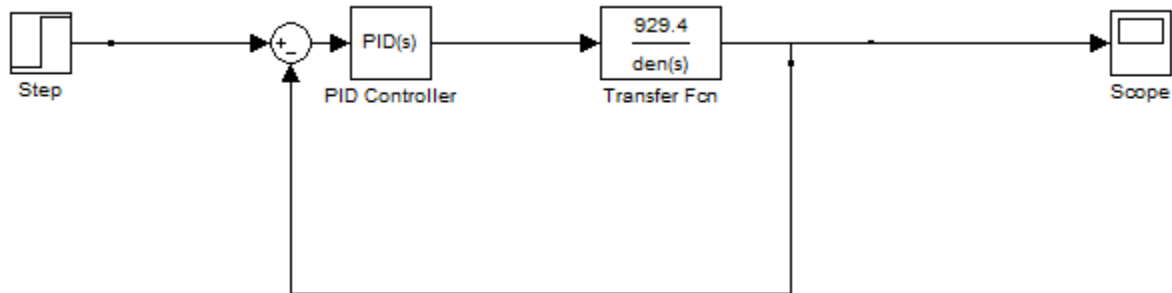


Figure 5.2 Matlab/simulink model for PID controller

A simple feedback control theory is utilized to represent the overall PID controlled system. This PID controller has the transfer function of the form:

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \dots\dots\dots (5.6)$$

It is observed that when the proportional gain alone is chosen arbitrarily, the response of the motor is not satisfactory. The same problem is experienced when the integral gain and the derivative gain alone are concentrated on. Therefore, in order to have the desired motor response, the PID controller has to be tuned.

Tuning of PID controller by means of trial and error method wastes time so tuning guide proposed by Ziegler-Nichols is adopted with aim of decreasing the rise time, eliminating the overshoot, speeding up the settling time of the system and reducing to a tolerable value the steady –state error which is difference between the steady -state output and desired output . When the PID controller is properly tuned and applied to a unit step input system.

5.3 FUZZY TUNING

Fuzzy logic can be defined as nonlinear mapping of an input data into a scalar output that is the output case decomposes into a anthology of independent multi- input/single-output systems . Fuzzy logic is used to imitate to the best of its ability of the human mind. Approximate behavior is the hallmark rather than its exactness. These flexibilities of fuzzy logic and its simplicity has make it desirable to be implemented especially in areas of dc motor control where proportional-integral-derivative (PID) controller is most popular. [23]

Fuzzy Logic Controller Algorithm

To successfully build a fuzzy logic controller system, the following steps are taken

- I. The input parameters are given to the fuzzy controller.
- II. Fuzzification is the process of taking crisp inputs or values and determine the degree to which those input belong to each of the appropriate fuzzy sets.
- III. Then these fuzzified sets are applied to the set of fuzzy rules. For this multiple antecedents, the fuzzy operator (AND) is used to find a single member that represents the result of the antecedent assessment such that

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \dots\dots\dots (5.7)$$

Where membership function for A is denoted by

$$\frac{\mu_A(x)}{\mu_B(x)} = \begin{cases} 1 & \text{if } x \text{ belongs to } A \\ 0 & \text{if } x \text{ not belongs to } A \end{cases} \dots\dots\dots (5.8)$$

For a continuous membership function of zero-one. Results applied to the membership function of the fuzzified set.

IV. Defuzzification is carried out by means of centroid technique to obtain the crisp output. This is expressed mathematically as

$$\text{COG} = \frac{\int_a^b \mu_A(x)xdx}{\int_a^b \mu_A(x)dx} \dots\dots\dots (5.9)$$

V. The algorithm is then implemented in Matlab with five member fuzzy inference system used for input parameters, which are error and the change in error and for the output also. A Mamdani-type fuzzy inference approach is opted.

The prepared rule for the controller is set up based on skilled knowledge and shown in table (5.2)

Table 5.2 Rules for fuzzy controller

| CHANGE IN ERROR (Δe) | ERROR (e) | | | | |
|--------------------------------|-----------|----|----|----|----|
| | NB | NM | ZR | PM | PB |
| NB | NB | NB | NM | NM | ZR |
| NM | NB | NM | NM | ZR | PM |
| ZR | NM | NM | ZR | PM | PM |
| PM | NM | ZR | PM | PM | PB |
| PB | ZR | PM | PM | PB | PB |

NB stands for a negative big with numerical equivalent of 0. NM stands for negative medium with numerical equivalent value 0.25. ZR stands for zero with numerical equivalent value 0.5, PM stands for positive medium with a numerical equivalent 0.75 and PB stands for positive big with numerical equivalent 1.

The simulation is run and the three-dimensional behavior of the controller generated is shown in figure 5.3

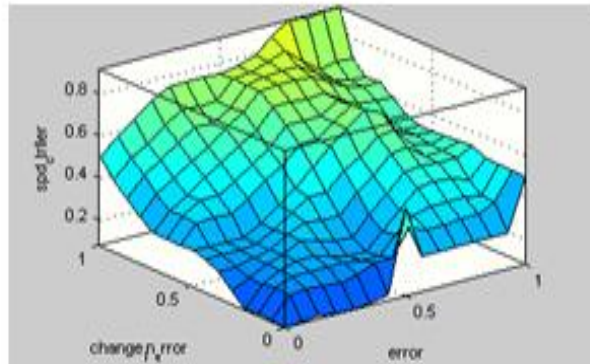


Figure 5.3 3D Behavior of Controller

Fuzzy logic generated is applied to the dc motor. The gain is introduced to achieve stability. The derivative term is used in feedback to make feedback signal in form of input signal. The set up in Matlab/simulink environment is shown figure 5.4

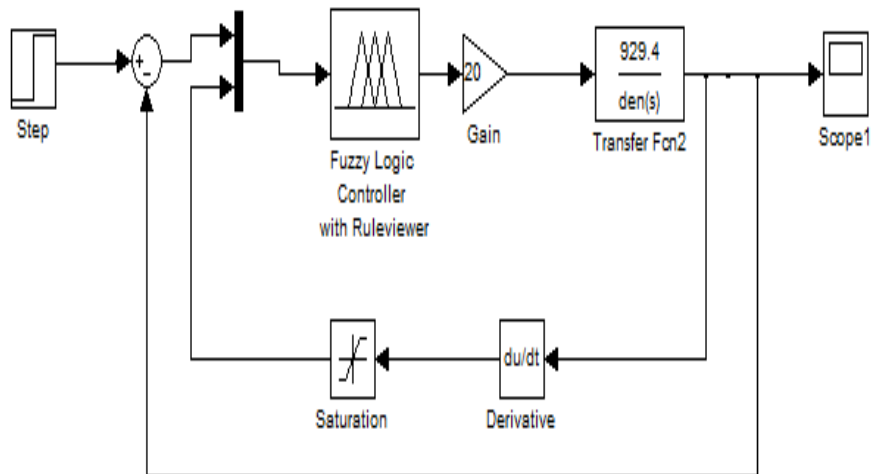


Figure 5.4 Matlab/simulink model for Fuzzy Controller

A unit step signal is applied to a closed-loop fuzzy logic motor controlled system. This signal corresponds to a supply voltage of 1 V and the system is run. The output is depicting the response of the dc motor is displayed via a scope after simulation.

5.4 PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle swarm optimization (PSO) is an evolutionary computation optimization technique (a search method based on a natural system) developed by Kennedy and Eberhart. The system initially has a population of random selective solutions. Each potential solution is called a particle. Each particle is given a random velocity and is flown through the problem space. The particles have memory and each particle keeps track of its previous best position (called P_{best}) and its subsequent fitness. There exist a number of P_{best} respective particles in swarm and for the particles with greatest fitness is called global best (G_{best}) of the swarm. The basic concept of the PSO technique lies in accelerating each particle towards its P_{best} and G_{best} locations, with a random weighted acceleration at each time step. [42]

The main steps in the particle swarm optimization and selection process are described as follows:

- a. Initialize a inhabitants of particles with random positions and velocities in D dimensions of the problem space and fly them.
- b. Evaluate the fitness of each particle in the swarm.
- c. For every iteration, compare each particle's fitness with its previous best fitness (P_{best}) obtained. If the current value is better than P_{best} , then update or set P_{best} value to current value and P_{best} location to current location in D dimensional space.
- d. Compare P_{best} of particles with each other and update the swarm global best location with greatest fitness value i.e. G_{best} .
- e. Change the velocity and position according to the above equations
- f. Calculate :

$$\text{Velocity} = \omega * \text{velocity} + c1 * (R1 * (\text{local_best_position} - \text{current_position})) + c2 * (R2 * (\text{globl_best_position} - \text{current_position})) \dots \dots \dots (5.10)$$

$$\text{Current_position} = \text{current_position} + \text{velocity} \dots \dots \dots (5.11)$$

- g. Repeat steps (a) to (e) until convergence is reached based on some desired single or multiple criteria.

PSO has many parameters and these are described as follows:

ω is called the inertia weight that controls the exploration and utilization of the search space because it dynamically adjusts velocity. V_{max} is the maximum permissible velocity for particles in case where velocity exceeds its maximum value. Thus, resolution and fitness of search depends upon V_{max} . If V_{max} is too high then particles will start moving beyond the good solution. On other hand if velocity is too low then particles are trapped in local minima. The constants C1 and C2 in equation 5.10 and 5.11 are known as cognition and social components. These are known as acceleration constant which changes the velocity of a particle towards P_{best} and G_{best} . The above figure shows the flow chat that how PSO algorithm works. [56]

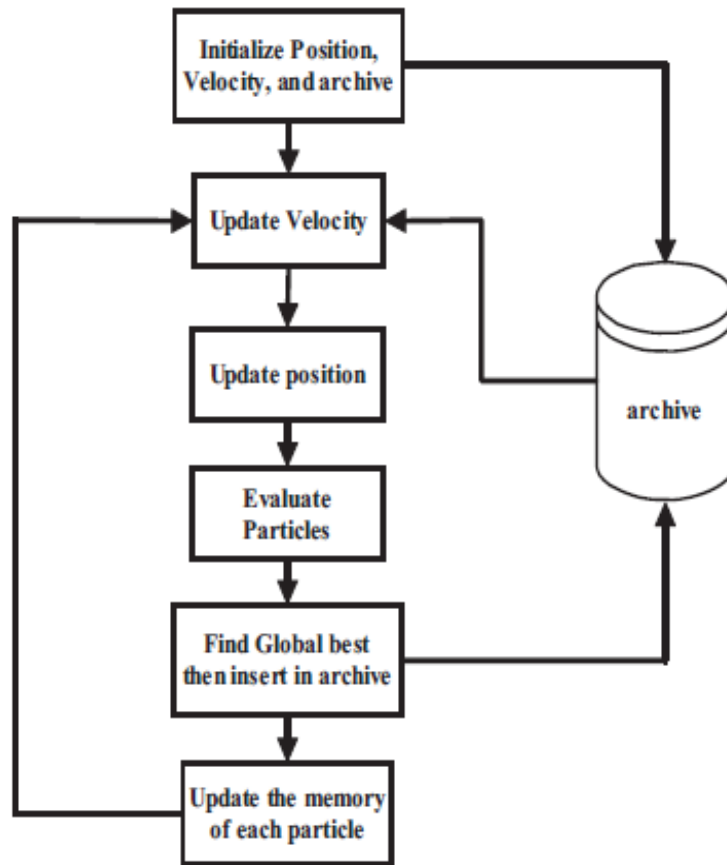


Figure 5.5 Flow chart of PSO algorithm

The main aim is that to minimize the errors using following function:

Minimize J where,

$$J = \alpha_1 (\text{rise time}) + \alpha_2 (\text{maximum overshoot}) + \alpha_3 (\text{settling time})$$

Where $\alpha_1 + \alpha_2 + \alpha_3 = 1$

This objective function J gives an operating point which in broad-spectrum is a trade off for the three weighted terms of the PID controller depending on the values of weights. [24]

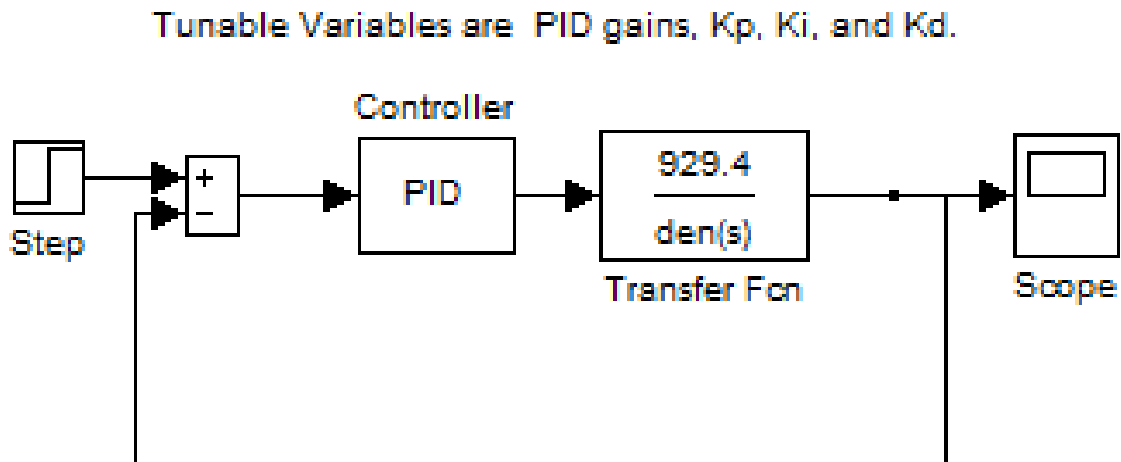


Figure 5.6 Matlab/simulink presentation of PSO

The values of K_p , K_i and K_d analogous to best solution obtained is used in following block diagram and simulation is carried out. The values of these PID parameters are used to study the response of the system.

5.5 COMPARISON

To contrast the responses of the proportional-integral-derivative controller (PID), fuzzy logic controller and PSO base controller the following Matlab/simulink arrangement is utilized.

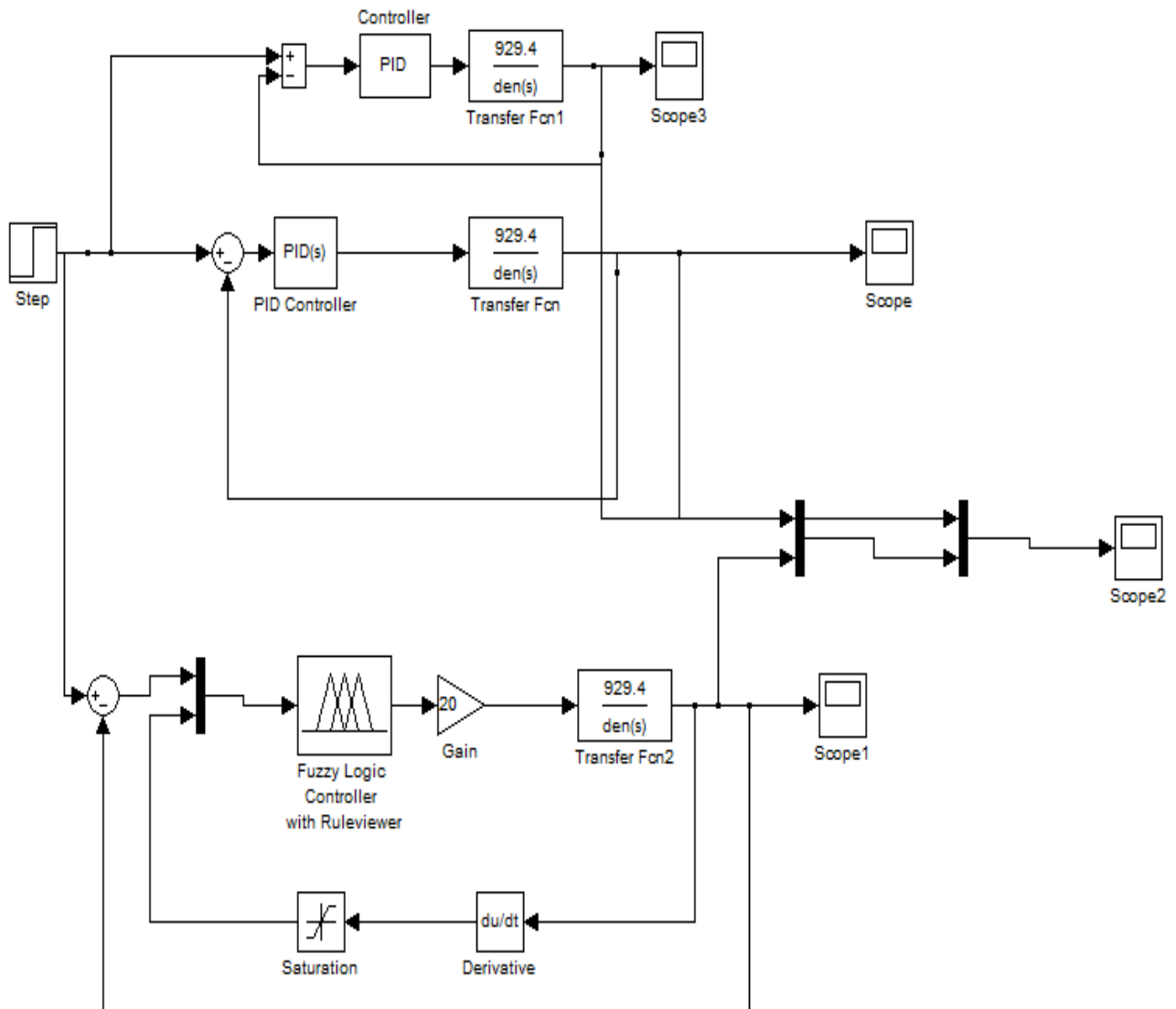


Figure 5.7 Matlab simulink model for PID, Fuzzy and PSO

A unit step input signal is applied and response is obtained for all three controllers.

CHAPTER 6
RESULTS AND DISCUSSIONS

In this analysis the results has been obtained from the developed methods using PID controller, fuzzy controller as well as particle swarm optimization (PSO) based controller which has already been discussed.

6.1 PID AND FUZZY SIMULATION

The parameters that are obtained from the simulation of PID controller and fuzzy controller are shown in table 6.1 in terms of rise time, maximum overshoot and settling time.

Table 6.1 Parameters of step response of PID and fuzzy controller

| Parameters | Rise time (sec) | Max. Overshoot (%) | Settling time (sec) |
|------------------|-----------------|--------------------|---------------------|
| PID controller | 0.267 | 6.457 | 0.227 |
| Fuzzy controller | 0.447 | 0.023 | 0.429 |

By analyzing the values obtained for considered parameters when step input is given. It is clear that fuzzy controller has an advantage over PID controllers. It can be clearly observed that overshoot has reduced when fuzzy controller is used but on the cost of increased settling and rise time. To improve rise time and settling time with no overshoot PID is tuned using PSO algorithm

STEP RESPONSE OF PID AND FUZZY CONTROLLER

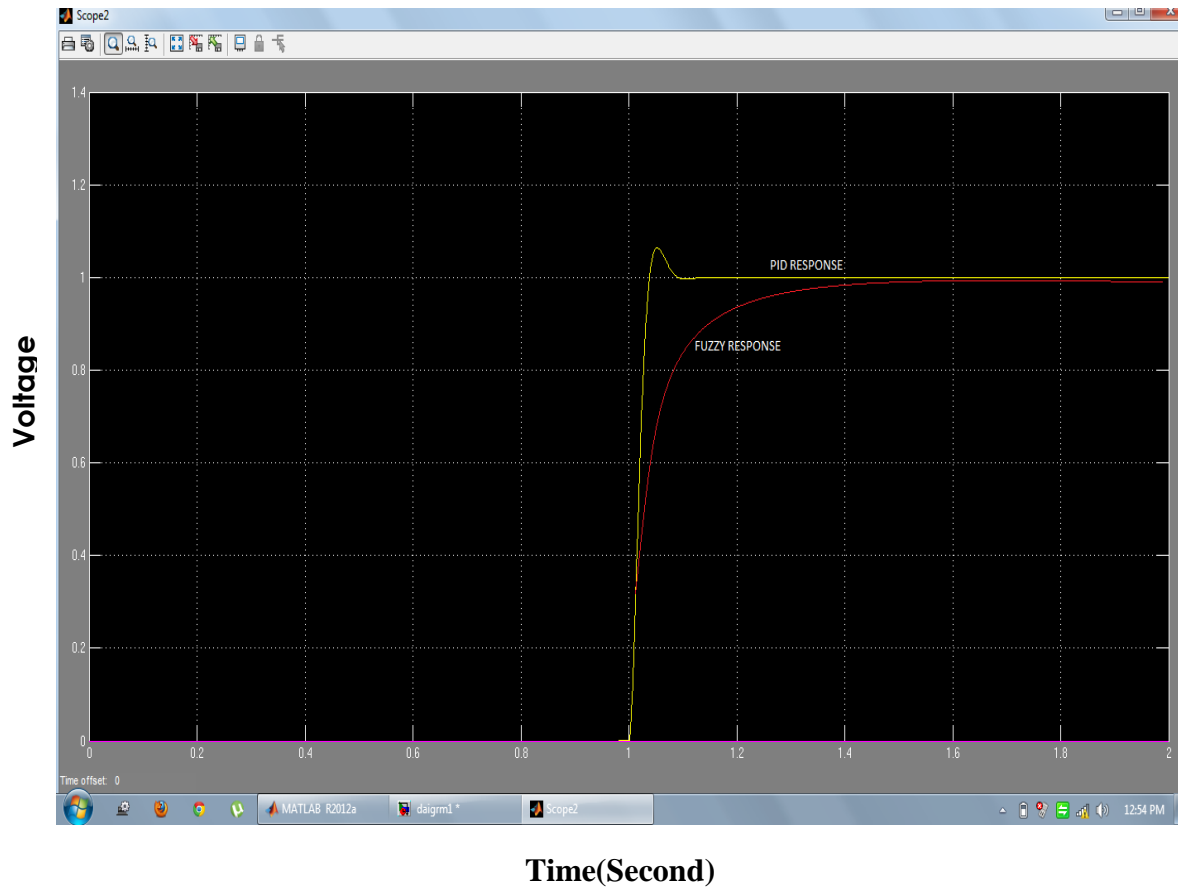


Figure 6.1 Step response of PID and Fuzzy controller

6.2 PSO SIMULATION

The following are results obtained after the simulation of program and are shown in the following tables. The result of objective is shown in table 6.2(a)-6.2(e).

Table 6.2(a) After 50 iterations

| | |
|------------|----------|
| α_1 | 0.206815 |
| α_2 | 0.257869 |
| α_3 | 0.195465 |

| | |
|-------|-----------------|
| K_P | 47.872518829792 |
| K_I | 38.528289832427 |
| K_D | 18.050645779785 |

Table 6.2(b) After 100 iterations

| | |
|------------|----------|
| α_1 | 0.256815 |
| α_2 | 0.305786 |
| α_3 | 0.155475 |

| | |
|-------|------------------|
| K_P | 42.5698645676543 |
| K_I | 27.6598463563434 |
| K_D | 17.789776456354 |

6.2(c) After 150 iterations

| | |
|------------|----------|
| α_1 | 0.273645 |
| α_2 | 0.318562 |
| α_3 | 0.138756 |

| | |
|-------|------------------|
| K_P | 36.896786456343 |
| K_I | 18.6784567645636 |
| K_D | 16.6787564352454 |

6.2(d) After 200 iterations

| | |
|------------|----------|
| α_1 | 0.283756 |
| α_2 | 0.327859 |
| α_3 | 0.127689 |

| | |
|-------|-------------------|
| K_P | 32.0128735263455 |
| K_I | 14.56747873283983 |
| K_D | 14.456467898090 |

6.2(e) After 250 iterations

| | |
|------------|----------|
| α_1 | 0.285768 |
| α_2 | 0.329354 |
| α_3 | 0.119867 |

| | |
|-------|------------------|
| K_P | 30.6574879438238 |
| K_I | 12.65747803912 |
| K_D | 16.097873562 |

The parameters obtained after simulation of particle swarm optimization are shown in table 6.3 in terms of rise time, maximum overshoot and settling time.

Table 6.3 Parameters of step response of particle swarm optimization simulation

| Parameters | Rise time (sec) | Max. Overshoot (%) | Settling time (sec) |
|---------------------|--------------------|--------------------|---------------------|
| After 50 iteration | 0.3907 | 0.042 | 0.6467 |
| After 100 iteration | 0.325 | 0.029 | 0.532 |
| After 150 iteration | 0.227 | 0.014 | 0.497 |
| After 200 iteration | 0.198 | 0 | 0.383 |
| After 250 iteration | 0.153 | 0 | 0.257 |

Table 6.3 shows the parameters of step response when PID is tuned using PSO algorithm. A performance-dependent PSO based PID controller is designed for speed control of DC motor using methodology discussed in chapter 5. Figure 5.10 present the comparative response of PID controller, fuzzy controller and PSO algorithm for DC motor. Table 6.3 represents better rise time (0.153), overshoot (0) and settling time (0.257) as compare to corresponding values obtained from PID and Fuzzy controller as shown in table 6.1.

CHAPTER 7

CONCLUSION

The use of PID controller for dc motor is no doubt is robust simple and certain especially for a well tuned PID controller. But this method does not prove satisfactory for many systems as it causes high overshoot. So the use of fuzzy logic seems to be particularly appropriate as it allows us to make use of the human intelligence. It is clear from the results that the use of soft-computing technique gives better outputs. The level of overshoot for the output response is successfully decreased using the fuzzy logic controller. But the improved performance is at the cost of increased rise-time and settling-time. On other hand PSO provides a good response over both PID and Fuzzy controllers. PSO dynamic search and optimization criterion is based on total error minimization. The main objective functions to be minimize rise time, maximum overshoot, and the settling time. The optimal PID controller parameters are dependent on selection of the weighting factors. So the main challenger was to choose these specific weighting factors. The weighting factors as active optimizing parameters within the particle swarm optimization were considered as optimization and global selection of PID controller as well as weighting parameters. The multiple objectives are converted into a single objective function in a weighted functional. The optimization solution gives a value that reflects a concession between all key objectives. The Computer simulation results show that the choice of optimal control gains is essential for effective response.

FUTURE SCOPE

The single optimization PSO process may not be acceptable for systems with multiple conflicting objectives function. So, future work is to use the Discrete Multi-Objective Particle Swarm Optimization, DMOPSO and extend the problem by incorporating two objective functions. Better results can be obtained by combining PSO with other techniques like ACO, GA.

CHAPTER 8

CHECK FOR ORIGINALITY

The dissertation report presented here has been checked for its originality using online plagiarism checker “Paper Rater”, available at http://www.paperrater.com/plagiarism_checker. Various theoretical concepts are explained as per the references from different technical books which I studied during my engineering graduation and post graduation studies. Thanks to all those who are already present in my references text.

REFERENCES

- [1] A. A. and I. D. Landaq “A method of the auto calibration of PID controllers”, *Autonmlica*, Vol. 3, No. 1, pp. 41-53, 1995.
- [2] Adel M. Sharaf, Adel A. A. El-Gammal “A Novel Particle Swarm Optimization PSO Tuning Scheme for PMDC Motor Drives Controllers”, Vol. 5, No. 1, pp. 56-61, 2009.
- [3] Adel A. A. El-Gammal1 Adel A. El-Samahy “Adaptive Tuning of a PID Speed Controller for DC Motor Drives Using Multi- Objective Particle Swarm Optimization” Vol.1, No. 1, pp. 18-22, 2009.
- [4] Ali Bekir Yildiz, “Electrical equivalent circuit based modeling and analysis of direct current motors” *Proceedings of the IEEE*, Vol. 2, No. 5, pp. 80-87,2012
- [5] *Advance in Electronic and Electric Engineering*. ISSN 2231-1297, Vol 3, No. 9, pp. 1209-1220, 2013.
- [6] Ashry, M.M. ; Kamalova, Z.Z. ; Breikin, T.V. “Tuning of digital PID controller parameters using local optimal control”, *Proceedings of the IEEE*, Vol. 23, No. 8, pp. 35-53, 2008.
- [7] B. Allaoua, A. Abderrahmani, B. Gasbaoui , A.Nasri. “The Efficiency of Particle Swarm Optimization Applied on Fuzzy Logic DC Motor Speed Control”, *Serbian Journal*
- [8] C.Ou and W. Lin, “Comparison between PSO and GA for parameters optimization of pid controller,” in *Mechatronics and Automation*, *Proceedings of the 2006 IEEE International Conference on*, pp. 2471 –2475, 2006.
- [9] C. Xia, M. Zhang, Y. Wang, and D. Liu, “The direct torque control for permanent magnet brushless dc motors” *Proceedings of the Chinese Society of Electrical Engineering*, Vol. 28,pp. 104–109, 2008.
- [10] *Control system engineering* by I.J. Nagrath.
- [11] D. P. Kothari, I. J. Nagrath, “*Electric Machines*”, Tata McGraw – Hill Education Private Limited, ISBN-13: 978-0-07-058377-1, ISBN-10: 0-07-058377-3, 2004.

- [12] DC Motors dynamic model and control techniques by Luca Zaccarian.
- [13] Fuzzy controller http://en.wikipedia.org/wiki/Fuzzy_control_system
- [14] Farzan Rashidi', Mehran Rashidi: and Arash Hashemi-Hoseinit ,“Speed Regulation of DC Motors Using Intelligent Controllers”, IEEE, pp.925-930
- [15] F. Valdez, P. Melin, O. Castillo, "An improved evolutionary method with fuzzy logic for combining Particle Swarm Optimization and Genetic Algorithms". Appl. Soft Comput. 11(2): p.p. 2625-2632, 2011.
- [16] G. Haung and S. Lee, “PC Based PID speed control of DC motor,” in Proceedings of International Conference on Audio, Language and Image Processing (ICALIP-2008), pp. 400-407 2008,.
- [17] Gupta, S.K. Varshney, P. “Fractional Fuzzy PID Controller for Speed Control of DC Motor”, Proceedings of the IEEE, Vol. 5, No. 8, pp. 53-59,2013.
- [18] Husain Ahmed “Controlling of D.C. Motor using Fuzzy Logic Controller” Proceedings of the IEEE, Vol. 8, No. 3, pp. 45-57, 2012
- [19] H.E.A. Ibrahim, F.N. Hassan, Anas O. Shomer “Optimal PID control of a brushless DC motor using PSO and BF techniques” Proceedings of the IEEE, Vol. 33, No. 5, pp. 54-77, 2013.
- [20] https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&docid=0lYDbgxEqxF3bM&tbnid=hurtaGpkZOUmM:&ved=0CAUQjRw&url=http%3A%2F%2Fmazsola.iit.unimiskolc.hu%2FDATA%2Fdiploma%2Fbrutoczki_kornelia%2Ffu_gz_03.htm
- [21] I-Hai Lin, P. Hwang, S.Chou, J. “Comparison on fuzzy logic and PID controls for a DC motor position controller”, Proceedings of the IEEE, Vol. 32, No. 9, pp. 251-263, 1994.
- [22] J.M. Mendel (1995) “Fuzzy Logic Systems for Engineering :A Tutorial”, Proceedings of the IEEE, Vol. 83, No. 3, pp. 345-377
- [23] J. Yen and R. Langari, Fuzzy logic: intelligence, control, and information” Proceedings of the IEEE, Vol. 28, No. 4, pp. 135-150, 1999.

- [24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Neural Networks, Proceedings of IEEE International Conference on, Vol. 4, pp. 1942–1948, 1995.
- [25] J.C. Basilio and S. R. Matos, "Design of PI and PID Controllers with Transient Performance Specification", IEEE Transactions Education, Vol. 45, Issue No. 4, pp. 364-370, 2002.
- [26] J. G. Ziegler and N. B. Nichols, "Optimum setting for automatic controllers," Transactions American Society of Mechanical Engineers, Vol. 64 , pp. 759–768, 1942.
- [27] J. Mendes and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space". IEEE Transactions on Evolutionary Computation, 6(1):58-73, 2002.
- [28] J. Mendes and J. Kennedy, "Population structure and particle swarm performance". Proceedings of IEEE conference on Evolutionary Computation, pp. 1671-1676. 2002.
- [29] K. A. Naik and P. Shrikant, "Stability Enhancement of DC Motor using IMC Tuned PID Controller," (IJAEST) International Journals of Advanced Engg. Science and Technologies, Vol. 4, Issue No. 1, pp. 092-096, 2011.
- [30] K. Venu Ch. Rushikesh V. Rajasekhar "Design and Analysis of DC Motor with PID Controller - A State Space Approach". Proceedings of the IEEE, Vol. 2, No.1, pp. 45-57, 2013.
- [31] K. Astrom and T. Hagglund, "PID Controller: Theory, Design and Tuning" 2nd edition, Library of Congress Cataloging-in-Publication Data, pp. 120–134, 1994.
- [32] K.H. Ang, G. Chong and Y. Li, "PID control system analysis, design and technology" IEEE transaction on Control System Technology, Vol.13, No.4, pp. 559-576, 2005.
- [33] M. Islam, S. Mir, and T. Sebastian, "Issues in reducing the cogging torque of mass-produced permanent-magnet brushless dc motor", IEEE Transactions on Industry Applications, Vol. 40, pp. 813 – 820, 2004.
- [34] Montiel, O. ; Sepulveda, R. ; Melin, Patricia ; Castillo, Oscar ; Porta, M.A. ; Meza, I.M. "Performance of a Simple Tuned Fuzzy Controller and a PID Controller on a DC Motor" Proceedings of the IEEE, Vol. 18, No.4, pp. 87-95, 2007.

- [35] M. Ruderman, J. Krettek, F. Hoffmann, T. Bertram “Optimal State Space Control of DC Motor” Proceedings of the IEEE, Vol. 7, No.5, pp. 57-67, 2008.
- [36] N. Kamaruddin, Z. Janin, Z. Yusuf and M. N. Taib, “PID Controller Tuning for Glycerin Bleaching Process Using Well-Known Tuning Formulas- A Simulation Study” in Proceedings of 35th Annual Conference of IEEE on Industrial Electronics (IECON-2009), pp. 1682-1686, 2009.
- [37] O. Montiel, R. Sepulveda, P. Melin, O. Castillo, M. A. Porta and I. M. Meza, “Performance of a Simple Tuned Fuzzy Controller and a PID Controller on a DC Motor” in Proceedings of IEEE Symposium on Foundation of Computational Intelligence , pp. 531-537, 2007.
- [38] K.Ogata, Modern Control Engineering, 3rd edition, 1997
- [39] PID Design method for the DC Motor, www.library.cmu.edu/ctms/examples/motor/pid2.htm [accessed on 15 May, 2014].
- [40] PID controller http://en.wikipedia.org/wiki/PID_controller
- [41] Particle swarm optimization http://en.wikipedia.org/wiki/Particle_swarm_optimization
- [42] QI Peng an improved PSO-based optimization BLDC motor”, IEEE Transactions, Vol. 2, pp. 88 –92, 2005.
- [43] R. Rahmani, A. Khairuddin, S. Cherati, and H. Pesaran, “A novel method for optimal placing wind turbines in a wind farm using particle swarm optimization (pso)”, in IPEC, 2010 Conference Proceedings, pp. 134 –139, 2010.
- [44] R. Krohling and J. Rey, “Design of optimal disturbance rejection pid controllers using genetic algorithms,” Evolutionary Computation, IEEE Transactions on, Vol. 5, pp. 78 –82, 2001.
- [45] Rohit G. Kanojiya, P. M. Meshram, Optimal Tuning of PI Controller for Speed Control of DC motor drive using Particle Swarm Optimization, Vol. 21, pp. 56-60, 2012.
- [46] R.Arulmozhiyal, R.Kandiban “Design of Fuzzy PID Controller for Brushless DC Motor”, Proceedings of the IEEE, Vol. 12, No.8, pp. 47-53, 2012.

- [47] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," Proceedings of 6th International Symposium on Micro Machine and Human Science, pp. 39-43, October 4-6, 1995.
- [48] R. Eberhart and Y Shi, "Particle swarm optimization: developments, applications and resources" Proceedings of the 2001 Congress on Evolutionary Computation, Vol. 1, pp. 81 -86, 2001.
- [49] S. Tong, H. Li, and W. Wang, "Observer-based adaptive fuzzy control for siso nonlinear systems", Fuzzy sets and systems, Vol. 148, No. 3, pp. 355–376, 2004.
- [50] S. Ghoshal, "Optimizations of pid gains by particle swarm optimizations in fuzzy based automatic generation control", Electric Power Systems Research, Vol. 72, No. 3, pp. 203 – 212, 2004.
- [51] Visioli A., "Fuzzy Logic Based Set-Point Weight Tuning of PID Controllers", IEEE Transaccions on systems, man and cybernetics – Part A: Systems and Humans, Vol. 29, No. 6, 1999.
- [52] Waleed Abd El-Meged El-badry, "Design and implementation of Real Time DC motor speed control using Fuzzy logic", Proceedings of the IEEE, Vol. 23, No.7, pp. 25-37, 2007
- [53] Y. Ma, Y. Liu and C. Wang Elissa, "Design of Parameters Self-tuning Fuzzy PID Control for DC Motor", Proceedings of Second International Conference on Industrial Mechatronics and Automation (ICIMA), vol. 2, pp. 345-348, 2010.
- [54] Y. Shi and R. Eberhart, "Parameter Selection in Particle Swarm Optimization", Proceedings of Seventh Annual Conf. on Evolutionary Programming, pp. 591-601, 1998.
- [55] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization" Proceedings of the 1999 Congress on Evolutionary Computation, Vol. 3, 1999.
- [56] Zadeh, L. A., Fuzzy Sets. Information and Control, 338-353, 1965.