Comparative Performance Study of ACO & ABC Optimization based PID Controller Tuning for Speed Control of DC Motor Drives

A Thesis report

submitted towards the partial fulfilment of the

requirements of the degree of

Master of Engineering

in

Power System and Electric Drives

submitted by

Jitender Kaushal

(801041011)

Under the supervision of

Mr. Souvik Ganguli

Assistant Professor, EIED

DEPARTMENT OF ELECTRICAL AND INSTRUMENTATION ENGINEERING

THAPAR UNIVERSITY

PATIALA – 147 004

JUNE, 2012

CERTIFICATE

*I hereby declare that the report entitled **"Comparative Performance Study of ACO and ABC Optimization based PID Controller Tuning for Speed Control of DC Motor Drives"** is an authentic record of my own work carried out as requirements for the award of degree of M.E. (Power System & Electric Drives) at Thapar University, Patiala under the guidance of Mr. Souvik Ganguli (Assistant Professor, EIED) during January to June, 2012.*

Date: 15-06-2012
Place: Thapar University, Patiala

Jitender
Jitender Kaushal
Roll No. 801041011

It is certified that the above statement made by the candidate is correct to best my knowledge and belief.

Ganguli
Mr. Souvik Ganguli
Assistant Professor, EIED
(Supervisor)
Thapar University, Patiala

S. Ghosh 6/7/12
Dr. Smarajit Ghosh
Professor & Head of Department, EIED
Thapar University, Patiala

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

ACKNOWLEDGEMENT

I would like to express my gratitude and sincere thanks to Mr. Souvik Ganguly, Assistant Professor whose guidance, support and continuous encouragement helped me being motivated towards excellence throughout the course of this work. I wish to extend my gratitude towards Dr. Smarajit Ghosh, Professor and Head, EIED, who has given me an inspiration throughout this work.

I would also thank to all of them who have been helpful and were associated with me directly and indirectly throughout the work.

Finally, no words to thanks my dearest parents whose support and take care me during this workout.

Jitender Kaushal
(801041011)

ABSTRACT

The aim of this thesis is to design a speed controller of a DC motor by selection of PID parameters using Ant Colony Optimization (ACO) and Artificial Bee Colony Algorithm (ABC). These algorithms are come under the category of bio-inspired optimization techniques. The model of a DC motor is considered as a second order system for speed control and third order system for position control. Here, there is a comparison between conventional tuning methods and optimization techniques of parameters for PID controller. In some cases, it was found that the proposed PID parameters adjusted by optimization technique is better than the conventional techniques like a Ziegler-Nicholls' method. These proposed optimization methods could be applied for higher order system also to provide better system performance with minimum errors. It is decided to create an objective function which will evaluate the optimum PID gains based on the controlled systems and overall error.

This tries to explore the potential of using optimization techniques in controllers and their advantages over conventional methods. PID controller is the most widely used controller in the industry applications, need efficient methods to control the different parameters of the DC motor. The conventional approach is not very efficient due to the presence of non-linearity in the system. The output of the conventional PID system has a quite high overshoot and settling time. The main aim this is to apply two ACO and ABC techniques to design and tuning of PID controller to get an output with better dynamic and static performance. The application of ACO and BA to the PID controller imparts it the ability of tuning itself automatically in an on-line process while the application of optimization algorithm to the PID controller makes it to give an optimum output by searching for the best set of solutions for the PID parameters.

ORGANISATION OF THESIS

Chapter 1: It includes the information about the thesis.

Chapter 2: This chapter elaborates the different literature reviews related to the subject.

Chapter 3: Basic block diagram, mathematical model and overall transfer function of the separately excited DC motor.

Chapter 4: This chapter contains basic theory of PID controller and giving the explanation about classical PID tuning methods for open-loop and closed-loop system.

Chapter 5: This shows about the optimal tuning of PID controller using error integrals.

Chapter 6: The basic theory and algorithm has been discussed about the optimal tuning of PID controller using Ant Colony Optimization.

Chapter 7: The basic theory and algorithm has been discussed about the optimal tuning of PID controller using Artificial Bee Colony Algorithm.

Chapter 8: Thesis has been concluded with future scope in this chapter.

TABLE OF CONTENTS

CONTENTS **PAGE NO.**

CONTENTS **PAGE NO.**

CONTENTS **PAGE NO.**

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DC	Direct Current
AC	Alternating Current
PID	Proportional-Integral-Derivative
PI	Proportional-Integral
ACO	Ant Colony Optimization
TSP	Travelling Salesman Problem
QAP	Quadratic Assignment Problem
JSSP	Job-Shop Scheduling Problem
COP	Combinatorial Optimization Problem
BA	Bees Algorithm
ABC	Artificial Bee Colony
ACSA	Ant Colony Search Algorithm
AVR	Automatic Voltage Regulator
SAPF	Shunt Active Power Filter
IAE	Integral Absolute Error
ISE	Integral Squared Error
ITAE	Integral Time Absolute Error
ITSE	Integral Time Squared Error
RBEMCE	Reference Based Error with Minimum Control Effort
Z-N	Ziegler-Nichols
IMC	Internal Model Control
IFT	Iterative Feedback Tuning

PSO	Particle Swarm Optimization
CABC	Chaotic search Artificial Bee Colony
GA	Genetic Algorithm
FOPID	Fractional Order Proportional-Integral-Derivative
ANN	Artificial Neural Network
AMIGO	Approximate M-constrained Integral Gain Optimization
FOPTD	First Order Plus Time Delay
C-C	Cohen-Coon
CHR	Chien-Hrones-Reswick
BN	Employed Bees
SN	Food Source
MCN	Maximum Cycle Number

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter includes a brief introduction about the DC motor model, basic theory and parameters for PID controller and finally different biological inspired optimization techniques i.e. Ant Colony Optimization and Artificial Bee Colony Algorithm. The basic ideas for both optimization techniques depend upon their respective foraging behaviour and the solution can be made as per the plant model.

1.2 DC Motor

DC motor drives are widely used in applications requiring adjustable speed, good speed regulations and frequent starting, braking and reversing. Some important applications are rolling mills, paper mills, mine winders, hoists, machine tools, traction, printing presses, textile mills, excavators and cranes. Fractional horsepower DC motors are widely used as servo motors for positioning and tracking. Although, it is being predicted that AC drives will replace DC drives, however, even today the variable speed applications are dominated by DC drives because of lower cost, reliability and simple control. As per the control of DC motor, there are lot of methods to control the speed and position of the motor. The purpose of a motor speed controller is to take a signal representing the demanded speed and to drive a motor at that speed.

1.3 PID Controller

PID (proportional-integral-derivative) control is one of the earlier control strategies. Its early implementation was in pneumatic devices, followed by vacuum and solid state analog electronics, before arriving at today's digital implementation of microprocessors. It has a simple control structure which was understood by plant operators and which they found relatively easy to tune. Since many control systems using PID control have proved satisfactory, it still has a wide range of applications in industrial control. PID control is a control strategy that has been successfully used over many years. Simplicity, robustness, a wide range of applicability and near-optimal performance are some of the reasons that have made PID controller so popular in the academic and industry sectors. Recently, it has been noticed that PID controllers are often poorly tuned and some efforts have been made to systematically resolve this matter. PID control has been an active research topic for many years; since many process plants controlled by PID controllers have similar dynamics it has been found possible to set satisfactory controller parameters from less plant information than a complete mathematical model. These techniques came about because of the desire to adjust controller parameters with a minimum of effort, and also because of the possible difficulty and poor cost benefit of obtaining mathematical models.

The PID controller calculation (algorithm) involves three separate parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. 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 or the power supply of a heating element. Heuristically, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. By tuning the three constants in the PID controller algorithm, the controller can provide 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 setpoint and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

Controller Response	**Rise Time**	**Overshoot**	**Settling Time**	**Steady State Error**
K_p	Decrease	Increase	Small Change	Decrease
T_i	Decrease	Increase	Increase	Eliminate
T_d	Small Change	Decrease	Decrease	Small Change

Table 1.1: Effect of each controllers K_p, T_i and T_d on a closed-loop system

The PID controller is the most common general purpose controller in the chemical process industry today. It can be used as a stand-alone unit, or it can be part of a distributed computer control system. Over 30 years ago, PID controllers were pneumatic-mechanical devices, whereas nowadays they are implemented in software in electronic controllers. The electronic implementation is much more flexible than the pneumatic devices since the engineer can easily re-program it to change the configuration of things like alarm settings, tuning constants etc. Once we have programmed the PID controller, and have constructed something, either in software or hardware to control, we must tune the controller.

For many industrial process control requirements, only proportional control is unsatisfactory since the offset cannot be tolerated. Consequently the PI controller is probably the most common controller, and is adequate when the dynamics of the process are essentially first or damped second order. PID is satisfactory when the dynamics are second or higher order. However the derivative component can introduce problems if the measured signal is noisy. If the process has a large time delay, the derivative action does not seem to help much. In fact PID control finds it difficult to control processes of this nature, and generally a more sophisticated controller such as a dead time compensator or a predictive controller is required. Processes that are highly under damped with complex conjugate poles close to the imaginary axis are also difficult to control with a PID controller. Processes with this type of dynamic characteristics are rare in the chemical processing industries, although more common in mechanical or robotic systems comprising of flexible structures.

Proportional Action:

In the case of pure proportional control, the control law of Equation (1), the control action is simply proportional to the control error. The variable u_b is a bias or a reset. When the control error e is zero, the control variable takes the value $u(t)' = u_b$. Bias u_b is often fixed to ("u_{max} + "u_{min})/2, but can sometimes be adjusted manually so that the stationary control error is zero at a given setpoint.

$$u(t) = Ke(t) + u_b \qquad \text{... (1)}$$

Integral Action:

The main function of the integral action is to make sure that the process output agrees with the setpoint in steady state. With proportional control, there is normally a control error in steady state. With integral action, a small positive error will always lead to an increasing control signal, and a negative error will give a decreasing control signal no matter how small the error is. The following simple argument shows that the steady-state error will always be zero with integral action. Assume that the system is in steady state with a constant control signal (u_0) and a constant error (e_o). The control signal is then clearly contradicts the assumption that the control signal u_0 is constant. A controller with integral action will always give zero steady-state error.

Derivative Action:

The purpose of the derivative action is to improve the closed-loop stability. The instability mechanism can be described intuitively as follows. Because of the process dynamics, it will take some time before a change in the control variable is noticeable in the process output. Thus, the control system will be late in correcting for an error. The action of a controller with proportional and derivative action may be interpreted as if the control is made proportional to the predicted process output, where the prediction is made by extrapolating the error by the tangent to the error curve. The basic structure of a PID controller is A Taylor series expansion. The control signal is thus proportional to an estimate of the control error, where the estimate is obtained by linear extrapolation.

1.4 Ant Colony Optimization

Ant Colony Optimization (ACO) is an evolutionary meta-heuristic algorithm based on the collective behavior emerging from the interaction of the different search threads that has proved effective in solving combinatorial optimization problems. The gains of the controller are tuned by trial and error method based on the experience and plant behavior. The ACO algorithm is used to optimize the gains and the values are applied into the controller of the plant. The objective of this algorithm is to optimize the gains of the PID controller for the given plant. The proportional gain makes the controller respond to the error while the integral derivative gain help to eliminate steady state error and prevent overshoot respectively. ACO is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 and, since then, many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a information about the structure of previously obtained good solutions. Metaheuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic: either a constructive heuristic starting from a null solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one. ACO is founded on the foraging behaviour of ants and their indirect communication based on pheromones, and has been applied to several combinatorial problems.

The ACO was inspired from natural behavior of the ant colonies on how they find the food source and bring them back to their nest by building the unique trail formation. In the ACO algorithm for a d-variabled optimization problem, a population of ant are put into the d-dimensional search space with randomly chosen positions knowing their best values so far and the position in the d-dimensional space. The distance of each city, adjusted according to its own level of pheromone and the other ant’s level of pheromone. The Conventional fixed gain PID controller is well known technique for industrial control process. The design of this controller requires the three main parameters, Proportional gain (K_p), Integral time constant (T_i) and derivative time constant (T_d). The gains of the controller are tuned by trial and error method based on the experience and plant behavior. In proposed ACO-PID controller, ACO algorithm is used to optimize the gains and the values are applied into the controller of the plant. The objective of this algorithm is to optimize the gains of the PID controller for the given plant. The proportional gain makes the controller respond to the error while the integral derivative gain help to eliminate steady state error and prevent overshoot respectively.

The main idea of ACO is to model the problem as the search for a minimum cost path in a graph that base the evolutionary meta-heuristic algorithm. The behavior of artificial ants is inspired from real ants. They lay pheromone trails and choose their path using transition probability. Ants prefer to move to nodes which are connected by short edges with a high among of pheromone. The algorithm has solved travelling salesman problem (TSP), quadratic assignment problem (QAP) and job-shop scheduling problem (JSSP) and so on.

Ant algorithm is a new nature-inspired optimization technique used especially in combinatorial optimization problems (COP). In ant algorithm, there is an iterative process in which a population of simple agents (ants) repeatedly creates candidate solutions of the given problem. There are two mechanisms in probabilistically guided solution creation process. These are heuristic information on the given problem and memory containing experience gathered by ants in the previous iterations (the pheromone trails). The communication between the ants is mediated by the deposition of pheromone to the elements of good solutions. Then the elements with a higher quantity of pheromone become more attractive for the other ants. The quantity of pheromone deposited on each element is a function of the quality of the solution.

1.5 Artificial Bee Colony Algorithm

The bees algorithm (BA) is a population-based search algorithm first developed in 2005. It mimics the food foraging behaviour of swarms of honey bees. In its basic version, the algorithm performs a kind of neighbourhood search combined with random search and can be used for both combinatorial optimization and functional optimisation. The BA is an optimisation algorithm inspired by the natural foraging behaviour of honey bees to find the optimal solution. The algorithm requires a number of parameters to be set, namely: number of scout bees, number of sites selected out of visited sites, number of best sites out of selected sites, number of bees recruited for best sites, number of bees recruited for the other selected sites, initial size of patches which includes site and its neighbourhood and stopping criterion. Bee-based algorithms are novel methods in engineering optimization and there are few works in the field of control and electrical engineering. The Artificial Bee Colony (ABC) algorithm is employed to optimize the given process model. Natural behavior of bees and their collective activities in their hives has been fascinating researchers for centuries. Recently, the studies focused on swarm intelligence followed by developing swarm optimization methods have unbelievably extended our knowledge about animal societies especially insect colonies. Ants and bees are the most important social insects inspiring efficient problem-solving algorithms.

Similar to other nature-based algorithms, ABC models honey bees but not necessarily precisely. In this model, the honey bees are categorized as employed, onlooker and scout. An employed bee is a forager associated with a certain food source which she is currently exploiting. She memorizes the quality of the food source and then after returning to the hive, shares it with other bees waiting there via a peculiar communication called waggle dance. An onlooker bee is an unemployed bee at the hive which tries to find a new food source using the information provided by employed bees. A scout, ignoring the other’s information, searches around the hive randomly. In nature, the recruitment of unemployed bees happens in a nearly similar way. In addition, when the quality of a food source is below a certain level, it will be abandoned to make the bees explore for new food sources.

In the last decades, design engineers have concentrated on evolutionary based approaches to improve the existing design theories and find the best design results to tune the parameters of PID controllers. The properly selection of the PID parameters is so important that the closed loop system must meets design requirements. The design requirements can include minimum overshoot, rise time, steady state error and settling time in the step response of the closed loop system. The Artificial Bee Colony (ABC) and Bees algorithms (BA) have been used successfully to solve many problems and applied to constrained and unconstrained single objective function optimizations. In this work, the ABC and the BA were employed to optimize the parameters of PID controllers. To indicate the effectiveness and efficiency of the proposed optimization methods, the step responses of closed loop systems were compared with the ABC and the BA.

In the ABC algorithm, the collective intelligence searching model of artificial bee colony consists of three essential components: employed, unemployed foraging bees, and food sources. The employed and unemployed bees search for the rich food sources, which close to the bee's hive. The employed bees store the food source information and share the information with onlooker bees. The number of employed bees is equal to the number of food sources and also equal to the amount of onlooker bees. Employed bees whose solutions cannot be improved through a predetermined number of trials, specified by the user of the ABC algorithm and called “limit”, become scouts and their solutions are abandoned. The model also defines two leading modes of behavior which are necessary for self-organizing and collective intelligence: recruitment of foragers to rich food sources resulting in positive feedback and abandonment of poor sources by scout causing negative feedback.

1.6 Conclusion

Brief descriptions about DC motor model, PID tuning control, Ant Colony Optimization and Artificial Bee Colony Algorithm are thus discussed in this chapter. The operating principle, construction and working of the dc motor and different parameters of the controller are also elucidated. By tuning the three constants in the PID controller algorithm, the controller can provide 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 setpoint and the degree of system oscillation.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the past decades, control theory has gone through major developments. Advanced and intelligent control algorithms have been developed over the years. However, the PID-type controller remains the most popular in industry; studies even indicate that approximately 90% of all industrial controllers are of the PID-type. Reasons for this are the simplicity of this control law and the few tuning parameters. Hundred of tools, methods and theories are available for this purpose. However, finding appropriate parameters for the PID controller is still a difficult task, so in practice control engineers still often use trial and error for the tuning process. This literature overview gives an impression of a number of the available methods for PID control design and discusses the advantages, disadvantages and their applicability.

2.2 Literature Review

N. Navidi *et al.* presented an ant colony search algorithm (ACSA) method for determining the optimal PID controller parameters for speed control of a linear brushless DC motor. The proposed approach has superior features, including easy implementation, stable convergence characteristic and good computation efficiency. The proposed method was more efficient in improving the step response characteristics such as reducing the steady-state error, rise time, settling time and maximum overshoot in speed control of a linear brushless DC motor [1].

A. Soundarrajan *et al.* described PID tuning for AVR in autonomous power generating systems. The proposed method overcomes the drawbacks of conventional fixed gain controller and improvement of settling time, oscillations and overshoot. The designed controller adapt themselves to varying loads and provide better performance as compared to conventional PID, fuzzy, Genetic Algorithm and Particle Swarm Optimization. The proposed ACO-PID controller provided a satisfactory stability between frequency overshoot and transient oscillations with zero steady state error [2].

An efficient algorithm to tuning PI-controller parameters for shunt active power filter using ACO were developed by Brahim Berbaoui *et al.* The proposed ACO-PI controller compensates currents to shunt active power filter (SAPF) under balanced voltages conditions which is applied to eliminate line current harmonics and compensate reactive power. This approach was not only easy to implanted but also very effective in reducing the unwanted harmonics and compensating reactive power [3].

Huseyin Atakan Varol and Zafer Bingul employed minimization of integral absolute error (IAE), integral squared error (ISE) and a new proposed cost function called reference based error with minimum control effort (RBEMCE) and results were compared with Ziegler-Nichols (ZN), Integral Model Control (IMC) and Iterative Feedback Tuning (IFT) methods. From this method, a faster settling time, less or no overshoot and higher robustness were achieved. ANT-RBEMCE PID tuning process: having no overshoot even if the system is perturbed in several ways and having noise rejection even if very high noise variance exists [4].

Ying-Tung Hsiao and Cheng-Long Chuang developed an approach to design PID controller, to obtain good load disturbance response by minimizing the integral absolute control error and minimizing the overshoot, settling time and rise time of step response. The algorithm based on ACO technique to determine the parameters of the PID controller. The feature of presented technique is that it can be implemented quite easily, it allows a more flexible problem formulation and allows to find a global optimum solution for the problem of the design PID controllers [5].

Hong He *et al.* were described the conventional techniques are fairly time consuming and PID controller have not self-adaptive ability and can only depend on artificial optimization parameters. PID controller based on ant colony algorithm is effective and feasible, simple, robust, easy-to-parallel and is a highly efficient method of optimization. ACO not only improved the quality of control system design but also designed to reduce the degree of difficulty [6].

Duan Hai-bin *et al.* developed a novel approach to nonlinear PID parameter optimization using ACO algorithm. To optimize the parameters nonlinear PID controller, an objective function based on position tracing error was constructed and elitist strategy adopted in the improved ACO algorithm. Under this, it has high precision of control and quick response. It also possesses good control and robust performance and can be used to control different kinds of object and process [7].

Dr. S. M. GiriRajkumar *et al.* developed an ACO algorithm to optimize the parameters in the design of a PID controller for a highly nonlinear conical tank system. The proposed approach discussed about the ACO and its application over the parameter tuning of a PI controller in a real time process. The various results presented prove the betterness of the ACO tuned PI setting than IMC tuned ones. ACO presented advantages to a designer by operating with a reduced number of design methods to establish the type of the controller, configuring the dynamic behaviour of the control system with ease, design with a reduced amount information about the controller under the problem of designing a control system for a plant of a first-order system with a time delay [8].

Mahdi Abachizadeh *et al.* developed an efficient method based on Artificial Bee Colony (ABC) metaheuristic for PID controller tuning. The performance indices and time delays were controlled by PID controller with optimum gains. The study clearly demonstrate that the employed method has outperformed better than other techniques such as fuzzy and GA with minimum error, overshoot and settling time. [9].

Ozden ERCIN and Ramazan COBAN presented the performance of the Artificial Bee Colony (ABC) and the Bees Algorithm (BA) and were compared for proportional-integral-derivative (PID) controller tuning. The results obtained show good stability, set-point tracking performance and robustness. However, performance of the ABC is better than BA for PID controller tuning. To indicate the effectiveness and efficiency of the proposed optimization methods, the step responses of closed loop systems were compared with the ABC and the BA. Although the ABC consists of less control parameters, it has a better tuning performance than the BA which consists of many control parameters and also ABC is faster than BA [10].

H. Gozde, *et al.* studied a new artificial intelligence based optimization method to optimize the gains of PID controller for Automatic Voltage Regulator (AVR) system. The dynamic performance of the controller which was optimized by ABC algorithm was compared with Particle Swarm Optimization (PSO). The maximum overshoots and the settling times of the control system which was optimized with ABC algorithm are as small as about 50% of PSO algorithm. The study explained that ABC algorithm showed better performance than the other population based optimization algorithm [11].

Gaowei YAN and Chuangqin LI presented an effective artificial bee colony optimization algorithm based on chaotic search and application for PID control tuning. The chaotic local search method was applied to solve the accuracy problem of global optimal value. The convergence issue of the artificial bee colony algorithm has been improved by increasing the number of scout and rational using the global optimal value and chaotic search. The chaotic search ABC (CABC) algorithm has a great adaptability to the actual control system and PID control system based on CABC can realize the optimization and tuning of parameters. CABC algorithm has a higher accuracy and the PID control system with optimal parameters can obtain the quality that low rise time, overshoot and quickly achieve the steady state [12].

K. Sundareswaran developed a DC motor speed controller design using a colony of honey bees. An optimization model for the controller design for speed control of DC motor was developed and solution was sought through bees foraging algorithm. The foraging behavior of a colony of bees develop an optimization algorithm and same was used to design closed loop speed controller for variable speed DC motor drive with its PID parameters. The proposed method was compared with traditional and GA based controller designs and it studied that the new bees optimization is superior and provides excellent dynamic response [13].

Fei Gao *et al.* were implemented an novel optimal PID tuning and on-line tuning based on artificial bee colony algorithm. An ABC algorithm approach was introduced in PID tuning and on-line tuning as a novel technique for optimum adaptive control in a non-Lyapunov way. The PID parameters were converted to a series of multi-modal non-negative functions' minimization and whose minimum values were optimally determined by ABC. At the end, ABC is efficient and robust for PID control tuning and tuning on-line [14].

Anguluri Rajasekhar *et al.* were proposed a new technique for designing feedback control of a DC motor speed using fractional order proportional-integral-derivative (FOPID) controller. The controller was formulated as a single objective optimization problem and based on integral time absolute error (ITAE) criterion; finally, a modified ABC algorithm was used to tune the FOPID controller parameters. The proposed approach was demonstrated by comparing it with conventional and the ABC algorithm [15].

2.3 Conclusion

This chapter gives an idea about the ACO and ABC based PID tuning methods as found in literature. Many methods have been presented based on the work and use and use very simple process models to derive tuning rules. These methods are very easy to use and do not require extensive knowledge of the process. In the forthcoming chapter, dc motor drive modeling has been carried out.

CHAPTER 3

DC MOTOR

3.1 Introduction

As per the control of DC motor, there are lot of methods to control the speed and position of the motor. The purpose of a motor speed controller is to take a signal representing the demanded speed and to drive a motor at that speed. The controller may or may not actually measure the speed of motor. If it does, it is called a Feedback Speed Controller of Closed Loop Speed Controller, if not it is called Open Loop Speed Controller. Feedback speed controller is better, but it is more complicated.

3.2 DC Motor Model

In armature control of separately excited DC motors, the voltage applied to the armature of the motor is adjusted without changing the voltage applied to the field. Figure 3.1 shows a DC motor equivalent model.

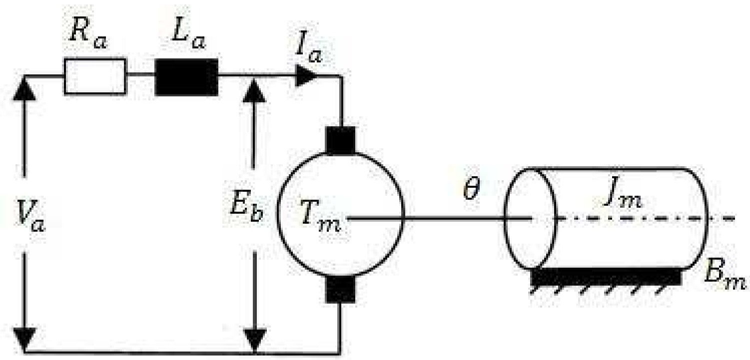

Figure 3.1: DC Motor Model

Some useful relations are $$v_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t) \quad ... (2)$$

$$e_b(t) = K_b \omega(t) \quad ... (3)$$

$$T_m(t) = K_t i_a(t) \quad ... (4)$$

$$T_m(t) - T_L(t) = J_m \frac{d\omega(t)}{dt} + B_m \omega(t) \quad ... (5)$$

where V_a = armature voltage (V), R_a = armature resistance (Ω), L_a = armature inductance (H), I_a = armature current (A), E_b = Back emf (V), ω = angular speed (rad/sec), T_m = motor torque (Nm), T_L = load torque (Nm), θ = angular position of rotor shaft (rad), J_m = rotor inertia (kgm^2), B_m = viscous friction coefficient (Nms/rad), K_t = torque constant (Nm/A), K_b = Back emf constant (V_s/rad) [16].

Figure 3.2 showing the basic block diagram of DC motor model including their transfer functions. V_a is the input supply, T_L is load torque and ω is angular speed.

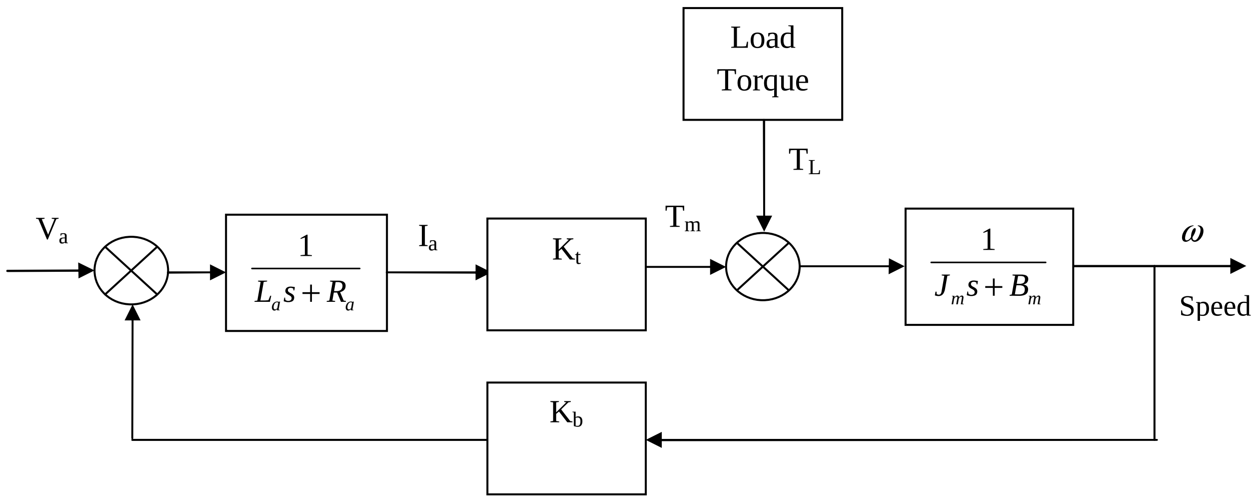

Figure 3.2: Basic block diagram of DC Motor Model

3.3 Speed Control of DC Motor

Substitute (3) in (2) and (4) in (5), we get

$$v_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K_b \omega(t) \quad \text{... (6)}$$

$$K_t i_a(t) = J_m \frac{d\omega(t)}{dt} + B_m \omega(t) \quad \text{... (7)}$$

Taking Laplace transform of equations (5) and (6),

$$V_a(s) = R_a i_a(s) + sL_a I_a(s) + K_b \omega(s) \quad \text{... (8)}$$

$$K_t I_a(s) = sJ_m \omega(s) + B_m \omega(s) \quad \text{... (9)}$$

There are two possible conditions:

When $T_L = 0$

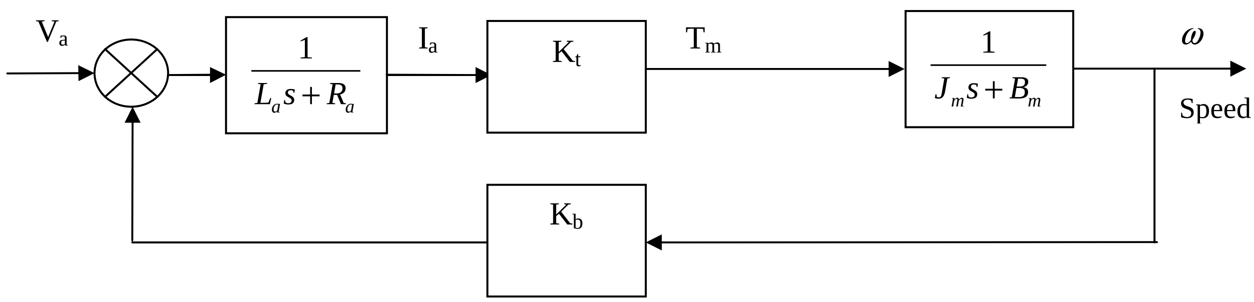

Figure 3.3: Block diagram of DC Motor Model when T_L = 0

Figure 3.3 shows that the DC motor is running under no-load condition (ideal) i.e. $T_L = 0$. Now find the transfer function of ω(s) with respect to $V_a(s)$.

So, the relation between motor speed and applied voltage is given by the transfer function,

$$\frac{\omega(s)}{V_a(s)} = \frac{K_t}{L_a J_m s^2 + (R_a J_m + L_a B_m)s + (R_a B_m + K_b K_t)} \quad \text{... (10)}$$

and when $V_a = 0$

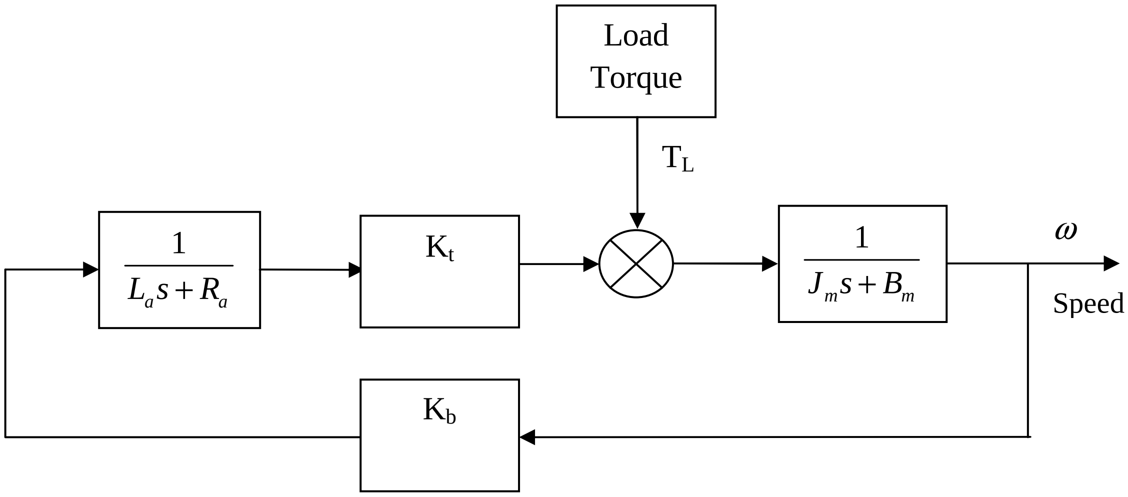

Figure 3.4: Block diagram of DC Motor Model when $V_a = 0$

Figure 3.4 shows the DC motor model when supply voltage (V_a) is 0 and the transfer function of ω(s) is with respect to $T_L(s)$.

Here, the relation between motor speed and load torque is given by the transfer function,

$$\frac{\omega(s)}{T_L(s)} = \frac{-(L_a s + R_a)}{L_a J_m s^2 + (R_a J_m + L_a B_m)s + (R_a B_m + K_b K_t)} \quad \text{... (11)}$$

3.4 Position Control of DC Motor

When $T_L = 0$

For a closed-loop system, the relation between position and speed is given as,

$$\theta(s) = \frac{1}{s}\omega(s) \quad \text{... (12)}$$

$$\frac{\theta(s)}{V_a(s)} = \frac{K_t}{L_a J_m s^3 + (R_a J_m + L_a B_m)s^2 + (R_a B_m + K_b K_t)s} \quad \text{... (13)}$$

and when $V_a = 0$

For a closed-loop system, the relation between position and load torque is given as,

$$\frac{\theta(s)}{T_L(s)} = \frac{-(L_a s + R_a)}{L_a J_m s^3 + (R_a J_m + L_a B_m)s^2 + (R_a B_m + K_b K_t)s} \quad \text{... (14)}$$

3.5 Conclusion

Due to its excellent speed control characteristics, the DC motor has been widely used in industry even though its maintenance costs are higher than the induction motor. As a result, position control of DC motor has attracted considerable research and several methods have evolved. Proportional-Integral Derivative (PID) controllers have been widely used for speed and position control of DC motor. In the next chapter only speed control of dc motor drive has been considered. Few classical tuning techniques has been applied to test the performance of the drive system.

CHAPTER 4

CLASSICAL PID TUNING METHODS

4.1 Introduction

The PID controller is the most common general purpose controller in the today's industries. It can be used as a single unit or it can be a part of a distributed computer control system. Over 30 years ago, PID controllers were pneumatic-mechanical devices, whereas nowadays they are implemented in software based techniques like ANN, Fuzzy Logic, Genetic Algorithm and most popular Optimization techniques.

After implementing the PID controller, now we have to tune the controller; and there are different approaches to tune the PID parameters like P, I and D. The Proportional (P) part is responsible for following the desired set-point while the Integral (I) and Derivative (D) part account for the accumulation of past errors and the rate of change of error in the process or plant, respectively.

4.2 PID Control

PID controller consists of three types of control i.e. Proportional, Integral and Derivative control [17].

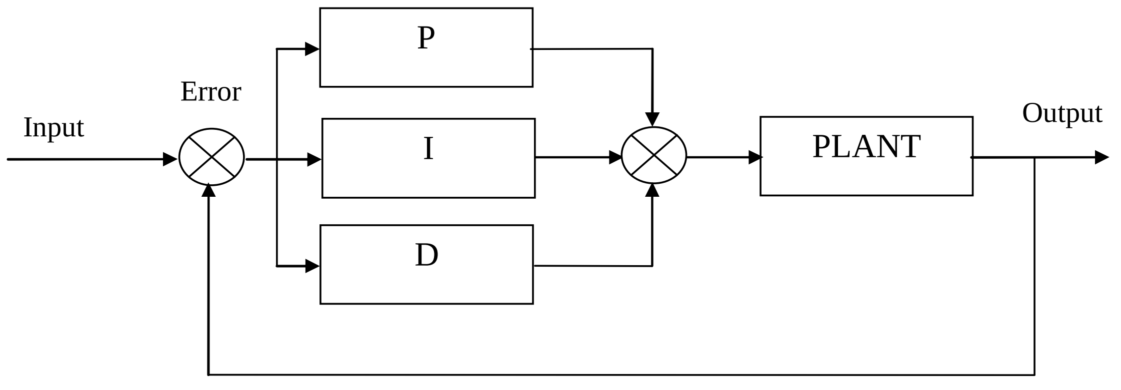

Figure 4.1 Schematic of PID Controller

4.2.1 Proportional Control (P)

The proportional controller output uses a 'proportion' of the system error to control the system. However, this introduces an offset error into the system or plant [17].

$$P_{term} = K_p \times Error \quad \text{... (15)}$$

4.2.2 Integral Control (I)

The integral controller output is proportional to the amount of time; there is an error present in the system. The integral action removes the offset introduced by the proportional control but introduces a phase lag into the system [17].

$$I_{term} = K_i \times \int Errordt \quad \text{... (16)}$$

4.2.3 Derivative Control (D)

The derivative controller output is proportional to the rate of change of the error. Derivative control is used to reduce or eliminate overshoot and introduces a phase lead action that removes the phase lag introduced by the integral action [17].

$$D_{term} = K_D \times \frac{d(Error)}{dt} \quad \text{... (17)}$$

4.2.4 Standard Transfer Function

The system transfer functions in continuous s-domain are given as

For $P = K_p$, $I = K_i/s$ and $D = K_d s$

$$\therefore \quad G_c(s) = P + I + D = K_p + \frac{K_i}{s} + K_d s \quad \text{... (18)}$$

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad \text{... (19)}$$

where K_p is the proportional gain, K_i is the integration coefficient and K_d is the derivative coefficient.

T_i is known as the integral action time or reset time and T_d is the derivative action time or rate time [17].

4.3 Open Loop Tuning Methods

4.3.1 Introduction

Open loop tuning methods are where the feedback controller is disconnected and the experimenter excites the plant and measures the response. The key point here is that since the controller is now disconnected the plant is clearly now no longer strictly under control. If the loop is critical, then this test could be hazardous. Indeed if the process is open-loop unstable, then we will be in trouble before we begin. Notwithstanding for many process control applications, open loop type experiments are usually quick to perform, and deliver informative results. If the system is steady at setpoint, and remains so, then we have no information about how the process behaves.

There are various tuning strategies based on an open-loop step response. While they all follow the same basic idea, they differ in slightly in how they extract the model parameters from the recorded response, and also differ slightly as to relate appropriate tuning constants to the model parameters. There are four different methods, the classic Ziegler-Nichols open loop test, the Cohen-Coon test, Internal Model Control (IMC) and Approximate M-constrained Integral Gain Optimization (AMIGO). Naturally if the response is not sigmoidal or 'S' shaped and exhibits overshoot, or an integrator, then this tuning method is not applicable.

This method implicitly assumes the plant can be adequately approximated by a first order transfer function with time delay,

$$G_p = \frac{Ke^{-\theta s}}{Ts+1} \quad \text{... (20)}$$

where K is gain, θ is the dead time or time delay, and T is the open loop process time constant. Once we have recorded the open loop input/output data, and subsequently measured the times T and θ, the PID tuning parameters can be obtained directly from the given tables for different classical methods.

Figure 4.2 Block diagram of plant with variable output

The method is based on computing the times t_1 and t_2 at which the 35.3% and 85.3% of the system response is obtained as shown in the figure:

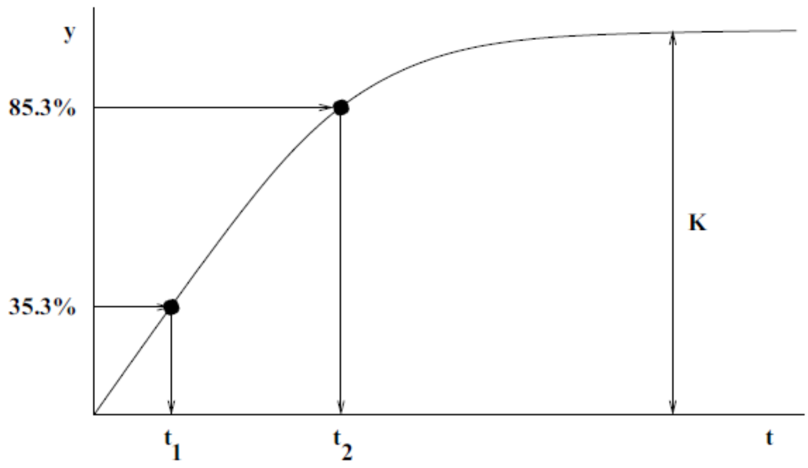

Figure 4.3 System response for first order time delay transfer function

After computing the t_1 and t_2 times, the time delay (θ) and process time constant (T) can be obtained from the following equations:

$$\theta = 1.3t_1 - 0.29t_2$$
$$T = 0.67(t_2 - t_1) \quad \ldots (21)$$

4.3.2 Ziegler-Nichols Tuning Method

The PID tuning parameters as a function of the open loop model parameters K, T and θ from equation (20) as derived by Ziegler-Nichols.

They often form the basis for tuning procedures used by controller manufacturers and process industry. The methods are based on determination of some features of process dynamics. The controller parameters are then expressed in terms of the features by simple formulas. The method presented by Ziegler and Nichols is based on a registration of the open-loop step response of the system, which is characterized by two parameters. first determined, and the tangent at this point is drawn. The intersections between the tangent and the coordinate axes give the parameters T and θ. A model of the process to be controlled was derived from these parameters. This corresponds to modelling a process by an integrator and a time delay. Ziegler and Nichols have given PID parameters directly as functions of T and θ. The behaviour of the controller is as can be expected. The decay ratio for the step response is close to one quarter. It is smaller for the load disturbance. The overshoot in the setpoint response is too large.

Controller		K_p	T_i	T_d
Ziegler-Nichols Method (Open loop)	P	$\frac{T}{K\theta}$	-	-
	PI	$\frac{0.9T}{K\theta}$	$\frac{\theta}{0.3}$	-
	PID	$\frac{1.2T}{K\theta}$	2θ	0.5θ

Table 4.1 Ziegler-Nichols open loop method

4.3.3 Cohen-Coon Tuning Method

Cohen and Coon based the controller settings on the three parameters θ, T and K of the open loop step response. The main design criterion is rejection of load disturbances. The method attempts to position closed loop poles such that a quarter decay ration is achieved.

The PID tuning parameters as a function of the open loop model parameters K, T and θ from equation (20) as derived by Cohen-Coon:

Controller		K_p	T_i	T_d
Cohen-Coon Method (Open loop)	P	$\frac{1}{K}\frac{T}{\theta}\left(1+\frac{\theta}{3T}\right)$	-	-
	PI	$\frac{1}{K}\frac{T}{\theta}\left(0.9+\frac{\theta}{12T}\right)$	$\theta\left(\frac{30+3\theta/T}{9+20\theta/T}\right)$	-
	PID	$\frac{1}{K}\frac{T}{\theta}\left(\frac{4}{3}+\frac{\theta}{4T}\right)$	$\theta\left(\frac{32+6\theta/T}{13+8\theta/T}\right)$	$\theta\left(\frac{4}{11+2\theta/T}\right)$

Table 4.2 Cohen-Coon open loop method

Although, one more parameter is used in this method, the results are not much better than the Ziegler-Nichols settings, mainly because of the decay ration being too small, leading to low damped closed-loop systems

4.3.4 Internal Model Control (IMC) Tuning method

The objective of the internal model control (IMC) tuning rule (Morari and Zafiriou, 1989) is to match the control performance of the PID controller with that of the IMC controller. The FOPTD model can approximate the usual overdamped processes. The model can be obtained by various process identification methods. The IMC tuning rule determines the tuning parameters using the formulas, where $\lambda \geq 0.25\theta$ for PID controller and $\lambda \geq 1.7\theta$ for PI controller. If a smaller value of λ is chosen, then a faster closed-loop response is obtained. However, too small a λ value results in an oscillatory or unstable closed-loop response. If the model is accurate, then the tuning parameters with $\lambda \geq 0.25\theta$ show good control performances and robustness for the step setpoint change.

The IMC tuning rule shows excellent control performances for a step setpoint change. Meanwhile, it shows sluggish control performances for step input disturbance rejection. Here, the step input disturbance is the step-type disturbance added to the process input. The FOPTD model has a structural limitation in representing underdamped or high-order processes. Thus, the IMC tuning rule based on the FOPTD model shows poor control performances for unusual processes, such as underdamped or high-order processes.

The PID tuning parameters as a function of the open loop model parameters K, T and θ from equation (19) as derived by Internal Model Control (IMC):

Controller		K_p	T_i	T_d
IMC Method (Open loop)	PI	$\frac{2T+\theta}{2\lambda}$	$T+\frac{\theta}{2}$	-
	PID	$\frac{2T+\theta}{2(\lambda+\theta)}$	$T+\frac{\theta}{2}$	$\frac{T\theta}{2T+\theta}$

Table 4.3 Internal Model Control open loop method

where $\lambda \geq 1.7\theta$ for PI controller

and $\lambda \geq 0.25\theta$ for PID controller

4.3.5 Approximate M-Constrained Integral Gain Optimization (AMIGO) Tuning Method

The PID tuning parameters as a function of the open loop model parameters K, T and θ from equation (19) as derived by Approximate M-constrained Integral Gain Optimization (AMIGO):

Controller		K_p	T_i	T_d
AMIGO Method (Open loop)	PID	$\frac{1}{K}\left(0.2+0.45\frac{T}{\theta}\right)$	$\theta\left(\frac{0.4\theta+0.8T}{\theta+0.1T}\right)$	$\frac{0.5\theta T}{0.3\theta+T}$

Table 4.4 AMIGO open loop method

4.3.6 Chien-Hrones-Reswick Tuning Method

Chien, Hrones and Reswick (CHR) changed the step response method to give better damped closed-loop systems. They proposed to use "quickest response without overshoot" or "quickest response with 20% overshoot" as design criteria. They also made the important observation that tuning for setpoint response or load disturbance response are different. To tune the controller according to the CHR method, the parameters *T* and *θ* of the process model are first determined in the same way as for the Ziegler-Nichols step response method. The controller parameters for the load disturbance response method are then given as functions of these two parameters.

However, when the 0% overshoot design criteria is used, the gain and the derivative time are smaller and the integral time is larger. This means that the proportional actions, the integral action, as well as the derivative action, are smaller. In the setpoint response method, the controller parameters are not only based on *T* and *θ*, but also on the time constant *T*.

$$a = K\,\theta/T \qquad \dots (22)$$

Overshoot		0 %			20 %		
Controller		K_p	T_i	T_d	K_p	T_i	T_d
Chien-Hrones-Reswick Load Disturbance Response Method	P	0.3/a	-	-	0.7/a	-	-
	PI	0.6/a	4 θ	-	0.7/a	2.3 θ	-
	PID	0.95/a	2.4 θ	0.42 θ	1.2/a	2 θ	0.42 θ

Table 4.5 Chien-Hrones-Reswick load disturbance response method

4.3.7 Analyzing the Parameters of Different Classical Methods

Some important test parameters:

DC Motor Model [27]:

Parameters	Value
Rated Power	1 kW
Rated Voltage	500 V DC
Resistance of the stator (R_a)	21.2 Ω
Inductance of the stator (L_a)	0.052 H
Viscous Coefficient (B)	1×10^{-4} kg-ms/rad
Moment of Inertia (J)	1×10^{-5} kg-ms^2/rad
Back emf constant (K_b)	1 volts/rad
Motor Torque constant (K_t)	1 volts/rad
Number of poles pairs	2
Speed of the rotor (N)	3000 RPM
Rotor magnetic flux (Ø)	0.11 Weber

Table 4.6 DC Motor Model Specifications and Parameters

Time domain analyses of DC machine open loop (uncontrolled) are given as:

Rise time, T_r = 0.2197

Settling Time, T_s = 0.3937

Maximum Overshoot, M_p = 0.0

Servo responses for speed control of uncontrolled system are given as:

Rise time, T_r = 0.00084724

Settling Time, T_s = 0.0186

Maximum Overshoot, M_p = 61.9573

After applying FOPTD system the following parameters were found:

K = 468.8835, θ = 0.1891, T = 0.0972

The system first order time delay transfer function is given as:

$$\frac{468.9e^{-0.189s}}{0.09723s+1}$$

Controller	K_p	T_i	T_d	T_r	T_s	M_p
Z-N	0.0013	0.3782	0.0946	1.7090	3.1189	0.0
C-C	0.0020	0.2892	0.0508	0.8989	1.7138	0.0
IMC	0.0017	0.1918	0.0479	0.6412	1.1652	0.0
AMIGO	0.00091998	0.7716	0.0597	4.8168	8.8075	0.0
CHR*	0.00065791	0.0972	0.0946	0.7248	1.1401	0.0

Table 4.7 Open Loop System Responses with Classical PID Tuning Methods

*For set point

It is found that for DC motor model, the robustness of the system has uncertainty and doesn't have frequency domain analysis within permissible limits.

Open loop Response:

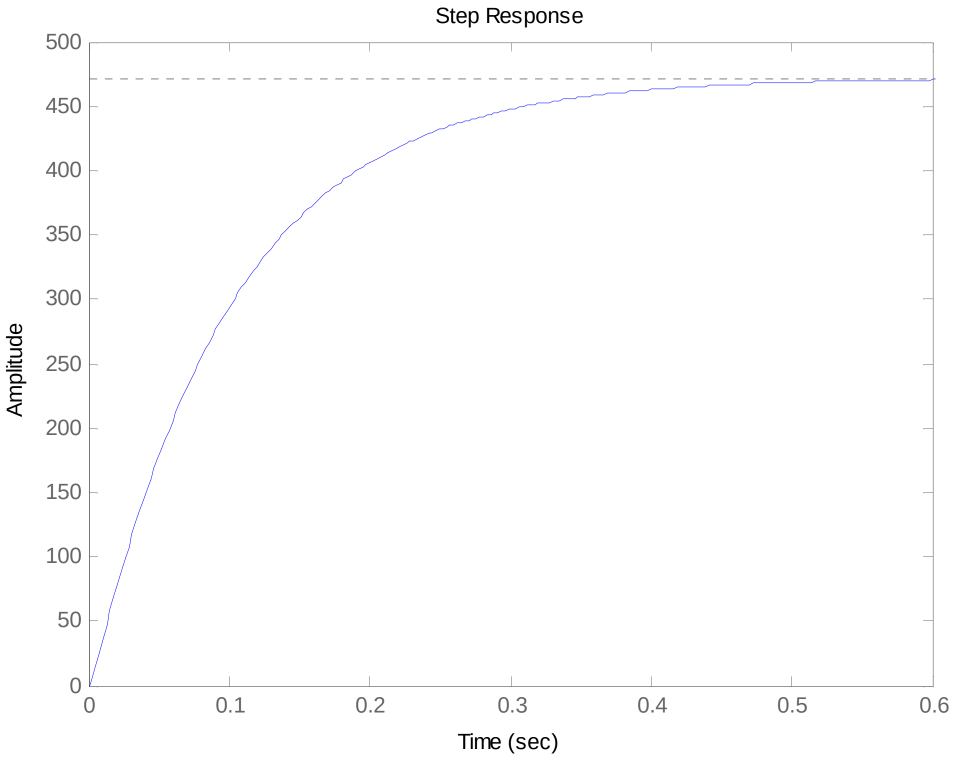

Figure 4.4 DC motor open loop response

Step response of system tuned with open loop PID methods:

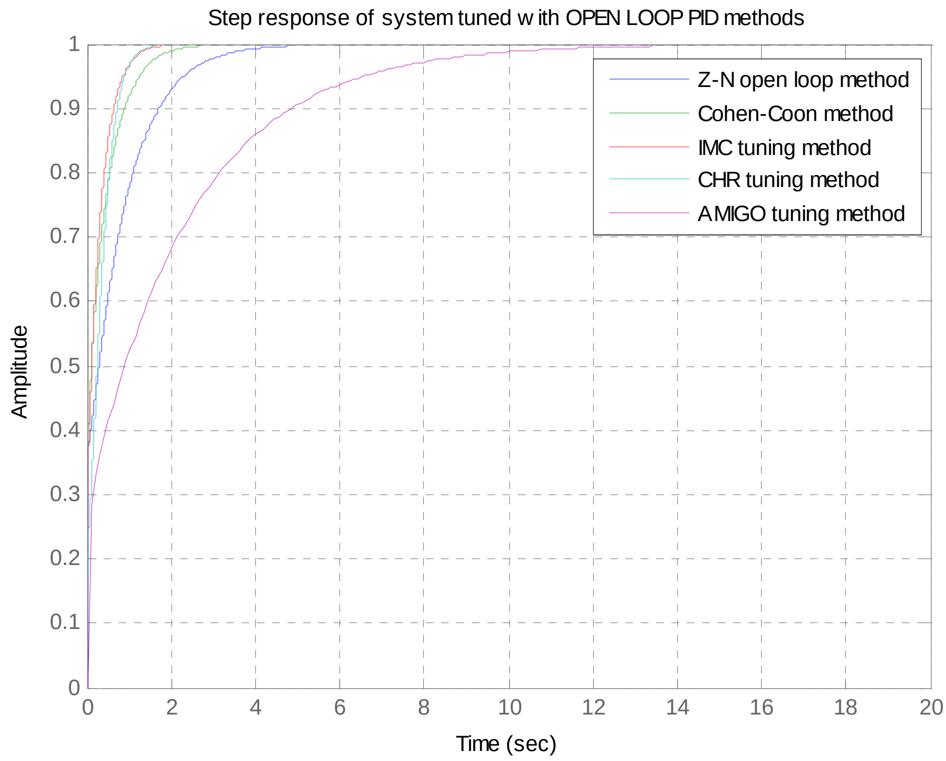

Figure 4.5 DC motor step response of system tuned with open loop PID methods

4.4 Closed Loop Tuning Methods

4.4.1 Ziegler-Nichols Tuning Method

The control system performs poor in characteristics and even it becomes unstable, if improper values of the controller tuning constants are used. So it becomes necessary to tune the controller parameters to achieve good control performance with the proper choice of tuning constants. Controller tuning involves the selection of the best values of K_p, T_i and T_d (if a PID algorithm is being used). This is often a subjective procedure and is certainly process dependent. It is widely accepted method for tuning the PID controller. The method is straightforward. First, set the controller to P mode only. Next, set the gain of the controller (K_p) to a small value. Make a small set point (or load) change and observe the response of the controlled variable. If K_p is low the response should be sluggish. Increase K_p by a factor of two and make another small change in the set point or the load. Keep increasing K_p (by a factor of two) until the response becomes oscillatory. Finally, adjust K_p until a response is obtained that produces continuous oscillations. This is known as the ultimate gain (K_u). Note the period of the oscillations (P_u). The steps required for the method are given below. We have to set the integral and derivative coefficients are zero. Gradually increase the

proportional coefficient from 0 to until the system just begins to oscillate continuously. The proportional coefficient at this point is called the ultimate gain K_u. And the period of oscillation at this point is called ultimate period P_u.
The K_u is the gain margin of the system and;

$$P_u = \frac{2\pi}{\omega cg} \quad \dots (23)$$

where, the ωcg is the gain crossover frequency. Gain margin is the reverse of amplitude ratio.

The Ziegler-Nichols continuous cycling method is one of the best known closed loop tuning strategies. The controller gain is gradually increased (or decreased) until the process output continuously cycles after a small step change or disturbance. At this point, the controller gain is selected as the ultimate gain, K_u, and the observed period of oscillation is the ultimate period, P_u. Ziegler and Nichols originally suggested PID tuning constants as a function of the ultimate gain and ultimate period.

Controller		K_p	T_i	T_d
Ziegler-Nichols Method (Closed loop)	P	$0.5K_u$	-	-
	PI	$0.45K_u$	$P_u/1.2$	-
	PID	$0.6K_u$	$P_u/2$	$P_u/8$

Table 4.8 Ziegler-Nichols Closed Loop Method

4.4.2 Modified Ziegler-Nichols Tuning Method

Controller		K_p	T_i	T_d
Modified Z-N Method (Closed loop) PID Controller	No Overshoot	$0.2K_u$	$P_u/2$	$P_u/2$
	Some Overshoot	$0.33K_u$	$P_u/2$	$P_u/3$

Table 4.9 Modified Ziegler-Nichols Closed Loop Method

4.4.3 Tyreus-Luyben Tuning Method

Controller		K_p	T_i	T_d
Tyreus-Luyben Method (Closed loop)	PI	$0.31K_u$	$2.2P_u$	-
	PID	$0.45K_u$	$2.2P_u$	$P_u/6.3$

Table 4.10 Tyreus-Luyben Closed Loop Method

4.4.4 Astrom-Hagglund Tuning Method

An improvement of the Ziegler-Nichols method is given by Astrom and Hagglund. They propose to use a relay feedback. This nonlinear feedback includes a limit cycle oscillation. The period of this oscillation is T_u and a good estimate for the ultimate gain can be calculated from the oscillation amplitude *a* with:

$$K_u = \frac{4d}{\pi a} \quad \dots (24)$$

The major advantage of using relay feedback is that the system is not driven to instability. Further, it offers the possibility to identify different points on the Nyquist curve which gives more information about the course of the Nyquist plot.

Controller		K_p	T_i	T_d
Astrom-Hagglund Method (Closed loop)	PI	$0.32K_u$	0.94	-

Table 4.11 Astrom-Hagglund Closed Loop Method

4.4.5 Closed loop frequency domain analysis

Frequency domain analyses of DC machine close loop are given as:

Gain margin, G_m = 1.5076

Phase Margin, P_m = 167.3819

Wcg = 1.4386×10^3

Wcp = 92.4975

Controller	$\mathbf{K_p}$	$\mathbf{T_i}$	$\mathbf{T_d}$	$\mathbf{T_r}$	$\mathbf{T_s}$	$\mathbf{M_p}$
Z-N	0.9046	0.0022	0.0005459	0.00083556	0.0092	40.8949
Modified Z-N	0.4975	0.0022	0.0015	0.0011	0.0115	21.1820
Tyreus Luyben	0.6784	0.0096	0.00069325	0.0011	0.0050	21.3540
Astrom Hagglund	0.4824	0.9400	0	0.0013	0.0174	49.2041

Table 4.12 Closed Loop System Responses with Classical PID Tuning Methods

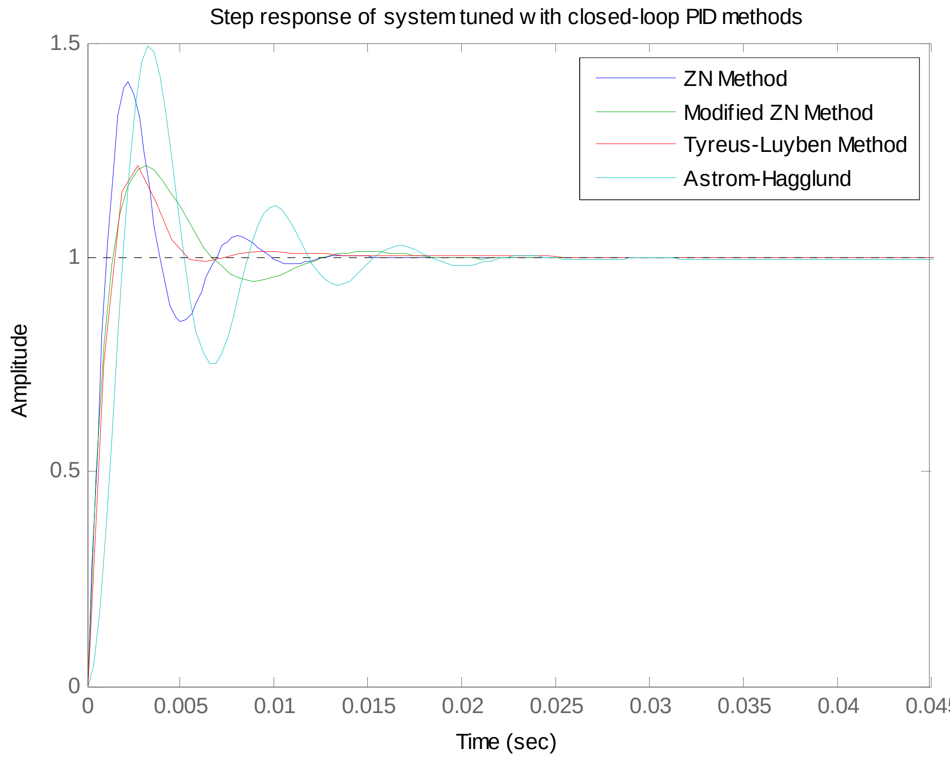

Figure 4.6 DC motor step response of system tuned with closed-loop PID methods

Rise Time(t_r) is defined often as the time required for a response to go from 10 % to 90 % of its desired final value, or as the time interval given by the intersection points of the inflexion tangent with the 0 % and 100 % lines. Maximum Overshoot (M_p) is defined as the maximum value of the response at time t_{max} in relation to its desired final value. It can be considered to be a measure of the relative stability of the system. It increases as the damping ratio decreases. Settling Time (t_s) is the time after which the response remains within a band of ±%ε about the desired final value, where ε is selected between 2 % and 5 %.

4.5 Conclusion

The open loop and closed loop DC motor speed control system are discussed with different PID tuning rules. From the results, it is found that in case of open loop Z-N and AMIGO control system, the rise time and settling time does not come in required range but for rest of the methods, rise time and settling time are coming within the range as per the system used here. For closed loop control, rise time and settling time are found in the required range but maximum overshoot does not fall in required range like in case of Z-N method maximum overshoot is 40.8949 and same case for rest of the conventional methods. Also, the robustness of the system has uncertainty with this system that gain margin, phase margin, delay margin and modulus margin are not coming in required range. In the next chapter, optimal tuning of PID controller using error integrals is discussed.

CHAPTER 5

OPTIMAL TUNING OF PID CONTROLLER USING ERROR INTEGRALS

5.1 Introduction

The starting point for the design of a feedback-control system is to have a good plant model described either in the form of a differential equation or a transfer function. Once the plant model is given, the next step is to design an overall system that meets the described design specification.

It is important to note that different applications may require different specifications. Generally, the performance of feedback-control systems includes two tasks; steady-state performance which specifies accuracy when all the transients are decayed and transient performance which specifies the speed of response.

The performance specifications introduced are appropriate for evaluating the results of a control system design. However, they cannot be used directly as a starting point for designing a controller. Therefore, performance indices based on various functions $f_k[e(t)]$ of the error,

$$e(t) = y(t) - r(t) \quad \dots (25)$$

between the reference input r(t) and the controlled plant output y(t) are preferred. General performance indices covering an error function in [0, ∞) have been introduced as the integral

$$\int_0^\infty f_k[e(t)]dt \quad \dots (26)$$

where $f_k[e(t)]$ can take various forms as shown in table 5.1.

To obtain the PID tuning parameters one usually has to minimize a performance index and some performance indices are given in Table 5.1. If the closed loop systems has a steady-state error e_∞, then *e(t)* must be replaced by

$$e(t) - e_\infty \quad \dots (27)$$

and also *r(t)* being the reference input and *y(t)* the output of the system.

Sr. No.	Performance Index	Characteristic
1.	$I_1 = \int_0^\infty e(t)dt$	*Integral of total error* (ITE): Only appropriate for highly damped monotonic step responses of *e(t)*; simple mathematical treatment.
2.	$I_2 = \int_0^\infty \lvert e(t)\rvert dt$	*Integral of absolute error* (IAE): Appropriate for non-monotonic step responses. Not easy to track analytically.
3.	$I_3 = \int_0^\infty e^2(t)dt$	*Integral of square error* (ISE): Penalizes large errors more heavily than small ones; provides longer t_s as I_2. In many cases analytical tracking is possible.
4.	$I_4 = \int_0^\infty \lvert e(t)\rvert tdt$	*Integral of time multiplied absolute error* (ITAE): Provides similar results as I_2; puts less weight on e(*t)* for *t* small and more for *t* large.
5.	$I_5 = \int_0^\infty e^2(t)tdt$	*Integral of time multiplied square error* (ITSE): Provides similar results as combined with the same time weighting as for I_4.
6.	$I_6 = \int_0^\infty [e^2(t) + \alpha e^2(t)]dt$	Integral of generalized square error (IGSE): Better results as for I_3 are obtained, however, the selection of the weighting factor α is subjective.
7.	$I_7 = \int_0^\infty [e^2(t) + \beta u^2(t)]dt$	*Integral of square error and control effort* (ISECE): Provides a slightly larger e_{max}, but t_s become essentially smaller as for; however, the selection of β is subjective.

Table 5.1 Various Integral Performance Indices

5.2 Murrill Method

This method proposed by P.W. Murrill *et al.* for optimal tuning of PID controller using integral performance indices for Servo and Regulatory response [18].

5.2.1 Servo Response

If the tuning parameters are calculated for a set-point change (i.e. for Servo Response), the integral time will be longer and the derivative time will be shorter, and they will depend mostly on the time constant of the process.

The relationship between the controller settings based on integral criterion and the ratio t_o / τ is expressed by the tuning relationship given in equation (28)

$$Y = A\left(\frac{t_0}{\tau}\right)^B \qquad \text{... (28)}$$

where $Y = KK_c$ for proportional mode, τ / T_i for reset mode; A, B=constants for given controller and mode; t_0, τ =pure delay time and first-order lag time constant respectively.

Using these equations:

$$K_c = \frac{A}{K}\left(\frac{t_0}{\tau}\right)^B \quad \text{... (29)}$$

$$\frac{1}{T_i} = \frac{A}{\tau}\left(\frac{t_0}{\tau}\right)^B \quad \text{... (30)}$$

$$T_d = \tau * A\left(\frac{t_0}{\tau}\right)^B \quad \text{... (31)}$$

a. For PI Controller:

The table for coefficients A and B for PI controller optimal tuning for error indices is given in table below:

ERROR INDICES	**Controller**	**A**	**B**
IAE	P	0.758	-0.861
	I	1.020	-0.323
ITAE	P	0.586	-0.916
	I	1.030	-0.165

Table 5.2 Servo Response of IAE and ITAE with coefficients A and B for PI Controller (Murrill Method)

b. For PID Controller:

The following table shows the value of coefficient A and B for PID controller:

ERROR INDICES	Controller	A	B
IAE	P	1.086	-0.869
	I	0.740	-0.130
	D	0.348	0.914
ITAE	P	0.965	-0.855
	I	0.796	-0.147
	D	0.308	0.929

Table 5.3 Servo Response of IAE and ITAE with coefficients A and B for PID Controller (Murrill Method)

5.2.2 Regulatory Response

With tuning parameters calculated for load rejection (i.e. for Regulatory Response), the integral time (T_i) and derivative time (T_d) will depend mostly on the dead time (t_d) of the process.

a. For PI Controller:

Table below shows value of A and B for PI controller,

ERROR INDICES	Controller	A	B
IAE	P	0.984	-0.986
	I	0.608	-0.707
ITAE	P	0.859	-0.977
	I	0.674	-0.680
ISE	P	1.305	-0.959
	I	0.492	-0.738

Table 5.4 Regulatory Response of IAE, ITAE and ISE with coefficients A and B for PI Controller (Murrill Method)

b. For PID Controller:

Table below shows value of A and B for PID controller,

ERROR INDICES	Controller	A	B
IAE	P	1.435	-0.921
	I	0.878	-0.749
	D	0.482	1.137
ITAE	P	1.357	-0.947
	I	0.842	-0.738
	D	0.381	0.995
ISE	P	1.495	-0.945
	I	1.101	-0.771
	D	0.560	1.006

Table 5.5 Regulatory Response of IAE, ITAE and ISE with coefficients A and B for PID Controller (Murrill Method)

5.3 M. Zhuang Method

This method proposed by M. Zhuang *et al.* is used to obtain optimum PID controller settings for minimizing time weighted integral performance criteria. FOPDT model is considered here for optimization using this method [19].

5.3.1 Servo Response

The FOPDT model transfer function is given by:

$$G(s) = \frac{Ke^{-s\tau}}{Ts+1} \qquad \text{... (32)}$$

The following formulae gives the KK_c, T/T_i, T_d/T as the functions of τ/T.

(Range of τ/T is between 0.1 to 1.0)

$$K_c = \frac{a_1}{K}\left(\frac{\tau}{T}\right)^{b_1} \quad \text{... (33)}$$

$$T_i = \frac{T}{a_2 + b_2(\tau / T)} \quad \text{... (34)}$$

$$T_d = a_3 T\left(\frac{\tau}{T}\right)^{b_3} \quad \text{... (35)}$$

a. For PI Controller:

Parameters	**ISE**	**ISTE**
a_1	0.980	0.712
b_1	-0.892	-0.921
a_2	0.690	0.968
b_2	-0.155	-0.247

Table 5.6 Servo Response of ISE and ISTE with coefficients for PI Controller (M. Zhuang Method)

b. For PID Controller:

Parameters	**ISE**	**ISTE**
a_1	1.048	1.042
b_1	-0.897	-0.897
a_2	1.195	0.987
b_2	-0.368	-0.238
a_3	0.489	0.385
b_3	0.888	0.906

Table 5.7 Servo Response of ISE and ISTE with coefficients for PID Controller (M. Zhuang Method)

5.3.2 Regulatory Response

The tuning formulae for step disturbance input (i.e. regulatory response) are:

$$K_p = \frac{a_1}{K}\left(\frac{\tau}{T}\right)^{b_1} \quad \text{... (36)}$$

$$\frac{1}{T_i} = \frac{a_2}{T}\left(\frac{\tau}{T}\right)^{b_2} \quad \text{... (37)}$$

$$T_d = a_3 T\left(\frac{\tau}{T}\right)^{b_3} \quad \text{... (38)}$$

a. For PI Controller:

The values of coefficients for PI controller are:

Parameters	ISE	ISTE
a_1	1.279	1.015
b_1	-0.945	-0.957
a_2	0.535	0.667
b_2	-0.586	-0.552

Table 5.8 Regulatory Response of ISE and ISTE with coefficients for PI Controller (M. Zhuang Method)

b. For PID Controller:

The values of coefficients for PID controller are:

Parameters	ISE	ISTE
a_1	1.473	1.468
b_1	-0.970	-0.970
a_2	1.115	0.942
b_2	-0.753	-0.725
a_3	0.550	0.443
b_3	0.948	0.939

Table 5.9 Regulatory Response of ISE and ISTE with coefficients for PID Controller (M. Zhuang Method)

5.4 Analyzing the parameters of different optimal PID tuning rules using performance indices

After applying FOPTD system the following parameters are found as:

K = 1.0027, θ = 0.0063, T = 0.00078795

Performance Indices	K_p	T_i	T_d	T_r	T_s	M_p
IAE	0.2117	0.0042	0.0040	0.0016	0.0165	8.6861
ISE	0.5243	1.3986	0.0010	0.0012	0.0049	9.8561
ITAE	0.3038	0.0029	0.00081419	0.0016	0.0206	42.3340
ITSE	0.1957	0.0038	0.0024	0.0020	0.0216	16.4760

Table 5.10 Performance Indices for Closed Loop System

Step response of closed loop system:

Rise time, T_r = 0.00084724

Settling Time, T_s = 0.0186

Maximum Overshoot, M_p = 61.9573

Gain Margin, Gm = 1.0573

Phase Margin, Pm = 142.5792

ωcg = 446.7584

ωcp = 92.4975

Delay Margin = [0.0479   0.00022402]

Modulus Margin = 0.1184

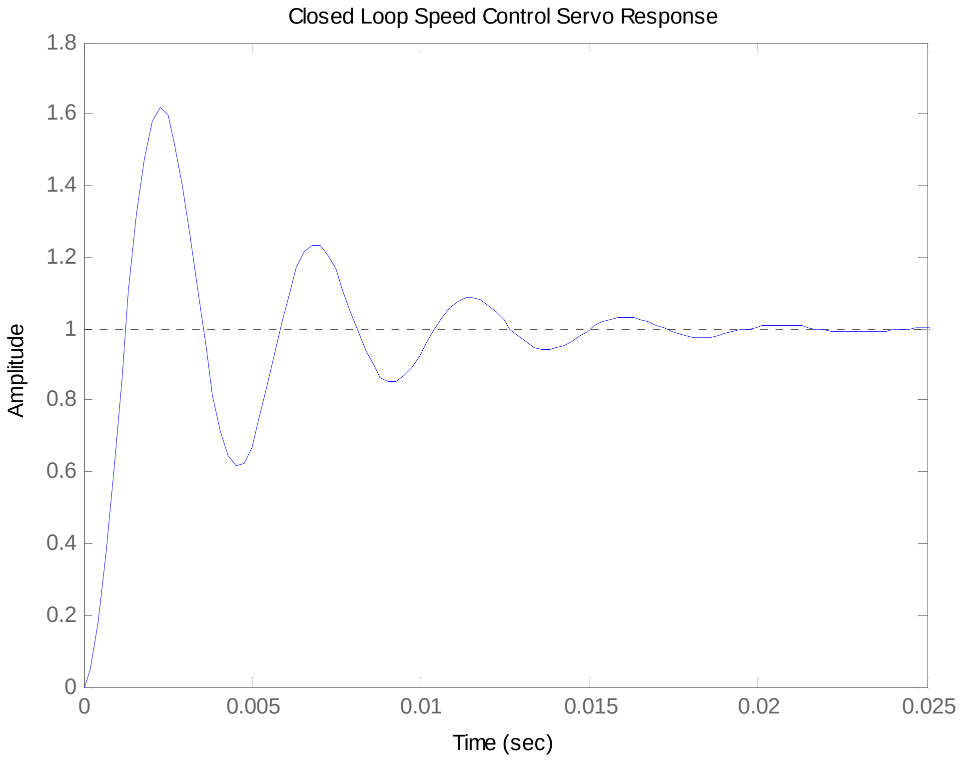

Figure 5.1 Servo response of DC motor closed-loop speed control

Step response of system tuned with closed-loop Error Integrals PID methods:

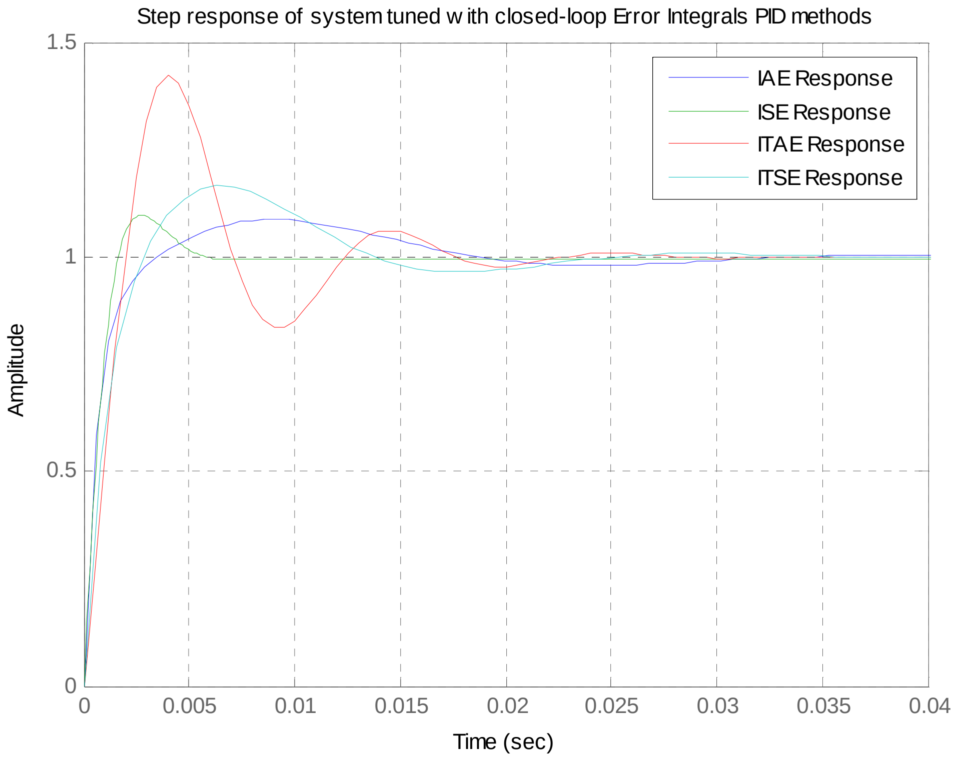

Figure 5.2 DC motor closed-loop speed control with different performance indices

5.5 Conclusion

The different error integrals are discussed like IAE, ISE, ITAE and ITSE with their system response and it is found that the error integrals (performance indices) have better results in terms of rise time, settling time and maximum overshoot as compared to classical approach for PID tuning. The rise time and settling time are coming well but maximum overshoot does not fall in required range. The minimum overshoot is found in case of IAE i.e. 8.6861% and rest of the error indices has large maximum overshoot. In this system, ISE has best results in terms of rise time and settling time. As far as this system, the robustness of the system has uncertainty with this system that gain margin, phase margin, delay margin and modulus margin are not coming in required range. In the next chapter, optimal tuning of PID controller using error integrals is discussed.

In the upcoming chapter, the basic theory, algorithm and optimal tuning of PID controller using ant colony optimization are discussed.

CHAPTER 6

OPTIMAL TUNING OF PID CONTROLLER USING ANT COLONY OPTIMIZATION

6.1 Introduction

Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 and, since then, many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a information about the structure of previously obtained good solutions. Metaheuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic: either a constructive heuristic starting from a null solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one. ACO is founded on the foraging behaviour of ants and their indirect communication based on pheromones, and has been applied to several combinatorial problems such as job scheduling and routing optimization in data [20, 21].

In computer science and operations research, the Bees Algorithm is a population-based search algorithm first developed in 2005. It mimics the food foraging behaviour of swarms of honey bees. The Bees Algorithm (BA) is a novel populace-based search algorithm. The algorithm imitates the food foraging behaviour of honeybees' colony. In its basic version, the algorithm executes a kind of vicinity search combined with random search and can be used for optimization. The Bees Algorithm is an optimisation method inspired by the natural foraging behaviour of honey bees for finding a quality food source. The algorithm starts with scout bees being placed randomly in the search space. The fitness values of the sites visited by the scout bees are then evaluated. A number of bees with the highest fitness values are chosen as “selected bees” and the sites visited by them are chosen for a neighbourhood search. The remaining bees in the population are assigned randomly around the search space looking for new potential solutions. These steps are repeated until a termination criterion is met [21, 22].

According to Karl O. Jones and et. al, by the use of PID controller, BA suggested that it does not provide any improvement over the ACO technique. In terms of the transient response of the controlled system, the decision on ACO offers an improved method related to the performance criteria required from the controlled response: minimum overshoot would suggested by ACO [21].

Fares Sayadi and et. al explored that the proposed algorithm uses less common control parameters: maximum iteration number and population size (i.e., it is a simple and flexible technique).From the results attained, it can be concluded that the intense trade-off between performance and computational complexity on multimodal and multivariable problems than other algorithms considered [22].

B. Nagaraj and et. al suggested that the optimal tuning by using ACO has a better control performance compared with the conventional schemes. Here the techniques are often criticized for two reasons: algorithms are computationally heavy and convergence to the optimal solution cannot be guaranteed. But according to them, computational complexity is not really an issue and thus ACO tuned system has good steady state response and performance indices [23].

6.2 Ant Colony Optimization (ACO) Technique

ACO's are especially suited for finding solutions to different optimization problems. A colony of artificial ants cooperates to find good solutions, which are an emergent property of the ant's co-operative interaction. Based on their similarities with ant colonies in nature, ant algorithms are adaptive and robust and can be applied to different versions of the same problem as well as to different optimization problems. The main traits of artificial ants are taken from their natural model. These main traits are artificial ants exist in colonies of cooperating individuals, they communicate indirectly by depositing pheromone they use a sequence of local moves to find the shortest path from a starting position, to a destination point they apply a stochastic decision policy using local information only to find the best solution. If necessary in order to solve a particular optimization problem, artificial ants have been enriched with some additional capabilities not present in real ants. An ant searches collectively foe a good solution to a given optimization problem. Each individual ant can find a solution or at least part of a solution to the optimization problem on its own but only when many ants work together they can find the optimal solution [23, 25].

Since the optimal solution can only be found through the global cooperation of all the ants in a colony, it is an emergent result of such this cooperation. While searching for a solution the ants do not communicate directly but indirectly by adding pheromone to the environment. Based on the specific problem an ant is given a starting state and moves through a sequence of neighbouring states trying to find the shortest path. It moves based on a stochastic local search policy directed by its internal state, the pheromone trails, and local information encoded in the environment. Ants use this private and public information in order to decide when and where to deposit pheromone. In most application the amount of pheromone deposited is proportional to the quality of the move an ant has made. Thus the more pheromone, the better the solution found. After an ant has found a solution, it dies; i.e.it is deleted from the system [23, 26].

ACO is depending upon the pheromone matrix $\tau = \{\tau_{ij}\}$ for the construction of good solutions. The initial values of τ are

$$\text{set } \tau_{ij} = \tau_0 \forall (i, j) \text{, where } \tau_0 > 0. \quad \ldots (39)$$

The probability $P_{ij}^A(t)$ of choosing a node *j* at node I is defined in the equation (40). At each generation of the algorithm, the ant constructs a complete solution using this equation, starting at source node.

$$P_{ij}^A(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{ij \in T^A} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}; i, j \in T^A \quad \ldots (40)$$

where $\eta_{ij} = \frac{1}{kj}, j = [p, i, d]$ represents the heuristic function.

α and β = constants that determine the relative influence of the pheromone values and the heuristic values on the decision of the ant.

T^A = the path effectuated by the ant A at a given time.

The quantity of pheromone $\Delta\tau_{ij}$ on each path may be defined as

$$\Delta\tau_{ij}^A = \begin{cases} \frac{L^{\min}}{L^A} & \text{if } i, j \in T^A \\ 0 & \end{cases} \quad \ldots (41)$$

else

where L^A = the value of the objective function found by the ant A

$L^{\min}$ = the best solution carried out by the set of the ants until the current iteration.

The pheromone evaporation is a way to avoid unlimited increase of pheromone trails and also it allows the forgetfulness of the bad choices.

$$\tau_{ij}(t) = \rho\tau_{ij}(t-1) + \sum_{A=1}^{NA} \Delta\tau_{ij}^{A}(t) \qquad \text{... (42)}$$

where *NA* = number of ants

ρ = the evaporation rate $0 < \rho \leq 1$.

6.2.1 Implementation Algorithm

Step I Initialize randomly potential solutions of the parameters K_p, K_i, K_d by using uniform distribution. Initialize the pheromone trail and the heuristic value.

Step II Place the A^{th} ant on the node. Compute the heuristic value associated on the objective (minimize the error).

Step III Use pheromone evaporation given by equation (42) to avoid unlimited increase of pheromone trails and allow the forgetfulness of bad choices.

Step IV Evaluate the obtained solutions according to the objectives.

Step V Display the optimum values of the optimization parameters.

Step VI Update the pheromone, according to the optimum solutions calculated at step V. Iterate from step II until the maximum of iterations is reached.

6.2.2 ACO Flowchart

The flowchart of the Ant Colony Optimization based PID control system is shown in figure 6.1 [23].

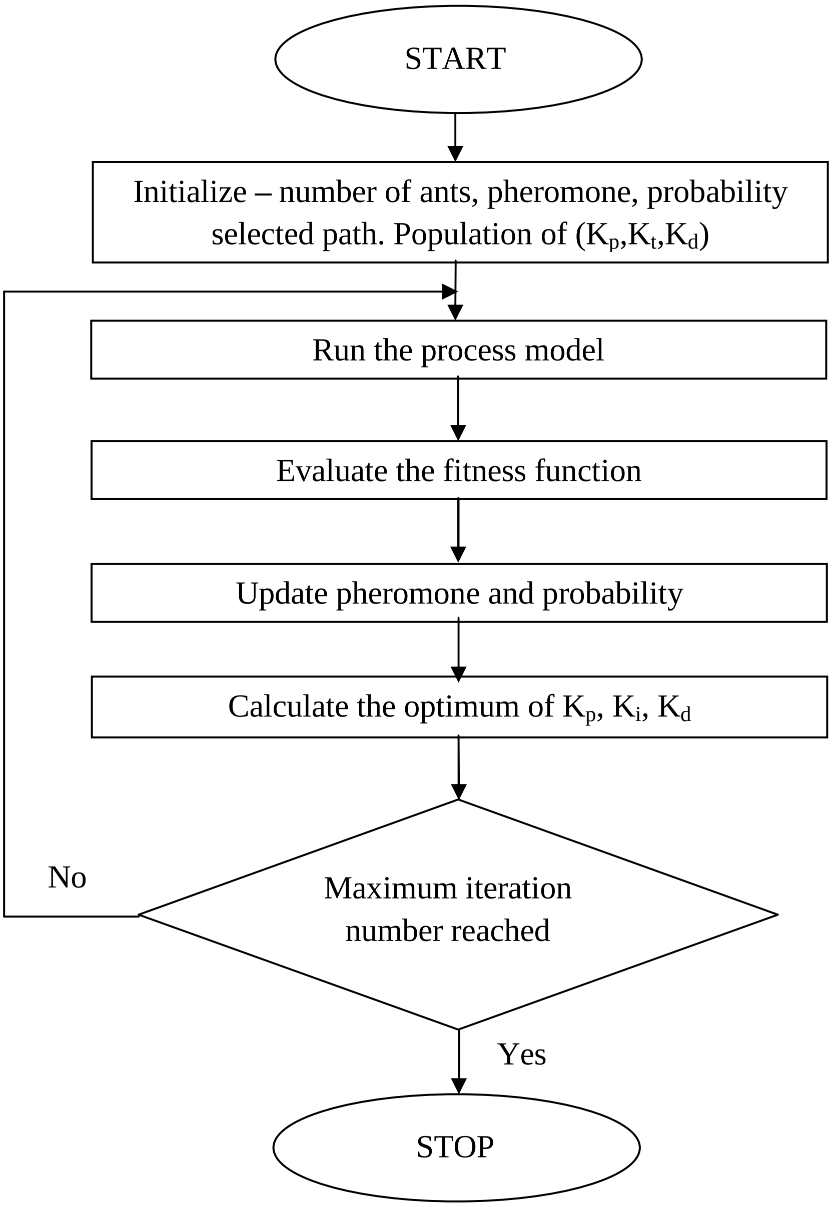

Figure 6.1 Flowchart of Ant Colony Optimization

6.3 ACO Result Parameters

The closed loop DC motor drive based on PID controller has been programmed in MATLAB and ACO algorithm has the following parameters:

$\alpha = 1$

$\beta = 2$

$\rho = 0.8$

Number of ants = 500

With ACO technique, the following parameters are found and given as:

System transfer function:

$$\frac{1.812s^2 + 0.8327s + 0.4787}{0.0000009045s^3 + 1.812s^2 + 0.8363s + 0.4787}$$

$K_p = 0.4787$

$T_i = 1.7394$

$T_d = 2.1760$

Rise time, $T_r = 0.0000010981$

Settling time, $T_s = 0.0000019577$

Overshoot, $M_p = 0.0$ %

Gain margin, G_m = Inf

Phase margin, $P_m = -180$

Delay margin = Inf

Modulus margin = -191.4396

Step response of system tuned with ACO technique:

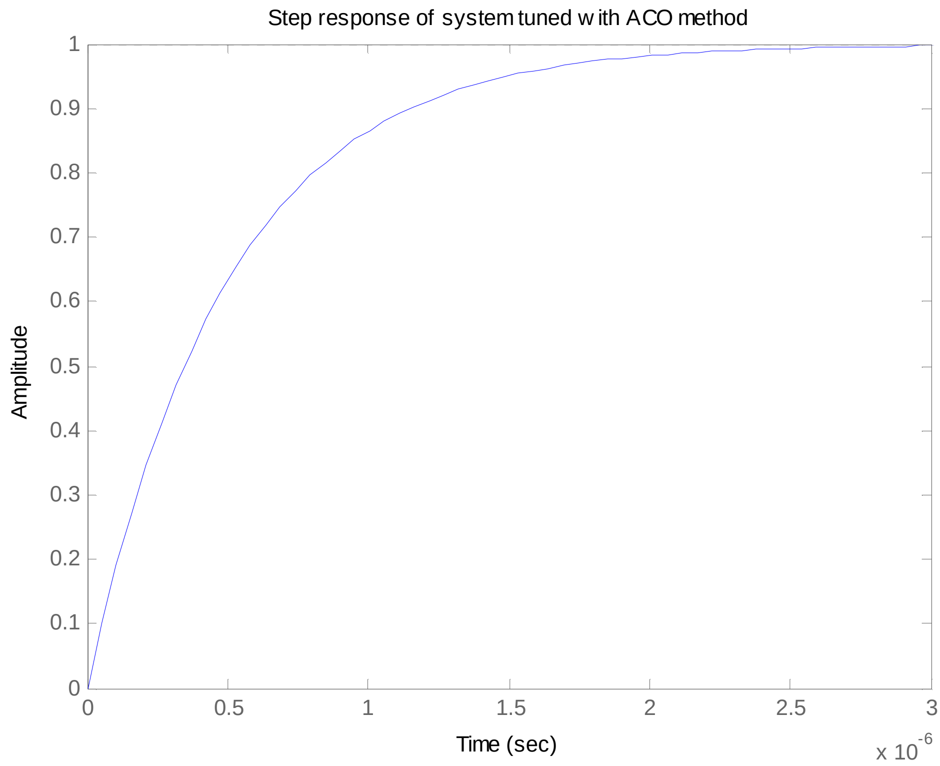

Figure 6.2 DC motor speed control response using ACO method

6.4 Conclusion

By using biological inspired ACO technique it is found that the closed loop system has very fast rise time, settling time and zero maximum overshoot to sustain the system stability under servo condition. From the transient response, it is observed that ACO method gives fast response. The ACO method is better as compared to conventional PID controller. It is also found that the modulus margin does not fall in range which is required for the same system. Also comparing with conventional PID tuning rules and optimal tuning of PID controller using error integrals, the ACO technique has the advantages of system being faster rise time, settling time and zero maximum overshoot; although the robustness has some issue with same system like gain margin is infinite, phase margin is -180^o and delay margin is also coming infinite. The modulus margin is -191.4396 which is also not in required range.

In the next chapter, the basic theory, algorithm and optimal tuning of PID controller using artificial bee colony algorithm are discussed.

CHAPTER 7

OPTIMAL TUNING OF PID CONTROLLER USING ARTIFICIAL BEE COLONY ALGORITHM

7.1 Introduction

Dervis Karaboga and et. al presented the comparison results on the performance of the Artificial Bee Colony (ABC) algorithm for constrained optimization problems. The ABC algorithm has been firstly proposed for unconstrained optimization problems and showed that it has superior performance on these kinds of problems. In this paper, the ABC algorithm has been extended for solving constrained optimization problems and applied to a set of constrained problems [24].

In Bees Algorithm, the colony of artificial bees consists of three groups of bees: employed bees, onlookers and scouts. First half of the colony consists of the employed artificial bees and the second half includes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose the food source has been abandoned by the bees becomes a scout. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population.

7.2 Bees Algorithm (BA) Technique

At the first step, the artificial bees colony generates a randomly distributed initial population P(G = 0) of SN solutions (food source positions), where SN denotes the size of population. Each solution x_i (i = 1, 2, ..., SN) is a D-dimensional vector. Here, D is the number of optimization parameters. After initialization, the population of the positions (solutions) is subjected to repeated cycles, C = 1, 2, ...,MCN, of the search processes of the employed bees, the onlooker bees and scout bees. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) and tests the nectar amount (fitness value) of the new source (new solution). Provided that the nectar amount of the new one is higher than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one in her memory. After all employed bees complete the search process; they share the nectar information of the food sources and their position information with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount.

As in the case of the employed bee, she produces a modification on the position in her memory and checks the nectar amount of the candidate source. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old on. An artificial onlooker bee chooses a food source depending on the probability value associated with that food source, p_i, calculated by the equation (43):

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad \text{... (43)}$$

where fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees (BN).

In order to produce a candidate food position from the old one in memory, the artificial BA uses the following expression (44):

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \qquad \text{... (44)}$$

where $k \in \{1, 2, ..., SN\}$ and $j \in \{1, 2, ..., D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. $Ø_{i,j}$ is a random number between [-1, 1]. It controls the production of neighbour food sources around $x_{i,j}$ and represents the comparison of two food positions visually by a bee.

As can be seen from (44), as the difference between the parameters of the $x_{i,j}$ and $x_{k,j}$ decreases, the perturbation on the position $x_{i,j}$ gets decrease, too. Thus, as the search approaches to the optimum solution in the search space, the step length is adaptively reduced. If a parameter value produced by this operation exceeds its predetermined limit, the parameter can be set to an acceptable value. In this work, the value of the parameter exceeding its limit is set to its limit value. The food source of which the nectar is abandoned by the bees is replaced with a new food source by the scouts.

In artificial BA, this is simulated by producing a position randomly and replacing it with the abandoned one. In BA, providing that a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of the artificial BA, which is called “limit” for abandonment. Assume that the abandoned source is x_i and $j \in \{1, 2,...,D\}$, then the scout discovers a new food source to be replaced with x_i. This operation can be defined as in (45).

$$x_j^i = x_{\min}^j + rand(0,1)(x_j^i - x_{\min}^j) \quad \text{... (45)}$$

After each candidate source position $v_{i,j}$ is produced and then evaluated by the artificial bee, its performance is compared with that of its old one. If the new food has equal or better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained in the memory. In other words, a greedy selection mechanism is employed as the selection operation between the old and the candidate one. It is clear from the above explanation that there are four control parameters used in the ABC: The number of food sources (SN) which is equal to the number of employed or onlooker bees, the value of limit, the maximum cycle number (MCN) [24].

7.2.1 Artificial Bee Colony Algorithm

Steps (pseudo-coding) to initialize the artificial BA:

1. Initialize the population of solutions $x_{i,j}$, i = 1. . .SN, j = 1. . .D.
2. Evaluate the population.
3. Cycle=1
4. Repeat
5. Produce new solutions $\upsilon_{i,j}$ for the employed bees by using (44) and evaluate them.
6. Apply the greedy selection process.
7. Calculate the probability values $P_{i,j}$ for the solutions $x_{i,j}$ by (43).
8. Produce the new solutions $\upsilon_{i,j}$ for the onlookers from the solutions $x_{i,j}$ selected depending on $P_{i,j}$ and evaluate them.
9. Apply the greedy selection process.
10. Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution $x_{i,j}$ by (45).
11. Memorize the best solution achieved so far.

12. Cycle = Cycle+1.

13. Until Cycle = MCN.

7.2.2 Basic Artificial Bee Colony Algorithm Flowchart

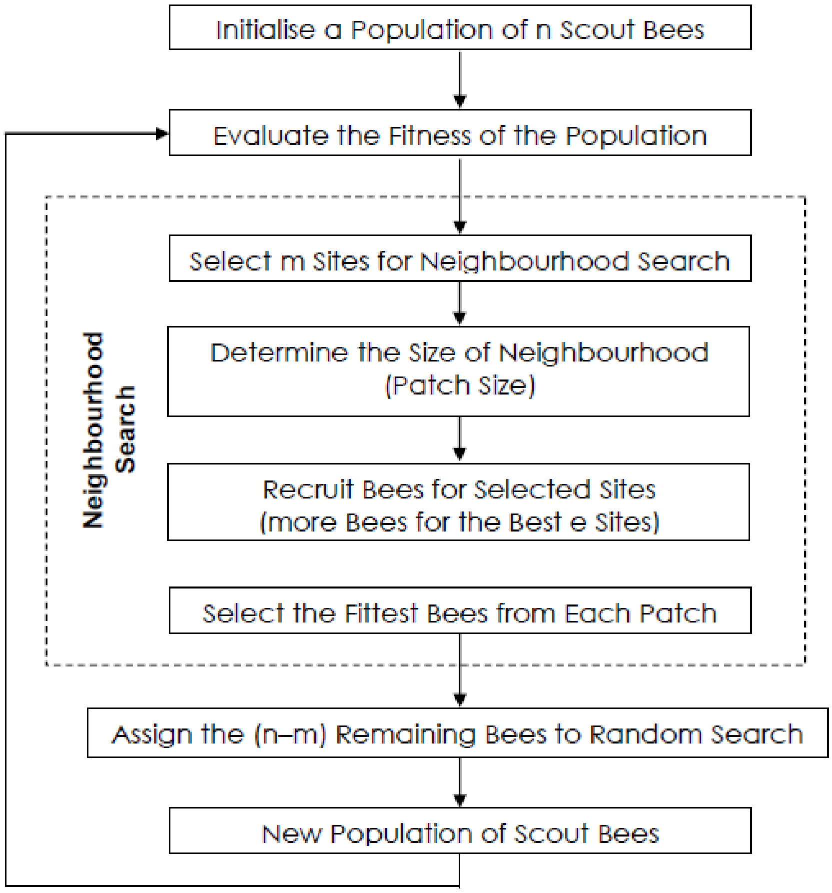

Figure 7.1 Flowchart of Artificial Bee Colony Algorithm

7.3 ABC Result Parameters

As per the ABC algorithm, the MATLAB coding results the following parameters and these are given as:

Closed loop system transfer function:

$$\frac{2.767s^2 + 2.4s + 1.474}{0.0000008464s^3 + 2.767s^2 + 2.403s + 1.474}$$

Number of scout bees = 30

Number of iterations = 15

Number of best selected patches = 20

Number of elite selected patches = 10

K_p = 1.4743

T_i = 1.6276

T_d = 1.1531

Rise time, t_r = 0.00000067267

Settling time, t_s = 0.0000011984

Maximum overshoot, M_p = 0.0 %

Gain margin, G_m = Inf

Phase margin, P_m = -180

Delay margin = Inf

Modulus margin = -71.9975

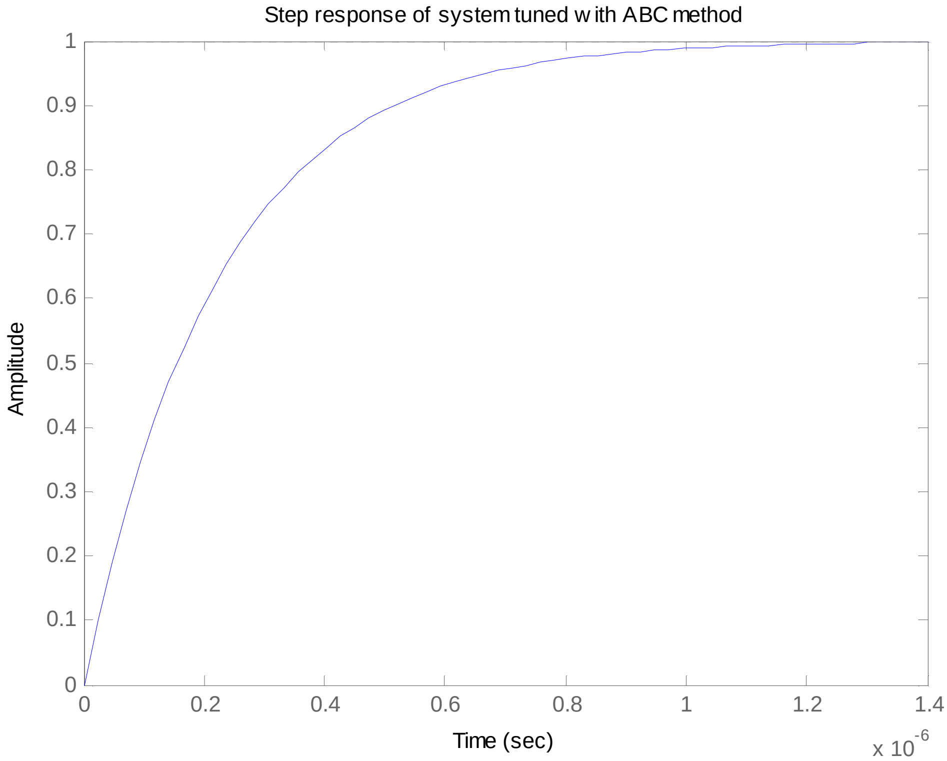

Figure 7.2 DC motor speed control response using ABC algorithm optimization technique

7.4 Conclusion

The ABC method optimized PID parameters as well as rise time, settling time and maximum overshoot. Results clearly expressed that this method has been successful in comparison to classical approach and optimal PID tuning using performance indices and can also be considered as a powerful tuning scheme for controllers. Although, the ABC consists less control parameters and it has a better tuning performance than the other optimization techniques. At the end of the analysis, the rise time, settling time and maximum overshoot of the control system are optimized with ABC algorithm; the overshoot is coming out zero for the same system. Finally, it may be said that the ABC algorithm gets better performance than the other tuning methods. Since, ABC has uncertainty with this model because of robustness of the system. For this model, system has gain margin as infinite, phase margin as -180^{o}, time delay is also infinite and modulus margin is -71.9975 which concluded that the all robust parameters does not fall in required range.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Ant Colony Optimization Technique

For the closed loop speed controller for DC motor drive, PID parameters are tuned with Ant Colony Optimization technique. By using biological inspired ACO technique it is found that the closed loop system has very fast rise time, settling time and zero maximum overshoot to sustain the system stability under servo condition. From the transient response, it is observed that ACO method gives fast response. The ACO method is better as compared to conventional PID controller. It is also found that the modulus margin is not fall in range which is required for the same system.

For the same model, the closed loop control system requires the controller for improvement of transient response of the error signal. Though the tuning of PID controller in real time is bit difficult and moreover it lacks the disturbance rejection capability. The proposed control strategy possesses good transient responses but has problem with good load disturbance response i.e. regulatory response.

Also comparing with conventional PID tuning rules and optimal tuning of PID controller using error integrals, the ACO technique has the advantages of system being faster rise time, settling time and zero maximum overshoot; although the robustness has some issue with same system like gain margin is infinite, phase margin is -180^o and delay margin is also coming infinite. The modulus margin is -191.4396 which is also not in required range. Hence, these are the uncertainty and problems with this DC motor model and also, ACO has more convergence time during its optimization process as compared to next upcoming biological inspired optimization technique.

Further for the future perspective we can make design of an efficient ACO based tuning of PID controller which may lead good robustness, uncertain and merge with the same system. There may be several techniques based on ACO algorithms to learn and optimize the controller parameters. The approach for designing of PID controller may be invented using different search techniques. The work can be extended in future by implementing the hybrid ACO search technique. Since, ACO having no overshoot even if the system is perturbed in several ways; especially, these features are important in robotic control applications and the off-line tuning process used in this work can also be extended to on-line tuning process. As a result, the PID controller tuning based on ant colony optimization technique in the field of control engineering has a very broad application prospectus.

Artificial Bee Colony Algorithm Technique

The artificial bee colony (ABC) algorithm was employed to tune PID controller for DC motor drive system. The ABC method optimized PID parameters as well as rise time, settling time and maximum overshoot. Results clearly expressed that this method has been successful in comparison to classical approach and optimal PID tuning using performance indices and can also be considered as a powerful tuning scheme for controllers. Although, the ABC consists less control parameters and it has a better tuning performance than the other optimization techniques.

The usage of ABC algorithm in order to optimize the parameters of PID controller, rise time, settling time and maximum overshoot which are used for DC motor drive system. The optimizing performance of the ABC algorithm in this system is compared with classical approach and optimal PID tuning using error indices. At the end of the analysis, the rise time, settling time and maximum overshoot of the control system are optimized with ABC algorithm; the overshoot is coming out zero for the same system. Finally, it may be said that the ABC algorithm gets better performance than the other tuning methods.

The ABC algorithm has a higher accuracy and the PID controller parameters can obtain the quantity that low rise time and low settling time as well as zero maximum overshoot in this DC motor model drive system. Since, ABC has uncertainty with this model because of robustness of the system. For this model, system has gain margin as infinite, phase margin as -180^{o}, time delay is also infinite and modulus margin is -71.9975 which concluded that the all robust parameters does not fall in required range. Here in this model, it is cleared from MATLAB coding result that ABC has fast convergence time during its optimization process as compared to ACO technique earlier.

The foraging behaviour of a colony of bees is exploited to develop an optimization algorithm and the same is used to design closed loop speed controller for a DC motor model drive system. As far as future concern, the researchers may focus on applying these bees' algorithm in designing of these kinds of systems with better approaches so robustness can achieve in required range for the same system.

[1] N. Navidi, M. Bavafa and S. Hesami, “A New Approach for Designing of PID Controller for a Linear Brushless DC Motor with Using Ant Colony Search Algorithm”, Power and Energy Engineering Conference, IEEE, 2009, pp. 1-5.

[2] A. Soundarrajan, S. Sumathi and C. Sundar, “Ant Colony Optimization Based PID Tuning for AVR in Autonomous Power Generating Systems”, International Journal of Recent Trends in Engineering and Technology, Vol. 3, No. 4, May 2010, pp. 125-129.

[3] Brahim Berbaoui, Chellali Benachaiba, Mustapha Rahli and Hamza Tedjini, “An efficient algorithm to tuning PI-controller parameters for shunt active power filter using ant colony optimization”, PRZEGLAD ELEKTROTECHNICZNY (Electrical Review), 6/2011, pp. 140-145.

[4] Huseyin Atakan Varol and Zafer Bingul, “A New PID Tuning Technique Using Ant Algorithm”, Proceeding of the 2004 American Control Conference Boston, Massachusetts June 30- July 2, 2004, pp. 2154-2159.

[5] Ying-Tung Hsiao and Cheng-Long Chuang, “Ant Colony Optimization for Designing of PID Controllers”, IEEE International Symposium on Computer Aided Control Systems Design, 2004, pp. 321-326.

[6] Hong He, Fang Liu, Li Li, Jin-Rong Yang, Lei Su and Yi Wu, “Study of PID Control for Ant Colony Algorithm”, Global Congress on Intelligent Systems, IEEE Computer Society, 2009, pp. 204-207.

[7] Duan Hai-bin, Wang Dao-bo and Yu Xiu-fen, “Novel Approach to Nonlinear PID Parameter Optimization Using Ant Colony Optimization Algorithm”, Journal of Bionic Engineering, Vol. 3 No. 2, 2006, pp. 73-78.

[8] Dr. S. M. GiriRajkumar, Dr. K. Ramkumar and Sanjay Sarma O. V., “Real Time Application of Ants Colony Optimization”, International Journal of Computer Applications, Volume 3 – No. 8, 2010, pp. 34-46.

[9] Mahdi Abachizadeh, Mohammad Reza Haeri Yazdi and Aghil Yousefi-Koma, “Optimal Tuning of PID Controllers Using Artificial Bee Colony Algorithm”, IEEE/ASME international Conference on Advanced Intelligent Mechtronics, pp. 379-384, 2010

[10] Ozden Ercin and Ramazan Coban, “Comparison of the Artificial Bee Colony and the Bees Algorithm for PID Controller Tuning”, Innovations in Intelligent Systems and Applications (INISTA) IEEE conference, pp. 595-598, 2011.

[11] H. Gozde, M. C. Taplamacioglu and I. Kocaarslan, “Applications of Artificial Bees Colony Algorithm in an Automatic Voltage Regulator (AVR) System”, International Journal on Technical and Physical Problems of Engineering (IJTPE), Volume 2 Number 3, pp. 88-92, 2010.

[12] Gaowei YAN and Chuangqin LI, “An Effective Refinement Artificial Bee Colony Optimization Algorithm Based On Chaotic Search and Application for PID Control Tuning”, Journal of Computational Information Systems, Issue 7:9, pp. 3309-3316, 2011.

[13] K. Sundareswaran, “D.C. Motor Speed Controller Design through a Colony of Honey Bees”, TENCON (IEEE Region) conference, pp. 1-6, 2008.

[14] Fei Gao, Yibo Qi, Qiang Yin and Jiaqing Xiao, “An Novel Optimal PID Tuning and On-line Tuning Based on Artificial Bee Colony Algorithm”, International Conference on Computational intelligence and Software Engineering, IEEE, pp. 1-4, 2010.

[15] Anguluri Rajasekhar and Ajith Abraham, “Fractional Order Speed Control of DC Motor Using Levy Mutated Artificial Bee colony Algorithm”, World Congress on Information and communication Technologies, IEEE, pp. 7-13, 2011.

[16] Manafeddin Namazov and Onur Basturk, “DC motor position control using fuzzy proportional-derivative controllers with different defuzzification methods”, Received: February 1, 2010 - Revised: April 2, 2010 – Accepted: May 26, 2010.

[17] Ian Griffin, “On-line PID Controller Tuning using Genetic Algorithm”, 22.08.2003.

[18] P.W. Muril, P.D. Schnelle Jr., B.G. Liptak, Gerry, J., Ruel, M., Shinskey, F.G., “Process Control and Optimization”, Volume-II, pp. 414-431, 2006

[19] Zhuang, M., Atherton, D.P., “Automatic Tuning of Optimum PID controllers”, IEE Proceddings-D, Vol. 140, No. 3, pp. 216-224, May 1993.

[20] Vittirio Maniezzo, Luca Maria Gambardella, Fabio de Luigi, “Ant Colony Optimization”, supported by the Future and Emerging Technologies unit of the European Commission through BISON (IST-2001-38923).

[21] Karl O. Jones, Andre Bouffet, “Comparison of Bees Algorithm, Ant Colony Optimization and Particle Swarm Optimization for PID Controller Tuning”, International Conference on Computer Systems and Technologies – CompSys Tech’08.

[22] Fares Sayadi, Mahamod Ismail, Norbahiah Misran, Kasmiran Jumari, “Multi-Objective Optimization Using the Bees Algorithm in Time-Varying Channel for MIMO MC-CDMA Systems”

[23] B. Nagaraj, P. Vijayakumar, “A Comparative Study of PID Controller Tuning using GA, EP, PSO and ACO”, 22nd February, 2011.

[24] Dervis Karaboga and Bahriye Basturk, “Artificial Bee Colony (ABC) Optimization Algorithm for solving Constrained Optimization Problems”, IFSA 2007, LNAI 4529, pp. 789-798, 2007.

[25] H. Ying-Tung, C. Cheng-Long, C. Cheng-Chih, “Ant Colony Optimization for Designing of PID Controllers”, IEEE International Symposium on Computer Aided Control Systems Design, Taipei, Taiwan, 24th September, 2004.

[26] I. Chiha, P. Borne, “Multi-Objective Ant Colony Optimization to Tuning PID Controller”, Proceedings of the International Journal of Engineering, Vol. III, issue no. 2, March, 2010.

[27] A. Purna Chandra Rao, Y.P. Obulesu and CH. Sai Babu, “Robust Internal Model Control Strategy Based PID Controller For BLDCM”, International Journal of Engineering Science and Technology, Vol. 2(11), pp. 6801-6811, 2010.