Comparative performance study of Evolutionary and Immune Algorithm based PID Controller Tuning for Speed Control of DC Motor Drives

A Thesis report

submitted towards the partial fulfillment of the

requirements of the degree of

Master of Engineering

in

Power System and Electric Drives

submitted by

Monu Tyagi

(801041033)

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 Evolutionary and Immune Algorithm 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 June 2012
Place: Thapar University, Patiala

Monu Tyagi
Roll No. 801041033

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

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

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.

Monu Tyagi
(801041033)

ABSTRACT

Biological immune system (BIS) and Biological Evolutionary system (BES) is a special type of control system that has shown strong robustness and self-adaptability. In this thesis report artificial immune system algorithm and biological evolutionary system used to develop are immune controller and evolutionary controller. The idea of immune controller and evolutionary controller is adopted and derived from biological vertebrate immune and evolutionary system, mimicking and imitating of biological immune and evolutionary system which is better known as the artificial immune and evolutionary system. Immune and Evolutionary are stochastic optimization techniques based on the principle of natural evolution. We describe the general functioning of EAs and BIS.

The aim of this subject is to design a speed controller for a DC motor by selection of PID parameters using “Evolutionary and Immune.” These algorithms 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. 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-Nichols’ 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. In this thesis we have taken into account the various configurations of PID controllers in Evolutionary and Immune algorithm to measure the performance of the controller for the dc motor drive system.

ORGANISATION OF THESIS

Chapter-1 In this chapter includes of the introduction of thesis

Chapter-2 It contains most of the previous work in this field which has been carried out till date are give

Chapter-3 DC motor and DC motor transfer function has been elaborated in this chapter.

Chapter-4 Classical PID tuning method has been discussed in this chapter.

Chapter-5 It contains optimal tuning using error indices in this chapter.

Chapter-6 Optimal tuning of PID controller using Evolutionary Algorithm with different controllers has been discussed in this chapter.

Chapter-7 Optimal tuning of PID controller using Immune Algorithm with different controllers has been elaborated in this chapter.

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

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURE

LIST OF SYMBOLS AND ABBREVIATIONS

Ti	Load torque
E_{ss}	Steady state error
K_p	Proportional gain
K_i	Integral gain
K_d	Derivative gain
T_i	Integral action time
T_d	Derivative action time
IMC	Internal mode control
C-C	Cohen-coon
Z-N	Ziegler-nicolas
T-L	Tyreus-Luyben
C-H-R	Chien-Hrones-Reswick
A-H	Astrom-Hagglund
IAE	Integral absolute error
ITAE	Integral time with absolute error
ISE	Integral square error
ISTE	Integral time with square error
EA	Evolutionary algorithm
DE	Differential error
BIS	Biological immune

CHAPTER 1
INTRODUCTION

1.1Introduction

The objectives of this thesis work are to study and analyze the mathematical model and algorithm for speed control of separately excited DC motor drive. Here are various types of mathematical model of applicable DC motor model can be found from different sources books, journal, papers, internet etc.

Here test the performance of some standard PID controller on DC motor drive system. After that compares those results.

When a separately excited motor is excited by a field current of i_f and an armature current i_a flows in the circuit, the motor develops a back emf and a torque to balance the load torque at a particular speed. The armature winding is connected to an external DC source; hence it plays the role of the current carrying conductor placed in the magnetic field. The i_f is independent of the i_a. Each winding are supplied separately. Any change in the armature current has no effect on the field current. The i_f is much less than the i_a. The direction of rotation depends on the direction of magnetic field produced by the field winding as well as the direction of armature current. The direction of rotation is decided by Fleming's left hand rule.

PID is the most common and most popular feedback controller used in Industrial Process today. A PID controller calculates an "error" value as the difference between a measured process_variable and a desired 'set-point'. PID controller is also known as three-term control:- the proportional(P), integral(I) and derivative(D). By tuning these three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements.

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.

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.

Evolutionary algorithms (EA) are stochastic optimization techniques based on natural evolution and survival of the fittest strategy found in biological organisms. Evolutionary algorithms have been successfully applied to solve complex optimization problem in business, engineering, and science. Some commonly used EAs are Genetic algorithms (GA's), Evolutionary Programming (EP), Evolutionary Strategy (ES) and Differential Evolution (DE). Each of these methods has its own characteristics, strengths and weaknesses. In general, an EA algorithm can generate a set of initial solutions randomly based on the given seed and population size. Afterwards, it will go through evolution operations such as cross-over and mutation before evaluated by the objective function. The wining entity in the population will be selected as the parents (or seed) of the next generation. The optimization iteration continues until the termination criteria are satisfied. Typically, either the evolution process reached users define maximum number of iteration or the improvement in objective function between the two generations converges.

The word "immune" comes from the Latin word for "protection" your body immune system is your built – in protection against attack by foreign substances known as antigen. An antigen is any substance that causes an immune response or causes the body to "attack."

Immune Algorithms is an improved algorithm based on biological immune mechanisms. In the course of immune response, biological immune system preserves part of the antibodies as memory cells. When the same antigen invades again, memory cells activated and a large number of antibodies are generated so that the secondary immune response is more quickly than the initial response. In the meanwhile, there are mutual promotion and inhibition between antibodies. Therefore, the diversity and immune balance of the antibodies are maintained. That is the self-regulatory function of the immune system. The Immune Algorithm simulates the process of the adaptive regulation of the biological antibody concentration, in which the optimal solution of the objective function corresponds to the invading antigens and the fitness $f(X_i)$ of solution X_i corresponds to the antibodies produced by the immune system. According to the concentration of the antibodies, the algorithm adaptive regulates the distribution of the search direction of the solutions and greatly enhances the ability to overcome the local convergence. Example of antigens includes bacteria or virus.

The main function of the immune system is to protect the body from pathogens and cancer. Vertebrate immune systems are more complex than the invertebrates. They are characterized by two important properties, which are memory and specificity. In the case of invertebrate, the immune system consist mainly of phagocytes which are nonspecific. This means that it will not remember any previous antigen, and will use the same attacking strategy each time. Phagocytes has no receptors for specific pathogens, which means that these cells will engulf and try to kill any pathogen. On the other hand, the vertebrate host has evolved more specialized cells called lymphocytes. These Lymphocytes are pathogen specific, which means that they have distinct receptors to interact with different pathogens. To combat antigens nature has provided us with the immune system. The blood, lymph nodes, and bone marrow act with the liver, spleen, thymus, and tonsils to produce and deliver specialized cells, including B-lymphocytes, T-lymphocytes, and phagocytes. These cells limit the severity and duration of colds, fight infections in the nose and throat, help wounds to heal, destroy some cancers, and much more.

CHAPTER 2

LITERATURE REVIEW

The following section describes the literature survey that is relevant with the work carried out for this thesis work.

A self-tuning PID controller is developed by Teng Fong-Chwee *et al*. based on pole assignment approach to overcome fractional dead time, constant and known dead time, plus time varying dead time. This method showed good results in handling dead time processes [10].

T. Back *et al*. present a overview of Evolutionary Algorithms for parameter optimization [11].

Evolutionary Computation is developed by D.B. Fogel. In this literature covering evolutionary programming, evolution strategies, and genetic algorithms from a perspective of achieving machine intelligence through evolution [12]

Kalyanmoy Deb *et al*. proposed a generic parent-centric recombination operator (PCX) and a computationally fast population-alteration model called G3 model. The performance of G3 model with PCX operator is found better in consistency and reliability when compared to the differential evolution technique and the quasi-Newton method. The proposed approach is found to be consistently and reliably performing better than all other methods used in study. A scale-up study with problem sizes up to 500 variables shows a polynomial computational complexity of the proposed approach.

Corina Rotar developed a new evolutionary algorithm for multi-objective optimization which is inspired by endocrine system and uses the Pareto non-dominance concept. Multi-objective optimization problems, due to their complexity, are suitable for interesting evolutionary approaches. Many evolutionary algorithms have been developed for solving multi-objective problems, appealing or not to the Pareto optimality concept. Although, evolutionary techniques for multi-objective optimization confront with several issues as: elitism, diversity of the population, or efficient settings for the specific parameters of the algorithm. In this paper, we propose a new evolutionary technique, which is inspired by the behavior of the endocrine system and uses the Pareto non-dominance concept. Therefore, the population's members are no more called chromosomes but hormones and even if they evolve according to the genetic principles (selection, crossover, and mutation), a supplementary mechanism, based on the endocrine paradigm, is connected with standard approach to deal with multi-objective optimization problems. Moreover, the proposed algorithm, in order to maintain population's diversity, uses a specific scheme of fitness sharing, eliminating the inconvenient of defining an appropriate value of sharing factor [14].

A robust tuning method for disturbance rejection of PID controller using Evolutionary algorithm is proposed by Dong Hwa Kim *et al*. which tunes the gain of PID controller using fitness value of immune algorithm. A PID Controller has been widely used to control industrial process loops because of its implementation advantages. However, it is very difficult to achieve an optimal PID gain with no experience, since the parameters of the PID controller has to be manually tuned by trial and error. This paper focuses on robust tuning of the PID controller using immune algorithm which has function such as diversity, distributed computation, adaptation, self-monitoring function. After deciding disturbance rejection condition for the given process, the gains of PID controller using fitness value of immune algorithm depending on disturbance rejection is tuned for the required response. To improve this suggested scheme, simulation results are compared with FNN based responses and genetic algorithm [15].

Gregory S.Hornby presented a method for automated antenna design and optimization based on Evolutionary algorithm. Whereas the current practice of designing antennas by hand is severely limited because it is both time and labor intensive and requires a significant amount of Evolutionary algorithms can be used to search the design space and automatically and novel antenna designs that are more effective than would otherwise be developed. Here we present automated antenna design and optimization methods based on evolutionary algorithms. We have evolved efficient antennas for a variety of aerospace applications and here we describe one proof-of-concept study and one project that produced flight antennas that flew on NASA's Space Technology 5 (ST5) mission [16].

A partial 3D reconstruction method based on Evolutionary algorithm is proposed by Monica Perez-Meza *et al*. for various vision stereo problems. This method can be used in a mobile robot with monocular vision for navigation tasks. When reconstructing a scenario, it is necessary to know the structure of the elements present on the scene to have an interpretation. In this work we link 3D scenes reconstruction to evolutionary algorithms through the vision stereo theory. We Consider vision stereo as a method that provides the reconstruction of a scene using only a couple of images of the scene and performing some computation. Through several images of a scene, captured from different positions, vision stereo can give us an idea about the three dimensional characteristics of the world. Vision stereo usually requires of two cameras, making an analogy to the mammalian vision system. In this work we employ only a camera, which is translated along a path, capturing images every certain distance. As we can not perform all computations required for an exhaustive reconstruction, we employ an evolutionary algorithm to partially reconstruct the scene in real time. The algorithm employed is the fly algorithm, which employ "flies" to reconstruct the principal characteristics of the world following certain evolutionary rules [17].

Xiufen Zou *et al*. proposed a new Evolutionary algorithm based on Loptimality for solving multi-objective optimization problems. In this researcher focused on the study of evolutionary algorithms for solving multi-objective optimization problems with a large number of objectives. First, a comparative study of a newly developed dynamical multi-objective evolutionary algorithm (DMOEA) and some modern algorithms, such as the indicator-based evolutionary algorithm, multiple single objective Pareto sampling, and non dominated sorting genetic algorithm II, is presented by employing the convergence metric and relative hyper volume metric. For three scalable test problems (namely, DTLZ1, DTLZ2, and DTLZ6), which represent some of the most difficult problems studied in the literature, the DMOEA shows good performance in both converging to the true Pareto-optimal front and maintaining a widely distributed set of solutions. Second, a new definition of optimality (namely, L-optimality) is proposed in this paper, which not only takes into account the number of improved objective values but also considers the values of improved objective functions if all objectives have the same importance. Researcher proves that L-optimal solutions are subsets of Pareto-optimal solutions. Finally, the new algorithm based on L-optimality (namely, MDMOEA) is developed, and simulation and comparative results indicate that well-distributed L-optimal solutions can be obtained by utilizing the MDMOEA but cannot be achieved by applying L-optimality to make *a posteriori* selection within the huge Pareto non dominated solutions. We can conclude that our new algorithm is suitable to tackle many-objective problems [18].

Exploring new horizons in evolutionary design of robots is proposed by Stephane Donicieus *et al*. using Evolutionary algorithm and work in robotics. This introduction paper to the 2009 IROS workshop “Exploring new horizons in Evolutionary Design of Robots” considers the field of Evolutionary Robotics (ER) from the perspective of its potential users: robot cists. The core hypothesis motivating this field of research will be discussed, as well as the potential use of ER in a robot design process. Three main aspects of ER will be presented: (a) ER as an automatic parameter tuning procedure, which is the most mature application and is used to solve real robotics problem, (b) evolutionary-aided design, which may benefit the designer as an efficient tool to build robotic systems and (c) automatic synthesis, which corresponds to the automatic design of a mechatronic device. Critical issues will also be presented as well as current trends and perspectives in ER [20].

Evolutionary computation: An Overview and Recent Trends proposed by Gunther Raidl. In this Gunther explained recent trends in evolutionary algorithms [22].

Dong Hwa kim, *et al* focuses on tuning scheme of the PID controller based on the required reference model using immune algorithms for a process. Up to this time, many sophisticated tuning algorithms have been tried in order to improve the PID controller performance under such difficult conditions [23].

In this paper author Dong Hwa Kim *et al* uses PID controller tuned by genetic algorithms. This paper addresses comparison of tuning method of the PID controller using immune based tuning technique and genetic algorithm based tuning technique on steam temperature process with long dead time [24].

This paper Researcher Dong Hwa Kim *et al* focuses on tuning of the PID controller with disturbance rejection using immune algorithms. To decide the performance of response, an ITSE is used in this paper. Up to this present time, PID controller has been used to operate this system because of its implementation advantages [25].

Dong Hwa Kim focuses on tuning PID controller gain/phase margin and immune algorithms. After deciding optimal gain/phase margin specification for the given process, the gains of PID controller using fitness value of Immune algorithm depending on error between optimal gain/phase margin and the gain/phase margin obtained by tuning is tuned for the required response [26].

Maryam Khoie *et al* focuses, a genetic-AIS algorithm is introduced for PID controller tuning using a multi-objective optimization frame work. The auto-tuned PID algorithm is then fused in an immune feedback law based on a nonlinear proportional gain to realize a new PID controller. Accordingly, this leads to a more effective Immune-based tuning than the Conventional PID tuning schemes benefiting a multi-objective optimization prospective [27].

Tian Xiao-min *et al* focuses in this paper traditional PID controller is used as a fractional order $PI^{\lambda}D^{\mu}$ controller . the application technology of traditional PID controller has been very mature, but in complex industrial production, the controller object often is non-linear or time-varying, traditional PID controller often can not reach the desired effect, which introduced the concept of fractional order $PI^{\lambda}D^{\mu}$ controller is still a problem to be resolved [28].

In this paper author Dong Hwa Kim proposed the immune network based on fuzzy set suggested here is applied for the PID controller tuning of multi-objective process with two inputs and one output. The result of study shows the artificial immune based on fuzzy set can effectively be used to tune the nonlinear process or the multivariable process, since it can more fit modes or parameters of the PID controller than that of the conventional tuning methods, against the noise or disturbance, various inputs, and coupling action between loops [29].

Young Jin Lee *et al* proposed in this paper, a adaptive PID control method based on humeral immune algorithm and neural network identifier technique. When HIA is applied to PID controller, there exists the case that the plant is damaged due to abrupt change of PID parameters. Since the parameters are almost adjusted randomly.

Dong Hwa Kim focuses on tuning of controller with disturbance rejection using immune based multi-objective approach, an ITSE is used decide performance of tuning results. Strictly maintaining the steam temperature can be difficult due to heating value variation to the fuel source, time delay changes in the main steam temperature, the change of dynamic characteristics in the reheater [30].

CHAPTER 3
DC MACHINE MODEL

3.1 Introduction

When a separately excited motor is excited by a field current of i_f and an armature current i_a flows in the circuit, the motor develops a back emf and a torque to balance the load torque at a particular speed. The armature winding is connected to an external DC source, hence it plays the role of the current carrying conductor placed in the magnetic field. The i_f is independent of the i_a . Each windings are supplied separately. Any change in the armature current has no effect on the field current. The i_f is much less than the i_a. The direction of rotation depends on the direction of magnetic field produced by the field winding as well as as the direction of armature current. The direction of rotation is decided by Fleming's left hand rule.

3.2 DC Machine Model (Separately Excited)

The design method uses the concepts of the system theory, such as signals and systems, transfer functions, direct and inverse Laplace transforms. This requires building the appropriate Laplace model for each component of the whole system.

In order to build the DC motor's transfer function, its simplified mathematical model has been used. This model consists of differential equations for the electrical part, mechanical part and the interconnection between them. The electric circuit of the armature and the free body diagram of the rotor are shown in figure 3.1.

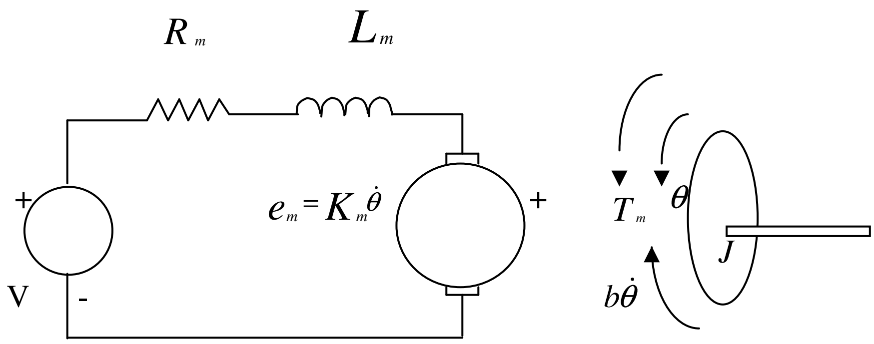

Figure 3.1:-The electric circuit of the armature and the free body diagram of the rotor for a DC motor

The motor torque T_m is related to the armature current, i, by a constant factor K_t. The back emf, e_m, is related to the rotational speed, $\dot{\theta}$ by the following equations:-

$$T_m = K_t i \qquad \text{... 3.1}$$

$$e_m = K_e \dot{\theta} \qquad \text{... 3.2}$$

Assuming that K_t (torque constant) = K_e (electromotive force constant) = K_m (motor constant).

From figure 1 and above known values, the following equations can be written based on Newton's law combined with Kirchhoff's law:

$$T_m - T_L = J \frac{d^2\theta(t)}{dt^2} + B \frac{d\theta(t)}{dt} \qquad \text{... 3.3}$$

$$J\ddot{\theta} + B\dot{\theta} = K_m i - T_L \qquad \text{... 3.4}$$

$$L_m \frac{di}{dt} + R_m i = V - K_m \dot{\theta} \qquad \text{... 3.5}$$

3.3 Transfer Function

Using Laplace transform, the above equations can be expressed in terms of s-domain.

$$(Js + B)\dot{\theta}(s) = K_m I(s) - T_L(s) \qquad \text{... 3.6}$$

$$(L_m s + R_m) I(s) = V(s) - K_m s\dot{\theta}(s) \qquad \text{... 3.7}$$

By eliminating $I(s)$, the following open-loop transfer function can be obtained, where the rotational speed $\dot{\theta}$ is the output and the voltage V is the input.

3.3.1 Speed Control

a)

Assuming T_L **(load torque) = 0**,

T_f (friction torque) = 0 and,

feedback constant $K_v = K_m$;

the transfer function is:-

$$\left.\frac{\dot{\theta}(s)}{V(s)}\right|_{T_L(s)=0} = \frac{K_m}{(Js+B)(L_m s+R_m)+K_m^2} \quad \text{... 3.8}$$

The block diagram obtained from this equation is shown in figure 3.2:

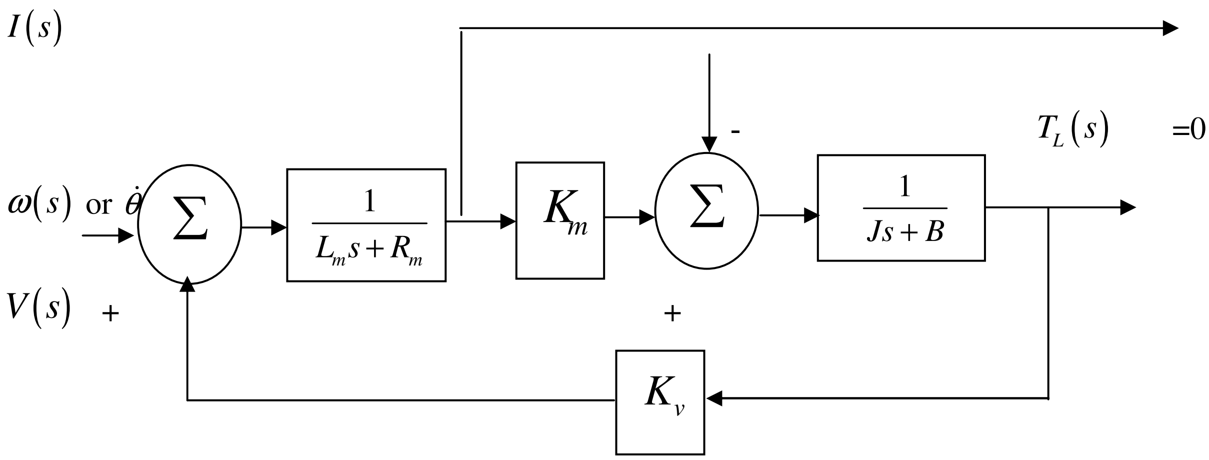

Figure 3.2: Block diagram representation for SPEED Response with T_L =0

b) Assume V(s) = 0;

For Speed Control, the transfer function is given (from equations 6 and 7):-

$$\left.\frac{\dot{\theta}(s)}{T_L(s)}\right|_{V(s)=0} = \frac{-(L_m s+R_m)}{\left[(Js+B)(L_m s+R_m)+K_m^2\right]} \quad \text{... (9)}$$

The block diagram obtain for this equation is shown in figure 3.3 drawn below:-

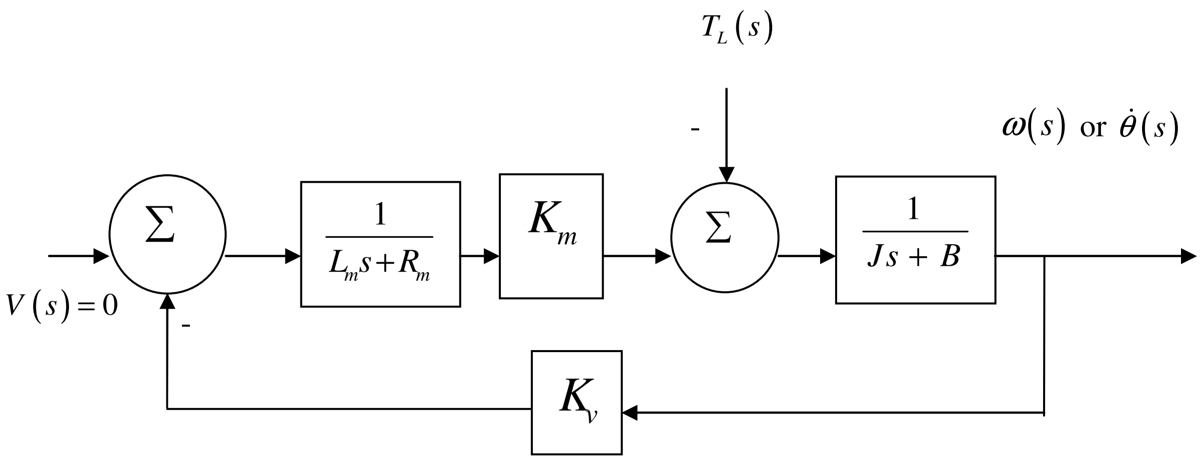

Figure 3.3:- Block diagram representation for SPEED Response with V(s) = 0

3.3.2 Position Control:-

a) Assuming T_L =0, for position control the transfer function is given as [2]:-

$$\left.\frac{\dot{\theta}(s)}{V(s)}\right|_{T_L(s)=0} = \frac{K_m}{s\left[(Js+B)(L_m s+R_m)+K_m^{\,2}\right]} \quad ..3.10$$

b) Assuming V(s)=0, for Position Control, the transfer function is given as:-

$$\left.\frac{\dot{\theta}(s)}{T_L(s)}\right|_{V(s)=0} = \frac{-(L_m s+R_m)}{s\left[(Js+B)(L_m s+R_m)+K_m^{\,2}\right]} \quad ..3.11$$

3.4 Design requirements

Since the most basic requirements of a motor are that it should rotate at the desired speed, the steady-state error e_{ss} of the motor speed should be less than 1%. The other performance requirement is that the motor must accelerate to its steady-state speed $\dot{\theta}_{ss}$ as soon as it turns on. In this case, it is desirable to have a settling time T_s in less than 2 sec. since speed faster than the reference may damage the equipment; a peak overshoot M_p of less than 5% is wanted.

3.5 PID control design method

Different characteristics of the motor response (steady-state error, peak overshoot, rise time, etc.) are controlled by selection of the three gains that modify the PID controller dynamics. Figure 3 shows how the PID controller works in the closed-loop system. The variable e represents the tracking error; the difference between the desired input value R and the actual output Y. This error signal will be sent to the PID controller computes both the derivative and the integral of this signal. Therefore, the PID controller is defined by the relationship between the controller input e and computes output u that is applied to the motor armature:

$$u = K_p e + K_i \int edt + K_d \frac{de}{dt} \quad ..3.12$$

Where K_p - Proportional gain

K_i - Integral gain

K_d - Derivative gain.

The signal u will be sent to the plant and the new output Y will be obtained and sent back to the sensor again to find the new error signal e. The controller takes e and computes its derivative and it's integral again.

This process goes on and on. By adjusting the weighting constants K_p, K_i and K_d, the PID controller can be set to give the desired performance.

(PID controller is discussed in detail in **Chapter 4**)

Taking Laplace Transform of equation (8) gives the following transfer function:

$$K(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s$$

$$= \frac{K_d s^2 + K_p s + K_i}{s} \quad ..3..$$

CHAPTER 4

CLASSICAL PID TUNING METHODS

4.1 INTRODUCTION

PID is the most common and most popular feedback controller used in Industrial Process today. A PID controller calculates an "error" value as the difference between a measured process_variable and a desired 'set-point'. PID controller is also known as three-term control:- the proportional(P), integral(I) and derivative(D). By tuning these three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements.

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 P+I+D CONTROLLER

As we have stated that PID controller is a sum of P, I and D controller, so now we must also discuss the function of each.

- Proportional (P) controller calculates the term proportional to the 'error'.
- Integral (I) controller calculates a term proportional to integral of 'error'. An integral controller gives zero SSE (Steady State Error) for a step input but reduces the speed of a system.
- Derivative (D) controller calculates a term proportional to derivative of 'error'. A derivative control terms often produces faster response.

4.2.1 Proportional (P) term

Proportional term makes a change to the output that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p, called the proportional gain.

The proportional term is given by:

$$P_{out} = K_p e(t) \qquad ...4.1$$

4.2.2 Integral (I) term

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain (K_i) and added to the controller output.

The integral term is given by:

$$I_{out} = K_i \int_0^t e(\tau) d\tau \qquad ... 4.2$$

4.2.3 Derivative (D) term

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain K_d. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, K_d.

The derivative term is given by:

$$D_{out} = K_d \frac{d}{dt} e(t) \qquad ... 4.3$$

4.2.4 PID term

The PID controller output can be obtained by adding the three terms.

$$G_c(s) = P + I + D = K_p + \frac{K_i}{s} + K_d s \qquad ... 4.4$$

Or

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \qquad ... 4.5$$

Where Kp= Proportional gain,

Ki= integration coefficient,

Kd= derivative coefficient,

Ti= integral action time,

Td= derivative action time

The block diagram of PID controller is shown below:-

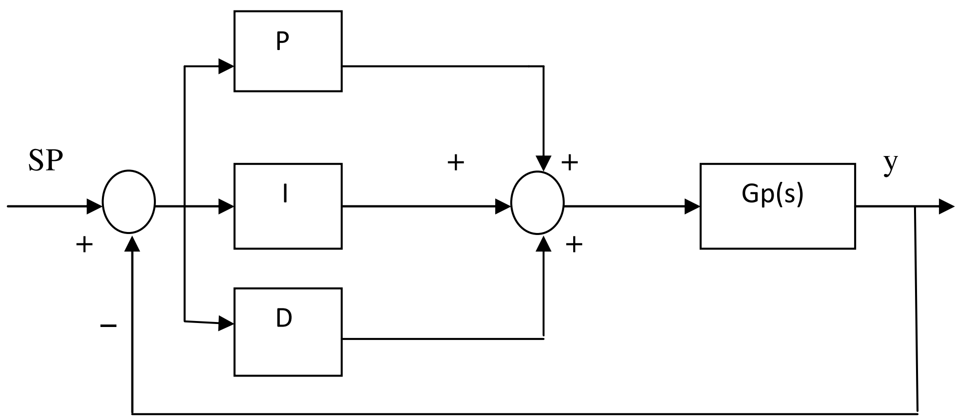

Figure 4.1: Block diagram representation of a PID control

4.3 CLASSICAL PID CONTROLLER TUNING METHODS

4.3.1 OPEN LOOP TUNING METHODS

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,

$$G_p = \frac{Ke^{-\theta s}}{Ts+1} \qquad \text{... 4.6}$$

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: First order system

The method is based on computing the times t1 and t2 at which the 35.3% and 85.3% of the system response is obtained as shown in the figure:

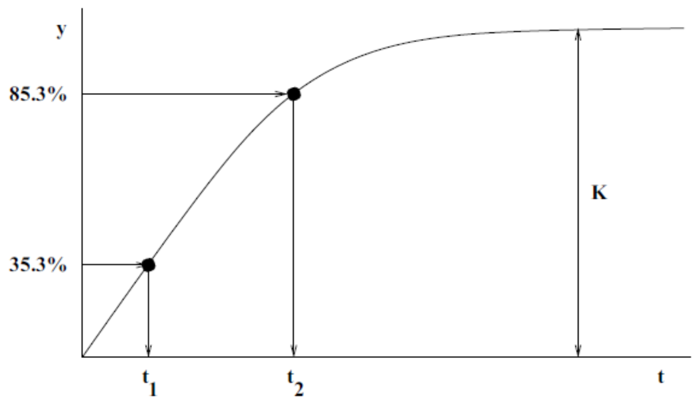

Figure 4.3: First order system response

After computing the t1 and t2 times, the time delay (θ) and process time constant (T) can be obtained from the following equations:

$$\theta = 1.3t_1 - 0.29t_2$$
$$T = 0.67(t_2 - t_1) \qquad \dots 4.7$$

Results

Table 4.1: Time domain analysis of DC machine open loop

Rise Time	Setting Time	Overshoot
0.3296	0.5902	0

Table 4.2: Servo speed control

Rise Time	Setting Time	Overshoot
0.0070	0.0226	10.3365

Table 4.3: Regulatory Speed Control

Rise Time	Setting Time	Overshoot
8.6457e-005	0.0271	1.9227e+003

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 (4.1) 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 modeling a process by an integrator and a time delay. Ziegler and Nichols have given PID parameters directly as functions of *T* and *θ*. The behavior 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 set-point response is too large.

Table 4.4: Ziegler-Nichols Tuning Method

<table>
<tr><td colspan="2">Controller</td><td>K_p</td><td>T_i</td><td>T_d</td></tr>
<tr><td rowspan="3">Ziegler-Nichols Method (Open loop)</td><td>P</td><td>$\frac{T}{K\theta}$</td><td>-</td><td>-</td></tr>
<tr><td>PI</td><td>$\frac{0.9T}{K\theta}$</td><td>$\frac{\theta}{0.3}$</td><td>-</td></tr>
<tr><td>PID</td><td>$\frac{1.2T}{K\theta}$</td><td>2θ</td><td>0.5θ</td></tr>
</table>

Results

Table 4.5: ZN comparison Transfer function and time delay forms for DC machine model

Form	Gain margin	Phase margin	Kp	Ti	Td	Rise time	Settling time	Overshoot	Modulus margin	Delay margin
T.F.	Inf	92.5126	0.0185	0.5774	0.1444	2.6232	4.7871	0	-445.611	0.0616

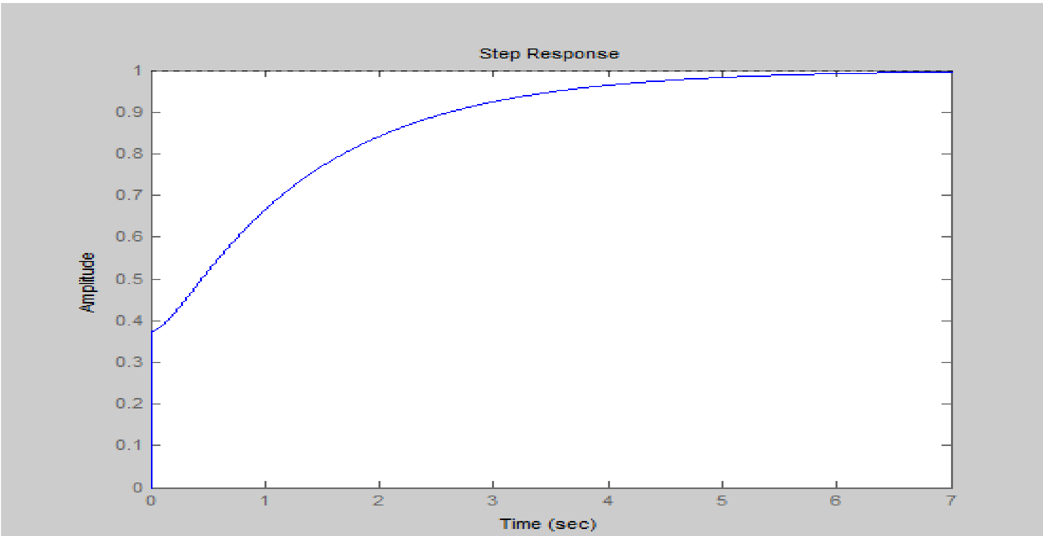

Figure 4.4 (a): Step response for T.F form

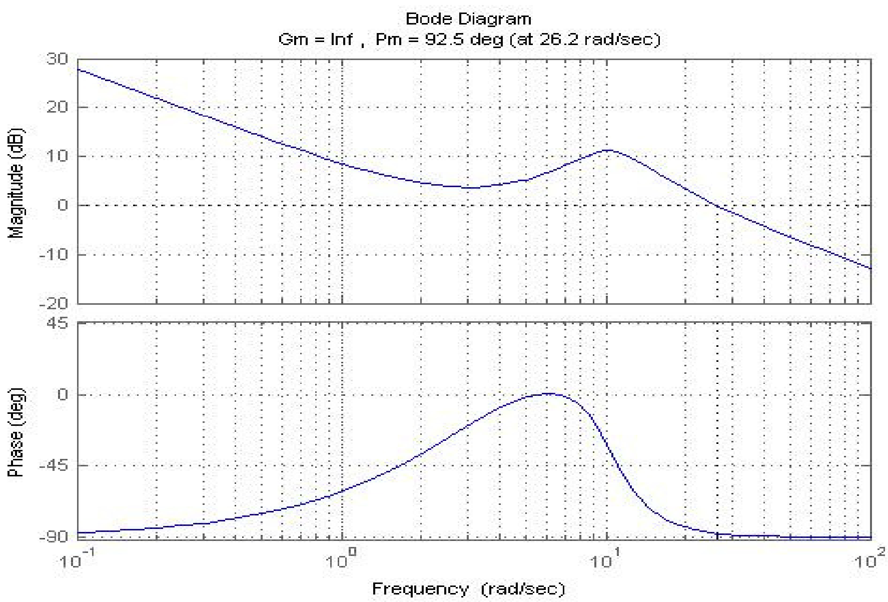

Figure 4.4 (b): Bode Plot for T.F form

4.3.3 COHEN-COON TUNING METHOD

The PID tuning parameters as a function of the open loop model parameters K, T and θ from equation (4.2) as derived by Cohen-Coon:

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.

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.

Table 4.6: C-C Tuning Method

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	$\frac{1}{K}\frac{T}{\theta}\left(\frac{4}{3}+\frac{\theta}{4T}\right)$	$\theta\left(\frac{32+6\theta/T}{13+8\theta/T}\right)$	$\theta\left(\frac{4}{11+2\theta/T}\right)$

Results

Table 4.7: C-C comparison Transfer function and time delay forms for DC machine model

Form	Gain margin	Phase margin	Kp	Ti	Td	Rise time	Settlng time	Overshoot	Modulus margin	Delay margin
T.F.	Inf	80.2498	0.0280	0.4414	0.0776	1.3812	2.6325	0	-1.8847e+003	0.0593

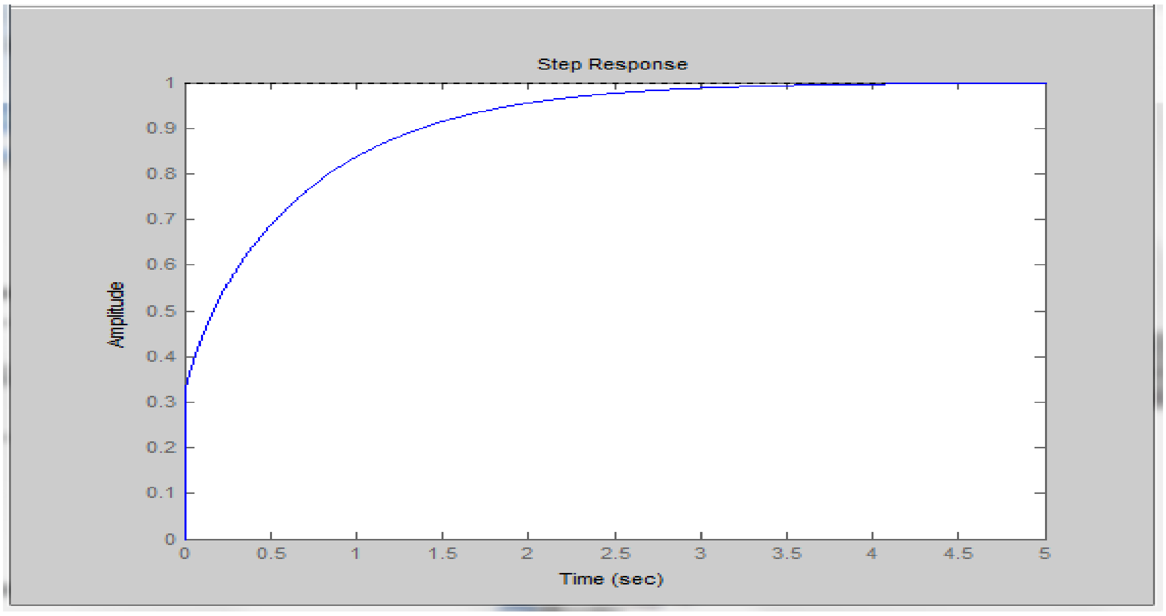

Figure 4.5(a): Step response for T.F form

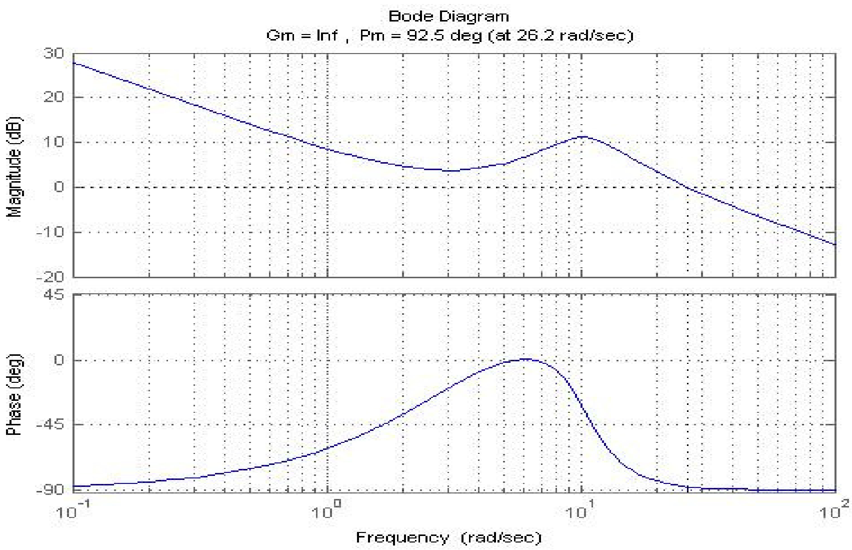

Figure 4.5(b): Bode Plot for T.F form

4.3.4 INTERNAL MODEL CONTROL (IMC) TUNING METHOD

The PID tuning parameters as a function of the open loop model parameters K, T and θ from equation (4.6) as derived by Internal Model Control (IMC).

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 FODPT 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.

Table 4.8: IMC Tuning Method

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

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

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

Results

Table 4.9: IMC comparison Transfer function and time delay forms for DC machine model Lambda=0.0722

Form	Gain margin	Phase margin	Kp	Ti	Td	Rise time	Settlng time	Overshoot	Modulus margin	Delay margin
T.F.	Inf	77.8535	0.0244	0.2928	0.0732	0.9864	1.7939	0	-8.5047e+003	0.0658

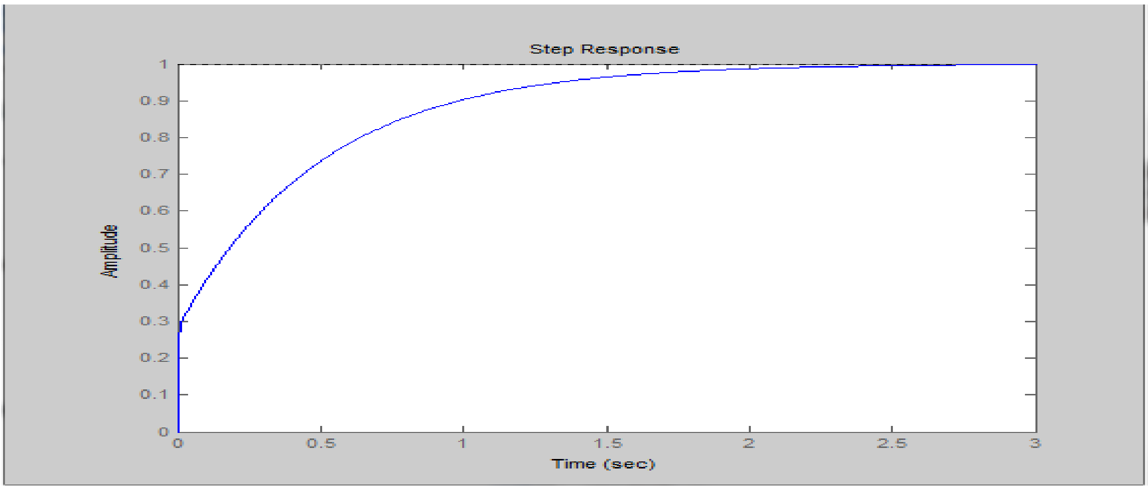

Figure 4.6(a): Step response for T.F. form

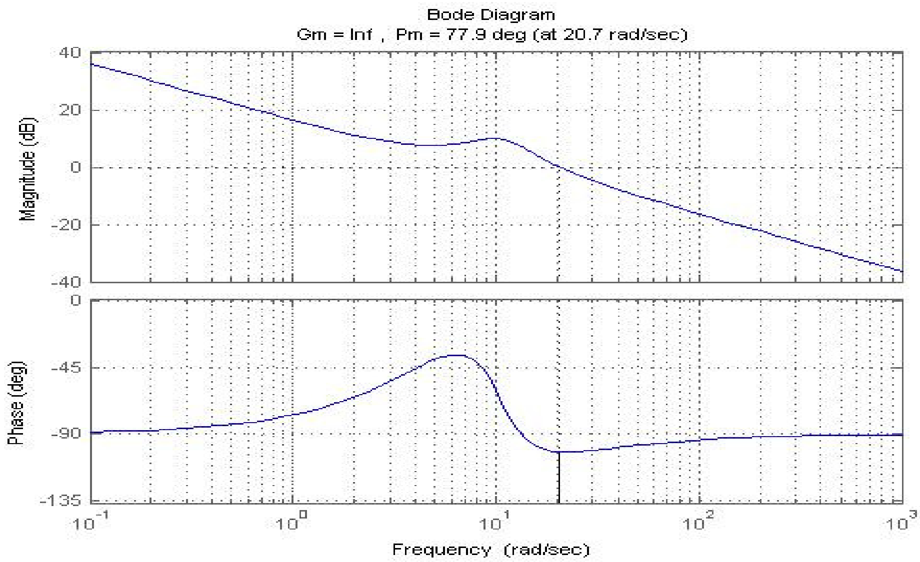

Figure 4.6(b): Step response for T.F. form

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 (4.6) as derived by Approximate M-constrained Integral Gain Optimization (AMIGO).

Table 4.10: AMIGO Tuning Method

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

Result

Table 4.11: AMIGO comparison Transfer function and time delay forms for DC machine model

Form	Gain margin	Phase margin	Kp	Ti	Td	Rise time	Settlng time	Overshoot	Modulus margin	Delay margin
T.F.	Inf	90.9352	0.0130	0.7717	0.0912	4.7751	8.7130	0	-1.1013e+004	0.0979

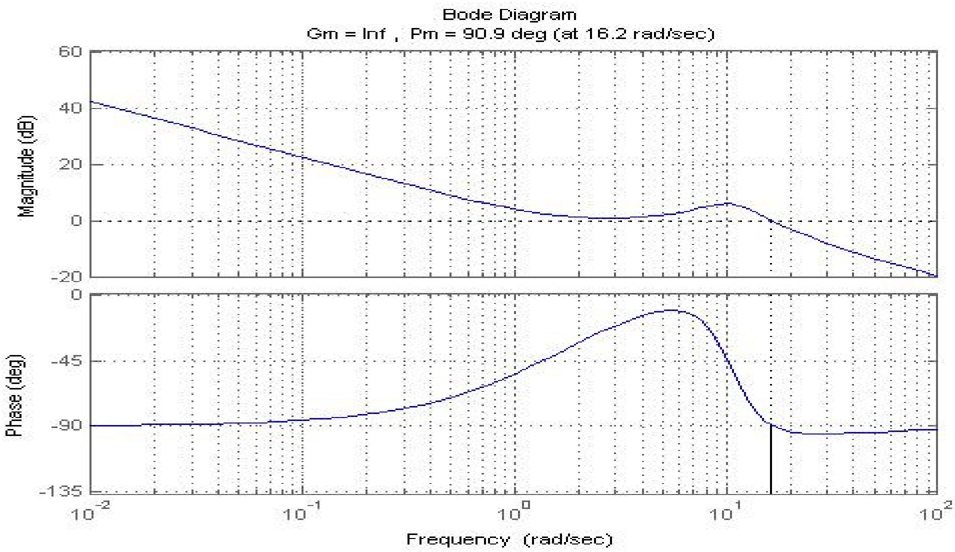

Figure 4.6(c): Bode Plot for T.F. form

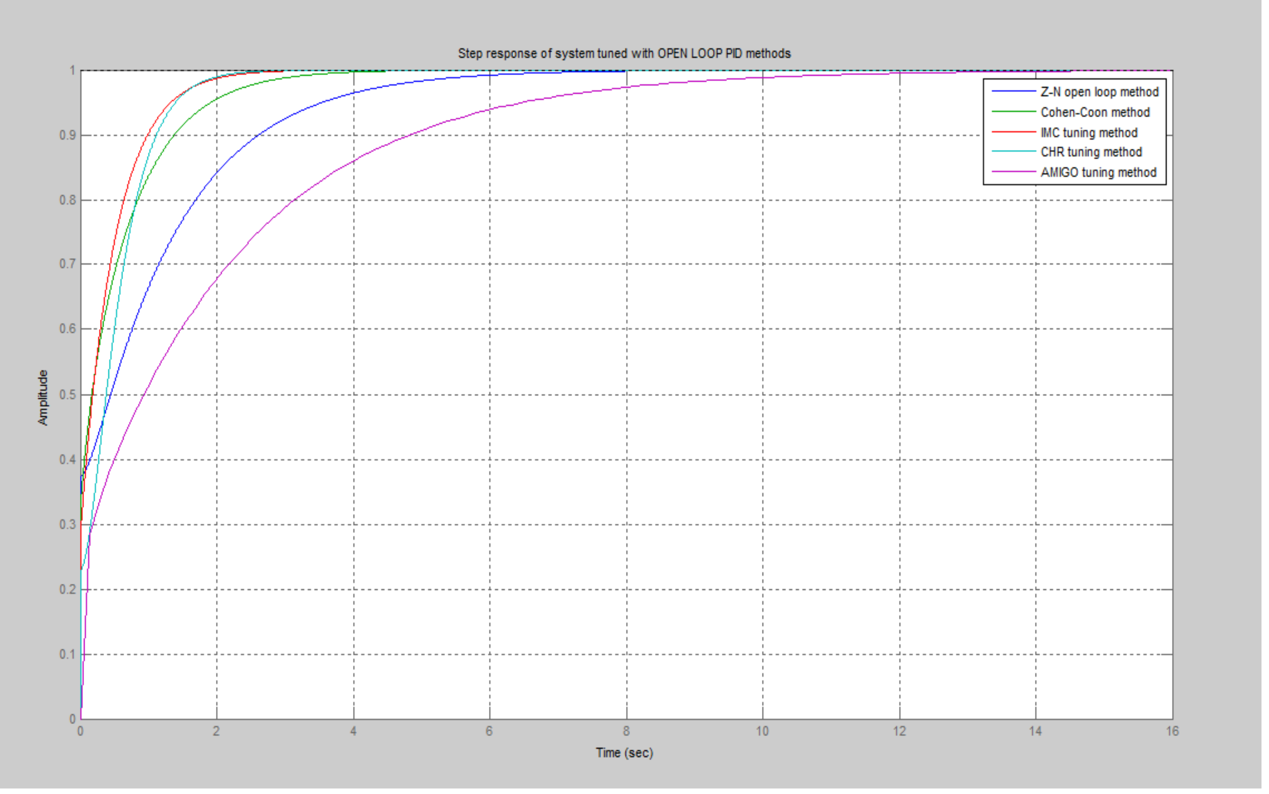

Figure 4.7: Step Response for T.F. Form

4.4 CLOSED LOOP TUNING METHODS

4.4.1 ZIEGLER-NICHOLS TUNING METHOD

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.

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, Ti and TD (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=gain margin of the system and the P_u=(2*pi)/wcg, where, the wcg is the gain cross over frequency. Gain margin is the reverse of amplitude ratio.

Table 4.12: Z-N Tuning Method

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	$0.6K_u$	$P_u/2$	$P_u/8$

Results

Table 4.13: ZN comparison Transfer function and time delay forms for DC machine model

Form	Gain margin	Phase margin	Kp	Ti	Td	Rise time	Settlng time	Overshoot	Modulus margin	Delay margin
T.F.	9.2204	-43.7651	0.9482	0.0071	0.0018	0.061	0.0612	39.3881	5.2104	0.0504

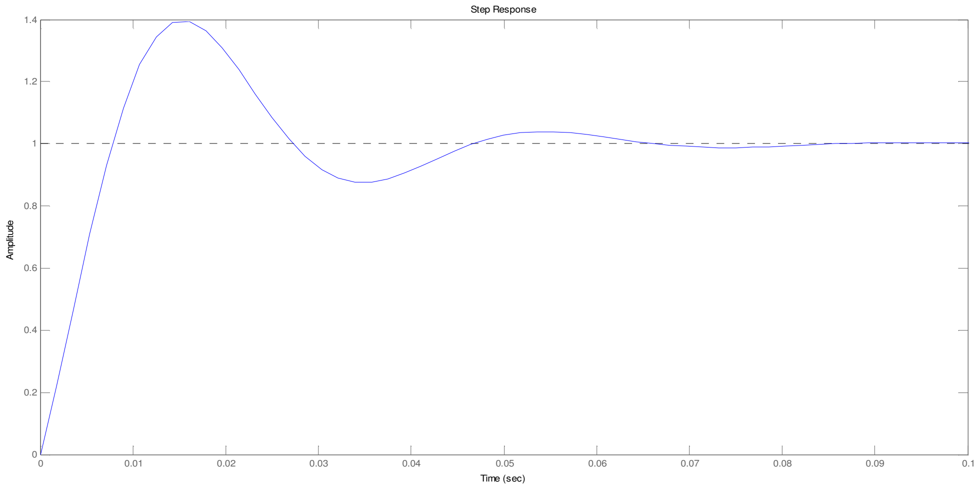

Figure 4.8(a): Step response of T.F. form

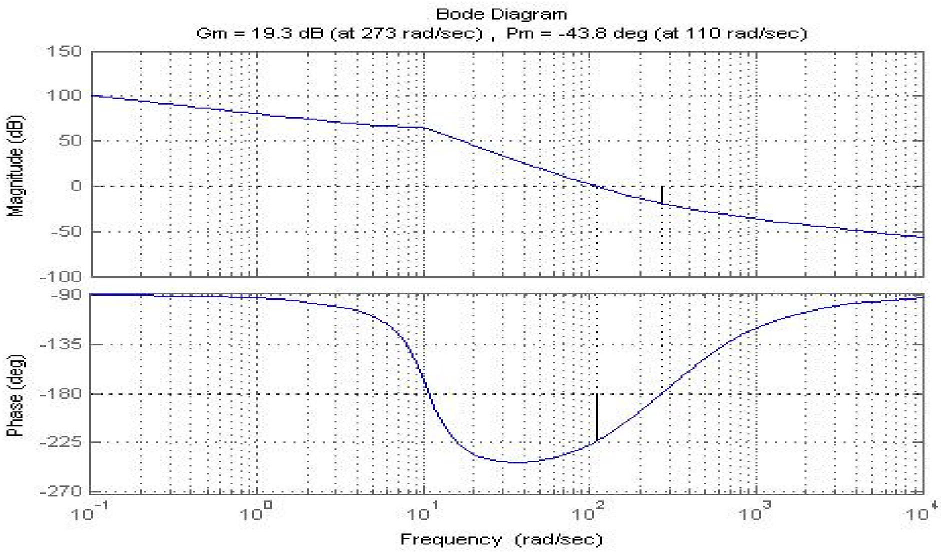

Figure 4.8(b): Bode Plot of T.F. form

4.4.2 MODIFIED ZIEGLER-NICHOLS TUNING METHOD

Table 4.14: Modified Z-N Tuning Method

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

Results

Table 4.15: Modified ZN comparison Transfer function and time delay forms for DC machine model

Form	Gain margin	Phase margin	Kp	Ti	Td	Rise time	Settlng time	Overshoot	Modulus margin	Delay margin
Transfer Function	6.9803	-50.6192	0.3161	0.0071	0.0071	0.0132	0.1875	37.8173	0.3192	0.0768

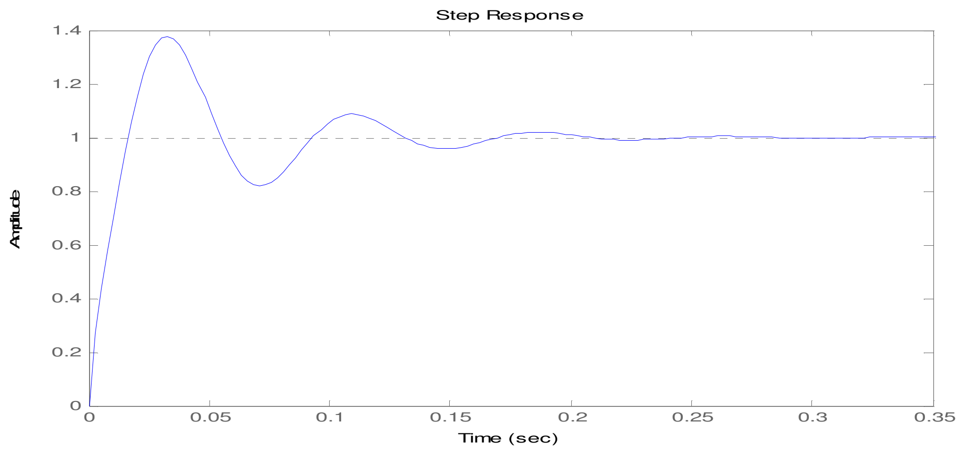

Figure 4.9(a): Step response of T.F. form

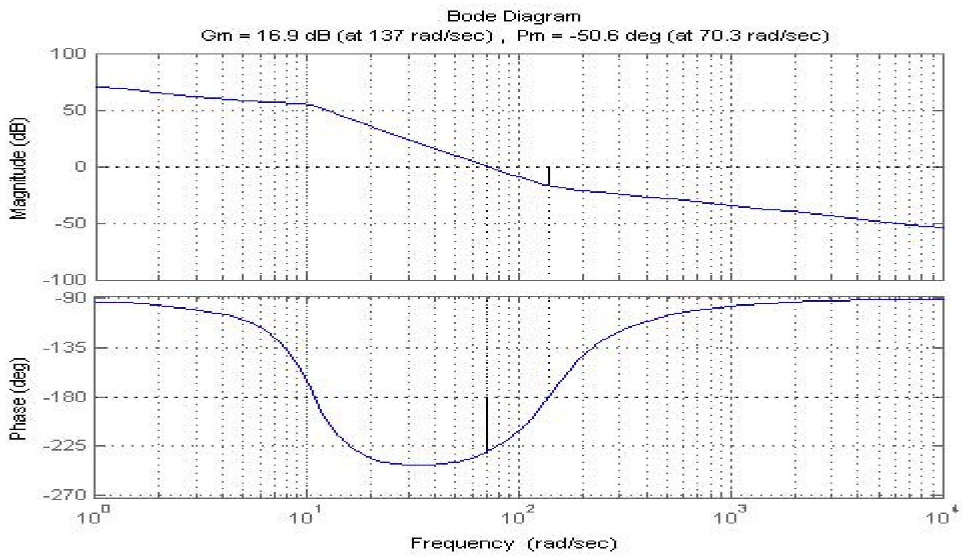

Figure 4.9(b): Bode Plot of T.F. form

4.4.3 TYREUS-LUYBEN TUNING METHOD

Table 4.16: T-L tuning Method

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

Results

Table 4.17: Tyreus-Luyben comparison Transfer function and time delay forms for DC machine model

Form	Gain margin	Phase margin	Kp	Ti	Td	Rise time	Settlng time	Overshoot	Modulus margin	DM
Transfer Function	1.7739	-7.3591	0.7112	0.0313	0.0023	0.0104	0.0721	11.3907	0.0553	0.0777

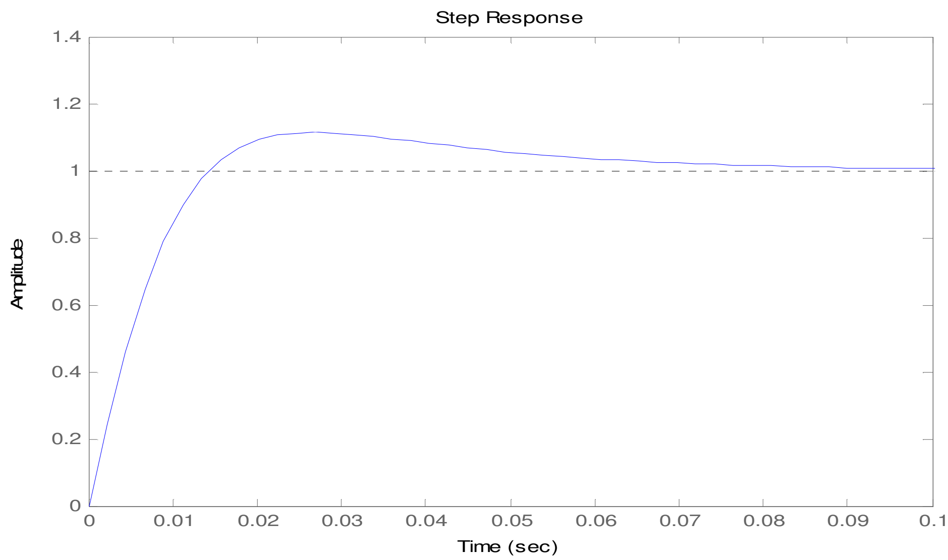

Figure 4.10(a): Step response of T.F. form

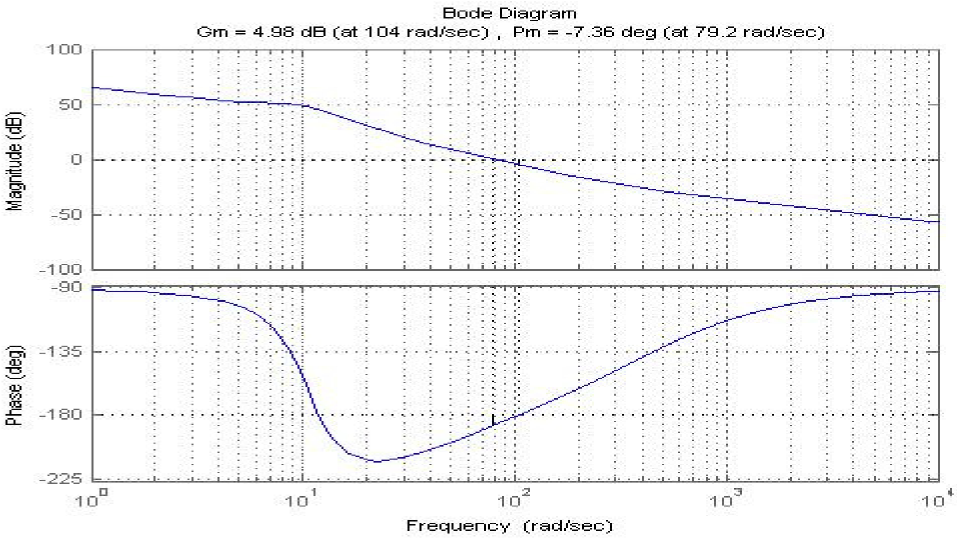

Figure 4.10(b): Bode Plot of T.F. form

4.4.4 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.

$$a = K\,\theta/T$$

Table 4.18: C-H-R Tuning Method

Controller		K_p	T_i	T_d
Chien-Hrones-Reswick Method (Open loop)	PI	$0.47K_u$	1	-

Results

Table 4.19: C-H-R comparison Transfer function and time delay forms for DC machine model

Form	Gain margin	Phase margin	Kp	Ti	Td	Rise time	Settlng time	Overshoot	Modulus margin	Delay margin
Transfer Function	inf	4.3487	0.7428	1	0	0.0115	0.9655	1.7198	0.0446	9.4907e-004

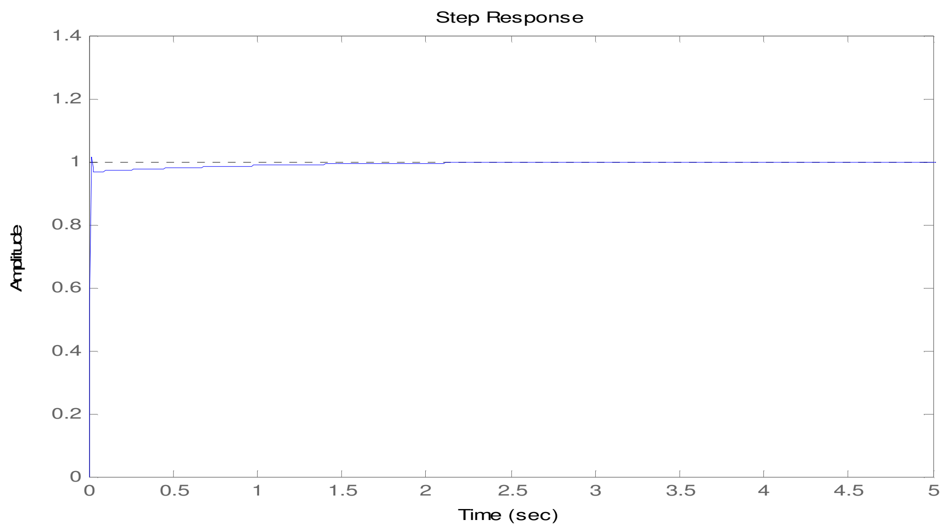

Figure 4.11(a): Step response of T.F. form

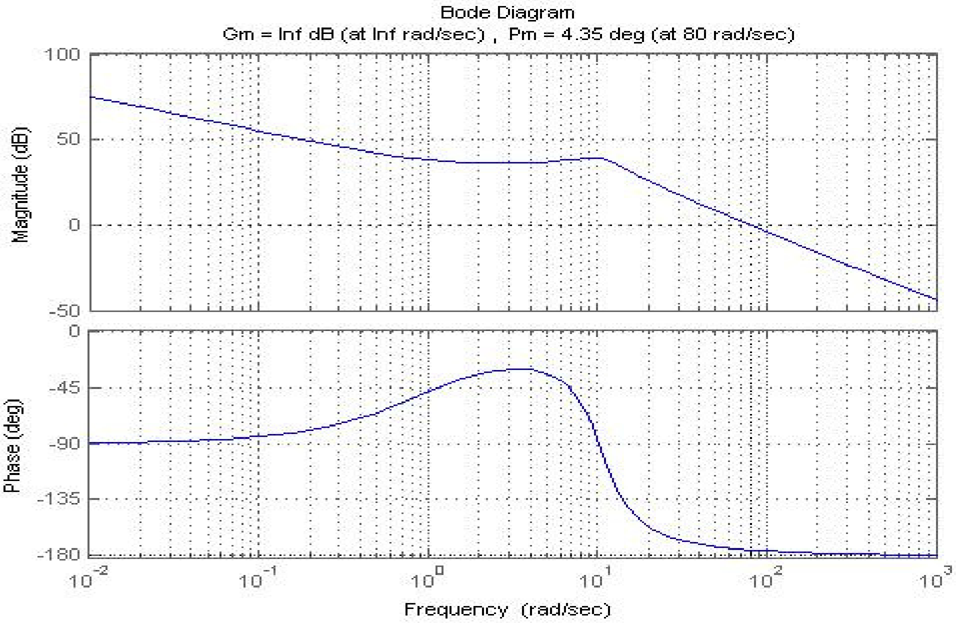

Figure 4.11(b): Bode Plot of T.F. form

4.4.5 ASTROM-HAGGLUND TUNING METHOD

An improvement of the Ziegler-Nichols method is given by Astrom and Haglund. 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:

$$K_u = \frac{4d}{\pi a}$$

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

Table 4.20: A-H Tuning Method

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

Result

Table 4.21: A-H comparison Transfer function and time delay forms for DC machine model

Form	Gain margin	Phase margin	Kp	Ti	Td	Rise time	Settlng time	Overshoot	Modulus margin	Delay margin
Transfer Function	Inf	0.0516	0.5057	0.9400	0	0.0173	0.8676	0	0.0164	1.3741e-007

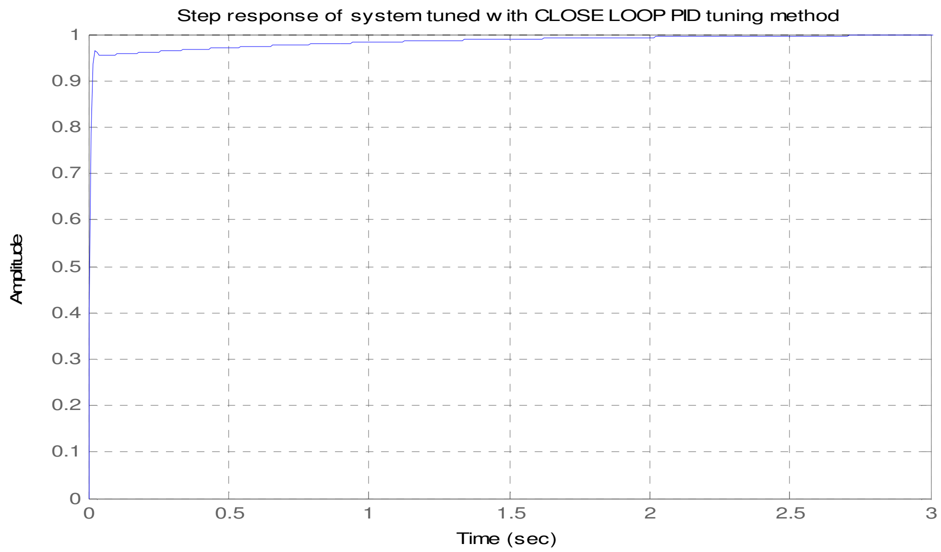

Figure 4.12(a): Step response of T.F. form

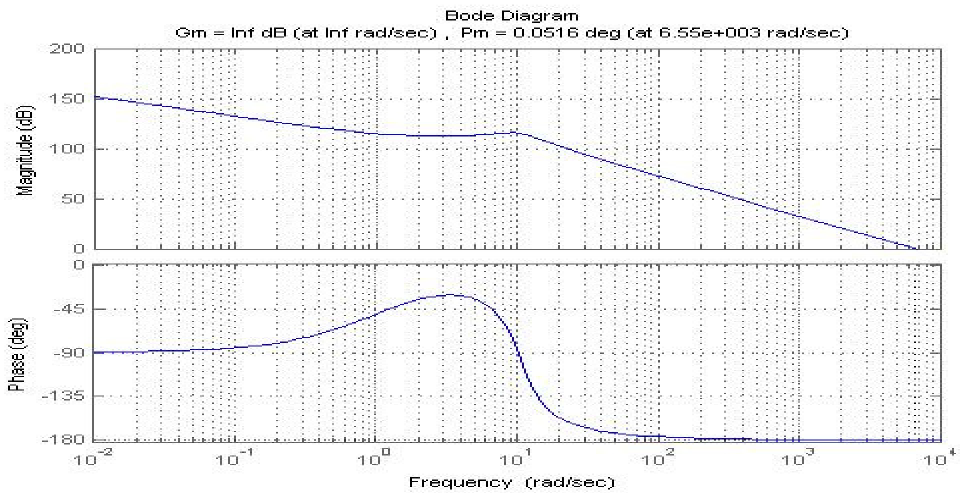

Figure 4.12(b): Bode Plot of T.F. form

CHAPTER 5

OPTIMAL TUNING USING ERROR INDICES

5.1 METHOD I

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

5.1.1) 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 5.1.1

$$Y = A\left(\frac{t_0}{\tau}\right)^B \qquad \text{.... 5.1.1}$$

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:

$$K_c = \frac{A}{K}\left(\frac{t_0}{\tau}\right)^B \qquad \text{.... 5.1.2}$$

$$\frac{1}{T_i} = \frac{A}{\tau}\left(\frac{t_0}{\tau}\right)^B \qquad \text{..... 5.1.3}$$

$$T_d = \tau * A\left(\frac{t_0}{\tau}\right)^B \qquad \text{.... 5.1.4}$$

a) For PI Controller:

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

Table 5.1: Servo Response for PI controller

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

b) For PID Controller:

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

Table 5.2: Servo Response 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

5.1.2) 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

Table 5.3: Regulatory Response 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

b) For PID Controller:

Table below shows value of A and B for PID controller

Table 5.4: Regulatory Response 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

5.2 METHOD II

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 method I.

5.2.1) SERVO RESPONSE

The FOPDT model transfer function is given by:

$$G(s) = \frac{Ke^{-s\tau}}{Ts+1} \qquad \text{... 5.2.1}$$

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)

$$K_c = \frac{a_1}{K}\left(\frac{\tau}{T}\right)^{b_1} \qquad \text{... 5.2.2}$$

$$T_i = \frac{T}{a_2 + b_2(\tau / T)} \qquad \text{... 5.2.3}$$

$$T_d = a_3 T\left(\frac{\tau}{T}\right)^{b_3} \qquad \text{... 5.2.4}$$

a) For PI Controller

The values of coefficients for optimal PI controller tuning is given in following table

Table 5.5: Servo Response 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

b) For PID Controller

The following table shows PID Controller optimal tuning parameter values:

Table 5.6: Servo Response 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

5.2.2) REGULATORY RESPONSE:

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

$$K_p = \frac{a_1}{K}\left(\frac{\tau}{T}\right)^{b_1} \qquad \text{... 5.2.5}$$

$$\frac{1}{T_i} = \frac{a_2}{T}\left(\frac{\tau}{T}\right)^{b_2} \qquad \text{... 5.2.6}$$

$$T_d = a_3 T\left(\frac{\tau}{T}\right)^{b_3} \qquad \text{... 5.2.7}$$

a) For PI Controller

The values of coefficients for PI controller are:

Table 5.7: Regulatory Response for PI controller

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

b) For PID Controller

The values of coefficients for PI controller are:

Table 5.8: Regulatory Response for PID controller

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

Results: Optimal PID Tuning of Servo System for Speed Control

Table 5.9: Optimal PID Tuning for Servo System IAE based

Kp	Ti	Td	Rise time	Settling time	Phase Margin	Overshoot	Delay Margin	Mod Marg.
0.0183	0.3047	0.0949	1.3282	2.3597	85.9274	0	0.0766	-2.0507e+003

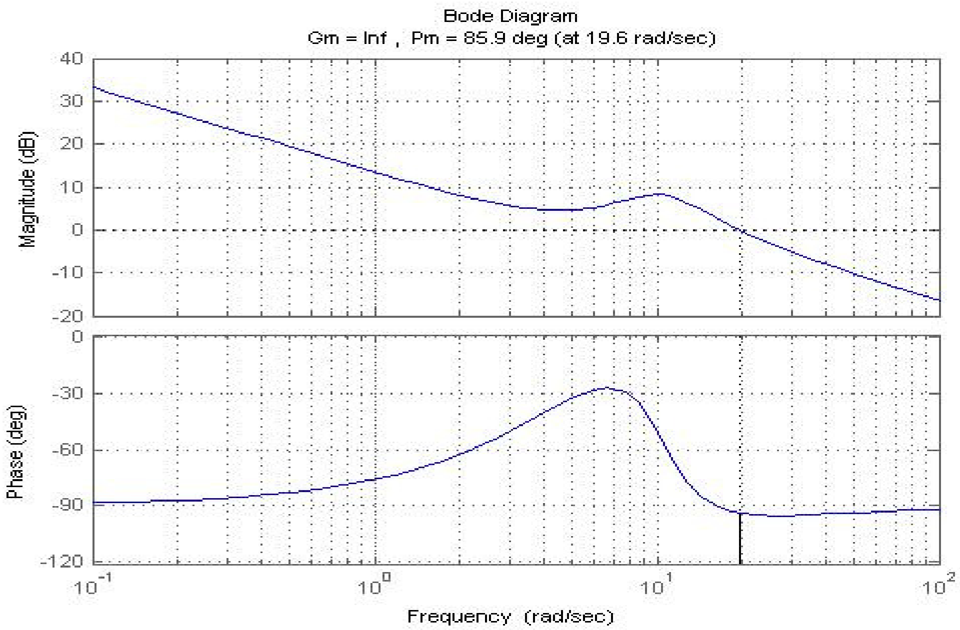

Figure 5.1: Bode Plot of T.F. form

Table 5.10: Optimal PID Tuning for Servo System ISE based

Kp	Ti	Td	Rise time	Settling time	Phase Margin	Overshoot	Delay Margin	Mod Marg.
0.0173	0.3097	0.1310	1.4267	2.4931	91.7641	0	0.0698	-667.6337

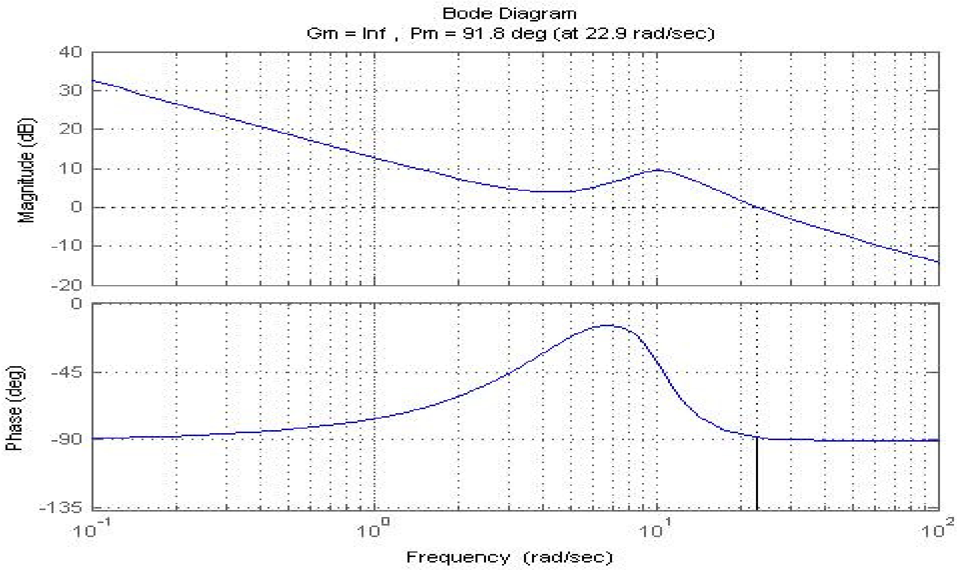

Figure 5.2: Bode Plot of T.F. form

Table 5.11: Optimal PID Tuning for Servo System ITAE based

Kp	Ti	Td	Rise time	Settling time	Phase Margin	Overshoot	Delay Margin	Mod Marg.
0.0165	0.2904	0.0848	1.3778	2.4397	84.1452	0	0.0844	- 8.6686e+003

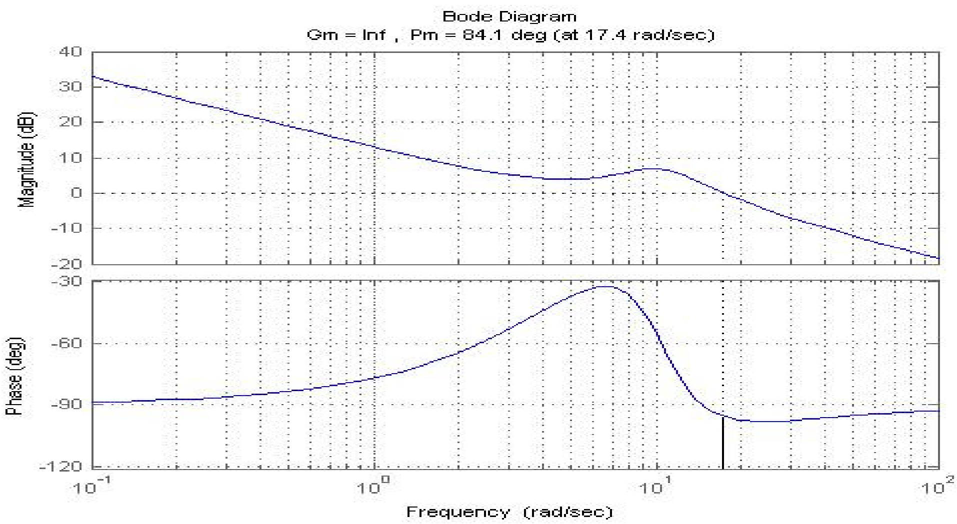

Figure 5.3: Bode Plot of T.F. form

Table 5.12: Optimal PID Tuning for Servo System ISTE based

Kp	Ti	Td	Rise time	Settling time	Phase Margin	Overshoot	Delay Margin	Mod Marg.
0.0172	0.2832	0.1044	1.2950	2.2678	88.2176	0	0.0780	-6.6442e-004

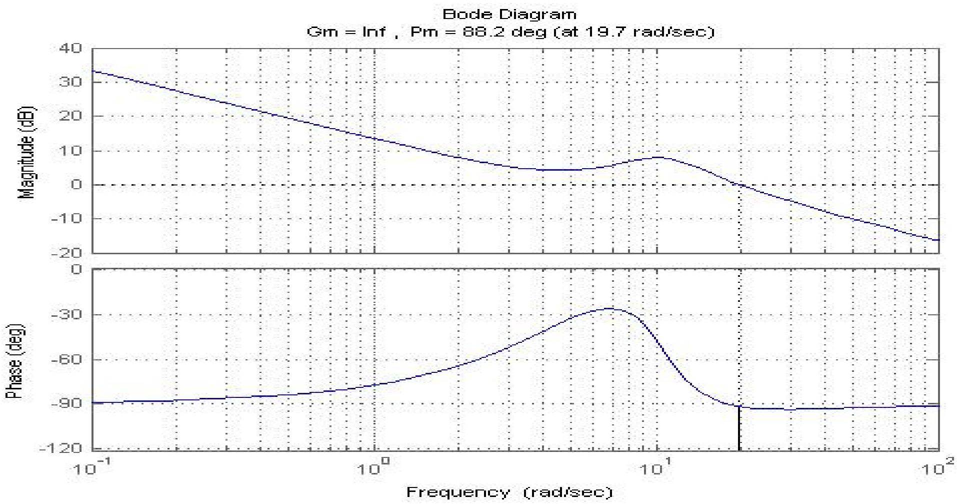

Figure 5.4: Bode Plot of T.F. form

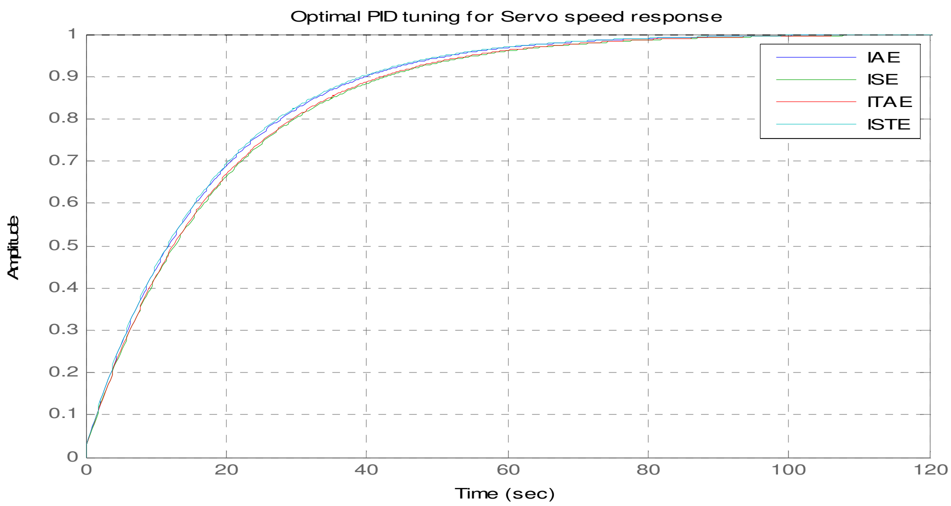

Figure 5.5: Optimal PID Tuning for Servo Speed Response

5.3 Conclusion

Among the results of various error indices calculated, ISTE performs better because here rise time and settling time is very less and overshoot is also zero. Table 5.12 and Figure 5.1 shows ISTE is more stable than the other methods.

In frequency domain only satisfied phase margin not gain margin in all four method and in case of robustness all four method satisfied modulus margin and delay margin.

CHAPTER 6

OPTIMAL TUNINIG OF PID CONTROLLER USING EVOLUTIONARY ALGORITHM

6.1 Introduction

Evolutionary algorithms (EA) are stochastic optimization techniques based on natural evolution and survival of the fittest strategy found in biological organisms. Evolutionary algorithms have been successfully applied to solve complex optimization problem in business, engineering, and science. Some commonly used EAs are Genetic algorithms (GA's), Evolutionary Programming (EP), Evolutionary Strategy (ES) and Differential Evolution (DE). Each of these methods has its own characteristics, strengths and weaknesses. In general, an EA algorithm can generate a set of initial solutions randomly based on the given seed and population size. Afterwards, it will go through evolution operations such as cross-over and mutation before evaluated by the objective function. The wining entity in the population will be selected as the parents (or seed) of the next generation. The optimization iteration continues until the termination criteria are satisfied. Typically, either the evolution process reached users define maximum number of iteration or the improvement in objective function between the two generations converges.

6.2 Aims of this topic:-

Te most important aim of this topic is to describe what an evolutionary algorithm is. This description is deliberated based on a unifying view presenting a general scheme that forms the common basis of all evolutionary algorithm variants.

6.3 What is an Evolutionary Algorithm?

As the history of the field suggest there are many different variants of Evolutionary Algorithm. The common underlying idea behind all these techniques is the same: given a population of individuals the environmental pressure causes natural selection (survival of the fittest) and this causes a rise in the fitness of the population. Given a quality function to be maximized we can randomly create a set of candidate solution, i.e. , elements of the function's domain, and apply the quality function as an abstract fitness measure – the higher the better. Based on this fitness, some of the better candidates are chosen to seed the next generation by applying recombination and /or mutation to them. Recombination is an operator applied to two or more selected candidates (the so-called parents) and results in one candidate and results in one new candidate (the children). Mutation is applied to one candidate and results in one candidate. Executing recombination and mutation leads to a set of new candidates (the offspring) that compete – based on their fitness (and possible age) – with the old ones for a place in the next generation.

In this process there are two fundamental forces that form the basis of evolutionary systems.

- Variation operators (recombination and mutation) create the necessary diversity and thereby facilitate novelty.
- Selection acts as a force pushing quality.

The combined application of variation and selection generally leads to improving fitness values in consecutive populations. It is easy to see such a process as if the evolution is optimizing, or at least "approximating", by approaching optimal values closer and closer over its course. Alternatively, evolution it is often seen as a process of adaptation. From this perspective, the fitness is not seen as an objective function to be optimized, but as an expression of environmental requirements. Matching these requirements more closely implies an increased viability, reflected in a higher number of offspring. The evolutionary process makes the population adapt to the environment better and better. The evaluation (fitness) function represents a heuristic estimation of solution quality and the search process is driven by the variation and the selection operators. Evolutionary algorithms (EA) posses a number of features that can help to position them within the family of generate-and-test methods:

- Ea are population based, i.e., they process a whole collection of candidate solution simultaneously.
- EAs mostly use recombination to mix information of more candidate solution into a new one.
- EAs are stochastic.

Typically, the candidate are represented by (I.e., the data structure encoding a solution has the form of) string over a finite alphabet in genetic Algorithm (GA), real-valued vectors in Evolutionary Programming (EP) and trees in Genetic Programming (GP).

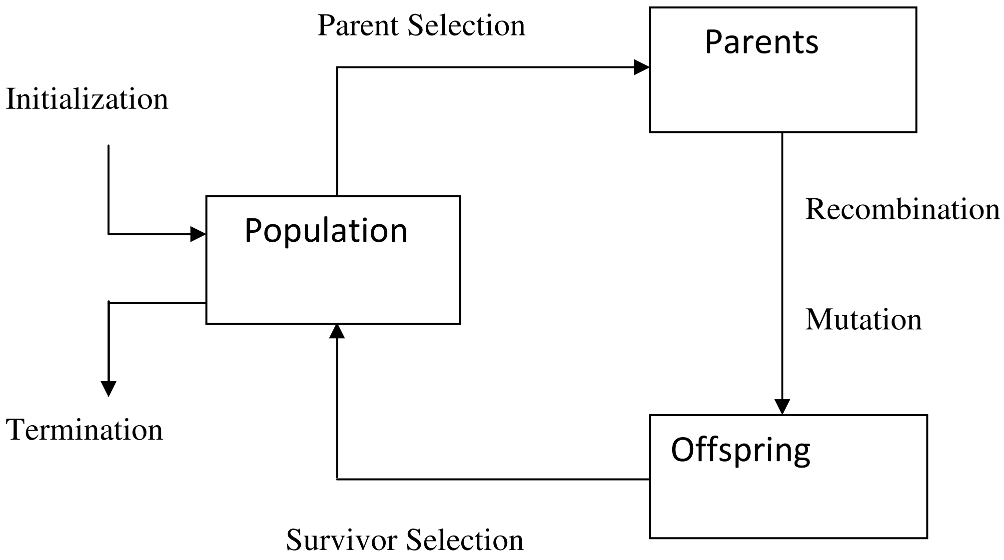

Figure 6.1: The general scheme of an Evolutionary Algorithm as a Flow-Chart

6.4 Components of Evolutionary Algorithms

In this section we discuss Evolutionary Algorithm in. EAs have a number of components, procedures or operators that must be specified in order to define a particular EA.

- Representation (definition of individuals)
- Evaluation function (or fitness function)
- Population
- Parent selection mechanism
- Variation operators, recombination and mutation
- Survivor selection mechanism (replacement)

6.4.1 Representation (Definition of individuals)

The first step in defining an EA is to link the “real world” to the “EA world”, that is to set up a bridge between the original problem context and the problem solving space where evolution will take place. Objects forming possible solutions within the original problem context are referred to as phenotypes, there encoding, the individuals within the EA, are called genotypes. The first design step is commonly called representation. Here one could decide to represent them by their binary code, hence 18 would be seen as a phenotype and 10010 as a genotype representing it. It is important to understand that the phenotype space can very different from the genotype space. A solution – a good phenotype- is obtained by decoding the best genotype after termination. To this hold end, it should hold that the (optimal) solution to the problem at hand – a phenotype – is represented in the given genotype space. It should be noted that the world "representation" is used in two slightly different ways. Sometimes it stands for the mapping from the phenotype to the genotype space. In this sense it is synonymous with encoding, e.g., one could mention binary representation or binary encoding of candidate solutions. The inverse mapping from genotype to phenotype is usually called decoding and it is required that the representation be invertible : to each genotype there has to be at most one corresponding phenotype. The word representation can also be used in a slightly different sense, where the emphasis is not on the mapping itself, but on the "data structure" of the genotype space. This interpretation is behind speaking about mutation operators for binary representation, for instance.

6.4.2 Evaluation Function (Fitness Function)

The role of the evaluation function is to represent the requirements to adapt to. It forms the basis for selection, and thereby it facilitates improvements. It represents the task to solve in the evolutionary context. Technically, it is a function or procedure that assigns a quality measure to genotypes. Typically, this function is composed from a quality measure in the phenotype space and the inverse representation. To remain with the above example, if we were to maximize x^2 On this integer, the fitness of the genotype 10010 could be defined as the square of its corresponding phenotype:$18^2=324$.

The evolution function is commonly called the fitness function in EC.

6.4.3 Population

The role of population is to hold (the representation of) possible solutions. A population is a multi-set of genotypes. The population forms the unit of evolutions. For instance, the best individual of the worst individual of the given population is chosen to seed the next generation, or the worst individual of the given population is chosen to be replaced by new one. In almost all EA applications the population size is constant, not changing during the evolutionary search.

The diversity of a population is a measure of the number of different solution present. No signal measure for diversity exists; typically people might refer to the number of different fitness values present, the number of different genotypes. Other statically measures, such as entropy ,are also used. Note that only one fitness value does not necessarily imply only one phenotype is present, and in turn only one phenotype does not necessarily imply only genotype. The reverse is however not true: one genotype implies only one phenotype and fitness value.

6.4.4 Parent Selection Mechanism

The role of parent selection or mating selection is to distinguish among individuals based on their quality, in particular, to allow the better individuals to become parents of the next generation. An individual is a parent if it has been selected to undergo variation in order to create

offspring. Together with the survivor selection mechanism, parent selection is responsible for pushing quality improvements.

6.4.5 Variation Operators

The role of variation operators is to create new individuals from old ones. In the corresponding phenotype space this amount to generating new candidate solutions.

6.4.6 Mutation

A unary variation operator is commonly called mutation. It is applied to one genotype and delivers a (slightly) modified mutant, the child or off spring of it. A mutation operator is always stochastic: its output – the child – depends on the outcomes of a series of random choices. It should be noted that an arbitrary unary operator is not necessarily seen as mutation.

A problem specific heuristic operator acting on one individual could be termed as mutation for being unary. However, in general mutation is supposed to cause a random, unbiased change.

6.4.7 Recombination

A binary variation operator is called recombination or crossover. The principal behind recombination is simple – that by mating two individuals with different but desirable features, we can produce an offspring which combines both of those features. This principal has a strong supporting case it is one which has been successfully applied for millennia by breeders of plant s and livestock, to produce species which give higher yields or have other desirable features. Evolutionary Algorithm create a number of offspring by random recombination, accept that some will have undesirable combinations of traits, most may be no better or worse than their parents, and hope that some have improved characteristics.

It is important to note that variation operators are representation dependent. That is, for different representations different variation operators have to be defined. For example, if genotype are bit – strings, then inverting a 0 to a 1(1 to a 0) can be used as a mutation operator. However, if we represent possible solutions by tree – like structures another mutation operator is required.

6.4.8 Survivor Selection Mechanism (Replacement)

The role of survivor selection or environmental selection is to distinguish among individuals based on their quality. In that it is similar to parent selection, but it is used in a different stage of the evolutionary cycle. The survivor selection mechanism is called after having created the offspring of the selected parents.

Survivor selection is also often called replacement or replacement strategy. In many cases the two terms can be used interchangeably. The choice between the two is thus often arbitrary.

6.4.9 Initialization

Initialization is kept simple in most EA applications: The first population is seeded by randomly generated individuals. In principal, problem specific heuristics can be used in this step aiming at an initial population with higher fitness. Whether this is worth the extra computational effort or not is very much depending on the application at hand.

6.5 Fields of application of EAs

- EAs have been thoroughly used in many domains. One of the most conspicuous fields in which these techniques have been utilized is combinational optimization (CO)
- Telecommunication is another field that has witnessed the successful application of EAs.
- EAs have been actively used in electronics and engineering as well.

6.6 Evolutionary Algorithms in Control System Engineering

With the increase of computational power in computers, evolutionary com-pupation has become more applicable to complex linear and nonlinear control problems. Complex design and optimization problems, uncertainties of the plant dynamics and lack of well known procedures of synthesis, especially in nonlinear control, have been the key factors for the spread of new synthesis and tuning techniques by means of evolutionary computation. A considerable revival of PID control in the last ten years, for instance, is also caused by the introduction of automatic tuning and new procedures to automatically optimize the performance of a given system (Astraom,Haagglund 2001). Fleming, Purshouse (2002) say that the evolutionary algorithm is a robust search and optimization methodology that is able to cope with ill-behaved problem domains, exhibiting attributes such as multi-modality, discontinuity, time-variance, randomness, and noise." They also explain that a control problem rarely has a single solution and it is generally suitable for a family of non-dominated solutions. 'These Pareto optimal (PO) solutions are those for which no other solution can be found which improves on a particular objective without detriment to one or more other objectives'. (Fleming, Purshouse 2002) Controlled systems appear to be strongly attracted by classes of Pareto optimal solutions. In most of the cases one solution is preferred to another for marginal reasons. If a control problem is specified with strict constraints, it is likely to have a unique optimum solution for the problem. Contrary, if some system specifications are left free to vary in a certain range, several solutions might be suitable for the control problem. Evolutionary computation has been proven to be an excellent tool to explore similar solutions given the parallel approach of the search. A broad range of applications is captured under two major classifications: online design and on-line optimization line application have proved to be the most popular and successful. The reason is related to the computational effort and the uncertainty of the results. EAs in control engineering evolve solutions through the simulation and evaluation of hundreds and often thousands of candidates. The process, in comparison to other synthesis techniques, is computationally very expensive and time demanding. Industrial controllers often require adaptive law or tuning and optimization procedures in the time scale of seconds. In that case, tasks that require intensive computation are often executed by dedicated hardware circuits designed for the purpose. However, EAs are in most of the cases too complex to be implemented in hardware. An others factor that makes on-line applications difficult is the uncertainty of the result of the search process caused mainly by the stochastic nature of the algorithm. While standard techniques are usually designed to obtained a solution with certain characteristic in a determined amount of time or calculations, EAs do not guarantee neither to reach a target solution nor to do it in a fixed amount of time. For mission-critical and safety-critical applications, such as aircraft control, navigation systems and every kind of processes that cannot be suspended, this uncertainty in the results excluded the use of EAs. Given this premise, it is clear how on-line computation has too strict requirements that are unlikely met by EAs. A third weakness of solutions obtained by EAs is related to the mechanism of synthesis itself. The solution found at the end of the computation is evolved applying the Darwinian principle of the survival of the fittest.

6.7 Results

In this chapter all result related to evolutionary algorithm on different controller. Here values of Kp, Ti and Td are obtained using evolutionary algorithm.

Ideal PID controller

Table 6.1: Ideal PID controller

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	Over shoot	Mod. margin	Delay Margin
T.F.	Inf	88.9723	.9478	.25	.0625	7.7878e-004	0.1400	0	-94.4513	0.0031

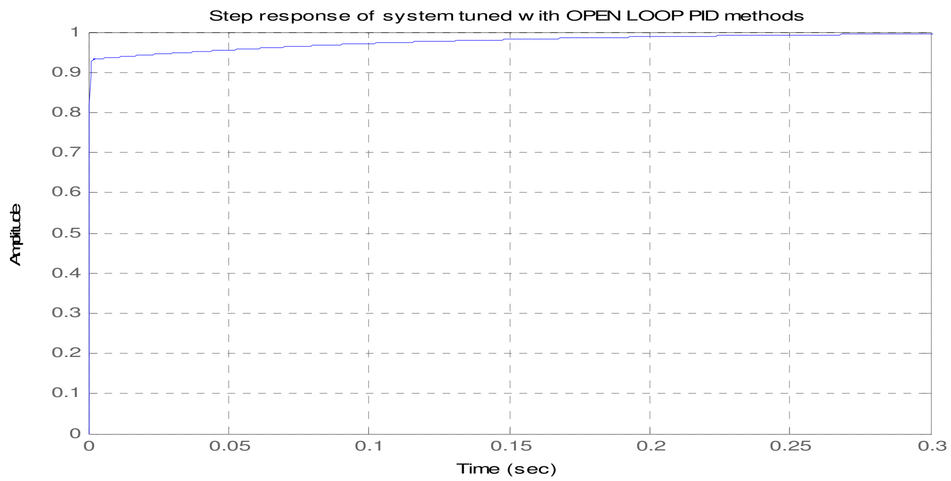

Figure 6.2(a): Ideal PID controller

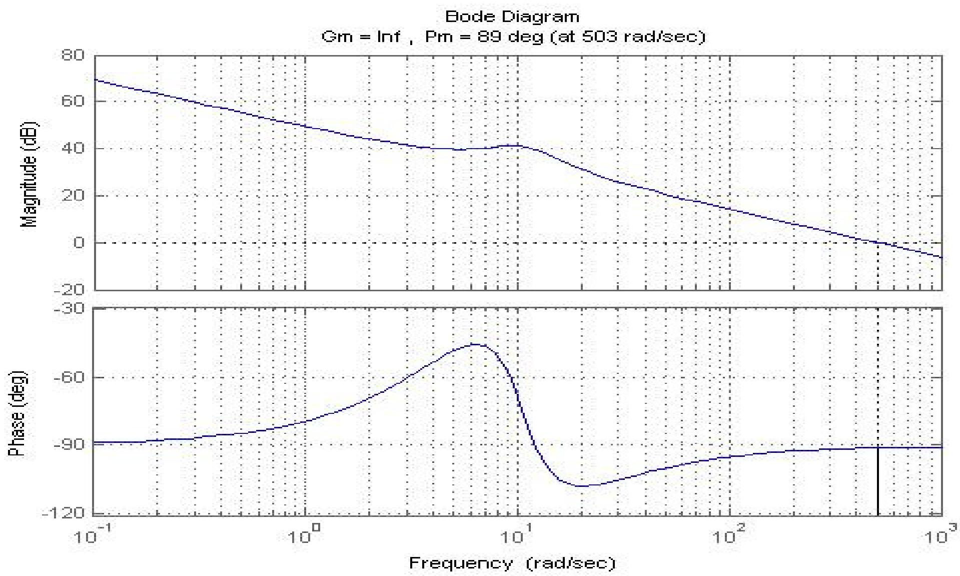

Figure 6.2(b): Bode Plot of T.F.

Ideal PID controller with first order filter

Table 6.2: Ideal PID controller with 1st order filter

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	Overshoot	Mod. margin	Delay Margin
T.F.	inf	89.4915	.9478	.25	.0625	3.2597e-004	0.0645	0	-2.3149e+003	0.0016

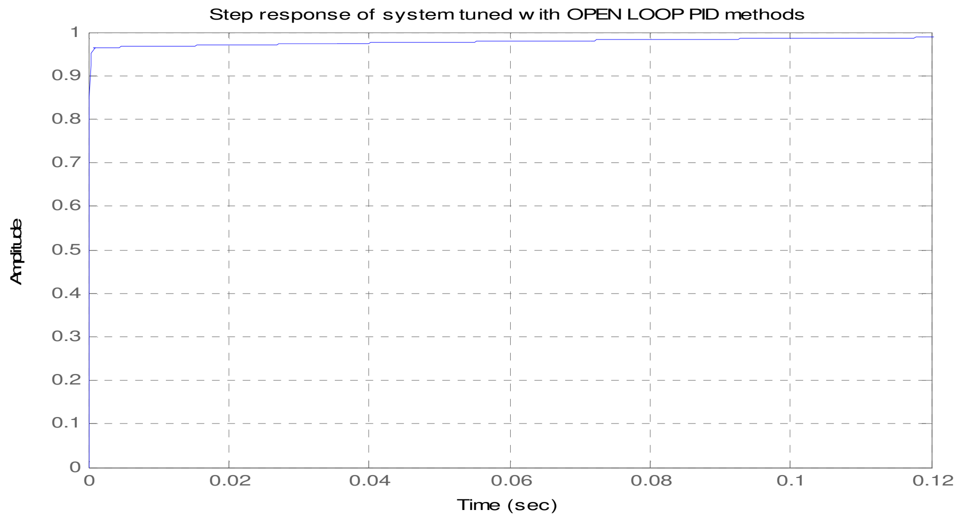

Figure 6.3(a): Ideal PID controller with 1st order filter

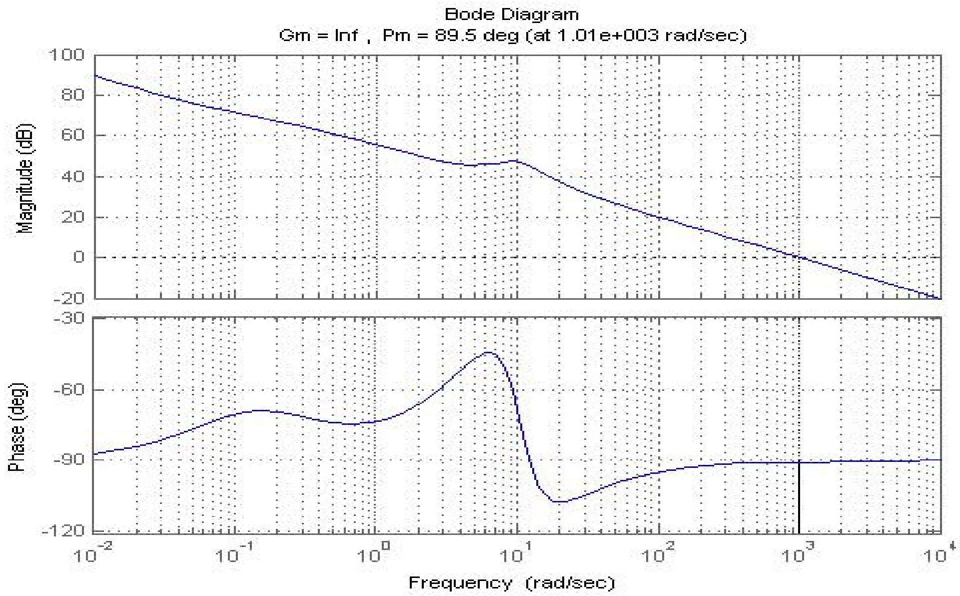

Figure 6.3(b): Bode Plot of Ideal PID controller with 1st order filter

Ideal PID controller with second order filter

Table 6.3: Ideal PID controller with second order filter

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	Overshoot	Mod. margin	Delay Margin
T.F.	0.0483	-9.2787	.9478	.25	.0625	0.0316	0.2559	10.8284	0.0633	0.1435

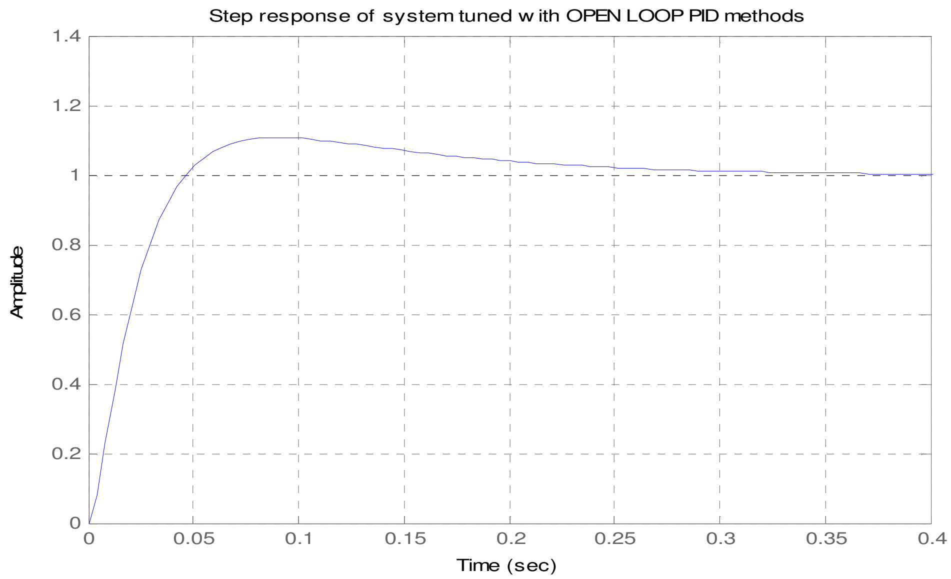

Figure 6.4(a): Ideal PID controller with second order filter

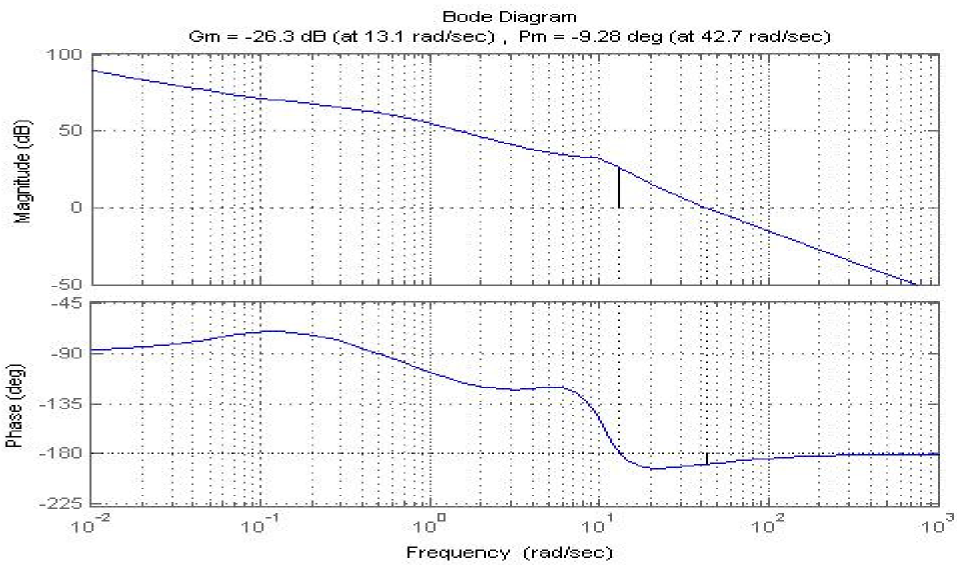

Figure 6.4(b): Bode Plot of Ideal PID controller with second order filter

Ideal PID controller in series with a first order lag

Table 6.4: Ideal PID controller in series with a first order lag

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	Overshoot	Mod. Margin	Delay Margin
T.F.	0.2711	-16.4063	.9478	.25	.0625	0.0316	0.2559	10.8284	0.0935	0.3107

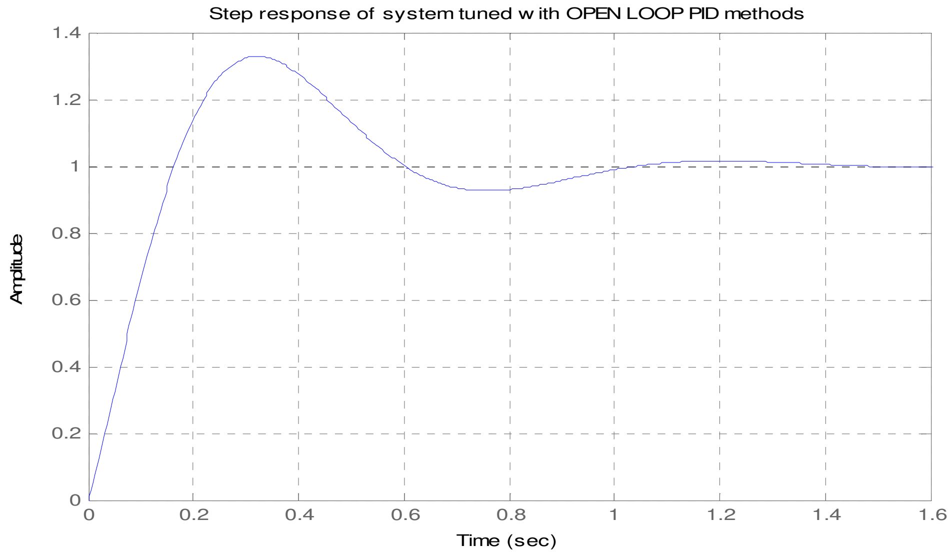

Figure 6.5(a): Ideal PID controller in series with a first order lag

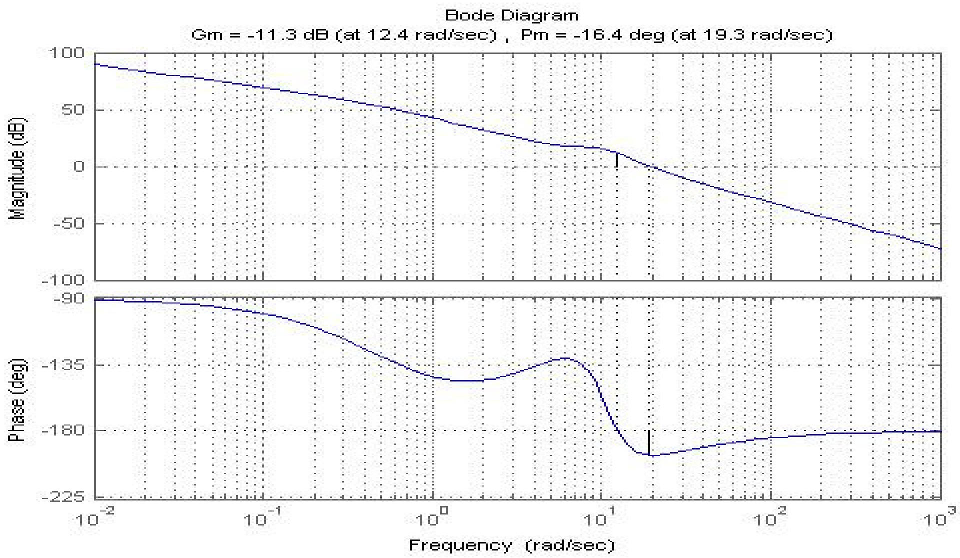

Figure 6.5(b): Bode Plot Ideal PID controller in series with a first order lag

Ideal PID controller weighted terms

Table 6.5: Ideal PID controller weighted terms

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	Overshoot	Mod. margin	Delay Margin
T.F.	inf	88.9723	.9478	.25	.0625	7.7878e-004	0.1400	0	-94.4513	0.0031

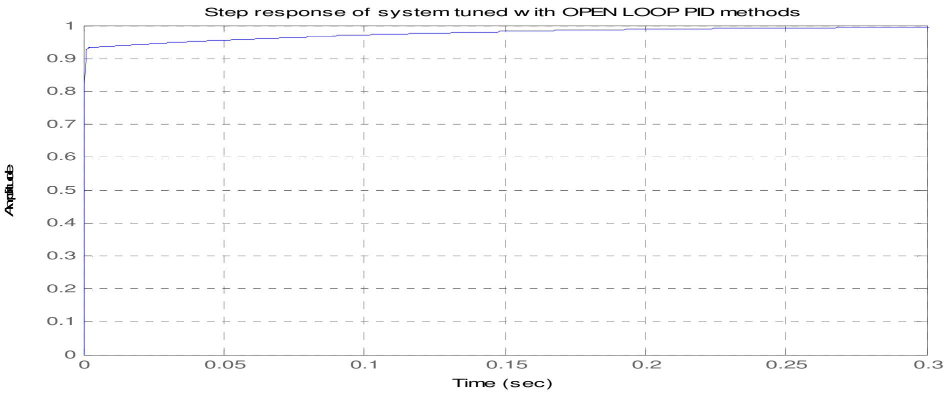

Figure 6.6(a): Ideal PID controller weighted terms

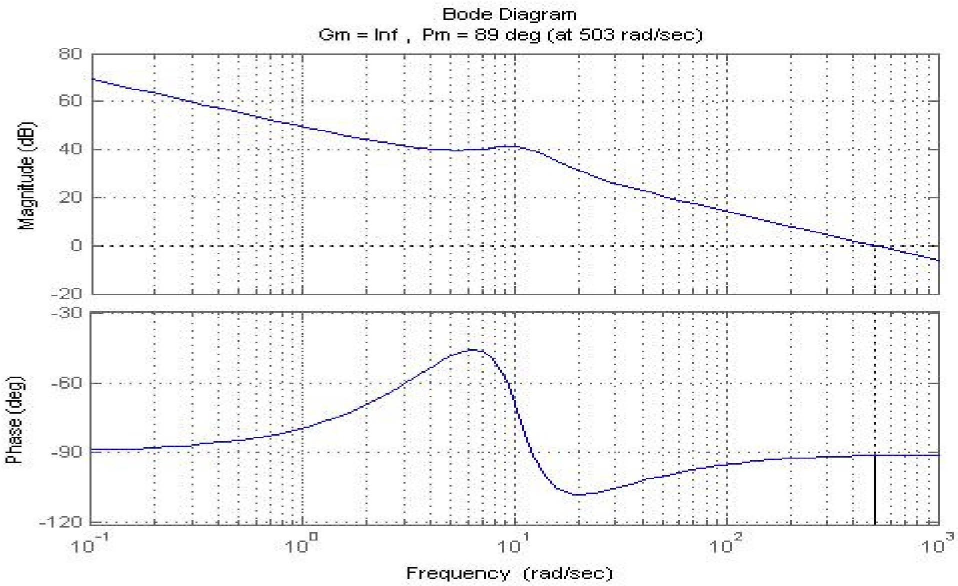

Figure 6.6(b): Bode Plot of Ideal PID controller weighted terms

Ideal PID controller with industrial controller

Table 6.6: Ideal PID controller with industrial controller

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	Overshoot	Mod. margin	Delay Margin
T.F.	inf	9.8126	.9478	.25	.0625	2.1270e-004	0.0348	86.8606	0.0652	9.5815e-005

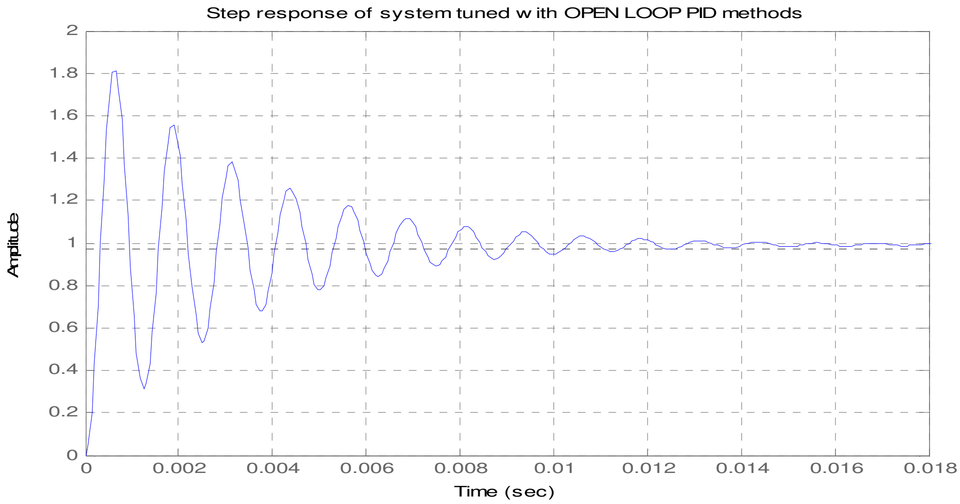

Figure 6.7(a): Ideal PID controller with industrial controller

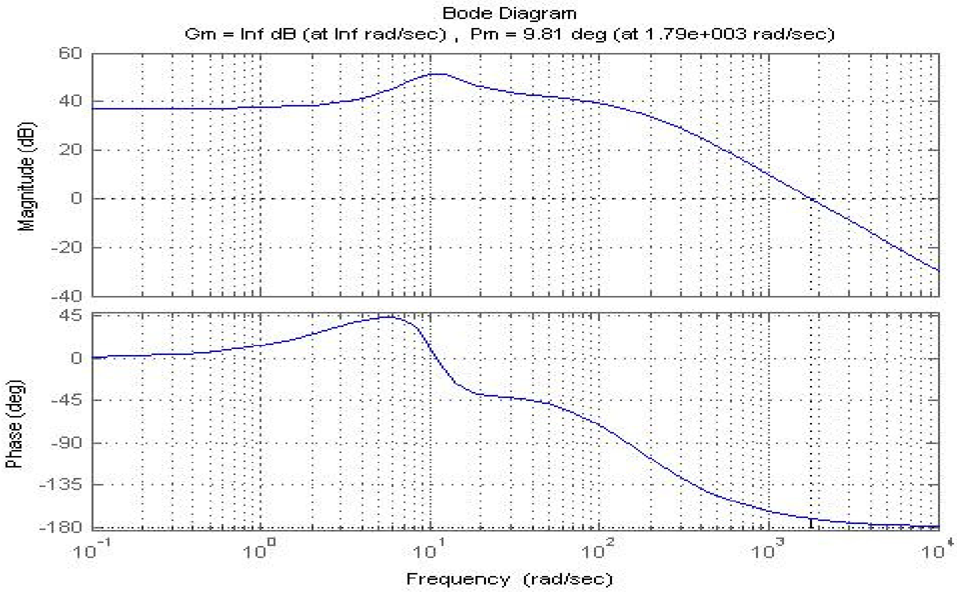

Figure 6.7(b): Bode Plot of Ideal PID controller with industrial controller

Series Ideal PID controller

Table 6.7: Series Ideal PID controller

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	Overshoot	Mod. margin	Delay Margin
T.F.	inf	88.5180	.9478	.25	.0625	7.7034e-004	0.1136	0	-96.0330	0.0031

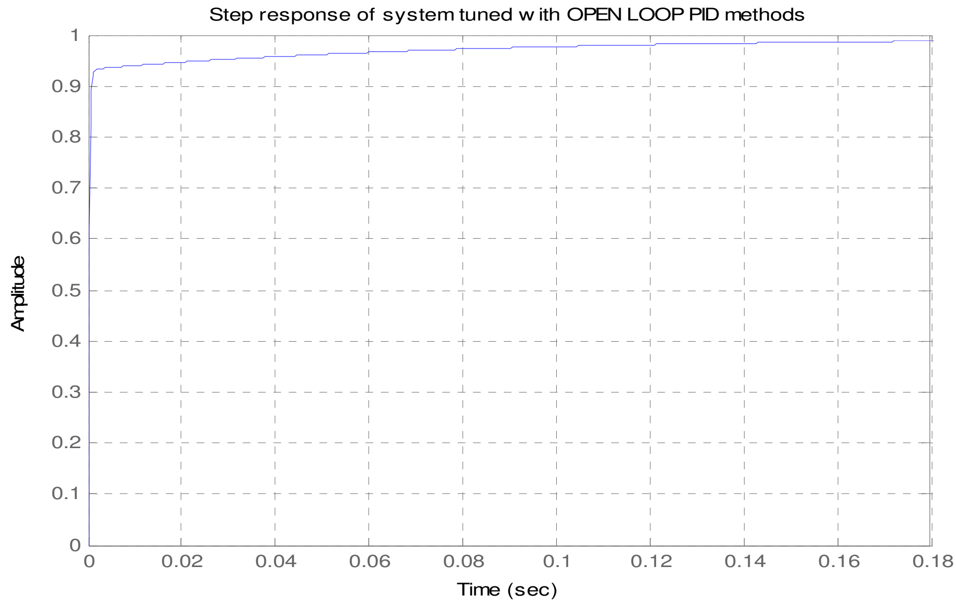

Figure 6.8(a): Series Ideal PID controller

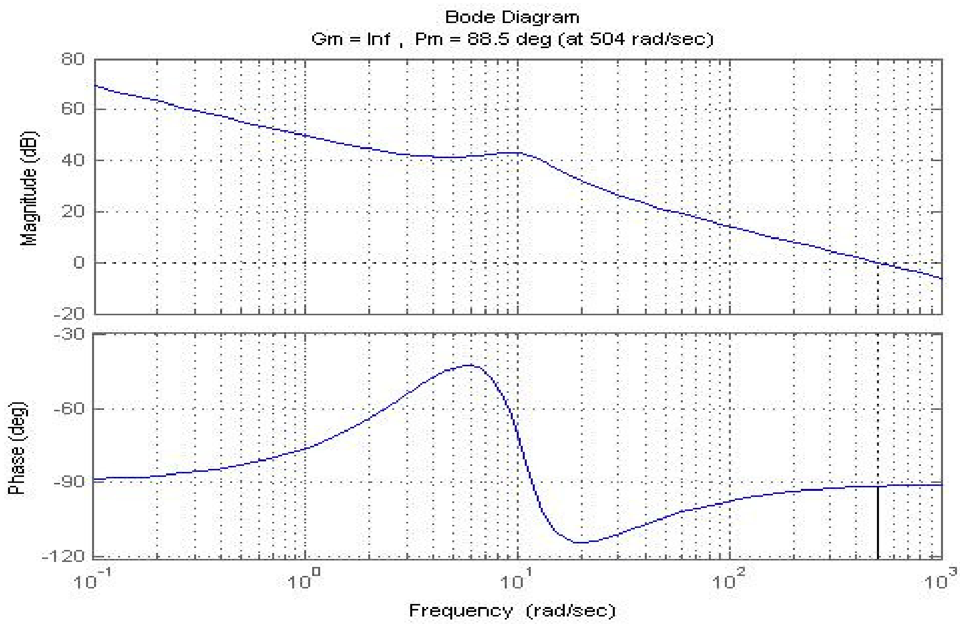

Figure 6.8(b): Series Ideal PID controller

6.8 Conclusion

In this chapter ideal PID controller with first order filter is better than the other controller because its rise time and settling time is very less and overshoot s zero and robustness also satisfied when ever frequency domain not satisfied. So by this results its better and also show table no 6.2 and figure no 6.3(a). So for evolutionary algorithm best controller is ideal PID with first order filter

CHAPTER 7

OPTIMAL TUNING OF PID CONTROLLERS USING IMMUNE ALGORITHM

7.1 Introduction

The word “immune” comes from the Latin word for “protection” your body immune system is your built – in protection against attack by foreign substances known as antigen. An antigen is any substance that causes an immune response or causes the body to “attack.”

Immune Algorithms is an improved algorithm based on biological immune mechanisms. In the course of immune response, biological immune system preserves part of the antibodies as memory cells. When the same antigen invades again, memory cells activated and a large number of antibodies are generated so that the secondary immune response is more quickly than the initial response. In the meanwhile, there are mutual promotion and inhibition between antibodies. Therefore, the diversity and immune balance of the antibodies are maintained. That is the self-regulatory function of the immune system. The Immune Algorithm simulates the process of the adaptive regulation of the biological antibody concentration, in which the optimal solution of the objective function corresponds to the invading antigens and the fitness $f(X_i)$ of solution X_i corresponds to the antibodies produced by the immune system. According to the concentration of the antibodies, the algorithm adaptive regulates the distribution of the search direction of the solutions and greatly enhances the ability to overcome the local convergence. Example of antigens includes bacteria or virus.

The main function of the immune system is to protect the body from pathogens and cancer. Vertebrate immune systems are more complex than the invertebrates. They are characterized by two important properties, which are memory and specificity. In the case of invertebrate, the immune system consist mainly of phagocytes which are nonspecific. This means that it will not remember any previous antigen, and will use the same attacking strategy each time. Phagocytes has no receptors for specific pathogens, which means that these cells will engulf and try to kill any pathogen. On the other hand, the vertebrate host has evolved more specialized cells called lymphocytes. These Lymphocytes are pathogen specific, which means that they have distinct receptors to interact with different pathogens.

To combat antigens nature has provided us with the immune system. The blood, lymph nodes, and bone marrow act with the liver, spleen, thymus, and tonsils to produce and deliver specialized cells, including B-lymphocytes, T-lymphocytes, and phagocytes. These cells limit the severity and duration of colds, fight infections in the nose and throat, help wounds to heal, destroy some cancers, and much more.

There are two types of immune models

1- Immune model based on the immune system theory (mainly clones choice theory nowadays).
   a- The somatic theory describes that somatic recombination and mutation contribute to increasing the diversity of antibody.
   b- The network hypothesis describe that a mutual recognition network among antibody contributes to control of the proliferation of clones.
2- Immune network model based on the immune network theory.
   a- All the continuous immune network models at present are the ordinary differential equation of time, which conforms to the real control system.
   b- The discrete immune network model is not the common discrete model based on time control system, but it means that the immune cells or molecules are separated among each others.

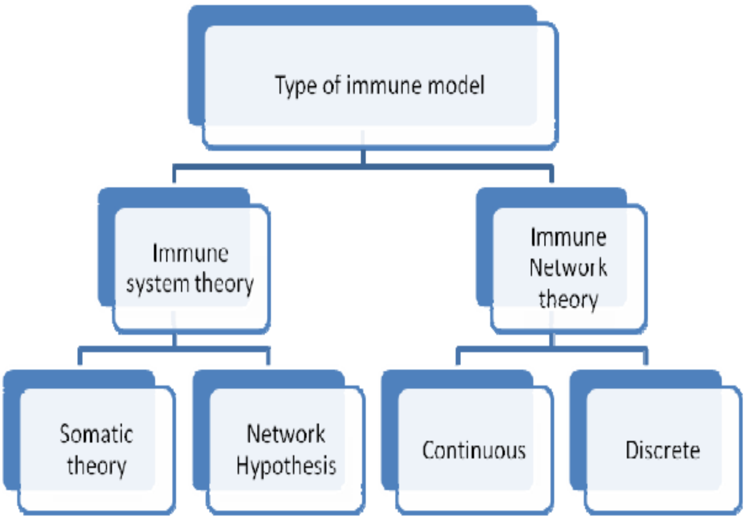

Figure 7.1: Types of immune model

7.2 Innate versus Acquired Immunity

There are two types of immunity, innate immunity and adaptive or acquired immunity. Also, the immune system response can be divided into humoral immunity, and cell mediated response.

7.2.1 Innate Immunity

The innate immunity can be regarded as natural resistance of the host to foreign pathogens. There are a number of external and internal lines of defenses in the innate immunity. As an examples we find Lysozymes in tears, and skin inflammation as a resistance to a penetrating pathogen. The innate immunity is the first line of defense against the foreign pathogens, and try to kill it. Some examples on the same line of defense are Macrophages, and Neutrophils. There are other types of cells that is called Neutrophils. There are other types of cells that is called Natural killer cells NK-cells that also use non-specific response to protect the host against the foreign pathogen.

7.2.2 Acquired Immunity

In contrast to the innate immune system, the acquired immune system uses a specific response to pathogens. The important advantage of the acquired immunity is the use of memory through lymphocytes. After getting rid of the foreign pathogen the lymphocytes change into memory cells. These memory cells will recognize rapidly the same pathogen when it evades the host again, and eliminate it before causing any damage. The two major types of lymphocytes are T-cells, and B-cells. B-cells have direct contact with the antigen when interacting with it. On the other hand, T-cell can bind to the antigen only after it is processed and presented by other cells. B-cells are the basic building block of the humoral immunity through the production of antibodies. Cell mediated immunity is contributed by T-cells mediated response. T-cells have many forms like the helper T-cells which helps either B-cells, or phagocytic macrophages. Another form that the T-cell can be is the cytotoxic T-cells, which recognize cells infected by virus or cancer, and eliminate them.

7.3 Antigens

An antigen (Ag) can be defined as a substance that triggers specific immune response. In vertebrates, the host system does not respond to its own proteins, and that is called tolerance. T-cells and B-cells that are capable of recognizing self-cells are eliminated during maturation phase. An antigen may carry several epitops, and consequently this will trigger the production of several antibodies, see figure 7.2. Generally, T or B cells do not recognize all of these eptopes, instead they recognize part of it. So, a single Ag may attract the attention of several T or B cells. Also, two different antigens may carry the same cross reactive epitopes, which means that an antibody antibody produced for that antigen can interact with another one.

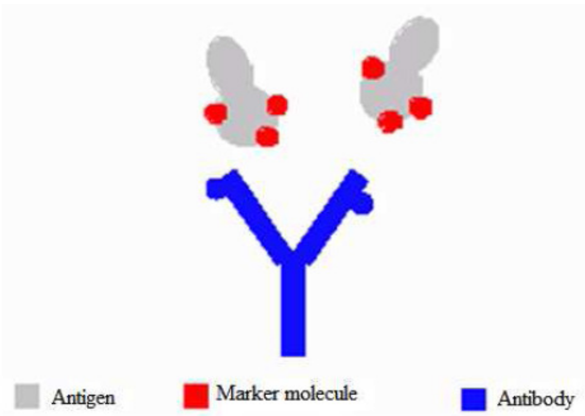

Figure 7.2: Antigen antibody interactions

7.4 Immune Cells

Cells destined to become immune cells are produced in the marrow. The descendants of some stem cells become lymphocytes, while others develop into a second group of immune cells known as phagocytes. The two major classes of lymphocytes are B cells and T cells. B cells complete their maturation in the bone marrow. On the other hand, T cells migrate to the thymus; an organ that lies high behind the breastbone. Each lymph node contains specialized compartments that house a great number of b lymphocytes, capable of presenting antigen to T cells. Thus, the lymph node brings together the several components neede to start an immune response.

7.5 Cells and Antibodies

B-cells is one of the major arms of the immune system mechanisms, and it is responsible for the humoral response. The name humoral comes from these fluids that circular around the body known as humors. Each B cells is programmed to make one specify antibody. When a B cells encounters its triggering antigen, it produces many large plasma cells. Every plasma cell is a factory for producing antibody. Each of the plasma cells descended from a given b cells produces millions of identical antibody molecules and pours them in to the bloodstream, see figure 5.3. A given antibody matches an antigen as akey matches a lock, and marks it for destruction.

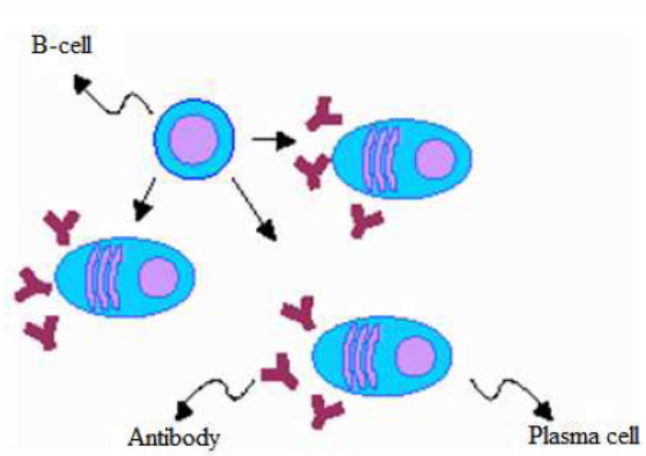

Figure 7.3: production of antibody

7.6 T-Cells And Lymphokines

T-cells play two rolls in the immune system defense. B cells can not make antibody against most substances with out regulatory T-cell help. On the other hand, Cytotoxic T-cells, directly attack body cells that are infected. Another important regulatory T cells actvate B cells are “helper” cells. Typically identifiable by the T4 cell marker, helper T cells activate B cells and other T cells as well as natural killer cells and macrophages. Another subset of T cells contributes by tuning or “suppress” these cells. T cells works by secreting cytokines or, Lymphokines which are considered to be chemical messengers.

7.7 Macrophages

Macrophages are responsible for carrying the initial attack against an invasion launched by antigens. Macrophages are distributed throughout body tissues, and they rid the body of worn-out cells and other debris. Foremost among the cells that “present” antigen to T cells, having first digested and processed it, macrophages play a crucial role in initiating the immune response. As secretary cells, Monocytes and Macrophages are essential to the regulation of immune response and the development of inflammation; they produce an array of powerful chemical called Monokines including-1 enzymes, complement proteins, and regulatory factors such as interleukin-1. Sometimes antigen change themselves, and that is why we continue to get sick.

7.8 An Overview of the Immune System

When foreign antigen enters the body, it triggers b-cells to produce antibodies, which bind to the antigen and clear it from the body; this is called humoral immune response. The cells-mediated response involves helper T-cells and T cytotoxic (CTL) cells. Helper T-cells (Th) can be divided into two sub fields: Th1 and Th2. Th1 cells help B-cells activate macrophages. CTL cells kill virtually infected or Cancer cells, see figure 7.4.

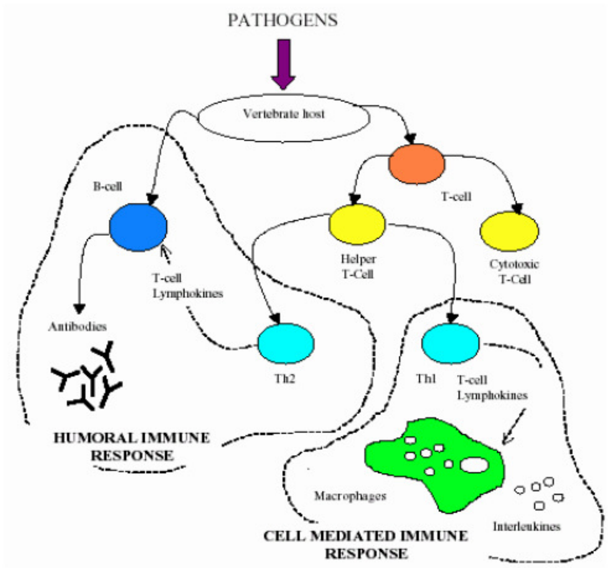

Figure 7.4: An overview of an Immune System

7.8.1 Humoral Response

When the B-cells proliferates, all of its descending will make this uniquely rearranged set of antibodies. B-cells continue to multiply, various mutants arise; these allow for the natural selection of antibodies that provide better and “fits” for antigens elimination. The result of this entire process is that a limited number of B-cells can respond to an unlimited number of antigens. Antibodies are triggered when a B-cells encounters its matching antigen, and digest it. Antigen fragments are displayed on B-cells distinctive markers. The combination of antigen fragments, and marker molecules attract the mature matching helping cells. T-cells secrete Lymphokines allow B-cells to multiply and mature into antibody producing plasma cell. Antibodies are released into the blood stream, and they lock into matching antigens. These antigen-body complexes are soon overcome either by the complement cascade, or by the liver and spleen, see Figure 7.5.

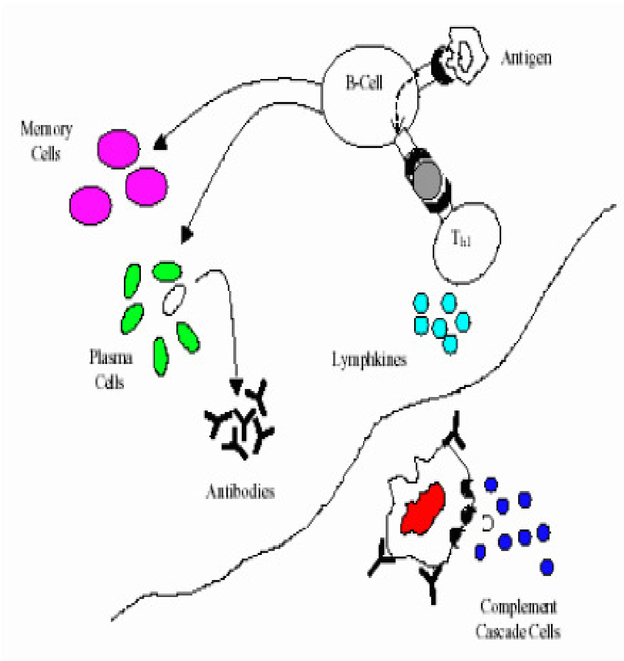

Figure 7.5: Humoral response

7.8.2 Cell Mediated Response

Macrophages initiate the cell mediated response, or by other antigen-presenting cell. The antigen-presenting cell digest the antigen, and then displays antigen fragment on its own surface. Bound to the antigen fragment is an MHC molecule. These fragments capture the T cell's attention. A T cell whose receptor fits this antigen binds to it.

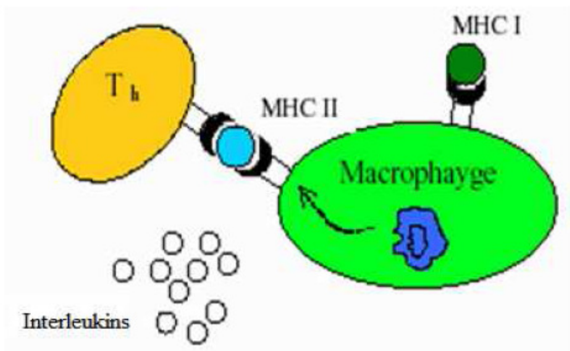

Figure 7.6: Cell Mediated Response

7.9 Immune PID Control

Introduction of immune PID controller are describe in the following section.

7.9.1 Feedback principle of Immune System

Immune is a characteristic physiological reaction of biological Immune system of biology could produce relative antibody to resist invading anti-source from extraneous. After antibody combines with anti source, a serial reaction will be brought to destroy antibody by swallowing effect or producing special enzyme. In immune system, there is a feedback mechanism that enables human survival of infection and disease. Fig. 7.7 presents the principle of feedback mechanism. The basic cells that are involved in the process are antigens Ag, antibodies Ab, B-cells B, helper T-cells T_H and suppressor T-cells T_S. According to Fig. 7.7, we know that antigens will be recognized by APC (Antigen Presenting Cell) when they invade into organisms, then, the message will be sent to T-cells. After receiving the message, B-cells will be stimulated by T-cells and create antibodies immediately to eliminate the antigen. When the number of antigens is increasing, the number of T_H-cells will increase and the human body can create more B-cells to protect itself. Along with the decrease of antigens, the amount of T_S-cells in the body would increase and the number of B-cell would reduce accordingly. After a period of time, the immune system inclines to balance. Table 1 summarizes the regulation action of T-cells in the process of the above immune response.

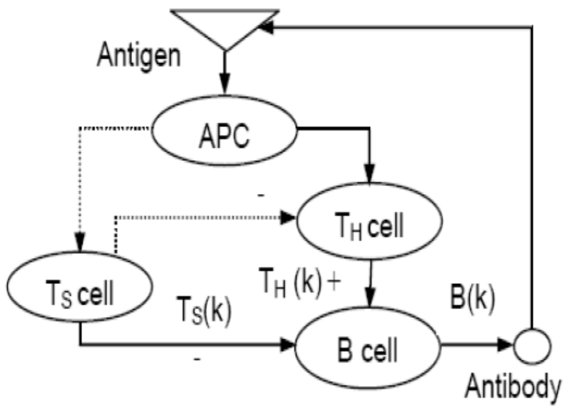

Figure 7.7: Schematic diagram of immune humeral response

Table 7.1: Regulation actions of T-cells in immune response

Immunity response process	**Antigen Consistency**	**Antibody Consistency**	**T-cells consistency**
Antigen invasion	High	Minimum	------------
Prophase	High	Low	Promotion
Anaphase	Low	High	Suppression
Telophase	Minimum	Low	-----------

As aforementioned, the T_S cells have the function of restraining the T_H cells and B-cells. This paper mainly focuses on the suppression action of the B-cells. For the invasion of the antigen, the B-cells are activated and restrained by the T_S cells. Therefore, the consistency of the kth generation B-cells can be given by

$$B(k)=T_H(k)-T_S(k) \quad \text{7.1}$$

$$T_H(k)=K1\varepsilon(k) \quad \text{7.2}$$

$$T_S(k)=K2\{f[\Delta B(k-d)]\}\varepsilon(k) \quad \text{7.3}$$

Where ε(k) is the consistency of antigen at the kth generation; K1 is the helper gene of T_H; K2 is the suppressor gene of T_S; ΔB(k) is the change of B-cell's consistency ΔB(k-d)=B(k-d-1), and d is the delay-time of immune response; *f*(x) is an nonlinear function that represents the interaction between antibody which emerge from B-cells and antigen.

From (7.1)~(7.3), we can obtained the relationship formula about the consistency of B-cells and antigen. It is shown as follows:

$$B(k)=K_1\varepsilon(k)-K_2\{f[\Delta B(k-d)]\}\varepsilon(k)$$

$$=K\{1-\eta f[\Delta B(k-d)]\}\varepsilon(k) \quad \text{7.4}$$

Where $\eta=K_2/K_1$ denotes the proportional coefficient of effecting between T_H and T_S.

Table 7.2: Comparison between artificial immune system and control system

Immune System	Control System
1) The kth generation reproduction of antigens and antibodies. 2) E(k) is the antigen concentration of the kth generation. 3) B(k) is the B cell concentration of the kth generation.	1) The kth sampling time discrete system. 2) e(k) is the deviation of the set value and output value at the kth sampling instant. 3) U(k) is the output value of the controller at kth sampling instant.

7.10 Immune PID Controller Design

The principal function of the appropriate immune response lies in ensuring the stability of the immune system and simultaneously responding to the antigen invasion in fast way, sinvce all the antigens attacking the biological body have to be removed. On the other hand, a high antibody consistency also does harm to the body, and must be controlled. Therefore, the general target of the immune system is to minimize the total injury of the biological body. In the dynamic regulation of a control system, it is requested that deviation should be slaked on the promise of the system stability, which is actually consistent with the target of immune system. Table 7.2 summarizes the comparison between the immune system and control system.

It is obvious that immune controller (IC) is a nonlinear P controller, its proportional gain will transform with the transformation of the output of controller. Therefore, immune P controller cannot compensate noises and errors arose by nonlinear disturbance. Thus, the fusion of the immune controller and conventional PID controller, namely immune PID controller, can learn from each other's strengths and overcome the weaknesses to improve the system performance.

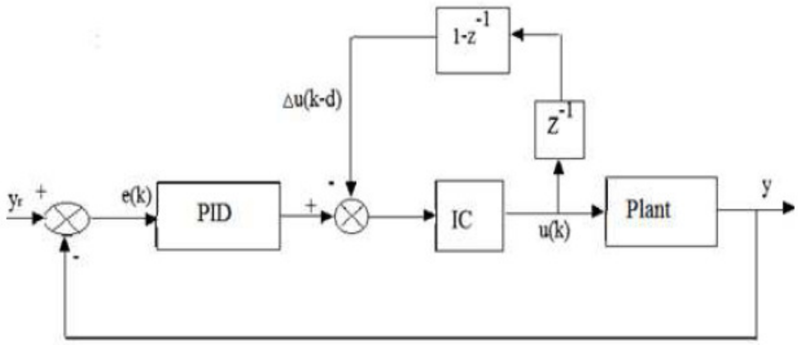

Figure 7.8: The structure of immune PID controller

7.11 Immune Algorithm Step

In general, the Immune Algorithm includes:

1. Antigens definition: Abstract the problem to the form of antigens which the immune system deals with and the antigen recognition to the solution of problem.
2. Initial antibody population generation: The antibody population is defined as the solution of the problem. The affinity between antibody and antigen corresponds to the evaluation of the solution, the higher the affinity, the better the solution.
3. Calculation of affinity: Calculate the affinity between antigen and antibody.
4. Various immune operations: The immune operations include selection, clone variation, auto-body tolerance, antibody supplementation and so on. The affinity and diversity are usually considered to be the guidance of these immune operations. Among them, select options usually refer to the antibody population selected from the population. Clone variation is usually the main way of artificial immune algorithm to generate new antibodies. Antibody supplementation is the accessorial means of the population recruitment.
5. Evaluation of new antibody population: if the termination conditions are not satisfied, the affinity is re-calculated and the algorithm restarts from the beginning. If the termination conditions are satisfied, the current antibody population is the optimal solution.
6. Evaluation of the antibody using standard genetic algorithm: crossover and mutation.

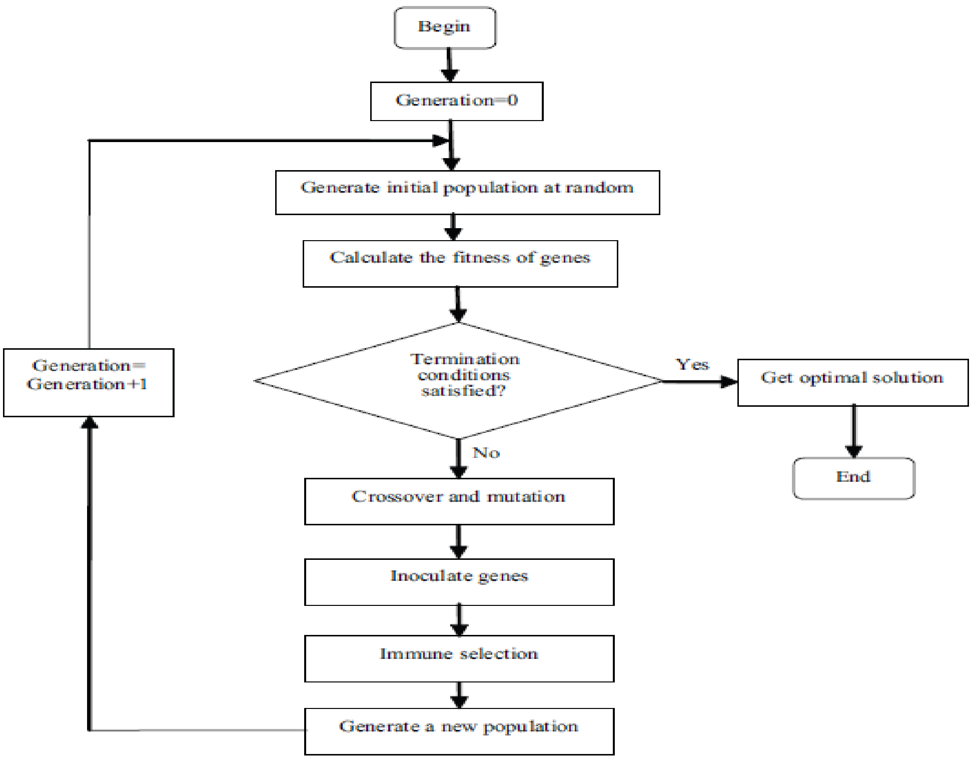

Figure 7.9: Flow chart of Immune algorithm

7.12: Results

In this chapter all result related to immune algorithm on different controller configurations are discussed. Here values of Kp, Ti and Td are obtained using immune algorithm.

Ideal PID with 1st order filter

Table 7.3: Ideal PID with 1st order filter

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	over shoot	Mod. margin	Delay Margin
T.F.	0.0205	90.0005	23.852	.000067	1.7733	3.8986e-007	6.9416e-007	0	-0.0298	2.1861e-006

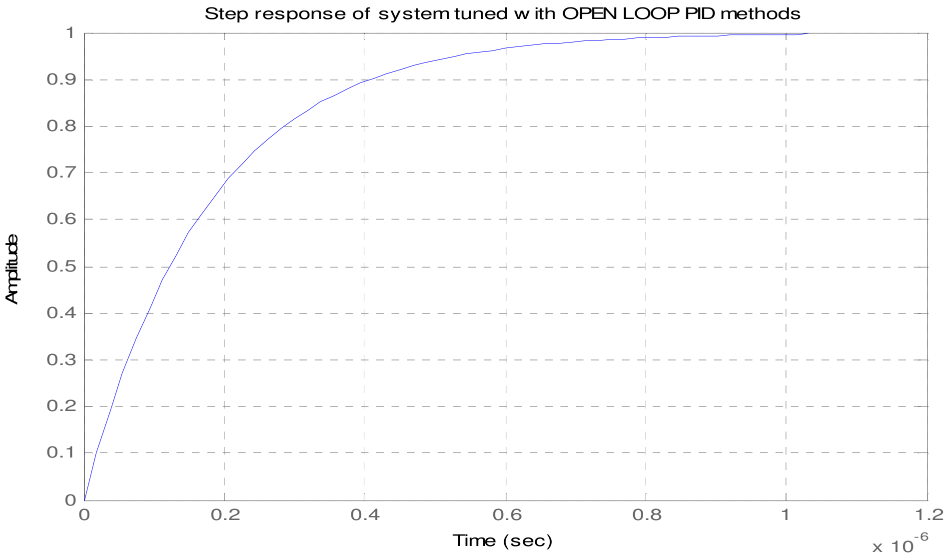

Figure 7.10(a): Ideal PID with 1st order filter

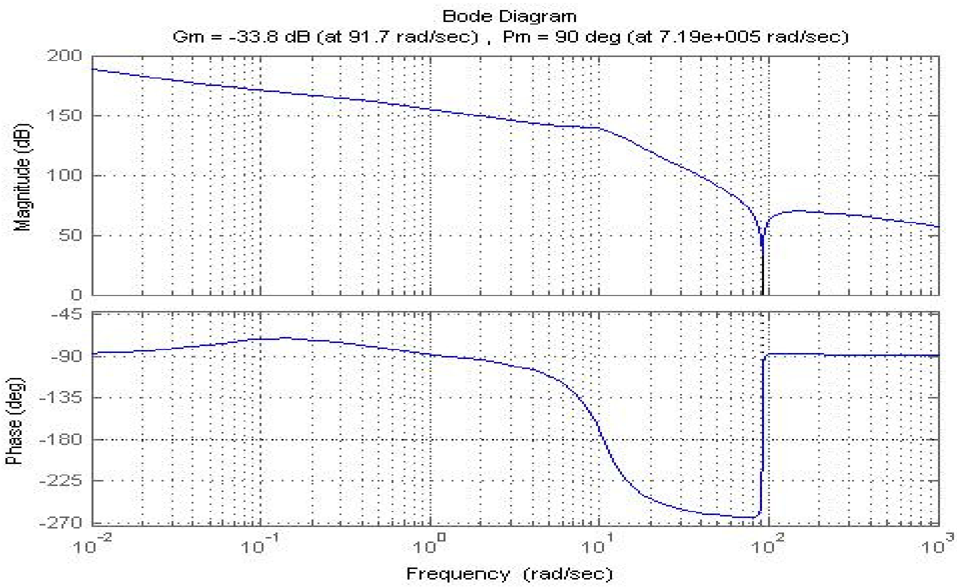

Figure 7.10(b): Bode Plot Ideal PID with 1st order filter

Ideal PID controller

Table 7.4: Ideal PID controller

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	overshoot	Mod. margin	Delay Margin
T.F.	0.0410	90.0010	23.852	.000067	1.7733	7.7985e-007	1.3892e-006	0	-0.0366	4.3723e-006

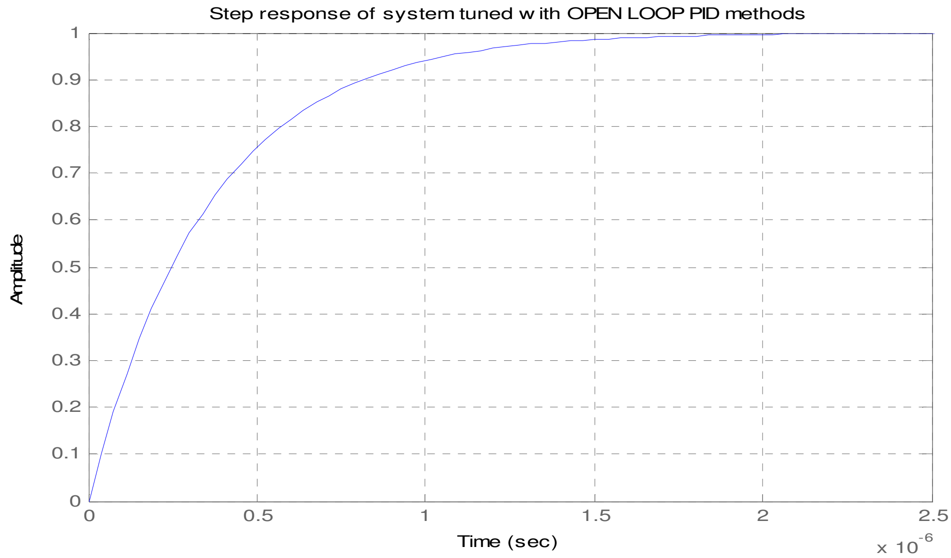

Figure 7.11(a): Ideal PID controller

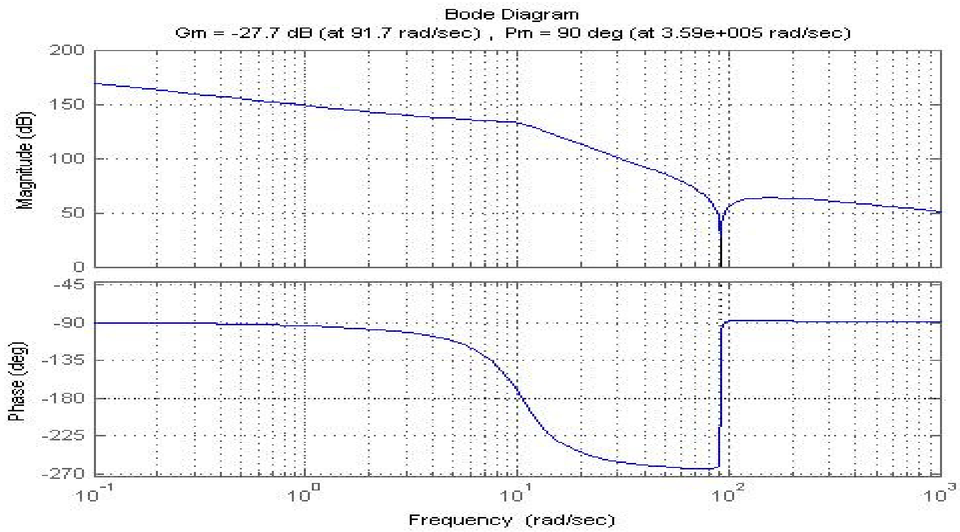

Figure 7.11(b): Bode Plot Ideal PID controller

Ideal PID controller with 2nd order filter

Table 7.5: Ideal PID with 2nd order filter

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	over shoot	Mod. margin	Delay Margin
T.F.	0.1122	0.4185	23.852	.000067	1.7733	3.5332e-004	0.0258	85.5273	0.0234	6.6982e-006

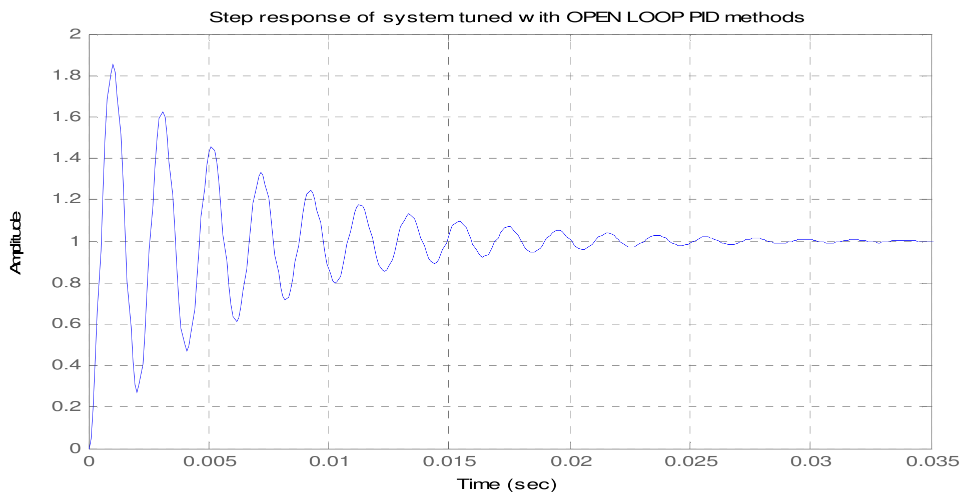

Figure 7.12(a): Ideal PID with 2nd order filter

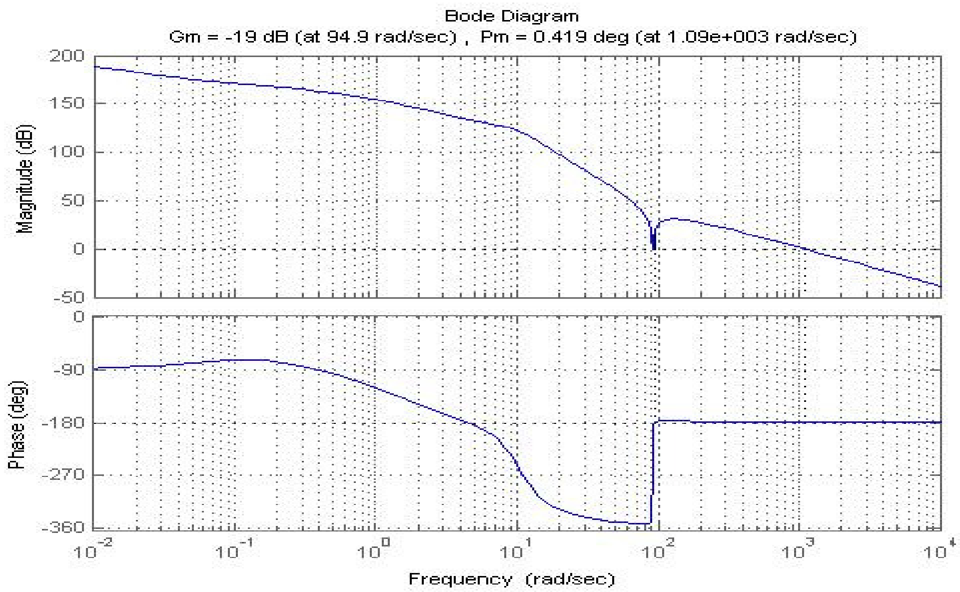

Figure 7.12(b): Bode Plot Ideal PID with 2nd order filter

Series Ideal PID controller

Table 7.6: Series Ideal PID controller

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	overshoot	Mod. margin	Delay Margin
T.F.	Inf	87.6242	23.852	.000067	1.7733	7.6769e-007	1.3149e-006	0.4952	-0.0087	4.2530e-006

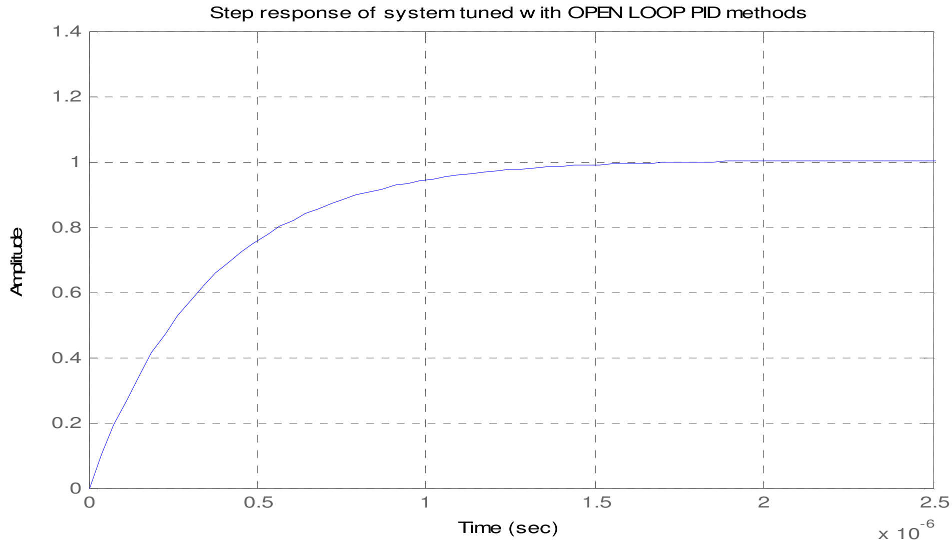

Figure 7.13(a): Series Ideal PID controller

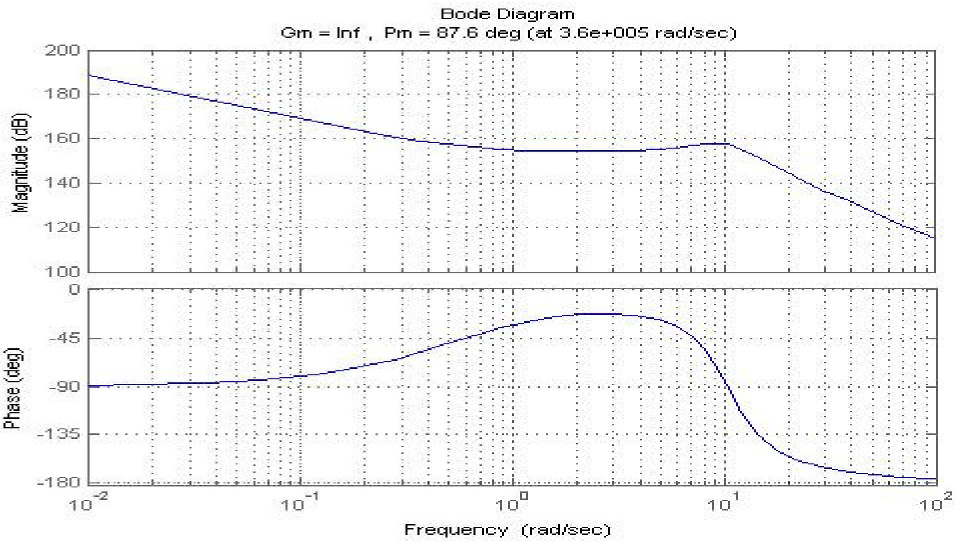

Figure 7.13(b): Bode Plot of Series Ideal PID controller

Ideal PID controller with weighted terms

Table 7.7: Ideal PID controller with weighted terms

Form	Gain margin	Phase Margin	Kp	Ti	Td	Rise time	Settling time	Overshoot	Mod. margin	Delay Margin
T.F.	0.0410	90.0010	23.852	.000067	1.7733	7.7985e-007	1.3892e-006	0	-0.0366	4.3723e-006

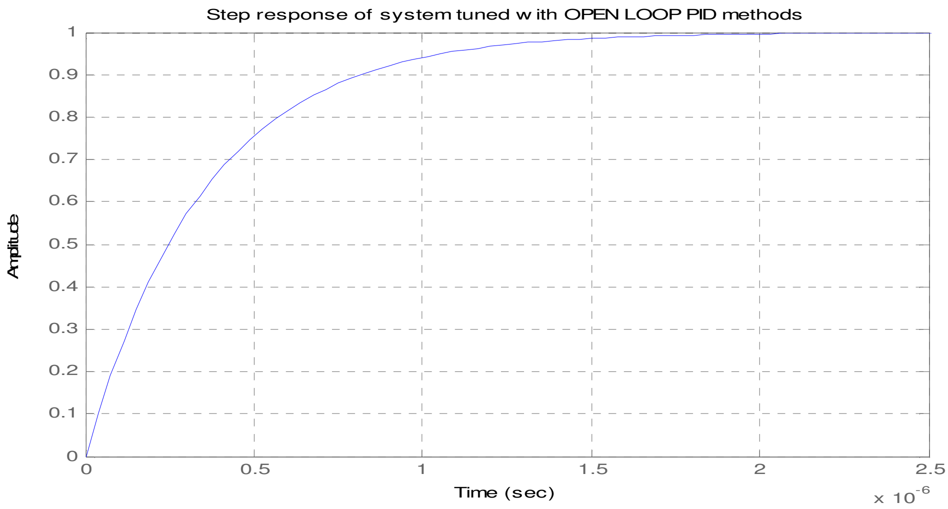

Figure 7.14(a): Ideal PID controller with weighted terms

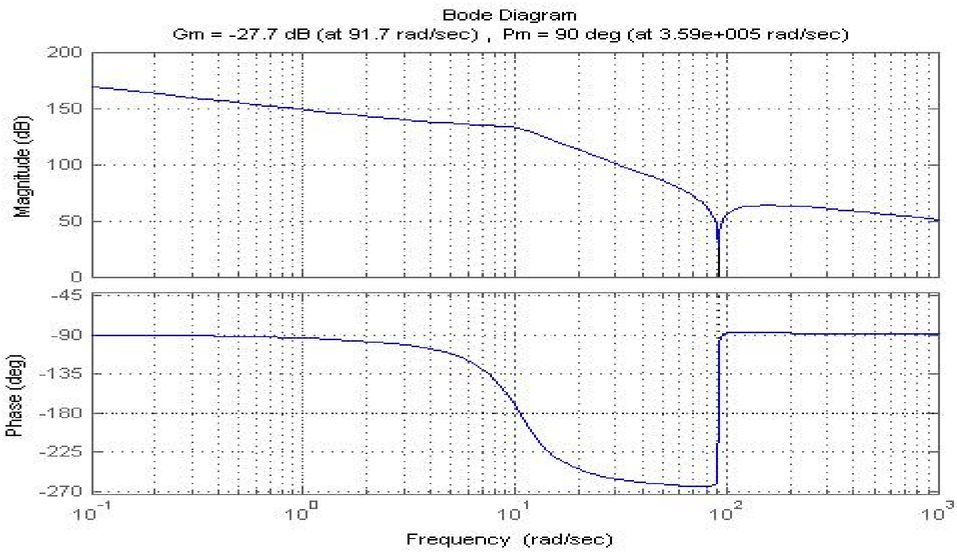

Figure 7.14(b): Bode Plot of Ideal PID controller with weighted terms

7.13: Conclusion

In this chapter ideal PID controller with 2nd order filter type controller is most efficient other then the controller. By table number 7.5 shows better result other then controller and also figure no 7.12(a) shows better results. Here mostly controller satisfied robustness and some frequency domain. Here K_p, T_i and T_d values concluded according immune algorithm. In table no.7.3 rise time and settling time is very less. So by these values ideal PID controller with 2nd order filter is best suitable for immune algorithm.

CHAPTER 8
CONCLUSION AND FUTURE WORK

In this thesis it is concluded that out of the different controllers considered using evolutionary and immune algorithms, ideal PID controller with first order filter performs best for DC motor speed control. Here all results show that evolutionary algorithm is good for different configuration of PID controller. The performance of a control system is not expressed only in terms of rapidity: the behavior of the control variable is as important as one of the output variable and the frequency response to disturbance might highly affect the feasibility. The EP method has the advantage of being flexible in evolving any controller structure. In this case, however, the method evolved a controller structure that makes use of a second derivative. In control engineering, the use of a first derivative is often avoided because of the sensitivity to noise. For this reason, the use of a second derivative is even more unlikely.

The EP and BIS approach is extremely computationally expensive and provides solutions of questionable quality. The EA and BIS approach was developed with the aim of using saturation and bang-bang control to achieve the best time domain performance, i.e. the maximum rapidity of control. The merit of the method is not given only by the quality of one solution, but also by the robustness proved in the statistical analysis of multiple runs. Two important aspects in the effectiveness of the search process were identified by the quality of the initial population and the application of heuristics. The quality of the initial population, that can be enhanced with seeds or randomized seeds, highly affects the speed of the search process and the quality of the results. The improvement of the standard EA and BIS by means of heuristics was studied. The heuristics implemented in this work do not rely on the domain knowledge. The purpose was to enhance GA with intelligent genetic operators without specializing it to a specific domain. The experimental results proved that the use of intelligent operators increases both the speed of the search and its robustness.

Each intelligent genetic operator that has been introduced in this work has proved to produce unsteadily high quality off-springs. However, the combination of all of them proved to be the key factor of a steady and enhanced performance of the overall algorithm. In particular, the global directional mutation provided high quality off-springs that justify. The good performance of the application was therefore achieved by combining the right choice of the controlled structure, the advantageous initialization with randomized seeds, and the use of heuristics such as randomization of the population and intelligent genetic operators. It can be concluded that the application of EA and BIS to the design of a control system is an approach that requires a certain effort in order to make it robust and reliable.

Future Works

The work and results presented in this thesis comprise a specific set of issues chosen in a wide range. During the progress of the work, as the study of the topic became more detailed, several directions of research were unfolding, making difficult the priority of keeping the work focused. For this reason, this section was written during the development of my work and was meant to keep track of the increasing list of new possible directions of investigation. Until the very last, none of the points listed below was excluded from being dealt in one section of the thesis. However, I had to compromise and leave the following points open for future works.

Testing on Different Control Problems

The application used for the experiment of this thesis is design to be used with different typologies of plant. The simplicity of the plant ordered a good bench mark to test the effectiveness of evolutionary and immune in the synthesis. To prove the effectiveness of the methods under more general conditions, it would be useful to use a more complex plant. Among them, it would certainly interesting testing the application for high order linear plants, plants that can be unstable for certain values of parameters, intrinsic nonlinear unstable plants. In particular, GAs have proved to be suitable to optimize highly nonlinear control systems.

Optimization of neural or fuzzy control

Evolutionary and immune computations are widely used to optimize neural network and fuzzy control. Hence, the method used in my thesis that has given good results on traditional control, can be applied and verified for other kind of control structures.

Interactive Computation

The task of defining an a priority fitness function might be complex and extremely problem dependent. Hence, the human interaction during the computation can substitute the initial laborious problem of the evolutionary and immune parameter tuning. For example, the quality and performance of a controlled system can be improved by interactive computation.

REFRENCES

[1] Neenu Thomas, Dr. P. Poongodi, ‘Position Control of DC Motor using Genetic Algorithm based PID Controller’, Proceedings of the World Congress on Engineering 2009 Vol. 2, WCE 2009, July 1-3, London, U.K.

[2] Ziegler, J.G. and *et.al*: ‘Optimum setting for automatic controllers’ ,Trans. ASME, 1942, pp.759-768

[3] Hagglund, T. and *et.al*: ‘Automatic tuning of PID controllers’ in ‘The control handbook’,1996, pp. 817-826

[4] Astrom, K.J., Hang, C.C., Persson, P., and Ho, W.K.: ‘Towards intelligent PID Control’, Automatica 1992, 28, (1), pp. 1-9

[5] Shafiel, Z., and *et.al*: ‘Frequency- domain design of PID Controllers for stable and unstable systems with time delay’, Automatica, 1997, 33, (12), pp, 2223-2232

[6] Cohen, G.H., Coon, G.A. ‘Theoretical consideration of retarded control’, Trans. ASME vol. 75, pp. 827-834, 1953

[7] O’Mahony, T., Downing, C.J., and Fatla, K.: ‘Genetic algorithms for PID parameter optimization: minimizing error criteria’. Cork Institute of Technology, Cork, Ireland, pp. 148-153

[8] P. W. Murril, P. D. Schnelle Jr., B. G. Liptak , J. Gerry, M. Ruel, F. G. Shinskey, ’Process Control and Optimization, VOLUME- II’, 2006, pp. no. 414-431

[9] M. Zhuang , D. P. Atherton , ‘Automatic Tuning of optimum PID controllers’, IEEE Proceedings-D, May 1993, Vol. 140, No.3, pp. no. 216-224

[10] Teng Fong-Chwee., “Self tuning PID controllers for dead time process”.,IEEE Trans., VOL. 35, NO. 1, pp. 119-125, 1988.

[11] T. Back and H.P. Schewefel., “An overview of Evolutionary Algorithms for parameter optimization”., IEEE International conference on Ecommerce (CEC03), VOL. 1, pp. 1-23, 1993.

[12] D.B. fogel., “Evolutionary Computation”, IEEE Conference of Evolutionary Computation., VOL. 1, pp. 193, 1995.

[13] J.B Yang, “Gradient Protection and Local region search for multiobjective optimization”., European journal of operational research., VOL. 112, NO. 2, pp. 432-459., 1999.

[14] Corina rato., "A New Evolutionary Algorithm for multiobjective optimization based on endocrine paradigm".,ICTAMI , pp. 271-284, 2003.

[15] Dong Hwa Kim, Jae Hoon cho., "robust tuning for Disturbance Rejection of PID controller using Evolutionary Algorithm"., IEEE Annual meeting of the fuzzy information, pp. 248-253, 2004.

[16] Gregory S. Horn by., "Automated Antenna Design with Evolutionary Algorithms"., American institute of aeronautics and astronutics, Moffett Field, CA 94035, 2006, pp. 1-8.

[17] Monica Perez-meza, and Redrigo Montufar-Chaveznava., "Partial 3D Reconstruction using Evolutionary algorithms"., International Journal of Engineering and applied science , pp. 107-112, 2007.

[18] Xiufen Zou, *et al*., "A New evolutionary algorithm for solving many optimization problems"., IEEE TRANS. ON SYSTEM MAN AND CYBERNETICS-PART B: CYBERNETICS, VOL. 38, NO. 5, pp. 1402-1412, 2008.

[19] D. Floreno and C. Claudio Mattiussi., "Bio inspired Artificial intelligence: Theories and mehods, and technologies ". Ser intelligent Robotics and automous agents., MIT press of artificial intelligence, pp. 335-398, 2008.

[20] Stephane Doncieux, *et al*., "Exploring New Horizons in Evolutionary Design of Robots"., IROS, pp. 20-29, 2009.

[21] Kezong Tong., *et al* "Linear Evolutionary Algorithms" cheapter-2 in Evolutionary Algorithms written by Eisuke kita., pp. 28- 38., 2011.

[22] Gunther Raidl., "Evolutionary Computation: An overview and Recent Trends"., FavoritenstraBe 9-11/1861, 1040 vienna, Austria, pp- 2-8, 2011.

[23] Jingui Lu and Min Xie., "Immune-Genetic Algorithm for Travelling Salesman Problem"., IEEE Trans., VOL. 35, NO. 1, pp. 84-86, 1988.

[24] Dong Hwa Kim, Won Pyo Hong., "Auto-Tuning of Reference Model Based PID Controller Using Immune Algorithm"., IEEE International conference on Evolutionary Computation, VOL. 1, pp. 483-488, 2002.

[25] Dong Hwa Kim, Jae Hoon Cho., "Intelligent Tuning of PID Controller With Distribution Function Using Immune Algorithm", IEEE Conference on Computational Intelligent for Measurement System and Application., VOL. 1, pp. 109-114, 2004.

[26] Dong Hwa Kim., "Tuning of PID Controller Using Gain/Phase Margin and Immune Algorithm"., IEEE Mid-Summer Workshop on Soft Computing in Industrial Application., pp. 69-74., 2005.

[27] Maryam Khoie, prof Ali Khaki Sodigh, Dr. Karim Salahshoor., "PID Controller Tuning using Multi-Objective Optimization based on fused Genetic-Immune Algorithm and Immune Feedback Mechanism"., International Conference on Mechatronics and Automation., pp. 2459-2464., 2011.
[28] Tion Xiao-min, Huang You-*rui*., "The Tuning of Fractional order PIλDμ controller Parameters based on Immune Clone Selection Algorithm"., International Symposium On Information Science and Engineering., pp. 354-357., 2010.
[29] Young Jin Lee, Jin Ho Suh, Jin Woo Lee ana Kwon Soon Lee., "Adaptive PID Control of an AGV System using Humoral Immune Algorithm and Neural Network Identifier Technique"., International Conference on Control Application., Vol. 2., pp. 1576-1581, 2004.
[30] Dong Hwa Kim, "Robust Power Plant Control Using Clonal Selection Of Immune Algorithm Based Multi-objective", Proceeding of the IEEE Fourth International Conference on Hybrid Intelligent System, pp. 450-455, 5-8 Dec. 2004.