Sudan University of Science
& Technology
College of Graduate studies

DC Motor Speed Control with PID Controller Auto-Tuning and Genetic Algorithm

التحكم في سرعة محرك التيار التيار المستمر مع الضبط التلقائي للمتحكم النسبي التكاملي التفاضلي والخوارزمية الجينية

A Proposal Submitted in Partial Fulfillment of the Requirement for the Degree of
M.Sc. in Electronics Engineering (computer &networks)

Prepared By:

Mussab Mohamed Yousif Adam

Supervised by:

Dr.Mohamed ELnour Abdullah

Nov 2022

الإستهلال

بسم الله الرحمن الرحيم

قال تعالي:

﴿ وَلَقَدْ خَلَقْنَا الإنسان مِنْ سُلَالَةٍ مِن طِينٍ * ثُمَّ جَعَلْنَاهُ نُطْفَةً فِي قَرَارٍ مَكِينٍ * ثُمَّ خَلَقْنَا النُّطْفَةَ عَلَقَةً فَخَلَقْنَا الْعَلَقَةَ مُضْغَةً فَخَلَقْنَا الْمُضْغَةَ عِظَامًا فَكَسَوْنَا الْعِظَامَ لَحْمًا ثُمَّ أَنْشَأْنَاهُ خَلْقًا آخَرَ فَتَبَارَكَ اللَّهُ أَحْسَنُ الْخَالِقِينَ ﴾

(سورة المؤمنون: 12 – 14)

Dedication

I dedicate this work

To the candle which burns to light my life

My mother

To the source of inspiration

My father

To those who have made this work it possible

My teachers

To those who encouraged me

Sisters, brothers and friends

Acknowledgments

Throughout the writing of this dissertation I have received a great deal of support and assistance

First of all I would like to give thanks to Allah

I want to thank my Supervisor

Dr. Mohammed Alnour

Who has given me his time and encouraged to do this work

Thank you to all my teachers at different educational stages

all of thanks go to my colleagues who supported me

to made this work possible

Finally I want to thank my friend ayman Adam

Who has helped me in the writing of this thesis

Abstract

Since DC motor plays a significant role in modern industry the aim of This research is to design a speed controller of a DC motor by selection of PID Parameters using Genetic algorithm. This algorithm is an approach to optimization and learning based loosely on principles of biological evolution, these are simple to construct, and its implementation does not require a large amount of storage, making them a sufficient choice for an optimization problems. Here conventional tuning techniques and optimization techniques for PID controller parameters were discussed and used to improve the performance of DC motor control speed because it play significant role in industrial application. Here comparison between different tuning methods was done. The main reason to use optimization algorithm with PID controller is to give an optimum output values for the PID parameters. The PID conventional controller had been applied and results were compared with the auto tuning PID-GA for DC motor speed control using Simulink of MATLAB. Finally Simulation results for the proposed method gave optimum values for PID controller parameters and contribute better transient response compare with the results that we got from conventional methods such as Ziegler-Nichols (ZN) & trial and error.

المستخلص

نظرًا لأن محرك التيار المستمر يلعب دورًا مهمًا في الصناعة الحديثة ، فإن الهدف من هذا البحث هو تصميم وحدة تحكم في السرعة لمحرك التيار المستمر عن طريق اختيار معلمات المتحكم التناسبي التكاملي التفاضلي باستخدامالخوارزمية الجينية. هذه الخوارزمية تعتبر طريقة للتحسين والتعليم و تعتمد بشكل فضفاض على مبادئ التطور البيولوجي ، وهي سهلة الإنشاء ، ولا يتطلب تنفيذها قدرًا كبيرًا من التخزين ، مما يجعلها خيارًا كافيًا لمشاكل التحسين. هنا تمت مناقشة تقنيات الضبط التقليدية وتقنيات التحسين لمعلمات وحدة المتحكم التناسبي التكاملي التفاضلي واستخدامها لتحسين أداء سرعة التحكم في محرك التيار المستمر لأنها تلعب دورًا مهمًا في التطبيق الصناعي. تم هناعمل مقارنة بين طرق الضبط المختلفة. السبب الرئيسي لاستخدام خوارزمية التحسين مع وحدة التحكم المتحكم التناسبي التكاملي التفاضلي هو إعطاء قيم الخرج المثلى لمعلمات المتحكم التناسبي التكاملي التفاضلي. تم تطبيق وحدة تحكم المتحكم التناسبي التكاملي التفاضلي التقليدية وتمت مقارنة النتائج مع الضبط التلقائي باستحدام الخوارزمية الجينية مع المتحكم التناسبي التكاملي التفاضلي للتحكم في سرعة محرك التيار المستمر باستخدام برنامج المحاكاة(الماتلاب). أخيرًا ، أعطت نتائج المحاكاة للطريقة المقترحة القيم المثلى لمعلمات تحكم المتحكم التناسبي التكاملي التفاضلي وساهمت في استجابة عابرة أفضل مقارنة بالنتائج التي حصلنا عليها من الطرق التقليدية.

Table of Contents

Chapter One: Introduction

Chapter two: literature review

Chapter Three: System Design

Chapter Four: Results And Discussion

Chapter Five: Conclusion and Recommendations

List of figures

List of tables

List of abbreviations

DC	Direct Current
PID	Proportional–Integral–Derivative
Z-N	Ziegler Nichol
GA	Genetic Algorithm
IE	Integral Error
ISE	Integral Square Error
ITSE	Integral Time square Error
IAE	Integral Absolute Error
ITAE	Integral Time Absolute Error
PSO	Particle swarm optimization
NSS	Non-Steady State
BLDC	Brushless DC motor
ACO	Ant Colony Optimization
PMDC	Permanent Magnet DC motor
AC	Alternate Current
MATLAB	Matrix Laboratory
FOA	Fruit fly Optimization Algorithm
SISO	Single –Input- Single-output
MIMO	Multi-Input-Multi-Output
T_r	Rise Time
T_S	Settling Time
E_{SS}	steady-State-Error

List of symbols

R_a: Armature resistance

L_a: Armature inductance

I_a : Armature current

V_c: Induced voltage

V_a: Voltage source

k_v : Velocity constant

$\boldsymbol{\omega_a}$: The rotational velocity of the armature

T_e : The electromagnetic torque

$T_{\omega'}$: The torque due to rotational acceleration of the rotor

T_ω : The torque produced from the velocity of the rotor

T_L : The torque of the mechanical load

k_t : The torque constant and like the velocity constant

J : The inertia of the rotor and the equivalent mechanical load

B : The damping coefficient

EMF: Back electromotive force

K_b : EMF constant

K_T: Torque constant

Kp: Proportional tuning constant

Ki: Integral tuning constant

Kd: Derivative tuning constant

Tu: Ultimate period.

K_u: Ultimate gain.
T_i : Integral time.
T_d : Derivative time.

Chapter One

Introduction

1.1Background

Direct current (DC) motors have been widely used in many industrial applications such as electric vehicles, steel rolling mills, electric cranes, and robotic manipulators due to precise, wide, simple and continuous control characteristics. DC drives, because of their simplicity, ease of application, reliability and favorable cost have long been a backbone of industrial application[1].

There are many types of controller used in the industry, such controller is PID controller. PID controller or proportional–integral–derivative controller is a Generic control loop feedback mechanism widely used in industrial control systems. PID controller attempts to correct the error between a measured process variable and a desired set point by calculating and then outputting a corrective action that can adjust the process accordingly. So by integrating the PID controller to the DC motor were able to correct the error made by the DC motor and control the speed or the position of the motor to the desired point or speed[2].

In This project we need to attempts to develop a new PID tuning method based on Genetic Algorithm (GA) to control speed of DC motor.

1.2 Problem statement

High performance motor drives are very important in industrial as well as other purpose applications; a high-performance motor drive system must have Good dynamic speed command tracking and load regulating response to perform task, traditional PID controller such as Ziegler and Nichols tuning method (Z-N), cannot achieve the desired control performance. To minimize speed error in DC motor several intelligent approaches PID controller were used because it is make good improvement the performance of the system than the traditional methods.

1.3 Proposed solution

Design of auto-tuning PID controller parameters for DC motor speed, using genetic algorithm based on MATLAB toolbox to enhance the capabilities of traditional PID parameters tuning techniques. This is because GA-PID can solve the searching and tuning problems of PID controller parameters more easily and quickly than other approaches.

1.4 Objectives

The main goal is to improve the DC motor speed behavior, the Objectives are:

- To enhance all parameters of PID controller.
- To improve the closed-loop step response characteriscts of the dc motor speed such as rise time, settling time, overshot and steady state error.
- To control the DC motor speed with traditional controlling (PID) Approaches.
- To control the DC motor speed with intelligent controlling (PID) Approaches such as Genetic Algorithm (GA-PID) controller.
- To compare the step response characteriscts of the DC motor speed using traditional techniques and intelligent techniques.

1.5 Methodology

PID controller was used to control the DC motor speed. Overall the closed-loop step response transfer function of the dc motor was simulated. Secondly the PID was tuned using try and error method. Thirdly the PID was tuned using traditional method such as Ziegler and Nichols tuning method (Z-N). Fourthly the PID was tuned using Genetic algorithm (GA). Simulation results of all above scenarios and comparisons between them were founded.

1.6 Scope of the work

This research is meanly focused on PID controller. Different tuning method will be covered, optimization method will be also covered and Genetic Algorithm will be highlighted. The system under study is DC motor.

1.7 Research layout

Chapter One: Introduced the general overview, the problems that solved by it and the objectives that will going to be achieved.

Chapter Two: Is talk about Literature Review, it gives a comprehensive study for all components used in the design of control system.

Chapter Three: Is about simulation of speed control system with conventional PID controller, PID controller tuned using try and error method and GA-PID auto-tuning controller using MATLAB/SIMULINK.

Chapter Four: Is about Simulation and Discussion results, the discussion focused on the results obtained from simulation and the comparisons between these results.

Chapter Five: Conclusion and Recommendations’ Concludes overall about the project, Obstacle faced and future recommendation was also discussed in this chapter.

Chapter Two

Literature Review

2.1 Overview

Control system is a system, which provides the desired response by controlling the output. The control system is represented by a single block. Since, the output is controlled by varying input; the control system got this name. We will vary this input with some mechanism. The following figure shows the simple block diagram of a control system.

Figure (2.1): control system[3]

Based on some parameters, we can classify the control systems as open loop or closed-loop.

In open loop control systems, output is not fed-back to the input. So, the control action is independent of the desired output. The following figure shows the block diagram of the open loop control system.

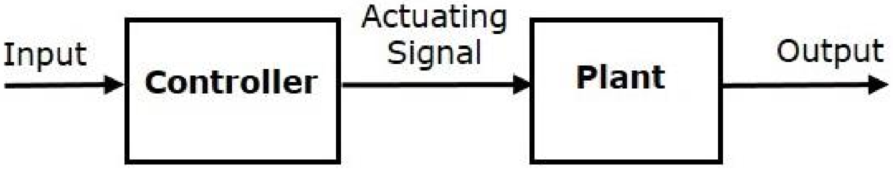


Figure (2.2): Open loop control system[3]

Here, an input is applied to a controller and it produces an actuating signal or controlling signal. This signal is given as an input to a plant or process which is to be controlled. So, the plant produces an output, which is controlled.

In closed loop control systems, output is fed back to the input. So, the control action is dependent on the desired output. The following figure shows the block diagram of closed loop control system.

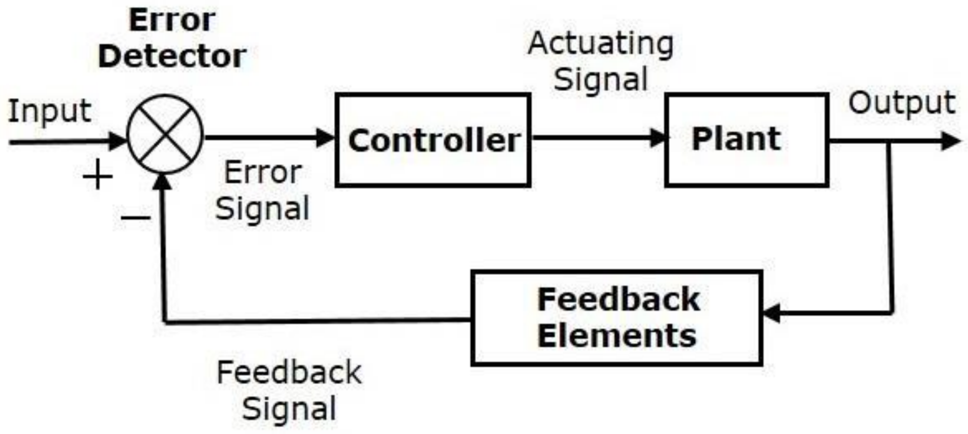


Figure (2.3): Closed loop control system[3]

The error detector produces an error signal, which is the difference between the input and the feedback signal. This feedback signal is obtained from the block (feedback elements) by considering the output of the overall system as an input to this block. Instead of the direct input, the error signal is applied as an input to a controller.

So, the controller produces an actuating signal which controls the plant. In this combination, the output of the control system is adjusted automatically till we get the desired response. Hence, the closed loop control systems are also called the automatic control systems.

The differences between the open loop and the closed loop control systems are mentioned in the following table.

Table (2.1): Comparisons open & closed loop control systems

Open loop system	**Closed loop system**
Control action is independent of desired out put	Control action is dependent of desired out put
Feedback path is not present	Feedback path is present
These are also called as non-feedback Control systems	These are also called as feedback Control systems
Easy to design.	Difficult to design.
These are economical.	These are costlier
Inaccurate.	Accurate.

Block diagrams consist of a single block or a combination of blocks. These are used to represent the control systems in pictorial form. The basic elements of a block diagram are a block, the summing point and the take-off point.

We can use the block diagram of a closed loop control system as shown in the following figure to identify these elements.

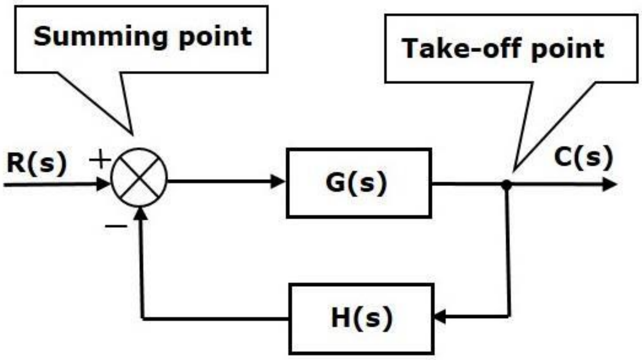


Figure (2.4): A basic elements of a block diagram[3]

The above block diagram consists of two blocks having transfer functions G(s) and H(s). It is also having one summing point and one take-off point. Arrows indicate the direction of the flow of signals. Ach block of a control system has a transfer function (represented by differential equations) and defines the block output as a function of the input.

The transfer function of a component is represented by a block. Block has single input and single output. The following figure shows a block having input X(s), output Y(S) and the transfer function G(S).

Figure (2.5): Block diagram transfer function[3]

Transfer function,

$$G(S) = \frac{Y(S)}{X(S)} \tag{2.1}$$

$$Y(S) = G\,(s)\,X\,(s) \tag{2.2}$$

The summing point is represented with a circle having cross (X) inside it. It has two or more inputs and single output. It produces the algebraic sum of the inputs. It also performs the summation or subtraction or combination of summation and subtraction of the inputs based on the polarity of the inputs. The take-off point is a point from which the same input signal can be passed through more than one branch. That means with the help of take-off point, we can apply the same input to one or more blocks, summing points[3].

2.2 Classical control system

A basic controls system is shown in Figure (2.6). The process (p) or plant is the object to the controlled. It is inputs are u (t), it is outputs are y(t), and reference inputs is r(t).

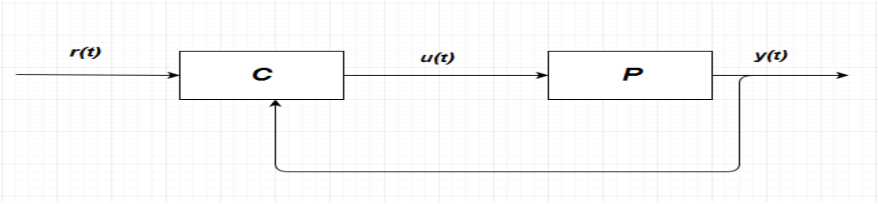


Figure (2.6): A basic control system [4]

2.3 Performance specifications

Control systems are designed to perform specific tasks. The specifications of control system must be given before the design process begins. The specifications may be given in terms of transient response requirements such as [4]:

- **Rise Time:** the time it takes for the plant output y to rise beyond 90% of the desired level for the first time.
- **Overshoot:** the time it takes for the plant output y to rise beyond 90% of the desired level for the first time.
- **Peak Time:** The peak time is the time required for the response to reach the first peak of the overshoot.
- **Peak Value:** The maximum value of the output, reached after application of the unit step input after time.
- **Settling Time:** the time it takes for the system to converge to its steady state.
- **Stability:** That characteristic of a system defined by a natural response that decays to zero as time approaches infinity.
- **Steady-State Error**: the deference between the steady-state output and the desired output.

2.4 PID controller

Proportional-Integral-Derivative (PID) controller is a generic control loop feedback mechanism widely used in industrial control systems, especially for systems with accurate mathematical models. The PID controller calculation involves three separate parameters: proportional, integral and derivative values. The proportional value calculates the value of the current error, the integral value determines the result of the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to be imported into the controlled system. The key issue for PID controllers is the accurate and efficient tuning of parameters. In practice, the controlled systems usually have some features, such as nonlinearity, time-variability and time delay, which make controller parameters tuning more complex. Moreover, in some cases, system parameters and even system structures can vary with time and environment. Thus, the goal of PID controller tuning is to determine parameters that meet the closed-loop system performance specifications over a wide range of operating conditions. Among the conventional PID tuning methods, the Ziegler-Nichols (ZN) method may be the most well known technique. For a wide range of practical processes, this tuning approach works quite well. However, sometimes it does not provide good tuning and tends to produce a big overshoot. To enhance the capabilities of traditional PID parameters tuning techniques, several intelligent approaches have been suggested[5].

PID control is often combined with logic, sequential functions, selectors, and simple function blocks to build the complicated automation systems used for energy production, transportation, and manufacturing. PID control represent an important ’component in every control engineer’s tool box. PID controllers have survived many changes in technology, from mechanics and pneumatics to microprocessors via electronic tubes, transistors’, integrated circuits. The microprocessor has had a dramatic influence on the PID controller. Practically all PID controllers made today are base on microprocessors. This has given opportunities to provide additional features like automatic tuning, gain scheduling, and continuous adaptation[6].

2.5 PID structure

The structure of PID includes proportional term, integral term and derivative term. The PID controller is mainly to adjust appropriate proportional gain(kP), integral gain (ki), and differential gain (kd) to achieve the optimal control performance. PID structure as shown in figure (2.7), r(t) is reference, e(t)is error, u(t)is controller output and y(t) system output.

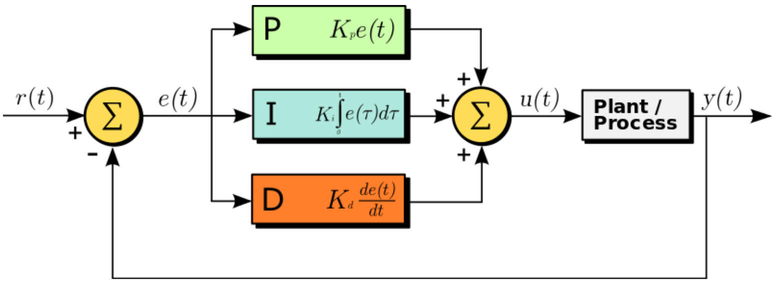


Figure (2.7): Structure of PID controller[7]

Relation between the input $e(t)$ and output $U(t)$ can be formulated in the following:

$$e(t) = r(t) - y(t) \tag{2.3}$$

$$U(t) = K_P . e(t) + K_I . \int_0^t e(t)dt + K_D . \frac{de(t)}{dt} \tag{2.4}$$

The transfer function is expressed (Laplace domain):

$$C(S) = \frac{U(S)}{E(S)} = K_P + \frac{K_i}{S} + K_D \ S \tag{2.5}$$

2.5.1 Proportional term

The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain constant. The proportional term is given by:

$$P_{out} \ = K_P e(t) \tag{2.6}$$

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. A small gain results in a small output response to a large input error, and a less responsive or less sensitive controller.

If the proportional gain is too low, the control action may be too small when responding to system disturbances[7] **.**

2.5.2 Integral 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 (Ki) and added to the controller output.

The integral term is given by:

$$I_{out} = K_I \int_0^t e(t)dt \tag{2.7}$$

The integral term accelerates the movement of the process towards set point and eliminates the residual steady-state error that occurs with a pure proportional controller. However, since the integral term responds to accumulated errors from the past, it can cause the present value to overshoot the set point value[7].

2.5.3 Derivative 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 Kd. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain (Kd).

The derivative term is given by:

$$D_{out} = K_D \frac{de(t)}{dt} \tag{2.8}$$

Derivative action predicts system behavior and thus improves settling time and stability of the system. An ideal derivative is not causal, so that Implementations of PID controllers include an additional low-pass filtering for the derivative term to limit the high-frequency gain and noise[7]**.**

2.5.4 Parallel PID controller

PID controller in Parallel form (also known as standard form, ISA form or non-interacting form), has control equation:

$$U = K\,[1 + \frac{1}{S.T_i} + \mathrm{S}.T_d\,].\mathrm{E} \tag{2.9}$$

The controller actions (P, I and D) act independently as can be seen in the Corresponding block diagram representation. Show figure (2.8)

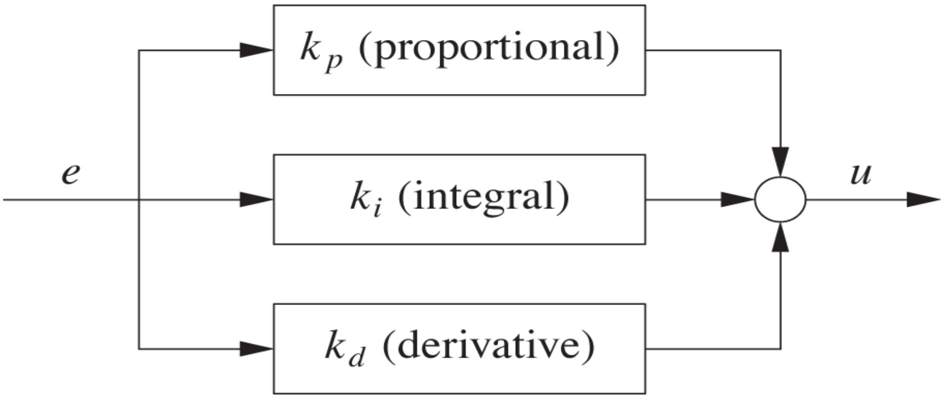


Figure (2.8): PID Controller in Parallel[7]

2.5.5 Series PID controller

PID controller in series form (also known as interacting form), has the control equation:

$$U = K\left[1 + \frac{1}{S.T_i} + S.T_d\right](1 + S.T_d) \quad (2.10)$$

The controller actions (P, I and D) act dependently as can be seen in the corresponding block diagram representation. Show figure (2.9)

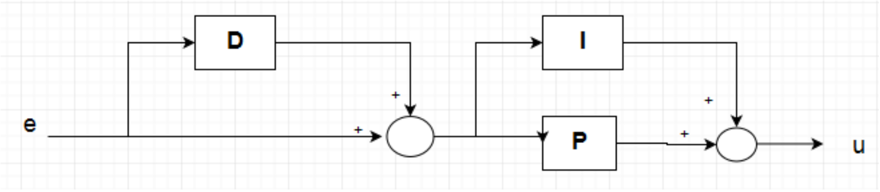


Figure (2.9): PID Controller in Series[7]

2.6 Tuning methods of PID controller

Tuning a control loop is the adjustment of its control parameters (Proportional gain, integral gain, derivative gain) to the optimum values for the desired control response. Stability is a basic requirement, but beyond that, Different systems have different behavior, different applications have different Requirements and requirements may conflict with one another. Some types of tuning methods in the following sections.

2.6.1 Manual tuning

If the system must remain online, one tuning method is to first set ki and kd values to zero. Increase the K_P until the output of the loop oscillates, then the K_P should be set to approximately half of that value for a "quarter amplitude decay” type response. Then increase ki until any offset is corrected in sufficient time for the process. However, too much ki will cause instability. Finally, increase K_D if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, too much K_D will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the set point more quickly[7] .

The effects of increasing each of the controller parameters KP, KI and KD can be summarized as follows:

Table (2.2): Effects of Increasing Parameter Independently

Response	***Rise Time***	***Overshoot***	***Settling Time***	***Steady State Error***
K_P	Decrease	Increase	NT	Decrease
K_I	Decrease	Increase	increase	Eliminate
K_D	NT	Decrease	Decrease	NT

NT: No definite trend. Minor change.

2.6.2 Zigler_Nichols tuning method

There have been various types of techniques applied for PID tuning, one of the earliest being the Ziegler Nichols technique. First, set the controller to P mode only. Next, set the gain of the controller (K_P) to a small value. If is (K_P) low the response should be Sluggish. Increase (K_P) by a factor of two and Keep(K_P) increasing (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 that the period of the oscillations is known as ultimate period (T_U).The steps required for the method are given below:

- The integral and derivative coefficients have to set (gains) to zero.
- Gradually increase the proportional coefficient from zero to until the system just begins to oscillate continuously (sustained oscillation). 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(T_U).

These two parameters (K_U) & (T_U) are used to find the loop-tuning constants of the PID controller using the following figure.

Table (2.3): ZN-PID Tuning Method

Control Type	K_P	K_I	K_D
P	$0.5K_U$		
PI	$0.45K_U$	$0.54K_U$ / T_U	
PID	$0.6K_U$	$1.2K_U$ / T_U	$0.6K_U T_U$ /8

These gains apply to the ideal, parallel form of the PID controller. When applied to the standard PID form, the integral and derivative time parameters Ti and Td are only dependent on the oscillation period Tu **.**

These 3 parameters are used to establish the correction u(t) from the error e(t) via the equation:

$$U(t) = K_P(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}) \tag{2.11}$$

2.6.3 Trial and error method

The trial and error tuning of PID controllers is said to be widespread in industrial practice. The method starts from some intelligently guessed PID controller parameters. Using these initial controller parameters, the output of the closed-loop system is observed and the controller parameters are modified until the desired output is obtained. Success is highly dependent on the tuning skill of the industrial control engineer and the knowledge available about the process to be controlled.

2.6.4 PID Software tuning method

Most modern industrial facilities no longer tune loops using the manual calculation methods shown above. Instead, PID tuning and loop optimization software are used to ensure consistent results. These software packages will gather the data, develop process models, and suggest optimal tuning. Some software packages can even develop tuning by gathering data from reference changes. Mathematical PID loop tuning induces an impulse in the system, and then uses the controlled system's frequency response to design the PID loop values. In loops with response times of several minutes, mathematical loop tuning is recommended, because trial and error can take days just to find a stable set of loop values. Optimal values are harder to find. Some digital loop controllers offer a self-tuning feature in which very small set point changes are sent to the process, allowing the controller itself to calculate optimal tuning values. Other formulas are available to tune the loop according to different performance criteria. Many patented formulas are now embedded within PID tuning software and hardware modules. Advances in automated PID Loop Tuning software also deliver algorithms for tuning PID Loops in a dynamic or Non-Steady State (NSS) scenario. The software will model the dynamics of a process, through a disturbance, and calculate PID control parameters in response. [7]

2.7 Related work

PID controller is most commonly used in process control applications because of their relative ease of operation and satisfied performances. Users can modify the dynamic properties of this controller by adjusting the three parameters: proportional, integral, and derivative. Most of the previous study depends on rational transfer function or unreliable mathematical model, such as that of Chan and Hui, Zhang et al. They proposed an adaptive GA to tune the parameters of the PID controller and apply it in the sludge aeration process. Moradi designed a predictive PID tuning to search for the optimal the parameters of the PID controller. Proposed by Yu and Hwang, the methodology of linear quadratic regulator was utilized to search for the optimal parameters of the PID controller. The whole design idea has been successfully realized on the speed control of a BLDC motor. Zou et al. applied the particle swarm optimization (PSO) for PID parameters in hydraulic servo edger screw down system. Mookherjee designed an analysis model for a single-stage electro-hydraulic servo-valve[8] .

The tuning of PID controller parameters has been commonly researched and discussed for many years. The tuning process usually needs lots of human power, efforts and time, and in the worst case, the bad tuning parameters could lead to poor controlling performance or even crash in control system. As mentioned earlier, the main aim of PID tuning is to meet the closed-loop system performance specifications, such as peak overshoot, settling time, rising time, and the robust performance of the control loop over different conditions. However, it is often difficult to simultaneously achieve all of these desirable qualities practically .For example, if the PID controller is adjusted to provide a good transient response to a set point change, it usually results in a bad steady state characteristic. On the other hand, if the control system is made to a small steady state error by choosing conservative values for the PID controller, it may result in a slow closed loop response to a set point change. To solve this problem, more and more optimization methods have been used to tune PID parameters. Compared with conventional Ziegler-Nichols (Z- N) method, the optimization methods, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), have been proved their excellence in giving better results by improving the steady state characteristics and performance indices. Particle Swarm Optimization (PSO) is one of the optimization and evolutionary computation techniques. The basic PSO is developed from researches on swarm such as fish schooling and bird flocking. It was firstly introduced in 1995, and plenty of improved versions were proposed. Ant Colony Optimization (ACO) is a meta-heuristic algorithm for the approximate solution of combinatorial optimization problems which was inspired by the foraging behavior of real ant colonies.

Genetic Algorithm (GA) is a stochastic global search method that mimics the process of natural evolution, which is one of the important optimization methods. John Holland formally introduced this method in the United States in the 1970s at the University of Michigan. GA includes three major operators: selection, crossover, and mutation. The technique of GA is proposed as a means of auto-tuning PID controllers by A. Jones and P. Oliveira. The continuous genetic algorithm was proposed to improve the operation time and efficiency for PID adjustment. GA uses the chromosomes fitness value to create a new population consisting of the fittest members, and each chromosome consists of three separate strings constituting a P , I and D term[6].

2.8 Genetic Algorithms

2.8.1 Overview

Genetic algorithms are an approach to optimization and learning based loosely on principles of biological evolution, these are simple to construct, and its implementation does not require a large amount of storage, making them a sufficient choice for an optimization problems. Optimal scheduling is a Nonlinear problem that cannot be solved easily yet, a GA could serve to find a decent solution in a limited amount of time Genetic algorithms are inspired by the Darwin’s theory about the evolution “survival of fittest”, it search the solution space of a function through the use of simulated evolution (survival of the fittest) strategy. Generally the fittest individuals of any population have greater chance to reproduce and survive, to the next generation thus it contribute to improving successive generations However inferior individuals can by chance survive and also reproduce, Genetic algorithms have been shown to solve linear and nonlinear problems by exploring all regions of the state space and exponentially exploiting promising areas through the application of mutation, crossover and selection operations to individuals in the population [9].

The Genetic Algorithm is method for solving optimization problem that is based on natural selection, the process that drives biological evolution. The GAs repeatedly modifies a population of individual solutions. At each step the GAs select individual at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generation, the population evolves toward an optimal solution. Three main steps of GAs are : Selection selects the individuals, called parents that contribute to the next generation. Crossover combines two parents to form children for the next generation. Mutation apply random changes to individual children[15].

2.8.2 Description of Genetic Algorithm

GA has been recognized as an effective and efficient technique to solve optimization problems. GA starts with an initial population containing a number of chromosomes where each one represents a solution of the problem which performance is evaluated by a fitness function. Basically, GA consists of three main stages: Selection, Crossover and Mutation. The application of these three basic operations allows the creation of new individuals which may be better than their parents. This algorithm is repeated for many generations and finally stops when reaching individuals that represent the optimum solution to the problem. The Genetic Algorithm Process Architecture is shown in Figure (2.10) [10].

2.8.3 Genetic Algorithm process architecture

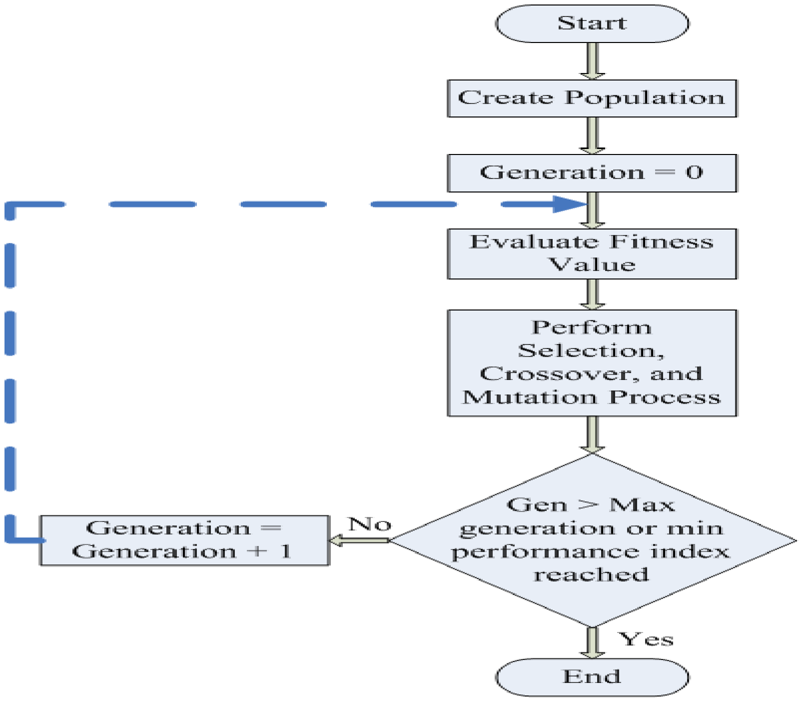


Figure (2.10): Genetic Algorithm process architecture[10]

2.9 DC Motor

The electric motor is a motor that convert electrical energy into mechanical energy. There are two types of motor which are AC motor, and DC motor. A simple DC motor use electricity and magnetic field for producing torque which rotate the motor. Permanent magnet DC motor (PMDC) outperforms to AC motor because it provides better speed control on high torque loads and use in wide industrial application.

DC motors are more usable as it designed to use with batteries and solar Cells energy sources, which provide portability where required it and thus Provide cost effective solution, because it is not possible to have AC power Supply in every place, DC motor shows its response at both voltage and current.

The applied voltage describes the speed of motor while current in the armature Windings show the torque. If applied load increased in the shaft of motor, then in order to sustain its speed motor draws more current from supply and if supply is not able to provide enough current then motor speed will be affected. Generally, it can be said that applied voltage affect speed while torque is controlled by current. DC motors provide more effective results if chopping circuit is used. Low power DC motor usually use in lifting and transportation purposes as low power AC motors do not have good torque capability.DC motor used in railway engines, electric cars, elevators, robotic applications, car windows and wide verify of small appliances and complex industrial mixing process where torque cannot be compromised. There are several types of DC motor but most common are brushed DC motor, brushless DC motor, stepper motor, and servo motor. These DC motors have three winding techniques such as shunt DC motor, series DC motor, and compound DC motor[11].

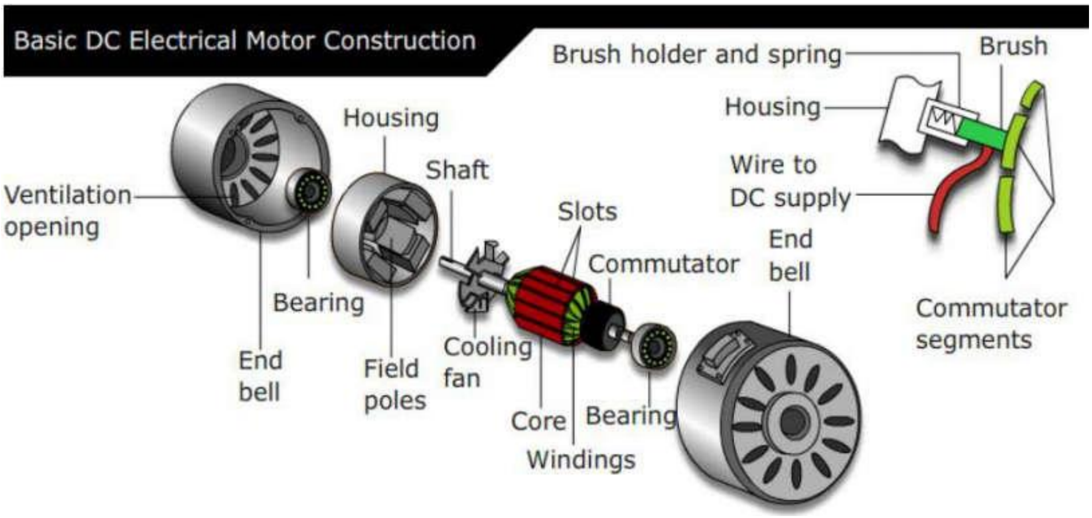


Figure (2.11): DC motor parts

The goal in the development of the mathematical model is to relate the voltage applied to the armature to the velocity of the motor. Two balance equations can be developed by considering the electrical and mechanical characteristics of the system**.**

2.9.1 Electrical characteristics

The equivalent electrical circuit of a dc motor is illustrated in Figure (2.12). It can be represented by a voltage source (Va) across the coil of the armature. The electrical equivalent of the armature coil can be described by an inductance (La) in series with a resistance (Ra) in series with an induced voltage (Vc) which opposes the voltage source.

The induced voltage is generated by the rotation of the electrical coil through the fixed flux lines of the permanent magnets. This voltage is often referred to as the back emf (electromotive force).

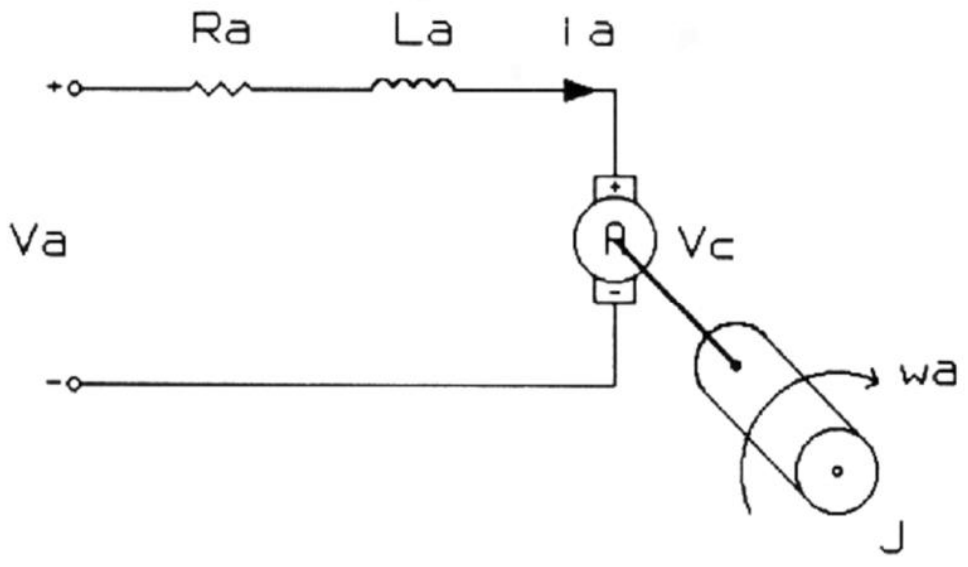


Figure (2.12): Electrical representation of DC motor

Differential equation for the equivalent circuit can be derived by using Kirchhoff’s voltage law around the electrical loop. Kirchhoff’s voltage law states that the sum of all voltages around a loop must equal zero, or

$$V_a - V_{Ra} - V_{La} - V_c = 0 \tag{2.12}$$

According to Ohm's law, the voltage across the resistor can be represented as:

$$V_{Ra} = i_a R_a \tag{2.13}$$

The voltage across the inductor is proportional to the change of current through the coil with respect to time and can be written as:

$$V_{La} = L_a \frac{d}{dt} i_a \tag{2.14}$$

Where L_a is the inductance of the armature coil. Finally, the back emf can be written as:

$$V_c = K_v\, \omega_a \tag{2.15}$$

Where K_v is the velocity constant determined by the flux density of the Permanent magnets, the reluctance of the iron core of the armature, and the number of turns of the armature winding $\omega_a$is the rotational velocity of the armature.
Substituting eqns. (2.13), (2.14), and (2.15) into eqn. (2.12) gives the following Differential equation:

$$V_a - i_a R_a - L_a \frac{d}{dt} i_a - K_v\, \omega_a = 0 \tag{2.16}$$

2.9.2 Mechanical characteristics

Performing an energy balance on the system, the sum of the torques of the motor must equal zero. Therefore,

$$T_e - T_{\omega'} - T_\omega - T_L = 0 \tag{2.17}$$

Where Te is the electromagnetic torque, $T_{\omega'}$ is the torque due to rotational acceleration of the rotor, T_ω is the torque produced from the velocity of the rotor, and T_L is the torque of the mechanical load. The electromagnetic torque is proportional to the current through the armature winding and can be written as:

$$T_e = K_t\, i_a \tag{2.18}$$

Where K_t is the torque constant and like the velocity constant is dependent on the flux density of the fixed magnets, the reluctance of the iron core, and the number of turns in the armature winding. $T_{\omega'}$ Can be written as:

$$T_{\omega'} = J \frac{d}{dt}\, \omega_a \tag{2.19}$$

Where J is the inertia of the rotor and the equivalent mechanical load. The torque associated with the velocity is written as:

$$T_\omega = B\, \omega_a \tag{2.20}$$

Where B is, the damping coefficient associated with the mechanical rotational system of the machine.

Substituting eqns. (2.18), (2.19), and (2.20) into eqn. (2.17) gives the following differential equation:

$$K_t\, i_a - J\frac{d}{dt}\,\omega_a - B\,\omega_a - T_L = 0 \tag{2.21}$$

2.9.3 Transfer Function Block Diagram

The differential equations given in eqns. (2.16) and (2.21) for the armature current and the angular velocity can be written as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{i_a} = -\frac{\mathrm{R_a}}{\mathrm{L_a}}\mathrm{i_a} - \frac{\mathrm{K_v}}{\mathrm{L_a}}\,\omega_\mathrm{a} + \frac{\mathrm{V_a}}{\mathrm{L_a}} \tag{2.22}$$

$$\frac{\mathrm{d}}{\mathrm{dt}}\omega_\mathrm{a} = \frac{\mathrm{K_t}}{\mathrm{J}}\mathrm{i_a} - \frac{\mathrm{B}}{\mathrm{J}}\,\omega_\mathrm{a} + \frac{\mathrm{T_L}}{\mathrm{J}} \tag{2.23}$$

A block diagram for the system can be developed from the differential equations given in eqns. (2.22) and (2.23). Taking the Laplace transform of each equation gives

$$\mathrm{sI_a(s)} - \mathrm{i_a(0)} = -\frac{\mathrm{R_a}}{\mathrm{L_a}}\mathrm{i_a(s)} - \frac{\mathrm{K_v}}{\mathrm{L_a}}\,\text{Ὠ}_\mathrm{a}\,\mathrm{(s)} + \frac{1}{\mathrm{L_a}}\mathrm{V_a(s)} \tag{2.24}$$

$$\mathrm{s}\text{Ὠ}_\mathrm{a}\,\mathrm{(s)} - \omega_\mathrm{a}\,(0) = \frac{\mathrm{K_t}}{\mathrm{J}}\mathrm{I_a(s)} - \frac{\mathrm{B}}{\mathrm{J}}\,\text{Ὠ}_\mathrm{a}\,\mathrm{(s)} - \frac{1}{\mathrm{J}}\mathrm{T_L(s)} \tag{2.25}$$

Around some steady state value are considered, the initial conditions go to zero and all the variables become some change around a reference state, and the equations can be expressed as follows:

$$\mathrm{I_a(s)} = \frac{-\mathrm{K_v}\,\text{Ὠ}_\mathrm{a}\,\mathrm{(s)} + \mathrm{V_a(s)}}{\mathrm{L_a(s)} + \mathrm{R_a}} \tag{2.26}$$

$$\text{Ὠ}_\mathrm{a}\,\mathrm{(s)} = \frac{-\mathrm{K_t I_a(s)} - \mathrm{T_L(s)}}{\mathrm{J(s)} + \mathrm{B}} \tag{2.27}$$

The equations (2.26) & (2.27) can then easily be put into block diagram form. The block diagram obtained from these equations for a permanent magnet dc motor is shown in Figure (2.13).

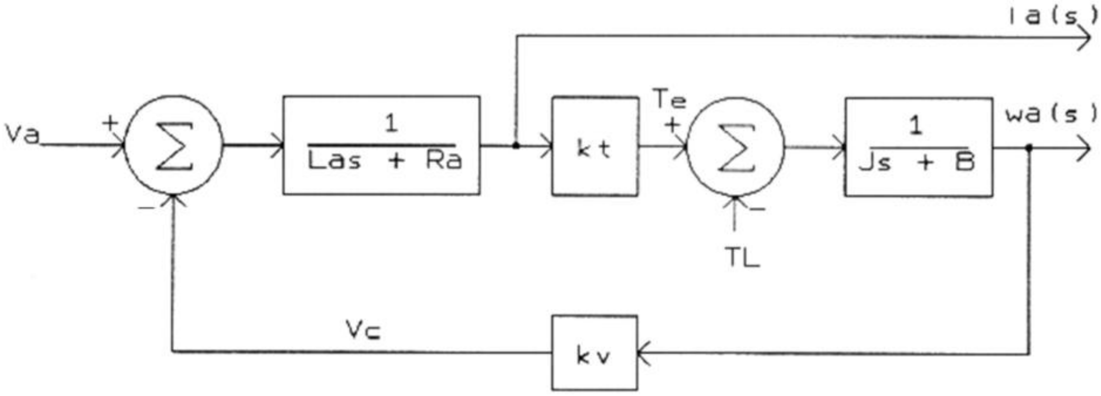


Figure (2.13): Block Diagram Representation of Eqns. (2.26) and (2.27)

The block diagram in Figure (2.13) can be simplified by making the assumption that the load torque is constant. In the case of a sun tracking servo system, the only load torque to be concerned with is the friction in the system, which is relatively constant while the motor is moving. Since the change in TL is zero, it does not need to appear in the block diagram. Also, if one only focuses on the angular velocity as the response of interest, the block diagram becomes as shown in Figure (2.14).

This block diagram is then easily reduced by block diagram algebra to an overall transfer function. Several steps in this process are shown in Figure (2.15), with the overall transfer function between the output angular velocity and input applied voltage given within the last block in Figure (2.15).

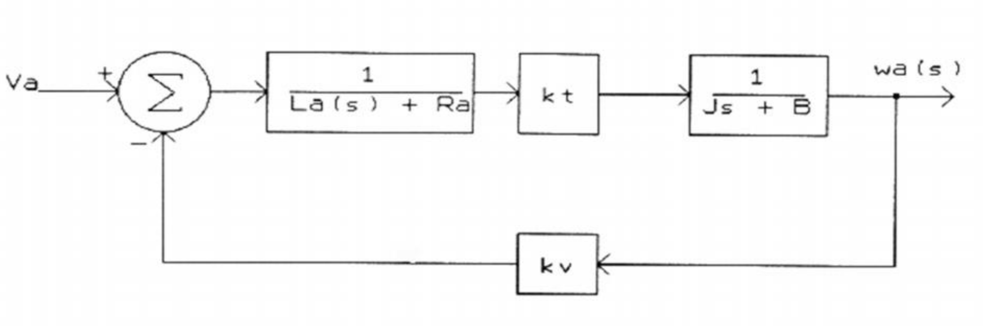


Figure (2.14): Block Diagram of the DC motor model

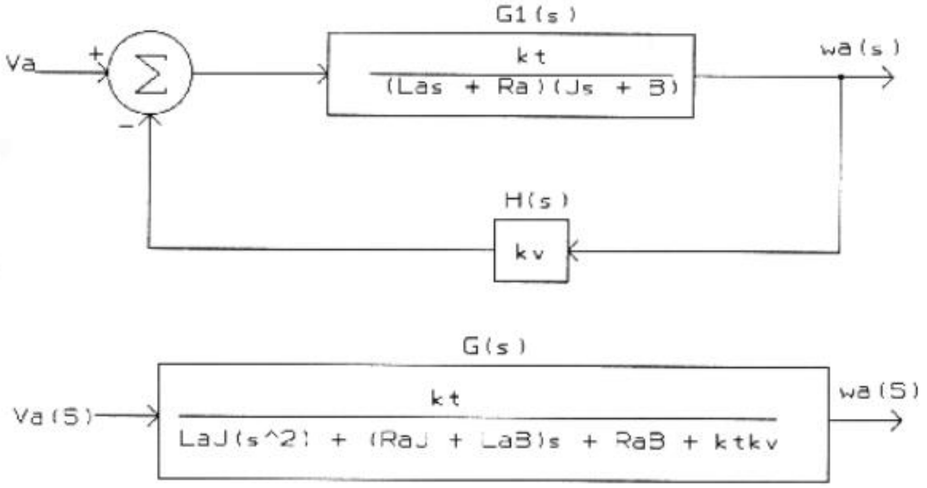


Figure (2.15): Overall Transfer Function for the DC Motor

From the above figure (2.14) the overall transfer function of dc motor can be written as follows:

$$\frac{\omega a(s)}{va(s)} = \frac{kt}{LaJ(S^2)+(RaJ+LaB)S+RaB+ktKv} \tag{2.28}$$

Assume the following values for the physical parameters:

- Moment of inertia of the rotor=0.02 $\frac{kg.m^2}{s^2}$.
- Damping ratio of the mechanical system B=0.2 Nms
- Electromotive force constant k= kt = Kv = 0.02 $\frac{Nm}{Amp}$.
- Electric resistance R=2 ohm.
- Electric inductance L=0.4H.
- Input (V): Source Voltage.
- Output (**ω**): rotational velocity of the armature.
- The rotor and shaft are assumed to be rigid

Replacing the given property values of the DC motor in equation (2.28) yields the open-loop transfer function with V(s) as input and ω(s) as output. This equation indicates the behavior of motor speed for given voltage.

$$\frac{\omega a(s)}{Va(s)} = \frac{0.02}{0.008S^2+0.12S+0.4004} \tag{2.29}$$

Design Criteria:

- **Settling time ≤ 1sec.**
- **Overshoot ≤5%.**
- **Steady-state error ≤ 0.4%** [12]**.**

Chapter Three

System Design

3.1 Importance of DC motors in practice

Easily controlled compared to AC motors. The making A DC motor is an electrical machine which converts direct current electrical power into mechanical power. We use DC motors in almost every aspect of our daily life like in Toys, Fans, Automobile drives. Small DC motors are used in toys, tools and appliances. Larger DC motors are used in electrical automobiles, propulsion systems and elevators. Industrially a good performing DC motor is required of high speed controllability; steady and transient state stability and good Torque-Speed characteristics. The speed of a DC motor is very of highly controlled motors is critical for Industrial purposes. For a satisfactory operation, a DC motor must have an excellent speed tracing and regulation of load. DC motors are easily constructed when compared to the AC motors which are bulky. DC motors are very economical when the requirement of horse power is high. An estimation states that more than 95% of controllers used for controlling speed of a DC motor are PID controllers[13].

The block diagram of a DC motor armature voltage control system is shown in figure (2.14), Simulink model of DC motor shown in figure (3.1)

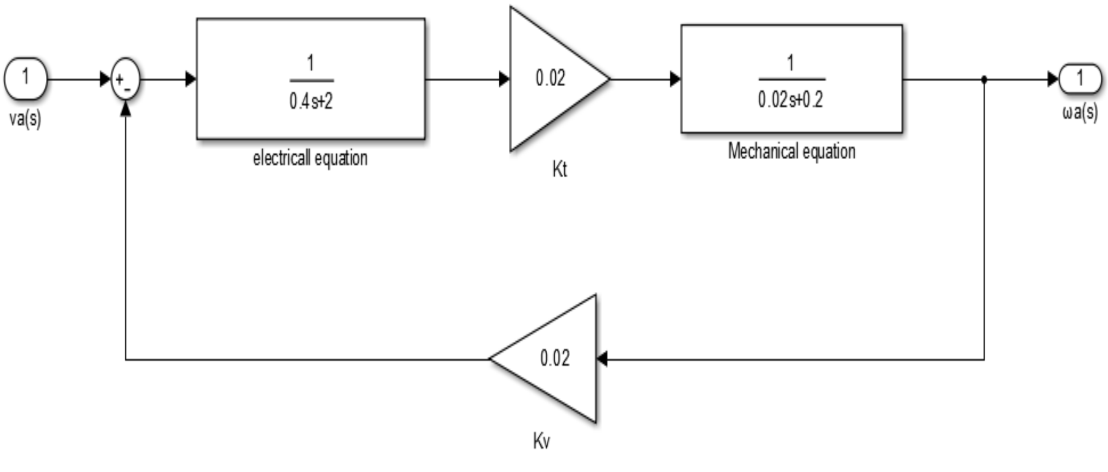


Figure (3.1): Model of DC motor transfer function from Simulink

3.2 PID controllers

Proportional-Integral-Derivative (PID) controller is generic control loop feedback mechanism widely used in industrial control systems, especially for systems with accurate mathematical models. The PID controller calculation involves three separate parameters: proportional, integral and derivative values. The proportional value calculates the value of the current error, the integral value determines the result of the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to be imported into the controlled system. The key issue for PID controllers is the accurate and efficient tuning of parameters. The goal of PID controller tuning is to determine parameters that meet the closed-loop system performance specifications over a wide range of operating conditions [5]. Block diagram of PID controller with DC motor model shown in figure (3.2)

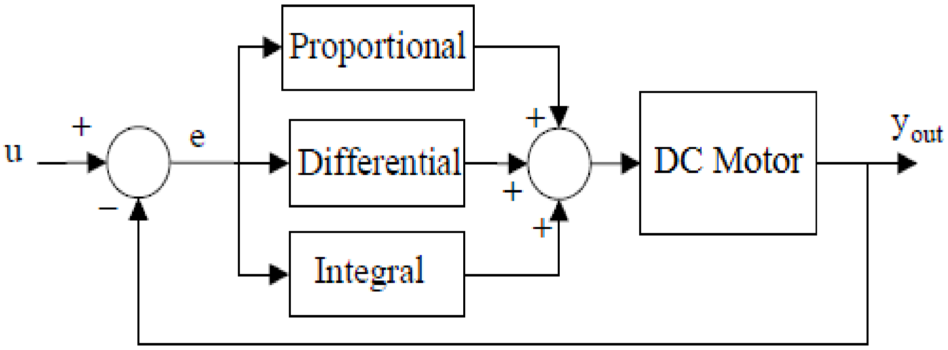


Figure (3.2): Model of DC motor with PID controller[5]

3.3 Tuning of PID controller Using Ziegler-Nichols method

AS we discussed in chapter two after calculating the value of (K_U) &(T_U) we can use it to find the loop-tuning constants of the PID controller using Table (2.3). PID controller system Using Ziegler-Nichols Tuning Method is shown in figure (3.3) [9].

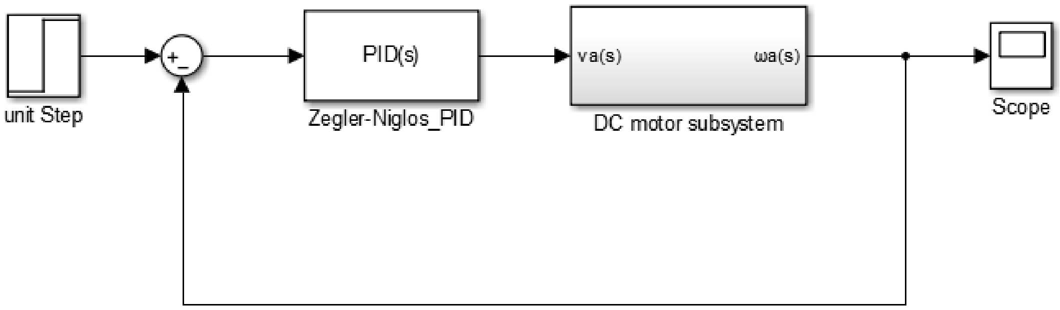


Figure (3.3): Simulink model of DC Motor with (PID-ZN)

3.4 Tuning of PID controller Using trial and error approach

Using a trial and error approach discussed in chapter two, the following gains were selected to tune PID parameters

Kp=70, Ki=170,Kd=5.

After using the above gains (**Kp=70; ki=170;kd =5**) the PID controller system Using trial and error Tuning Method is shown in figure (3.4)

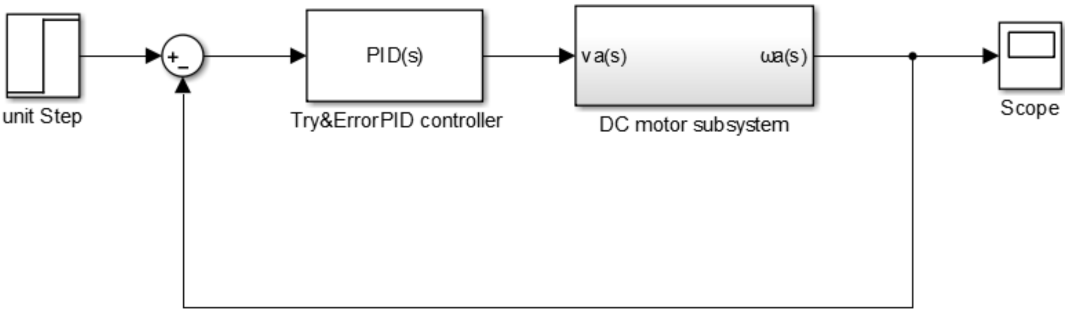


Figure (3.4): Simulink Model of DC Motor with (PID -try &error)

3.5 PID Control Gains using Genetic Algorithm (GA)

The objective of this research is to understand and investigate the efficiency of genetic algorithm in PID tuning because PID tuning, it is important to obtain the best solution so that the controller will have the fastest and stable response time.

3.5.1 The soft Computing technique of PID controller

The process of computing and setting the optimal values of Kp, Ki and Kd to get desired response from a control system, called tuning.PID controllers are mostly tuned by Ziegler-Nicholas method. Many model based controller techniques such as internal model control are used in conjunction with PID controller to improve the dynamic response of the process. Apart from conventional tuning methods there are many soft computing based intelligent tuning rules like Particle Swarm Optimization (PSO), Genetic Algorithms (GA) etc. The soft computing techniques for a PID controller considerably reduce the overshoot and rise time as compare to any other tuning method [15].

The soft computing techniques for a PID controller are usually used for data modeling and optimization of a cost function, and have been used in PID tuning. Some examples are neural networks (computational models to simulate complex systems), genetic algorithm and differential evolution. The optimization techniques require a cost function they try to minimize. There are four types of cost functions used commonly:

- Integral Absolute Error

$$\mathrm{IAE}= \int_0^{\tau}|e(t)| \tag{3.1}$$

- Integral square Error

$$\mathrm{ISE}= \int_0^{\tau}|e(t)|^2 \tag{3.2}$$

- Integral time Absolute Error

$$\mathrm{ITAE}= \int_0^{\tau} t\,|e(t)| \tag{3.3}$$

- Integral time square Error

$$\mathrm{ITSE}= \int_0^{\tau} t\,|e(t)|^2 \tag{3.4}$$

3.5.2 Genetic Algorithms based PID controller

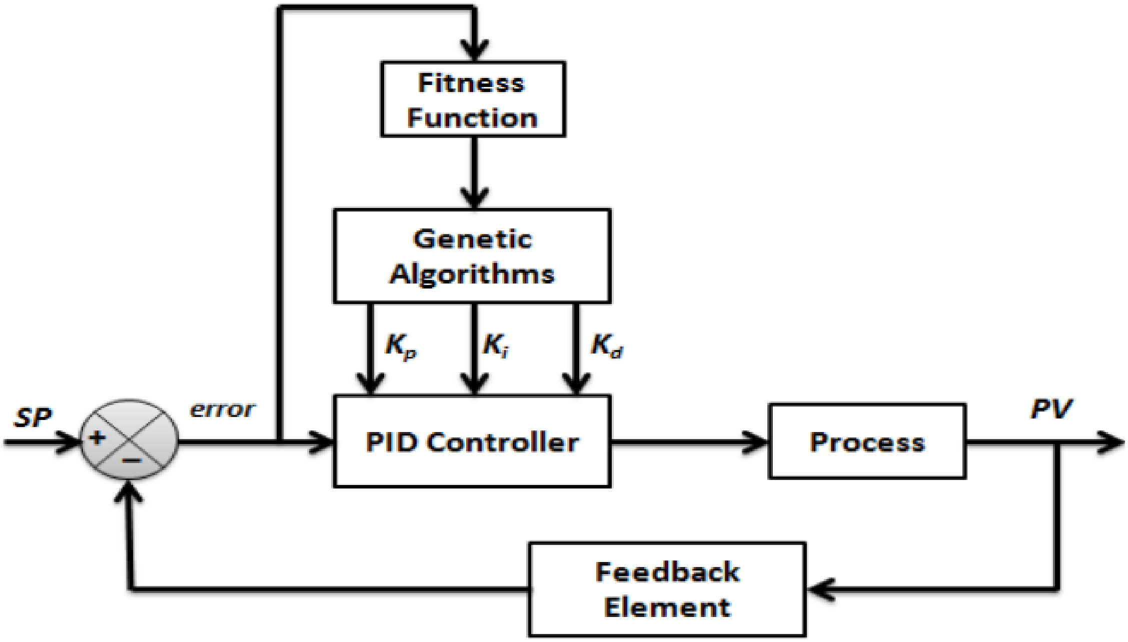


Figure (3.5): Block Diagram of Genetic Algorithms based PID controller

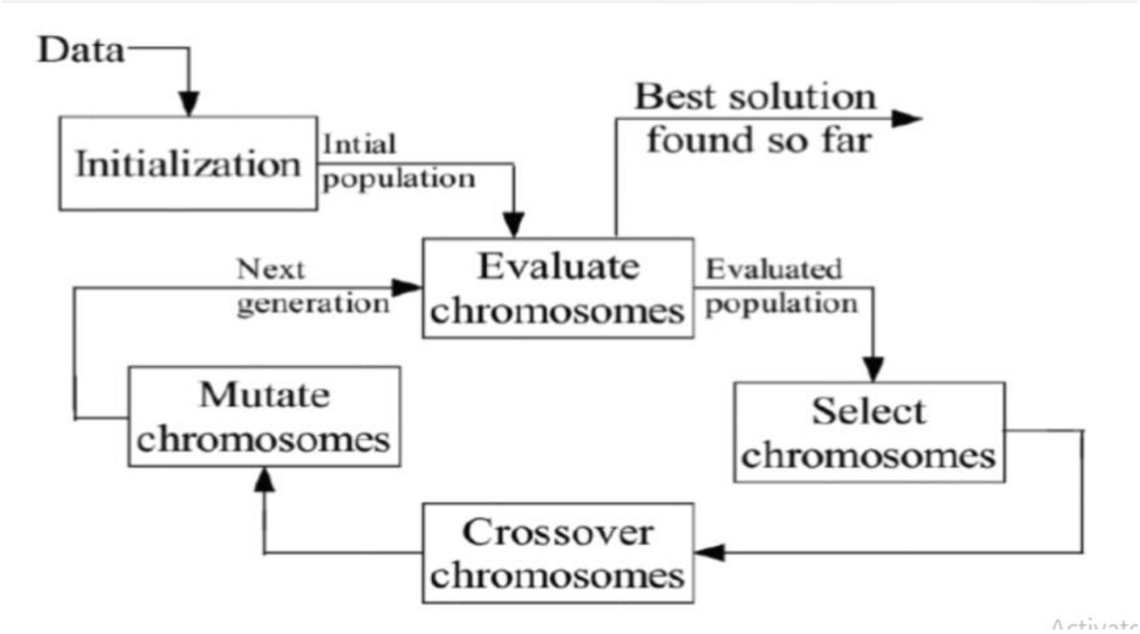


Figure (3.6): Optimization using Genetic Algorithm

3.5.3 Mathematical model of the fitness Function

In order to meet the requirements of PID control, the mathematical model of the fitness function is specified as follows:

1. The fitness function contains the absolute value of the system deviation e (t), and in order to achieve different emphasis control schemes, a certain proportion operation is performed on the deviation with a coefficient of.

2. The fitness function contains the control quantity input u (t) at that moment, and the square of the input is used as a factor of the fitness function, and a proportional operation is also performed on the item with a coefficient of.

3. The fitness function contains the rise time t(u) of the system, and the term is proportionally calculated with a coefficient of. U^2

4. Once the system has an overshoot, use the overshoot as an important component of the fitness function, and use a relatively large scale factor as a penalty function to avoid overshoot. From the above points, the objective function of the system can be obtained:

If ey(t)>>0

$$J = \int_0^\infty (\mathrm{w}_1|\mathrm{e(t)}\,| + \mathrm{w}_2\, u^2\,(t))dt + \mathrm{w}_3\,.\,\mathrm{t(u)} \tag{3.5}$$

If ey(t)<0

$$J = \int_0^\infty (\mathrm{w}_1|\mathrm{e(t)}\,| + \mathrm{w}_2\, u^2\,(t) + \mathrm{w}_4|\mathrm{ey(t)}|)\mathrm{dt} + \mathrm{w}_3\,.\,\mathrm{t(u)} \tag{3.6}$$

Where ey (t) =y (t)-y (t-1), ey (t) is the output, and the fitness function is set to the inverse of the objective function, so the fitness function is F = $\frac{1}{j}$ [16].

3.5.4 Steps of Genetic Algorithm

The basic process of genetic algorithm that being used in MATLAB can be outlined into 6 basic steps as follow:

Step 1: Start: Generate random population of chromosomes which represent the number of solutions that is suitable for the problem.

Step 2: Fitness: Evaluate the fitness of each chromosome in the population.

Step 3: New population: create a new population by repeating following steps until the new population is complete:

a) Selection: Select two parent chromosomes from a population according to their fitness. Better the fitness, the bigger chance to be selected to be the parent.

b) Crossover: crossover the parents to form new offspring that is children. If no crossover was performed, offspring is the exact copy of parents.

c) Mutation: mutate new offspring at each locus.

d) Reproduction: Place new offspring in the new population.

Step 4: Replace: Use new generated population for a further run of the algorithm.

Step 5: Test: If the end condition is satisfied, stop, and return the best solution in current population.

Step 6: [Loop] Go to step 2. [14].

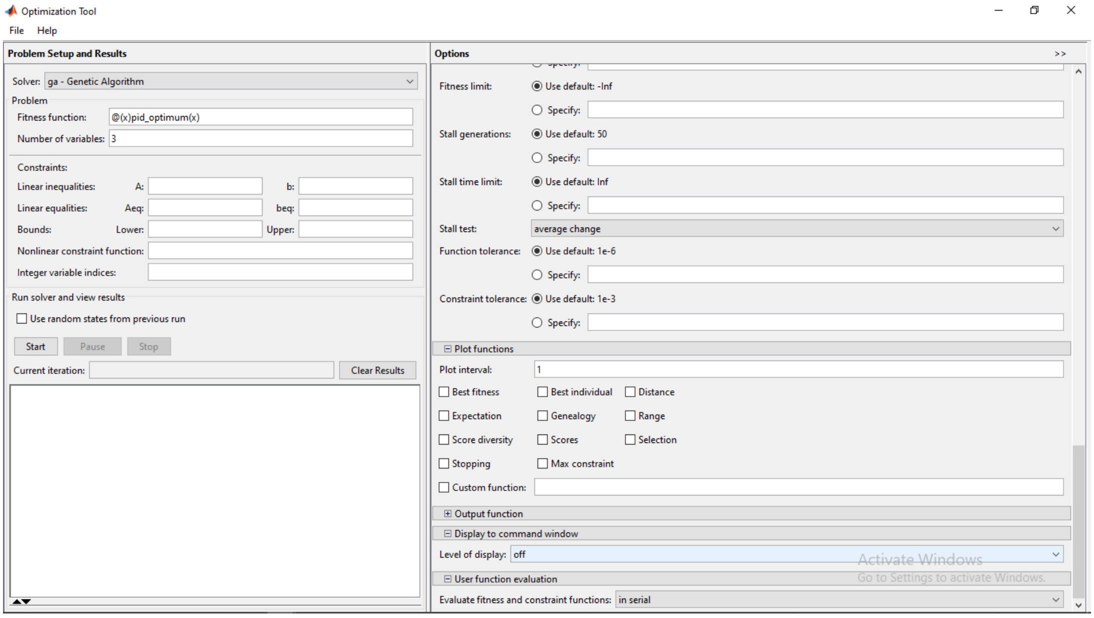
Values of parameter used in evaluating Genetic Algorithm to find optimize results in this research are given in table (3.1).

Table (3.1): Genetic Algorithm parameters

GA Property	Function Definition
Population Size	50
Maximum Generations	300
Performance Index	Integral time absolute error(ITAE)
Selection Function	Stochastic uniform
Crossover Method	Constraint Dependent
Crossover Fraction	0.8
Mutation FUNCTION	Constraint Dependent
Elite Count	2.5
Migration	forward

3.5.5 Genetic Algorithm for optimization using MATLAB Toolbox

The MATLAB package comes with sophisticated libraries for matrix operations, general numeric methods and plotting of data, therefore MATLAB become first choice of programmer to implement scientific, graphical and mathematical applications and for the GA implementation MATLAB is come with special tool that is GA-tool or Optimtool [9].



Solver: ga - Genetic Algorithm

Problem

Fitness function: @(x)pid_optimum(x)

Number of variables: 3

☐ Use random states from previous run

Fitness limit: ● Use default: -Inf ○ Specify:

Stall generations: ● Use default: 50 ○ Specify:

Stall time limit: ● Use default: Inf ○ Specify:

Stall test: average change

Function tolerance: ● Use default: 1e-6 ○ Specify:

Constraint tolerance: ● Use default: 1e-3 ○ Specify:

Plot interval: 1

☐ Best fitness ☐ Best individual ☐ Distance

☐ Expectation ☐ Genealogy ☐ Range

☐ Score diversity ☐ Scores ☐ Selection

☐ Stopping ☐ Max constraint

☐ Custom function:

Level of display: off

Evaluate fitness and constraint functions: in serial

Figure (3.7): GAs in Matlab’s Optimization Toolbox

Using Genetic Algorithm for optimization using MATLAB discussed in chapter two, the following gains represent final values to tune PID parameters **(kp= 271.6619; ki= 952.9365; kd= 18.111)**.

The PID controller system using Genetic Algorithm for optimization using MATLAB Tuning Method is shown in figure (3.8)

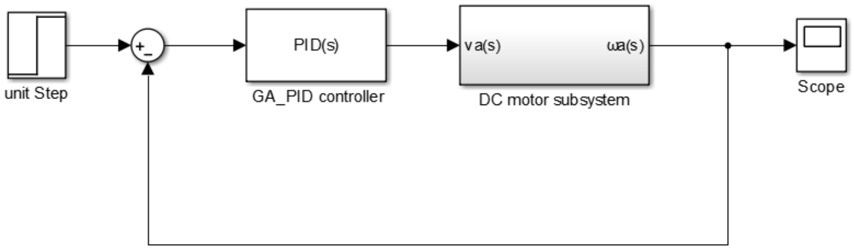


Figure (3.8): Simulink model of DC Motor with (PID-GA)

Chapter Four

Results and Discussion

4.1 Step response simulation results

After simulating the process control of speed control for DC motor model by Using different PID Controller tuning methods using MATLAB/SIMULINK, the the step response of the speed control of DC Motor using different tuning approaches in the following figures.

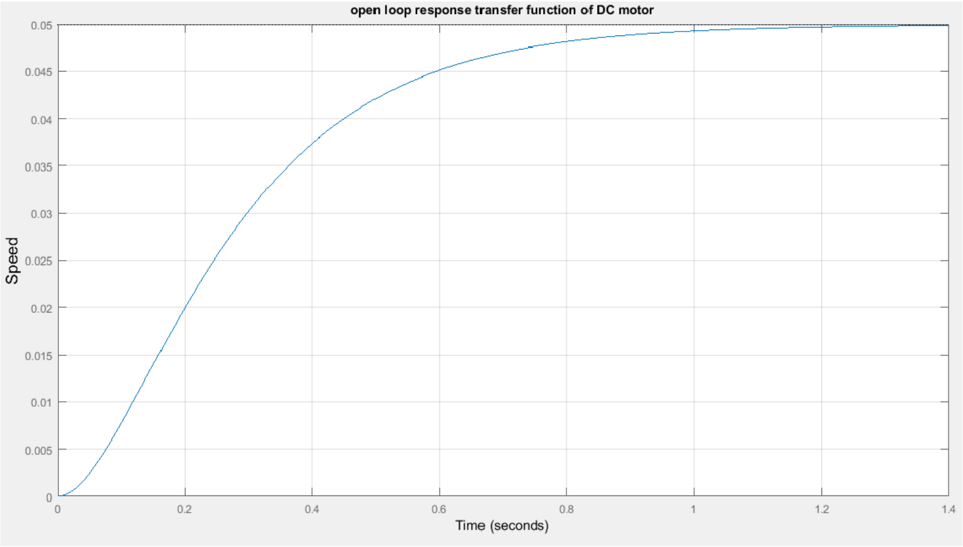


Figure (4.1): Open loop step response of Dc Motor

From above figure the step response specifications to control speed of DC motor it is too low compare with reference signal (Step Signal).

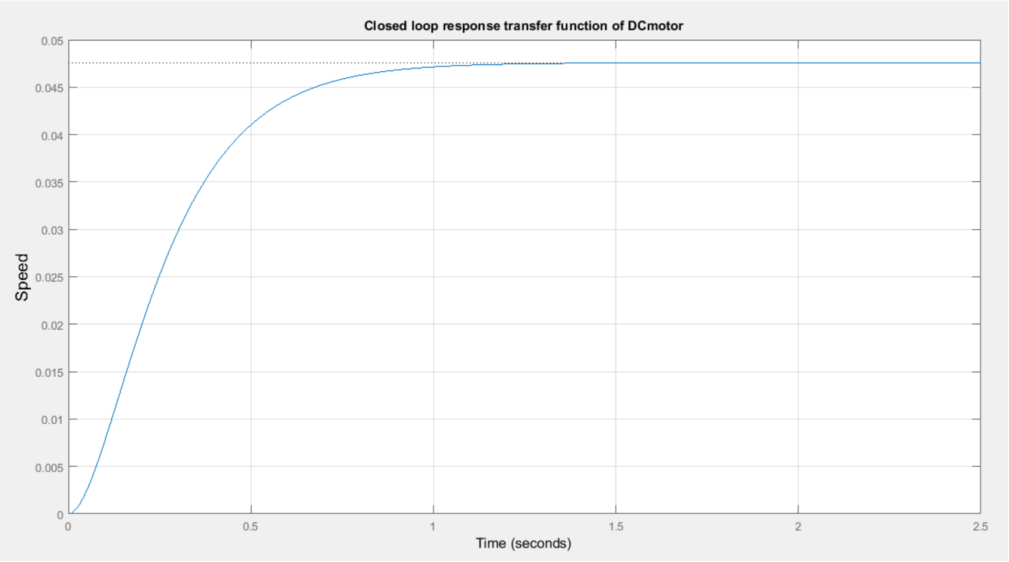


Figure (4.2): closed loop step response of Dc Motor

From above figure the step response specifications to control speed of DC motor it is better compare with open loop Step Response.

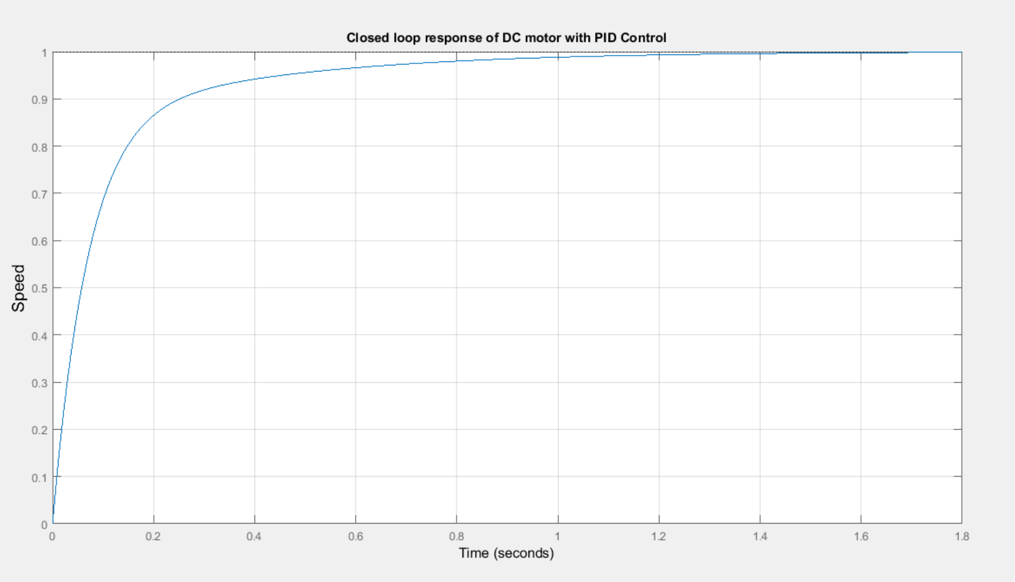


Figure (4.3): Step response of Dc Motor with PID controller

From above figure the Step Response specifications to control speed of DC motor using PID Controller it is better compare with using closed loop step response**.**

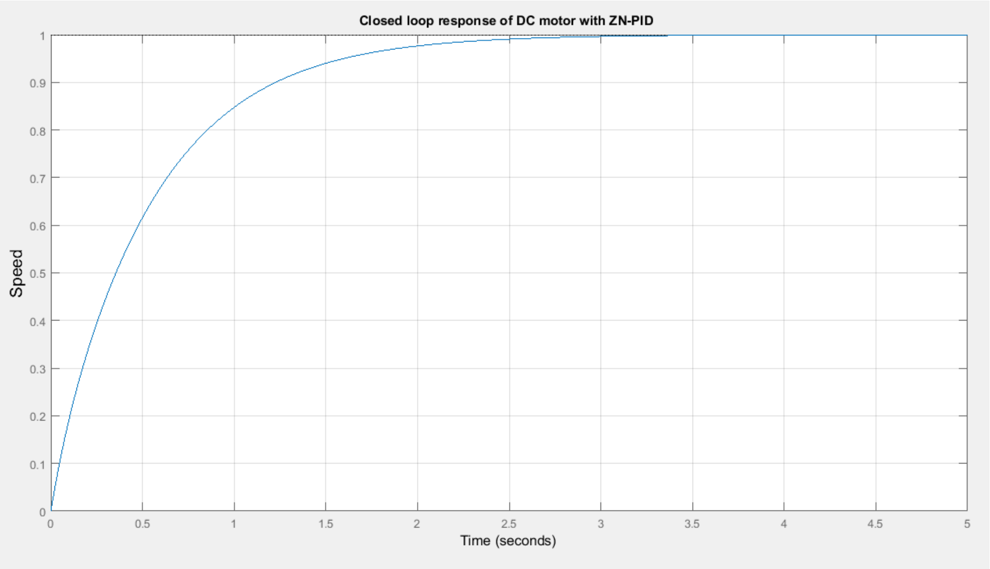


Figure (4.4) Step response of Dc Motor with (PID-ZN) controller

From above figure the step response specifications to control speed of DC motor using (PID-ZN) Controller it is better compare with using other methods.

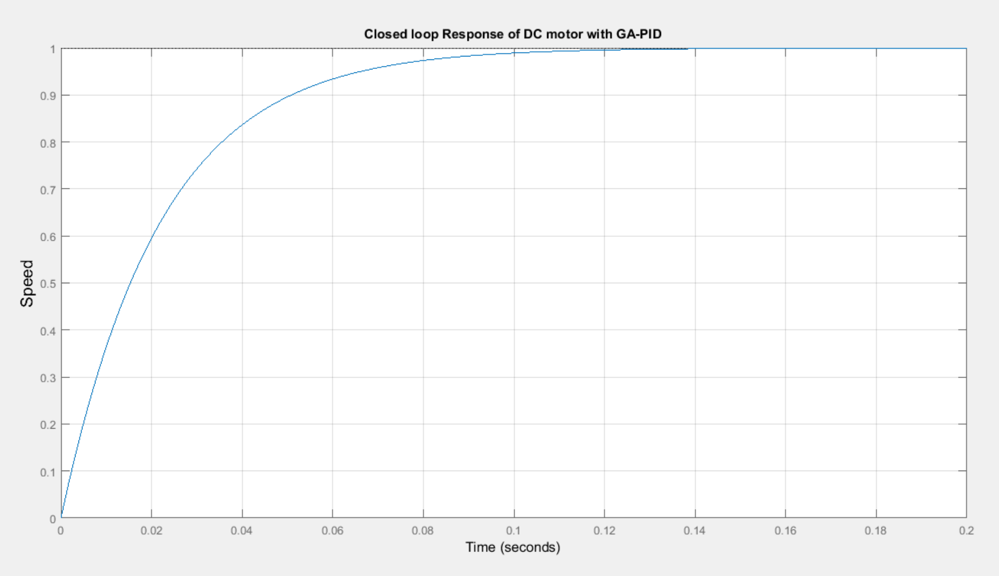


Figure (4.5): Step response of Dc Motor with (PID-GA) controller

From above figure the step response specifications to control speed of DC motor using (PID-GA) Controller it is better compare with using other methods

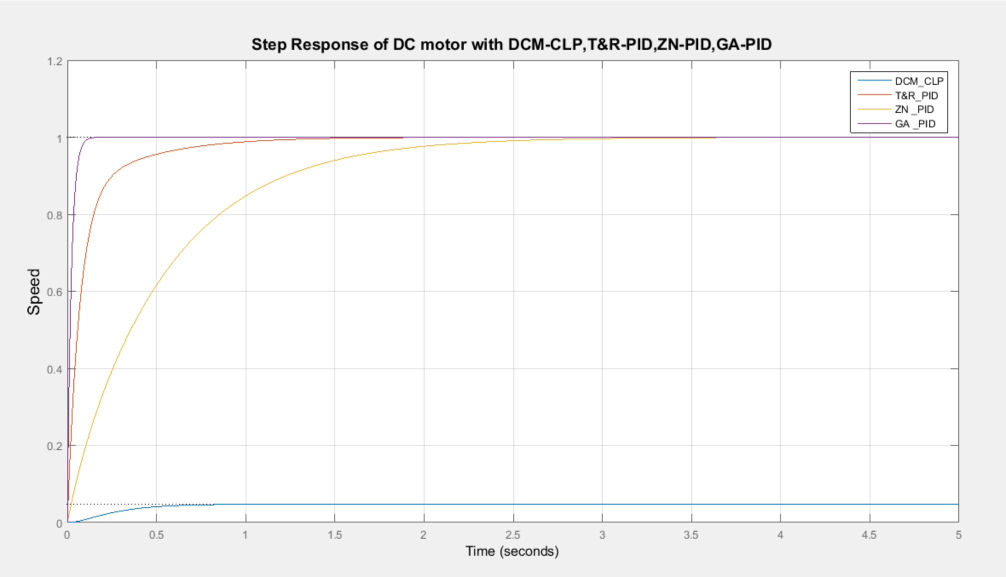


Figure (4.6): Step response of Dc Motor with different controllers

From above figure the step response specifications to control speed of DC motor using (PID-GA) Controller it is giving good results compare with using other methods

4.2 Comparisons of different tuning methods

As mentioned earlier, the main aim of PID tuning is to meet the closed-loop system performance specifications, such as peak overshoot, settling time, rising time, and the robust performance of the control loop such as integral error (IE), integral absolute error (IAE) , integral square error (ISE),integral time square error (ITSE) and integral time absolute error (ITAE). The comparisons between different tuning method based on closed-loop system performance specifications and performance criterion index are shown in the table (4.1) & table (4.2) respectively.

Table (4.1): Characteristics of the closed-loop step response

characteristics	**DCMCLP**	**PID-TR**	**PID-ZN**	**PID-GA**
Rise time(sec)	**0.487**	**0.244**	**1.18**	**0.0485**
settling time(sec)	**0.858**	**0.801**	**2.09**	**0.0864**
Peak amplitude	**0.0476**	**1**	**1**	**1**
%overshoot	**0**	**0**	**0**	**0**

Table (4.2): Performance criterion Index of the closed-loop

characteristics	DCM-CLP	PID-TR	PIDZN	PID-GA
IE	**8.585**	**0.1178**	**0.5234**	**0.02101**
IAE	**8.585**	**0.1177**	**0.5234**	**0.02306**
ISE	**8.19**	**0.04643**	**0.2552**	**0.01469**
ITSE	**44.93**	**0.00225**	**0.06811**	**0.0001437**
ITAE	**47.16**	**0.02694**	**0.2815**	**0.0007163**

From the simulation results tabulated in Table (4.1) & Table (4.2): it is concluded that by step response specifications to control speed of DC motor using PID-GA controller such as rise time, settling time, IE, IAE,ISE,ITSE &ITAE is better than PID-ZEG, PID-TR & DC motor closed loop controller (DCM-CLP).

Simulation results reflect that the Genetic Algorithm tuning method has a better control performance than the conventional methods. Genetic Algorithms based PID controller (PID-GA) results in faster rise time, quicker settling time, minimum of (IE, IAE, ISE, and ITSE & ITAE) which means good transient response and this led to better controllability and less sensitivity to change in system condition.

4.3 Comparisons between PID-GA tuning method and the PID controller using trial And error

The comparisons between the PID-GA auto tuning method and the PID Controller Using Trial And error mentioned In The Scientific Paper[12] for the controlling speed of DC motor in term of rise time ,settling time , IE, IAE, ISE,ITSE7ITAE are illustrated in the following figures and table(4.3)

From figure (2.14) the overall transfer function of dc motor can be written as follows:

$$\frac{\omega a(s)}{va(s)} = \frac{kt}{LaJ(S^2)+(RaJ+LaB)S+RaB+ktKv} \qquad (2.26)$$

Assume the following values for the physical parameters:

- Moment of inertia of the rotor=0.02 $\frac{kg.m^2}{s^2}$.
- Damping ratio of the mechanical system B=0.2 Nms.

- Electromotive force constant k= kt = Kv = 0.02 $\frac{Nm}{Amp}$
- Electric resistance R=2 ohm.
- Electric inductance L=0.4H.
- Input (V): Source Voltage
- Output (**ω**): rotational velocity of the armature.
- The rotor and shaft are assumed to be rigid

Replacing the given property values of the DC motor in equation (2.26) yields the open-loop transfer function with V(s) as input and ω(s) as output. This equation indicates the behavior of motor speed for given voltage.

$$\frac{\boldsymbol{\omega}\mathbf{a(s)}}{\mathbf{Va(s)}} = \frac{\mathbf{0.02}}{\mathbf{0.008}\boldsymbol{S}^{\mathbf{2}}\mathbf{+0.12}\boldsymbol{S}\mathbf{+0.4004}} \tag{2.27}$$

Design criteria

- **Settling time ≤ 1sec.**
- **Overshoot ≤5%.**
- **Steady-state error ≤ 0.4%** [12]**.**

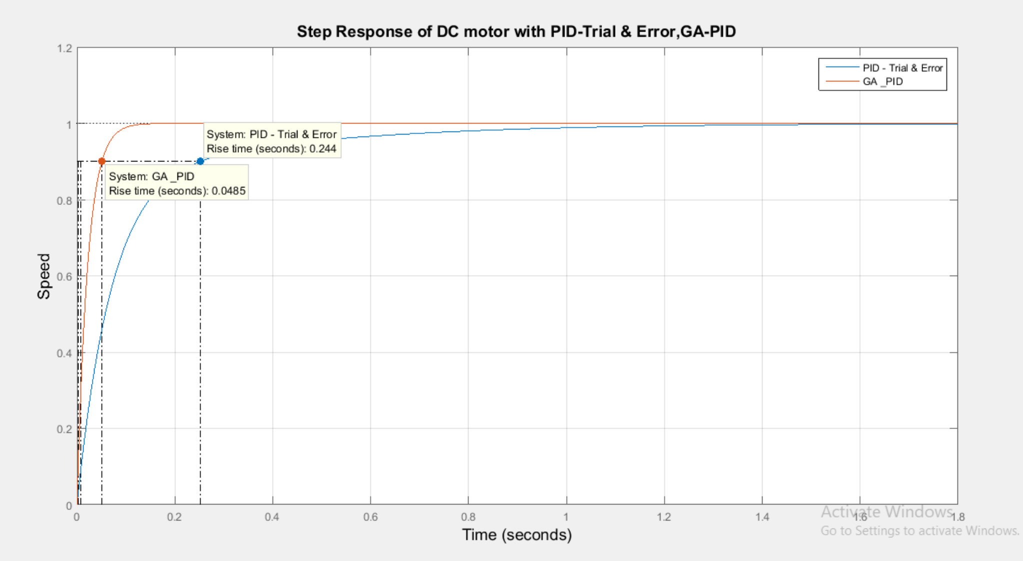


Figure (4.7): Comparison (T_r) between the (PID-GA) & (PID -T & E)

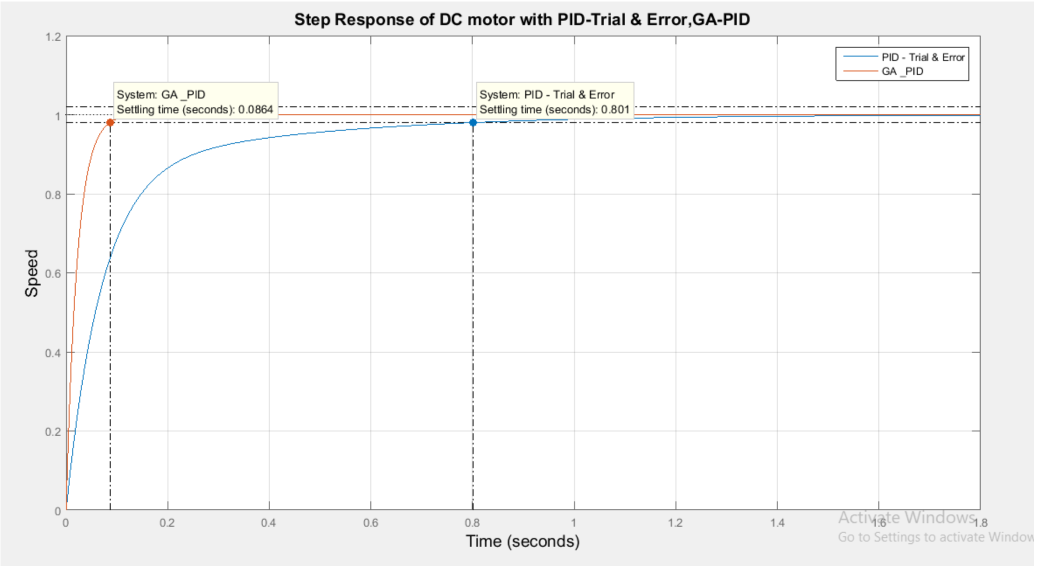


Figure (4.8): Comparison (T_S) between the (PID-GA) & (PID -T & E)

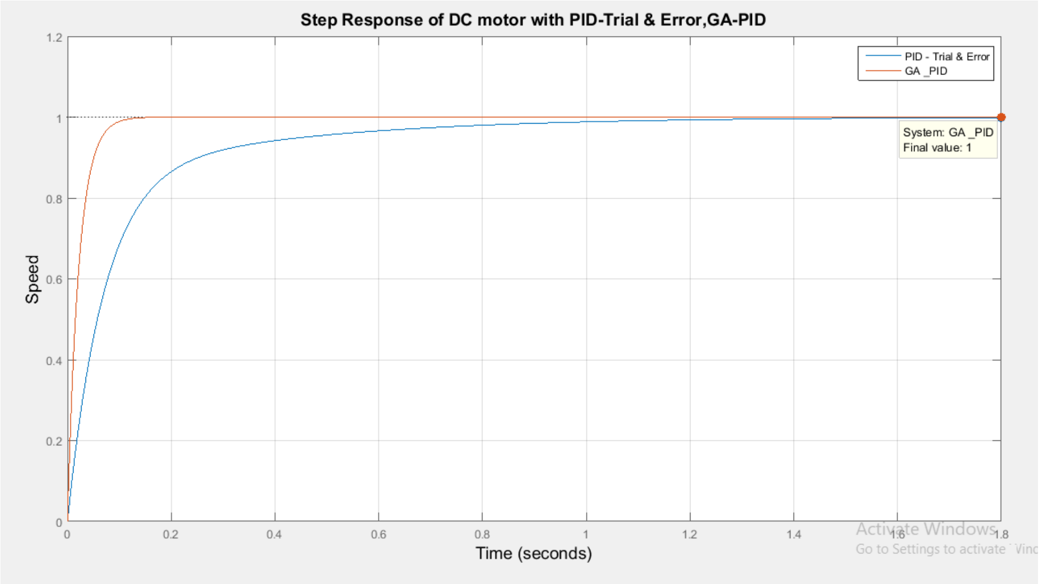


Figure (4.9): Comparison (E_{SS}) between the (PID-GA) & (PID -T & E)

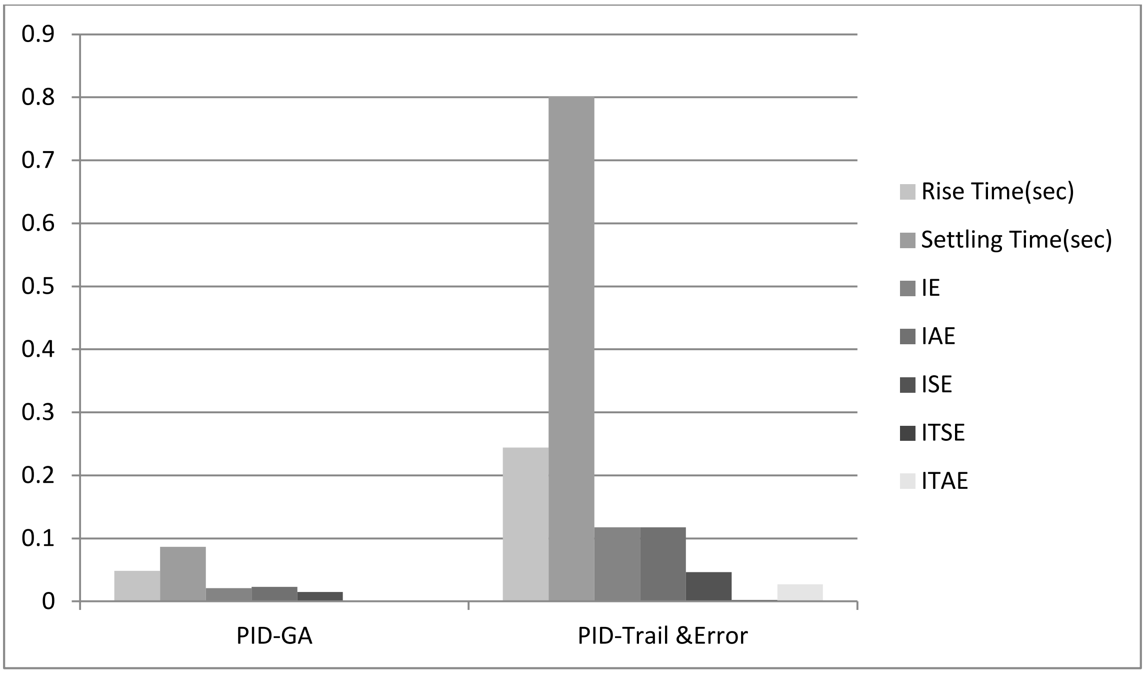


Figure (4.10): Comparison Between (PID-GA) & (PID -T & E)

Table (4.3): Comparison characteristics Between (PID-GA) & (PID -T& E)

characteristics	***(PID- T&r)***	***(PID-GA)***
Rise time(sec)	**0.244**	**0.02101**
Settling time(sec)	**0.801**	**0.0864**
Final value	**1**	**1**
IE	**0.1178**	**0.02101**
IAE	**0.1177**	**0.02306**
ISE	**0.04643**	**0.01469**
ITSE	**0.00225**	**0.0001437**
ITAE	**0.02694**	**0.0007163**

From the simulation results using MATLAB software illustrated in figures(4.7) , figures(4.8),figures(4.9), figures(4.10) & table(4.3) We note that the GA-PID auto tuning method have better performance in terms of rise time(tr),settling time(ts) ,integral error (IE),integral absolute error (IAE),integral square error(ISE),integral time square error(ITSE)& integral time absolute error (ITAE)compared with PID Controller Using Trial And error mentioned In The Scientific Paper[12].

The results illustrate that the rise time =0.244 sec and 0.0485sec, the settling time = 0.0864sec and 0.801sec and the final value=1(steady-state error=0) for auto tuning of PID controller using genetic algorithm (PID-GA) and PID Controller Using Trial And error (PID -Trial &Error) respectively. Minimum rise time and minimum settling time means better transient response and better transient response led to improve the performance for DC motor control speed. Finally we can say that the auto tuning of PID controller using genetic algorithm (PID-GA) contributes better response compared with conventional methods.

Chapter Five

Conclusion and Recommendations

5.1 Conclusion

PID controller has been widely used in the field of industrial control due to its simple principle, easy implementation and wide application. It has been the research direction of experts for how to determine PID parameters to make the system stronger. In this research PID parameters are tuned using three different methods firstly Tuning of PID Controller Using A. Ziegler-Nichols Method (PID-ZN), Secondly Tuning of PID Controller Using Trial and error approach and thirdly auto tuning of PID controller using genetic algorithm (PID-GA).the step response outcomes for three tuning methods are determined using MATLAB software. We found that the Transient response characteristics such as the rise time and settling time are very smaller when we used auto tuning of PID controller using genetic algorithm (PID-GA) compared with other two methods. Also performance criterion index such as integral error (IE), integral absolute error (IAE) and integral square error (ISE) for three tuning methods are evaluated but the results demonstrate that the auto tuning of PID controller using genetic algorithm (PID-GA) have better performance compared with other two mentioned methods such as conventional methods like A. Ziegler-Nichols Method and Trial and error method.

Better transient response and also better performance criterion index for the auto tuning of PID controller using genetic algorithm (PID-GA) compared with conventional methods such as A. Ziegler-Nichols Method and Trial and error method led to better controllability and less sensitivity to change in system condition.

5.2 Recommendations

- In this project auto tuning of PID controller using genetic algorithm (PID-GA) was used to determine the optimal PID parameters for SISO system, we can designed Similar One for MIMO system to evaluate its performance.
- Implementation of this tuning method for other types of motors.

- Adapting sophisticated control strategies such as neural network and neurofuzzy control techniques can be used.
- The parameters of PID controller can also be tuned based on other optimization algorithms such as fruit fly optimization algorithm (FOA).

References

[1] S. Of and I. Controllers, “Speed Control of Dc Motor Using Fuzzy Logic Technique Speed Control of Dc Motor Using Fuzzy Logic Technique,” no. January 2012, 2015, doi: 10.9790/1676-0364148.

[2] “Pid controller design for controlling dc motor speed using matlab application mohamed farid bin mohamed faruq universiti malaysia pahang.”

[3] C. Systems, “About the Tutorial Copyright & Disclaimer.”

[4] Rao, V.Dukkipati,” Analysis and Design of control systems using MATLAB”, 2006.

[5] J. Han, P. Wang, and X. Yang, “Tuning of PID controller based on fruit fly optimization algorithm,” in *2012 IEEE International Conference on Mechatronics and Automation, ICMA 2012*, 2012, pp. 409–413, doi: 10.1109/ICMA.2012.6282878.

[6] P. I. D. Control, “PID Control,” pp. 216–251, 2002.

[7] https://en.wikipedia.org/wiki/PID_controller#PID_controller_theory

[8] K. M. Elbayomy, J. Zongxia, and Z. Huaqing, “PID Controller Optimization by GA and Its Performances on the Electro-hydraulic Servo Control System,” vol. 21, pp. 378–384, 2008.

[9] M. Saraswat, “Optimization of Function by using a New MATLAB based Genetic Algorithm Procedure Available Online at www.ijarcs.info Genetic Algorithm for optimization using MATLAB,” no. July 2019, 2013, doi: 10.5120/10001-4212.

[10] M. A. Fkirin, M. Ahmed, and M. Hassan, “Speed Control of DC Motor Using PID Controller Based on Artificial Intelligence Techniques,” no. May 2015, 2013, doi: 10.1109/CoDIT.2013.6689543.

[11] “SPEED CONTROL OF DC MOTOR BY USING FUZZY LOGIC CONTROLLER,” no. June, 2012.

[12] S. V. Output, L. C. Pid, and C. Using, “EML 4312: Control of Mechanical Engineering Systems Fall 2009, University of Florida A Lecture Note on DC Motor Speed Control Dr. Redmond Ramin Shamshiri,” pp. 1–9, 2009.

[13] K. Biswal, “Speed Control of Separately Excited DC Motor using Fuzzy Logic Controller A thesis submitted in fulfilment of prerequisites of.”

[14] T. Ahmad, F. Ku, M. F. Atan, N. A. Rahman, F. Salleh, and N. A. Wahab, “Optimization of PID Tuning Using Genetic Algorithm,” vol. 2, no. 2, pp. 97–106, 2015.

[15] Singh, C. (2015). Genetic Algorithms Based PID controller Design. 3(3), 2– 5

[16] I. O. P. C. Series and M. Science, “Self-Tuning of PID Parameters Based on Adaptive Genetic Algorithm Self-Tuning of PID Parameters Based on Adaptive Genetic Algorithm,” 2020, doi: 10.1088/1757-899X/782/4/042028.

Appendix

- Tuning of PID parameters for DC motor control speed using different tuning methods:

```
%%%% Simulation Results for DC motor control speed using different PID tuning methods%%%

%Mussab Mohamed yousif Adam :( mussab_elex12@hotmail.com).

%MSC student, Electronic Eng. SUST (computer and networking).

Clear all
 Close all
clc
% The physical parameters for DC motor under study.
J=0.2;
b=0.2;
kt =0.02;
kv=0.02;
R=2;
L=0.4;
% Transfer function of dc motor.
num=kt;
den=[J*L J*R+b*L b*R+kv*kt ];
TF_DC=tf(num,den)
%0pen loop transfer function of DC motor.
disp('open loop transfer function of DCmotor')
figure
```

```
step(TF_DC), grid on
title('open loop response transfer function of DC motor')
xlabel('Time','fontsize',12)
ylabel('Speed','fontsize',14)
%Closed loop transfer function of DC motor.
H= 1;
disp('Closed loop transfer function of DC motor')
closed_loop =feedback (TF_DC, H)
figure
step(closed_loop), grid on
title('Closed loop response transfer function of DCmotor')
xlabel('Time','fontsize',12)
ylabel('Speed','fontsize',14)
%PID control gain, using trial and error method%
%PID control gain.
Kp =70;
Ki =170;
Kd =5;
% PID transfers function.
numPID=[kd  kp  ki];
denPID=[1 0];
% Closed loop TF of DC motor with PID controller.
num_DC_PID=conv (num, numPID);
den_DC_PID=conv (den, denPID);
[NumPID_CLP, DenPID_CLP]=cloop (num_DC_PID, den_DC_PID);
```

```
Disp ('Closed loop of DC motor with PID controller')

cloop=tf(NumPID_CLP,DenPID_CLP)

Figure

Step (cloop), grid on

title (‘closed loop response of DC motor with PID Control')

xlabel('Time','fontsize',12)

ylabel('Speed','fontsize', 14)

%%PID control gain, using Ziegler-Nichols method%%

%/*PID-ZN gain.

Kp= 12.012;

ki=38.25;

kd=0.9428;

Gc =pid(Kp,ki,kd);

disp('Closed loop of DC motor with ZN-PID ')

ZN= feedback(TF_DC*Gc,H)

Figure

step(ZN), grid on

title('Closed loop response of DC motor with ZN-PID ')

xlabe l('Time','fontsize',12)

ylabel ('Speed','fontsize',14)
```

%% /*Genetic Optimization Algorithm Simulation*/%%

%code of Auto Tuning PID Controller Based on Genetic Optimization Algorithm function using MATLAB %

% creation of function for calculating optimal PID parameters

```
Function M = pid_optimum(x)
s=tf('s');
```

% closed loop DC motor transfer function under study.

```
plant =0.02/(0.008*s^2+0.12*s+0.4204);
kp=x(1)
ki=x(2)
kd=x(3)
controller=kp+ki*1/s+kd*s;
 step (feedback(controller*plant,1))
```

% **Integral Time Absolute Error (ITAE) based objecting function to optimize values of the PID parameters.**

```
dt=0.01;
t=0:dt:1;
e=1-step(feedback(controller*plant,1),t);
M=sum(t'.*abs(e)*dt);
```

%/*PID-GA Gain from the Result of GA Using Matlab Toolbox by Using Calling Function (Function M = Pid_Optimum(x)) In To Matlab ToolBox to observe the step response of DC motor under study*/%.

```
disp ('Closed loop of DC motor with GA-PID ')

s=tf ('s');

Plant =0.02/(0.008*s^2+0.12*s+0.4204);

kp= 271.6619;

ki= 952.9365;

kd= 18.111;

Controller= kp +ki*1/s+kd*s

GA_PID=feedback (controller*plant,1)

Figure

Step (GA_PID), grid on
```

```
title ('Closed loop Response of DC motor with GA-PID')

xlabel('Time','fontsize',12)

ylabel('Speed','fontsize',14)
```

%%/*Representation of Different Tuning Methods to Control Speed of Dc Motor in One Figure*%/.

```
Figure

step(closed_loop), grid on,hold on

step(cloop), grid on,hold on

step(ZN), grid on

step(GA_PID), grid on

title('Step Response of DC motor with PID-Trial & Error,GA-PID','fontsize',14)

xlabel('Time','fontsize',14)

ylabel('Speed','fontsize',14)

legend('DCM_CLP','T&R_PID','ZN _PID','GA _PID','fontsize',14)
```

%/* Comparisons between PID-GA Tuning Method and the PID Controller Using Trial And error To Control Speed of Dc Motor in One Figure*/%.

```
Figure

step(cloop), grid on, hold on

Step (GA_PID), grid on

Legend ('T&R_PID’,'GA _PID','fontsize',14)
```

➢ values of the physical parameters for DC Motor Model under study

- Moment of inertia of the rotor,J=0.02 $\frac{kg.m^2}{s^2}$.
- Damping ratio of the mechanical system B=0.2 Nms
- Electromotive force constant k= kt = Kv = 0.02 $\frac{Nm}{Amp}$.
- Electric resistance R=2 ohm.
- Electric inductance L=0.4H.
- Input (V): Source Voltage.
- Output (**ω**): rotational velocity of the armature.
- The rotor and shaft are assumed to be rigid**[12].**