CHAPTER 1

Introduction

1.1 General

This chapter describes the literature review of PM Brushless Motor, PID controller, and the tuning methods for PID controller parameters. This chapter also describes the thesis objectives and outlines.

1.2 Literature Review

The literature review is divided into three sections. The first section shows an overview of PM Brushless Motor advantages and applications in recent researches, the second section describes the PID controller and the third section describes the commonly and recently used tuning rules for PID controllers.

1.2.1 PM Brushless Motors (PMBL)

In many applications, a wide range in speed and torque control for the electric motor is desired. The DC machine fulfils these requirements, but this machine needs periodic maintenance. The AC machines, like induction motors, and brushless permanent magnet motors do not have brushes, and their rotors are robust because commutator and/or rings do not exist. That means very low maintenance. This also increases the power-to-weight ratio and the efficiency. For induction motors, flux control has been developed, which offers a high dynamic performance for some applications like that in electric traction. However, this control type is complex and sophisticated. The development of brushless permanent magnet machines has permitted an important simplification in the hardware for much application control. Today, two kinds of brushless permanent magnet machines are the most popular: i) the Permanent Magnet Synchronous Motor (PMSM), which is fed with sinusoidal currents, and ii) the Brushless DC (BLDC) Motor, which is fed with quasi- square-wave currents. These two designs eliminate the rotor copper losses, giving very high peak efficiency compared with a traditional induction motor. Besides, the power-to-weight ratio of PMSM and BLDC is higher than equivalent squirrel cage induction machines. The aforementioned characteristics and a high reliability control make this type of machine a powerful. [1].

With the advent of high-energy permanent-magnet (PM) materials PM Brushless motor drive is becoming more and more attractive for industrial applications and electric vehicles. As compared with induction motor drives, they possess some distinct advantages such as higher power density, higher efficiency, and better controllability. PM brushless motor drives have sine wave and square wave versions. The sine wave PM brushless motor drive, also called the PM synchronous motor drive, is fed by sine wave current and uses continuous rotor position feedback signal to control the commutation. On the other hand, the square wave PM brushless motor drive, also called the PM brushless dc motor drive is fed by square wave current and uses discrete rotor position feedback signal to control the commutation. Since the interaction between the square wave current and square wave magnetic field in the motor can produce a large torque product than that produced by sine wave current and sine wave magnetic field, the PM brushless dc motor drive possesses higher power density than the PM synchronous motor drive. In most applications, particularly in electric vehicles, wide range speed control of motor drives is necessary [2].

Recent developments in permanent magnet (PM) materials, power electronics, fast digital signal processors (DSPs) and modern control technologies have significantly influenced the widespread use of permanent magnet brushless (PMBL) motor drives in order to meet the competitive worldwide market demands of manufactured goods, devices, products and processors. Large, medium, small as well as micro PMBL motors are extensively sought for applications in all sorts of motion control apparatus and systems. The marvelous increase in the popularity of the PMBL motor drives among engineers bears testimony to its industrial usefulness in terms of superior performance and relative Size [3].

1.2.2 PID Controller

During the past decades, great advances have been made in motion control techniques. Numerous control methods, such as adaptive control, fuzzy control and neural network control, have been introduced to the motion control field. Despite these advanced control techniques, the PID and its variations (P, PI, and PD) still are widely applied in the motion control. Behind the curtain of this prevalence lie some reasons, being immune to the incorrect model order assumption, good robustness, and easy implementation [4].

PID controller has been used widely for processes and motion control system in industry. Now more than 90% of control system is still using PID controller. The most critical step in application of PID controller is parameters tuning. Today self-tuning PID controller provides much convenience in engineering. The parameter settings of a PID controller for optimal control of a plant depend on the plant behavior [4].

The usefulness of PID controllers lies in their general applicability to most control systems. In the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not provide optimal control [5].

The typical PID control law in its standard form is:

$$u(t) = K_p[e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i}\int_0^t e(\tau)d\tau] \quad (1.1)$$

Where, $e(t) = y_{sp}(t) - y(t)$ is the system error (difference between the reference input $y_{sp}(t)$ and the system output $y(t)$, $u(t)$ is the control variable, K_p is the proportional gain, T_d is the derivative time constant, T_i is the integral time constant.

1.2.3 PID Controller Parameter Tuning

The PID controller with simple structure and stable characteristic has been usually utilized in the industrial circle. The performance of PID controller is related to the setting of parameters, i.e., the proportion (P) plus the integration (I) plus the derivation (D). However, the three parameters are mutual with each other such as the improvement in transient response with the PD controller in some cases yields deterioration in the improvement of the steady-state error with the PI controller, and vice versa. Thus, the over-design of controller system with respect to steady-state errors or transient response will cause more cost or produce other design problems. Traditionally, the optimization of PID controller was manually adjusted by the trial-and error approach so that the procedure consumes much time and manpower [5].

The methods of tuning PID parameters have the traditional PID and intelligent PID adjustments. The traditional PID proposed Ziegler and Nichols in 1942 suggests the adjustment formula based on the observation of the sensitivity, amplitude, and natural frequency of systems. However, the tuning algorithm is comparatively complicated and difficult to make the response optimized with the worse vibration and overshoot. Later on, a lot of researches were devoted to the intelligent PID controllers such as the fuzzy algorithm, but the fuzzy rules still need to be optimized. Thus, the biological optimization algorithms such as the evolutionary computing, swarm intelligence, and so on, were introduced to improve the optimization of PID parameters [6].

1.2.3.1 Traditional Methods

In this section, some of traditional PID tuning methods are discussed.

1.2.3.1.1 Ziegler-Nichols (ZN) Tuning Rule

Ziegler and Nichols proposed a methodology capable to find PID controller gain starting from identified characteristics of the system to be controlled. Basically there are three methodologies proposed by Ziegler and Nichols: the step response method, the frequency response method, and the modified method of Ziegler and Nichols [7].

i. Ziegler- Nichols step response method

This method is based on the step response to determine the two parameters *a* and *L* which describe the process model of an integrator with dead time as given by equation (1.2).

$$G_p(S) = \frac{a}{SL} e^{-SL} \tag{1.2}$$

where, $G_p(S)$ is the transfer function of the process model in frequency domain, S is the Laplace transform operator, L is the dead time, and $b = a / L$ is the slope of the step response c(t) as described graphically in figure (1.1), where, K_P is the static gain of the process, and T is the time constant. The PID tuning parameters obtained by the ZN step response method are shown in table (1.1) [9].

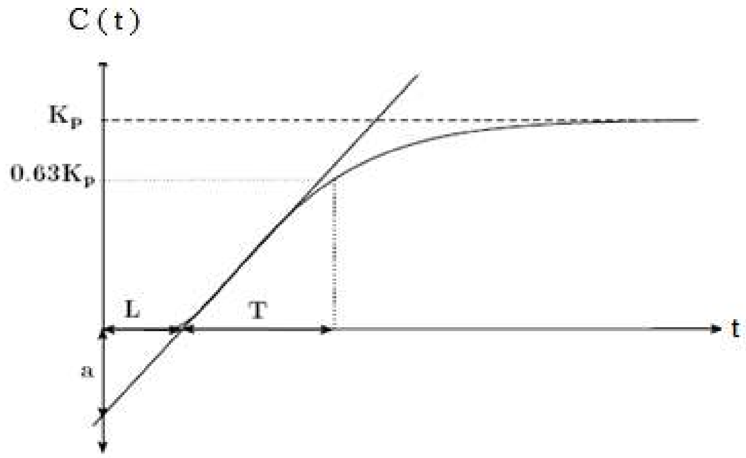


Figure (1.1): Determination of model parameters from a step response

Table (1.1): ZN PID step response tuning parameters

Controller	K_p	T_i	T_d
P	$1/a$		
PI	$0.9/a$	$3L$	
PID	$1.2/a$	$2L$	$L/2$

ii. Ziegler- Nichols frequency response method

This method is based on the knowledge of the intersection point between the Nyquist curve of the process and the negative real axis, known as the ultimate point as shown in figure (1.2), where, $a(\omega)$ is the amplitude function, $\varphi(\omega)$ is phase function, and $G(i\omega)$ is the process. This point can be determined using the frequency response of the closed-loop process under pure proportional controller. Here, the gain is increased until the closed-loop system becomes critically stable (the response of the system is a sustained oscillation). At this point the ultimate gain, K_u is recorded together with the corresponding period of oscillation, T_u, known as the ultimate period. Based on these values Ziegler and Nichols calculated the PID parameters as shown in table (1.2) [7].

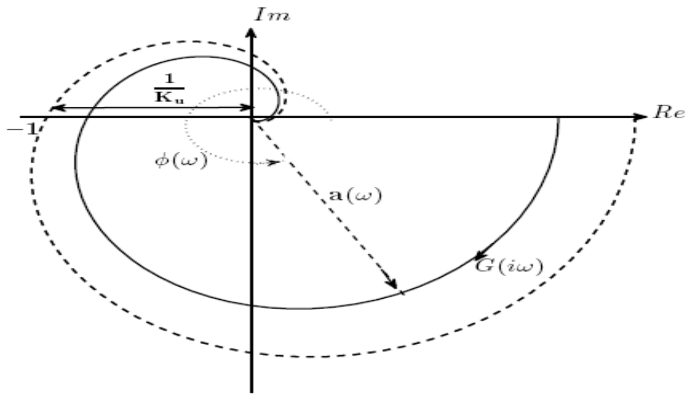


Figure (1.2): Nyquist curve for a process $G(i\omega)$

Table (1.2): ZN PID frequency response tuning parameters

Controller	K_p	T_i	T_d
P	$0.5K_u$		
PI	$0.4K_u$	$0.8T_u$	
PID	$0.6K_u$	$0.5T_u$	$0.12T_u$

iii. Modified Ziegler- Nichols frequency response method

The Ziegler-Nichols tuning rules are simple and intuitive and can be applied to a wide range of processes with little effort. The tuning rules were developed to give good load disturbance rejection, but this also gives a closed-loop system that is poorly damped and has a poor stability margins. The modified Ziegler-Nichols frequency response method can be interpreted as a method where the identified point on the Nyquist curve is moved to a new position. With a PID controller a given point on the Nyquist curve can be moved to an arbitrary position in the complex plane. With this knowledge, the identified point can be moved to give the desired amplitude and phase margins. To describe the method, a process model $G(i\omega_0) = r_a e^{i(\pi+\varphi_a)}$ (point A in the Nyquist curve) and a new position B ($r_b e^{i(\pi+\varphi_b)}$) are assumed as shown in figure (1.3) [7].

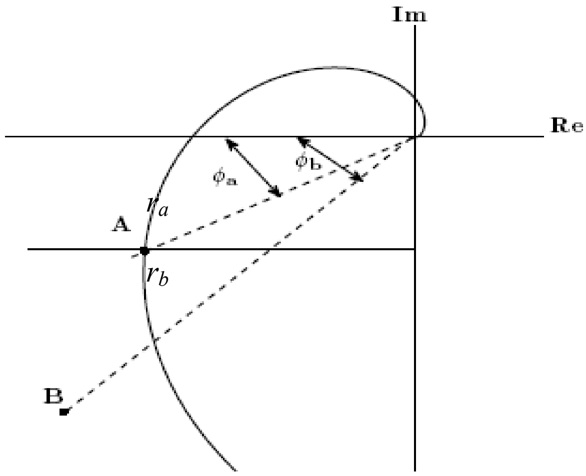


Figure (1.3): Nyquist curve of the process and the identified point A

With the Z-N frequency response method, to determine K_u and T_u, together with the modified Z-N method, the design rules are given by:

$$K_p = \frac{r_b}{r_a}\cos(\varphi_b - \varphi_a) \tag{1.3}$$

$$T_d = \frac{T_u}{4\pi}\left(\tan(\varphi_b - \varphi_a) + \sqrt{4\delta + \tan^2(\varphi_b - \varphi_a)}\right) \tag{1.4}$$

$$T_i = \frac{T_u}{4\pi\delta}\left(\tan(\varphi_b - \varphi_a) + \sqrt{4\delta + \tan^2(\varphi_b - \varphi_a)}\right) \tag{1.5}$$

Where, $\delta = \frac{T_d}{T_i}$, $\varphi_a = 0$ and $r_a = 1/K_u$ for ZN frequency response method or ideal relay feedback method, but have another values for relay with hysteresis method.

Although the experiment proposed by Ziegler and Nichols in the frequency response method is simple in the characterization of the system and tuning of PID, it is of difficult automation, once the oscillation amplitude must be maintained under control, since the operation of systems close to the unstable area is dangerous. Besides this limitation, the accurate determination of the critical gain is an arduous work in practical conditions. Relay feedback are often applied for parameter identification to overcome the above problem [7].

1.2.3.1.2 Relay Feedback Tuning Method

Astrom and Haggland [8] proposed a relay feedback test to determine the ultimate gain K_u and ultimate period T_u by replacing the proportional controller with a relay during the tuning procedures as shown in figure (1.4).

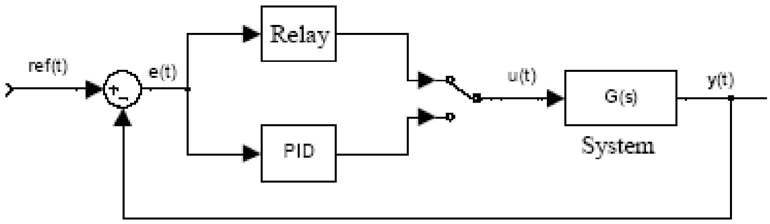


Figure (1.4) Scheme of the relay feedback test

The relay gives a square wave input signal to the process which will start to oscillate with opposite phase, meaning that the frequency of the oscillation is the ultimate frequency. The system and the relay outputs are shown in figure (1.5) [8].

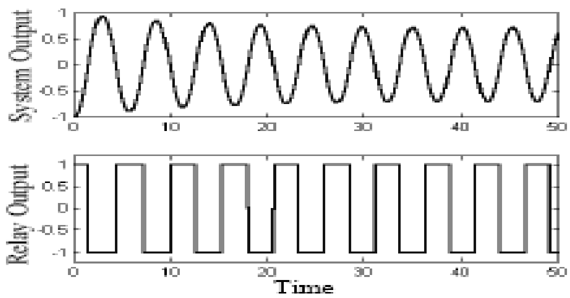


Figure (1.5) System output signal and output of relay feedback

The relay in a feedback system can be described by a gain $N(a)$, which depends on the amplitude a of the input signal. If the relay output amplitude is d , a Fourier series expansion of the relay output gives a first harmonic with amplitude $4d/\pi$ [8].

The describing function $N(a)$ for a relay is then given by equation (1.6), the relay can also assumed to have a hysteresis ε to avoid random relay switching on noisy signal and the describing function in this case is given by equation (1.7). The selection of the value of the hysteresis depends on the noise level [7, 8].

$$N(a) = \frac{4d}{\pi a} \tag{1.6}$$

$$N(a) = \frac{4d}{\pi a^2}(\sqrt{a^2 - \varepsilon^2} + i\varepsilon) \tag{1.7}$$

From the describing function analysis the limit cycle can be predicted to a point where the Nyquist curve intersects $-1/N(a)$, which is the negative inverse of the describing function for an ideal relay or relay with hysteresis as follows:

$$-\frac{1}{N(a)} = -\frac{\pi}{4d}\sqrt{a^2 - \varepsilon^2} + i\frac{\pi\varepsilon}{4d}$$

(1.8)

This also implies:

$$\left|G(i\omega_u) = \frac{1}{K_u} = \frac{\pi a}{4d}\right| \quad \text{and} \quad \varphi_a = \arcsin(\frac{\varepsilon}{a}) \tag{1.9}$$

The describing function $-1/N(a)$ can be described as a straight line parallel to the negative real axis in complex plane as shown in figure (1.6) [8].

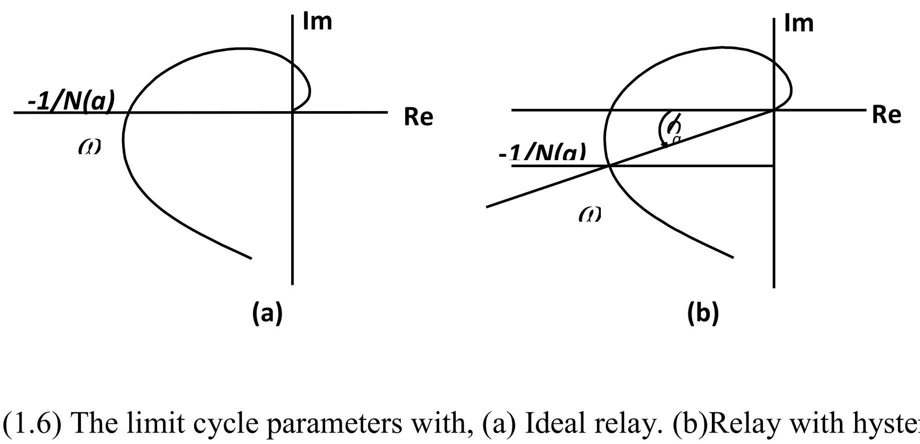


Figure (1.6) The limit cycle parameters with, (a) Ideal relay. (b)Relay with hysteresis.

By determined values of ultimate gain and ultimate period, the rules of ZN frequency response method or the modified one can be used to find the PID controller gains [8].

1.2.3.1.3 Kappa-Tau Tuning Rule

The kappa-tau tuning method is a PID design method developed by Astrom and Hagguland [9]. The idea of this method is to characterize the process by three parameters as given by equation (1.10) and previously shown in fig.(1) instead of two parameters used in ZN tuning method.

$$G_p(S) = \frac{K_P}{1+TS} e^{-SL} \tag{1.10}$$

Where, K_P is the static gain of the process, T is time constant, L is the dead time.

As in the ZN method, it comes in two versions. One is based on the step response, in which the PID controller parameters are given as a function of a new parameter tau (τ) as defined in equation (1.11). The second tuning rule is based on the frequency response, in which the PID controller parameters are given as a function of new parameter kappa (k) as defined in equation (1.12) [9].

$$\tau = \frac{L}{T+L} \tag{1.11}$$

$$k = \frac{1}{K_P K_u} \tag{1.12}$$

1.2.3.1.4 Pole Placement Tuning Rule

Analytical pole placement methods [10], are mostly used when the system under consideration is of low order. A common approach is to adopt a second-order model and then specify a desired damping ratio and natural frequency for the system. These specifications can then be fulfilled by locating the two system poles at positions that give the required closed loop performance. For example of a second order system, the system transfer function $G_p(s)$ is given by equation (1.13) and the PID controller by equation (1.14).

$$G_P(S) = \frac{K_P}{(1+ST_1)(1+ST_2)} \tag{1.13}$$

$$G_C(S) = \frac{K_p(1+ST_i+S^2T_iT_d)}{ST_i} \tag{1.14}$$

From equation (1.10) to equation (1.14) the system characteristic equation is:

$$S^3 + S^2\left(\frac{1}{T_1} + \frac{1}{T_2} + \frac{K_pK_PT_d}{T_1T_2}\right) + S\left(\frac{1}{T_1T_2} + \frac{K_pK_P}{T_1T_2}\right) + \frac{K_pK_P}{T_iT_1T_2} = 0 \tag{1.15}$$

The characteristic equation (1.15) can be compared with the, general, third order characteristics equation given by (1.16) and evaluate the three parameters of the PID controller.

$$(S+\alpha\omega)(S^2 + 2\zeta\omega S + \omega^2) = 0 \tag{1.16}$$

Where, ω is the natural frequency and ζ is the damping ratio of the system.

1.2.3.1.5 Dominant pole design

Dominant pole design [10] is another, simplified, pole placement technique employed when it is required to obtain a PID controller for high-order systems. This method is based on the positioning of the system dominant poles in the complex plane. In many cases, the dominant system dynamics can be approximated by the simple pole-zero configurations shown in figure (1.7). The pair of poles P_1, P_2 is known as the dominant pole. Poles and zeros which have real parts much more negative than those of the dominant poles have little influence on the overall system response.

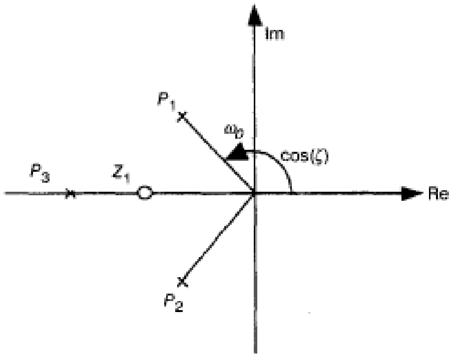


Figure (1.7) Dominant poles of closed loop system

1.2.3.1.6 Design Based On Gain and Phase Margin Specifications

In [10] Astrom presented a method to choose the coefficient of PID controller based on gain and phase margins, where, the phase margin is related to the damping of the system. The gain margin, A_m, is defined as the inverse of the process gain at the phase-crossover frequency ω_p and can be obtained by the solution of equations (1.17) and (1.18).

$$\arg[G_C(j\omega_p)G_P(j\omega_p)] = -\pi \tag{1.17}$$

$$A_m = \frac{1}{\left|G_C(j\omega_p)G_P(j\omega_p)\right|} \tag{1.18}$$

Where, $G_P(S)$ is the process transfer function, and $G_c(S)$ is the controller transfer function.

The phase margin is defined by ϕ_m and is a measure of how much the phase can be decreased before it reaches 180^0 and can be determined by the solution of the equations (1.19), (1.20).

$$\left|G_C(j\omega_g)G_P(j\omega_g)\right| = 1 \tag{1.19}$$

$$\phi_m = \arg\left[G_C(j\omega_g)G_P(j\omega_g)\right] + \pi \tag{1.20}$$

Where, ω_g is the gain crossover frequency.

It is apparent that, depending on the plant model, the solution of the above set of equations can be extremely difficult to carry out analytically, so numerical methods are usually employed.

1.2.3.2 Intelligent Methods

Conventional control depends on the mathematical model of the plant being controlled and when this model is uncertain, intelligent controllers promise more performance. The applications of artificial intelligence (AI) in industry have been increasing rapidly, among the AI technologies; fuzzy logic is the most popular choice in many high performance industrial control systems [11,12].

1.2.3.2.1 Incremental Fuzzy Expert PID Control (IFE)

In [12], the incremental fuzzy expert PID control method is proposed to scale the values of the three controller parameters, initially determined by the Ziegler-Nichols formula, during the transient response depending on the system error e and its rate $e^{\cdot}$. In other words, the current values of the proportional, integral and derivative gains are increased or decreased by means of a fuzzy inference system, according to the following relations:

$$K_p = K_p + CV\{e(t), e^{\cdot}(t)\}K_1 \tag{1. 21}$$

$$K_i = K_i + CV\{e(t), e^{\cdot}(t)\}K_2 \tag{1. 22}$$

$$K_d = K_d + CV\{e(t), e^{\cdot}(t)\}K_3 \tag{1. 23}$$

Where the basic tuning is the Ziegler-Nichols one, CV {e (t), e$^{\cdot}$ (t)} is the output of the fuzzy inference system, K_1 , K_2, and K_3 are constants.

The fuzzy inference system reflects the typical action of a human controller. For example, the integral action has to be increased at the beginning of the transient response to decrease the rise time and then it has to be decreased when the system error is negative, to reduce the overshoot. Finally, K_1, K_2, and K_3 are constant parameters that determine the range of variation of each term. The whole fuzzy system involves fourteen quantization levels for both error and change of error. It has to be stressed that the tuning of the three parameters K_1, K_2, and K_3, and of the two scaling factors that multiply the two inputs e and $e^{\cdot}$ is left to the user, and it might be a difficult task, as it is not clear how these parameters influence the performances of the overall controller, for a generic system.

1.2.3.2.2 Fuzzy PID Speed Controller

In [12], traditional PID controller of speed is replaced with a fuzzy PID controller. The speed error e(t) and speed error slope de(t)/dt are used to determine proportional (K_p) and integral (K_i) parameters. The rules for the parameters are determined by using trial and error methods. The fuzzy characteristic gives the controller higher flexibility in adjusting control parameters for better transient responses of speed during the operation while the traditional PID controller has fixed values of control parameters. The fuzzy characteristic gives the controller higher flexibility in adjusting control parameters for better transient responses of speed during the operation, while the traditional PID controller has fixed values of control parameters.

i. Fuzzy self-adapting PID controller design

In [13], the control algorithm of traditional PID controller can be described as

$$u(k) = k_p e(k) + k_i \sum e(k) + k_d e_c(k) \tag{1. 24}$$

Where k_p is the proportional factor; k_i is the integral factor; k_d is the differential factor. $e(k)$ is the speed error; $e_c(k)$ is the change rate of speed error. The control performance can become better through adjusting the k_p, k_i and k_d according to the changing control parameters condition. The design algorithm of PID controller in this paper is to adjust the k_p, k_i, and k_d parameters on line through fuzzy inference based on the current e and e_c to make the controlled object attain the good dynamic and static performances .The block diagram Of fuzzy self-adjusting PID controller is shown in figure (1.8) [13].

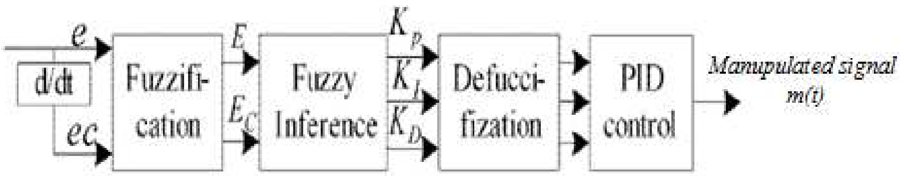


Figure (0.8): Block diagram of fuzzy self-adjusting PID controller

ii. Fuzzy set-point weighting (FSW)

The approach proposed by Visioli [13] consists of fuzzy the set-point weight, leaving fixed the other three parameters (again determined with the Ziegler-Nichols method to preserve good load disturbance attenuation).

In this way, the control law can be written as:

$$u(t) = K_p[b(t)y_{sp}(t) - y(t)] + K_d \frac{de(t)}{dt} + K_t \int_0^t e(\tau)d\tau \tag{1.25}$$

$$b(t) = w + f(t) \tag{1.26}$$

Where *w* is a positive constant parameter less than or equal to 1, and *f (t)* is the output of the fuzzy inference system, which consists of five triangular membership functions for each of the two inputs *e(t)* and $e^{\bullet}(t)$ and nine triangular membership functions for the output. Figure (1.9) shows the overall control scheme. It is worth stressing that in this method the role of the fuzzy mechanism parameters is somewhat intuitive, and it is very similar to the one in the typical fuzzy PD-like controller, for which tuning procedure have been established. Hence, the task of the user is simplified by a simple empirical procedure for the manual tuning of the fuzzy module.

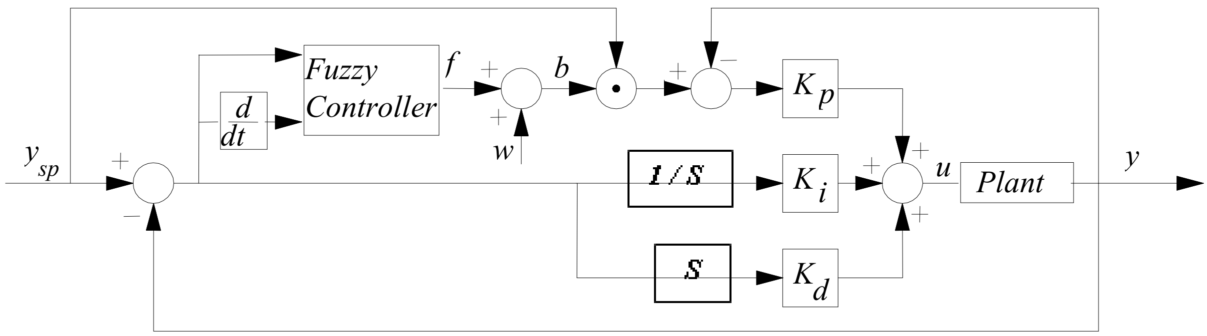


Figure (1.9): Control scheme of fuzzy set-point weighting (FSW)

1.2.3.3 Biological Optimization Algorithms

The biological optimization algorithms such as the evolutionary computing, and swarm intelligence were introduced to improve the optimization of PID parameters [13].

1.2.3.3.1 Genetic Algorithm (GA-PID)

The genetic algorithms (GA) techniques [14] are a rapidly expanding area in control systems design. A genetic tuning algorithm usually starts with no knowledge of the correct solution and depends on the responses from its environment to give an acceptable result. It has been shown that genetic algorithms are capable of locating optimal regions in complex domains avoiding the difficulties, or even erroneous results in some cases, associated with the gradient descent methods and with high-order systems. To obtain the PID tuning parameters one usually has to minimize a performance estimation function which can be; integrated absolute error *(IAE)*, or the integral of squared-error *(ISE)*, or the integral of time-weighted-squared-error *(ITSE)* because it can be evaluated analytically in the frequency domain [14].

In [16], a genetic algorithm based on binary coding is used. Each parameter of the PID controller (K_p, K_i, K_d) is represented by 16 bits and a single individual is generated by concatenating the coded parameter strings. It was demonstrated that genetic algorithms provide a much simpler approach to the tuning of such controllers than rather complicated non-genetic optimization algorithms previously proposed by Plak and Mayne, [17], and Gesing and Davison [15].

Though the GA methods have been employed successfully to solve complex optimization problems, recent research has identified some deficiencies in GA performance. This degradation in efficiency is apparent in applications with highly objective functions [i.e., where the parameters being optimized are highly correlated (the crossover and mutation operations cannot ensure better fitness of offspring because chromosomes in the population have similar structures and their average fitness is high toward the end of the evolutionary process)], Moreover, the premature convergence of GA degrades its performance and reduces its search capability [15].

1.2.3.3.2 Particle Swarm Optimization (PSO-PID)

Particle Swarm Optimization first introduced by Kennedy and Eberhart in 1995, is one of the modern heuristic algorithms. The method was proved to be of high computation efficiency, easy implementation and stable convergence by many works. Some forerunners began to apply this method to engineering problems and obtained the better results than the methods used before. PSO has an excellent performance in some nonlinear and constrained problems. It offers a potential approach in optimal controller design for nonlinear systems.

In [16], a particle swarm optimization (PSO) method for determining the optimal proportional-integral-derivative (PID) controller parameters is presented for speed control of a linear brushless DC motor. The proposed approach has superior features, including easy implementation, stable convergence characteristic and good computational efficiency. Figure (1.10) shows the block diagram of optimal PID control for the BLDC motor.

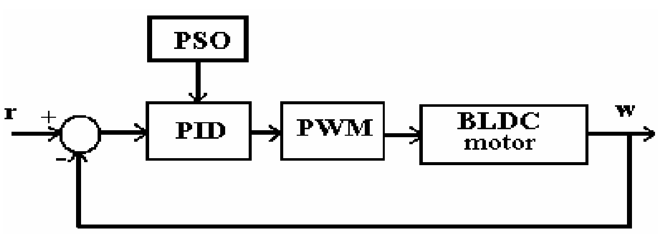


Figure (1.10): Optimal PID control

In PID controller design methods, the most common performance criteria are integrated absolute error (IAE), the integrated of time weight square error (ITSE) and integrated of squared error (ISE) that can be evaluated analytically in the frequency domain [20]. These three integral performance criteria in the frequency domain have their own advantages and disadvantages. For example, the disadvantage of the IAE and ISE criteria is that its minimization can result in a response with relatively small overshoot but a long settling time because the ISE performance criterion weighs all errors equally independent of time. Although the ITSE performance criterion can overcome the disadvantage of the ISE criterion, the derivation processes of the analytical formula are complex and time-consuming [15,16].

In this paper [16] a time domain criterion is used for evaluating the PID controller. A set of good control parameters P, I and D can yield a good step response that will result in performance criteria minimization in the time domain. These performance criteria in the time domain include the overshoot, rise time, settling time, and steady-state error. For example, the performance criterion is defined as follows

$$\min_{k.stablizing}\ w(k) = (1-e^{-\beta}).(M_p + E_{ss}) + e^{-\beta}(t_s - t_r) \quad (1.27)$$

Where *K* is any of *P*, *I*, or *D*, and β is the weighting factor. The performance criterion *W(K)* can satisfy the designer requirement using the weighting factor β value. β can be set to be larger than 0.7 to reduce the overshoot and steady states error, also can be set smaller than 0.7 to reduce the rise time and settling time [20]. The optimum selection of β depends on the designer's requirement and the characteristics of the plant under control. In BLDC motor speed control system the lower β would lead to more optimum responses. In this work β is set to 0.5 to optimize the step response of speed control system.

In [16], a novel PID controller design method is proposed for PMSM Servo system using Particle Swarm Optimization (PSO). The detailed procedures for optimal PID controller design are summarized in terms of the principle of Particle Swarm Optimization. In order to overall optimize the performance of the system step response, a new evaluation strategy (Fuzzy Hamming Distance) is introduced for evaluating the performance.

In [17], the particle swarm optimization algorithm is used to design an online self-tune framework of PID controller. The system is simulated in Matlab based on particle swarm optimization algorithm and several problems are concerned. The conclusions include that different fitness function can lead to different time response, and application system should initialize range of each particle as small as possible. Moreover, the conclusions also include that a modest generations for the online system with linearly inertia weight consume less times evolutionary generation, not a larger one. These conclusions can contribute mostly to application system concerning about calculation cost.

Different PSO optimization parameters are required for solving different problems in practical applications such as the number of individuals, weight factors, and the limit of velocity change; hence, how to select suitable parameters for the target problem is one of the recommendations to be in the future work.

1.2.3.3.3 Bacterial Foraging Optimization Algorithm (BFO-PID)

Recently, search and optimal foraging of bacteria have been used for solving optimization problems. To perform social foraging, an animal needs communication capabilities and over a period of time it gains advantages that can exploit the sensing capabilities of the group. This helps the group to predate on a larger prey, or alternatively, individuals could obtain better protection from predators while in a group [18].

In [19] the classical BFOA was compared with the adaptive BFOAs and a few other well-known evolutionary and swarm based algorithms over a test bed of 10 well-known numerical benchmarks. The performance metrics used for comparison were the solution quality, the speed of convergence, and the frequency of hitting the optimum. The adaptive BFOAs variants were shown to provide better results than their classical counterpart for all of the tested problems.

1.3 Thesis Objective

The main objective of this thesis is to design, implement, and test different optimization algorithms such as Particle Swarm Optimization (PSO) and Bacterial Foraging Optimization (BF), for tuning the PID controller parameters for the speed control of PM Brushless DC Motor to achieve a better performance.

1.4 Thesis Outlines

This thesis is organized into six chapters and three appendices

Chapter 1: Introduction

This chapter presents a general description of PM Brushless Motor and PID controller followed by a literature review of traditional, intelligent and biological tuning methods for PID controller.

Chapter 2: Fundamentals of Control System

This chapter presents an introduction to basic control system, and then followed by a description of each component, and it presents a survey of DC motors, drive configuration, the end of this chapter PID speed control of BLDC motor is illustrated.

Chapter 3: Mathematical Model of Brushless DC Motor

In this chapter the brushless DC motor is presented by equations. The block diagram equivalent to dynamics mathematical equations is depicted and simulated at MATLAB.

Chapter 4: Optimization Techniques

This chapter presents a survey of the classification of different optimization problems along with a brief discussion of their selection factors. The general mathematical formulation of multi objective optimization is described and followed by solution techniques such as, Particle Swarm Optimization and Bacterial Foraging Optimization. It demonstrates also the simulation results for simulated model with step input under the influence of the proposed controllers with the fitness function.

Chapter 5: Simulation Results

In this chapter the simulation results are established. The simulation results with (PSO-PID) and (BF-PID), controllers are investigated and discussed. Speed tracking and motor loading with the same controllers are also verified.

Chapter 6: Conclusion and Future Work

The main achievements and most important conclusion of the whole thesis are presented in this chapter along with the recommendations for future work on its subject.

Appendix A: Implemented Algorithms

Appendix B: (BLDC) Motor Parameters

CHAPTER 2

Fundamentals of Control System

A typical motion control system consists of a host computer to generate the command signal, motion controller, motor, motor drive and a position sensor. A typical motion control system shown in figure (2.1) is described in the following sections [20].

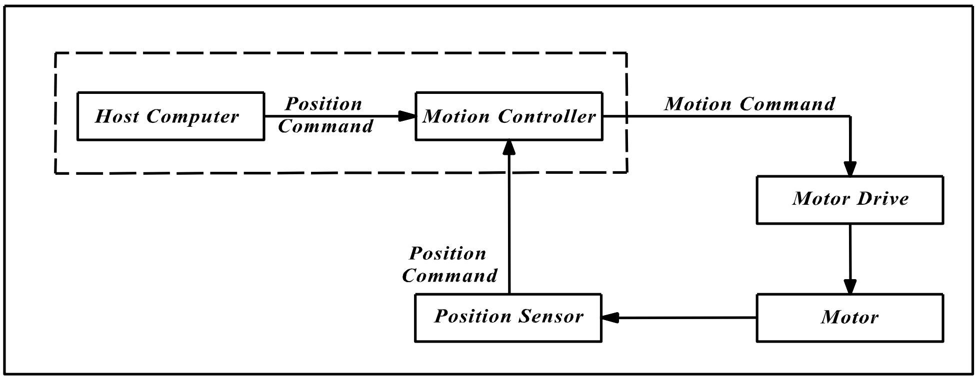


Figure (2.1): A typical motion control system

2.1 Motion controller

The input to the motion controller is the error signal between desired position and the actual position from the position sensor, the motion controller generates the control signal to the motor drive in order to drive the system for the desired position [20].

2.2 Transducers and Sensors

Transducers are devices which convert a physical quantity into another quantity of difference nature, which is often electric quantity, as electric signals can easily be measured or controlled. The major types of transducers used in motion control systems are position, velocity, and presence sensors [10, 20].

2.2.1 Position Transducers

The aim of a position transducers are to provide an electric signal proportional to the angular or linear displacement of the mechanical apparatus with respect to a given reference position. There are two kinds of position measurements [21].

- **Absolute position**: The sensor can measure the position of an object relative to a reference position.
- **Incremental position**: The sensor cannot measure the position of an object relative to a reference, but can keep track of the change in position of an object.

The position sensors may be also classified according to the type of the output signal into digital position sensors such as encoders and analog position sensors such as potentiometer and resolvers.

a) Resolver

Resolver is analog absolute position sensors which operate based on the transformer principle. The key to their operating principle is that the change in the position of the rotor element changes the electromagnetic coupling (magnetic flux linkage) between the two windings, primary and secondary windings. As a result, the induced voltage between the two windings changes in relation to the position. Hence, we have a well-defined relationship between the induced voltage and the position [20, 21].

In resolvers, the primary winding is located on the rotor, and the secondary winding on the stator. Either the rotor winding or the stator windings can be excited externally by a known voltage, and the induced voltage on the other winding is measured which is related to the position [21].

A simplified functional diagram of resolver and its corresponding signals over one mechanical revolution is depicted in figure (2.2) [21].

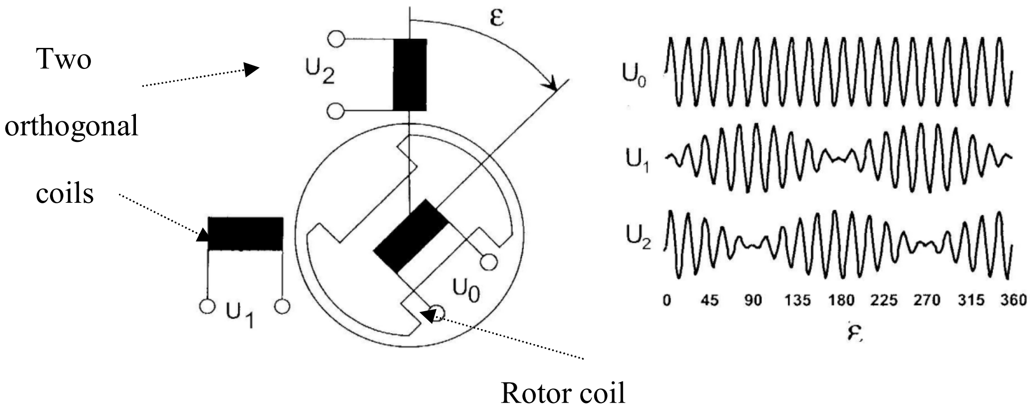


Figure (2.2): Resolver and corresponding signals

The resolver basically consists of a rotor coil, with *N* turns winding and two orthogonal stator coils with usually *N* or *N/2* turns winding. An alternating voltage (the reference signal), is coupled into the rotor winding and providing primary excitation. The reference signal is typically a fixed frequency signal in the range of 2k Hz to 10 k Hz [21].

The two orthogonal stator coils are wound, so that when the rotor shaft turns, the amplitude of the output signals is modulated with the sine and cosine of the shaft angle ε, hence the shape of the resolver output signal u_1 and u_2 is equal to the sine and the cosine of the mechanical angle, respectively.

If a reference voltage given by eq. (2.1) excites the rotor of a resolver, then the stator terminal voltages are given by equations (2.2) and (2.3) [21].

$$U_0 = V\sin(\omega t) \tag{2.1}$$

$$U_1 = V\sin(\omega t)\sin(\varepsilon) \tag{2.2}$$

$$U_2 = V\sin(\omega t)\cos(\varepsilon) \tag{2.3}$$

Then a resolver-to-digital converters transform the secondary voltages into a digital representation of the actual angle (ε).

b) Optical encoder

An encoder is a device that converts linear or rotary displacement into digital or pulse signals. The most popular type of encoders is the optical encoder, which consists of a rotating disk, a light source, and a photo detector. The disk, which is mounted on the rotating shaft, has patterns of opaque and transparent sectors coded into the disk as shown in figure (2.3). As the disk rotates, these patterns interrupt the light emitted onto the photo detector, generating a digital or pulse signal output [22].

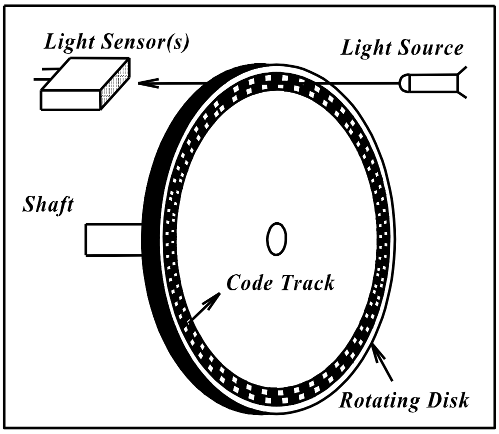


Figure (2.3): Optical encoder

The encoders are divided into two classes:

- Absolute encoder.
- Incremental encoder.

i) Absolute encoders

An absolute encoder generates a unique word pattern for every position of the shaft. The tracks of the absolute disk, generally four or six, commonly are coded to generate binary code, binary-coded decimal, or gray code outputs [22]. Figure (2.4) shows the components of an absolute encoder. Figure (2.5) shows the disc of a 16-position grey-coded disk absolute encoder and its typical output signal. Absolute encoders are most commonly used in applications where the device will be inactive for long periods of time, there is risk of power down, or the starting position is unknown. The resolution of the absolute encoder is determined by the number of photo detectors, each photo detector output represents a bit on the digitally coded position information. If the absolute encoder has eight photo detectors (8-bit), the smallest position change that can be detected is $360^\circ/(2^8) = 360^\circ/256^\circ = 1.4^\circ$ [21, 22].

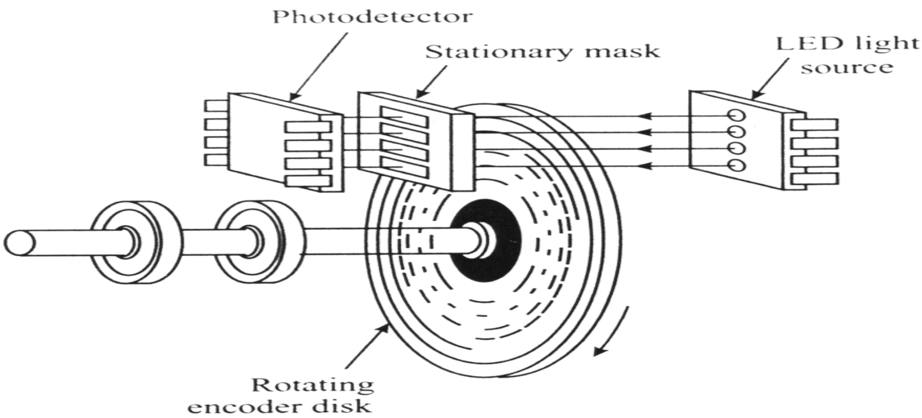


Figure (2.4): Components and operating principle of a rotary absolute encoder

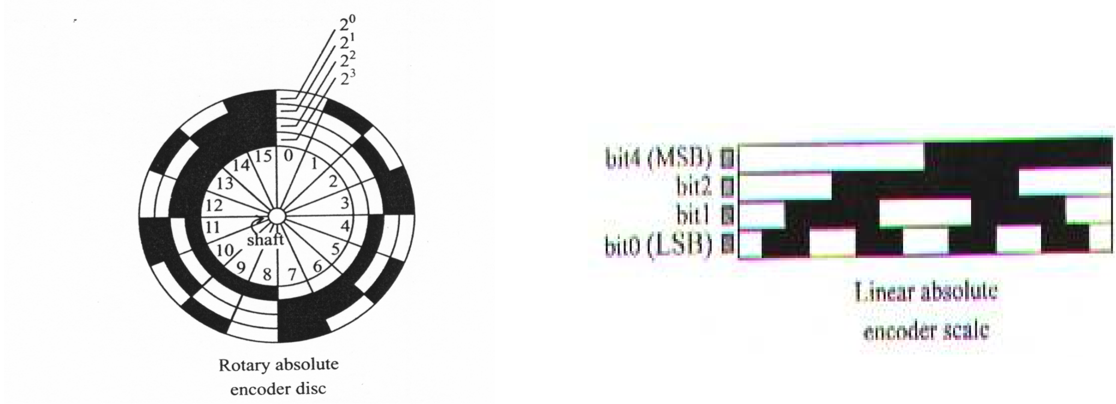


Figure (2.5): Disk of absolute encoder and its output signal

ii) Incremental encoder

An incremental encoder generates a pulse for each incremental step. Although the incremental encoder does not output absolute position, it does provide more resolution at a lower price. An incremental encoder with single code track, referred as a tachometer encoder, generates a pulse signal whose frequency indicates the velocity of the displacement However, the output of a single-channel encoder does not indicate direction. To determine direction, a two-channel, or quadrate, encoder uses two detectors and two code tracks with sectors positioned 90^0 out of phase to give two output channels (A and B) for indicating both direction and position [23].

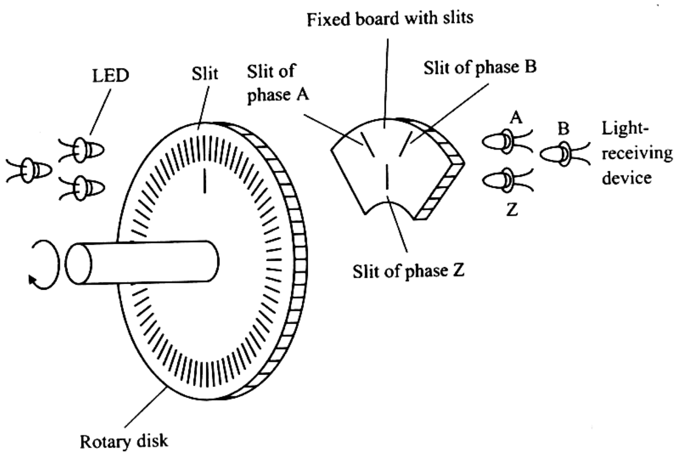


Figure (2.6): Components and operating principle of a rotary incremental encoder

In addition, some quadrature encoders include a third output channel, called a zero or reference signal, which supplies a single pulse per revolution. This pulse can be used for precise determination of a reference position, the component of incremental encoder is shown in figure (2.6) [22, 23].

The direction of rotation can be found by observing the phase shift between signal A and signal B. If signal A leads signal B (signal B is low at every rising edge of signal A) as shown in figure (2.7-a), the disk is rotating in clock wise direction. If signal B leads signal A (signal B is high at every rising edge of signal A) as shown in figure (2.7-b), then the disk is rotating in counter-clock wise direction. Therefore, by monitoring the number of pulses and the relative phase of signals A and B, the position and direction of rotation can be tracked [23].

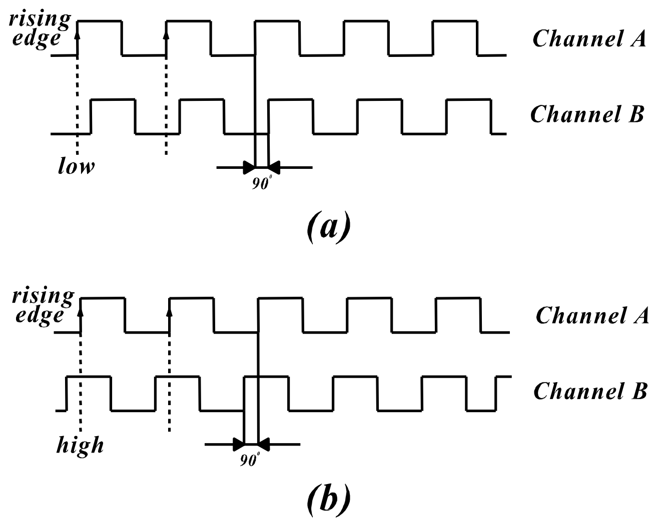


Figure (2.7): Quadrature encoder output channels A and B for (a) CW direction (b) CCW direction

Encoders may have a complementary channels ($\bar{A}, \bar{B}, \bar{C}$) for each output channel (A, B, C), which are used for protection against noise as shown in figure (2.8) [23].

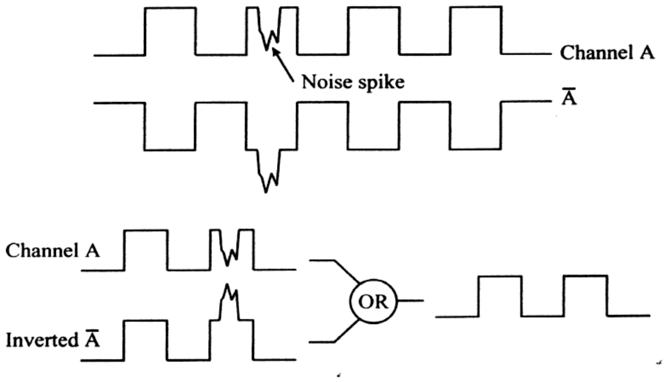


Figure (2.8): Usage of complementary channel Ā with channel A to cancel noise

When the quadrature encoder is used with the counter of data acquisition card which has three inputs (source, up-down and gate) and two outputs (out and interrupt), there are two choices. First, for simple application, we can connect the encoder directly to the counter, without any extra logic or signal conditioning as shown in figure (2.9). Although simple to implement, this configuration has the disadvantage of not being able to discern between stationery vibration of the encoder and real rotation. Second, we can interface the encoder to the counter using a quadrature clock converter IC. This method provides higher measurement resolution [23].

A quadrature clock converter IC has another function which is providing TTL and CMOS compatible outputs if the input is of different signal level to prevent damage of the DAQ.

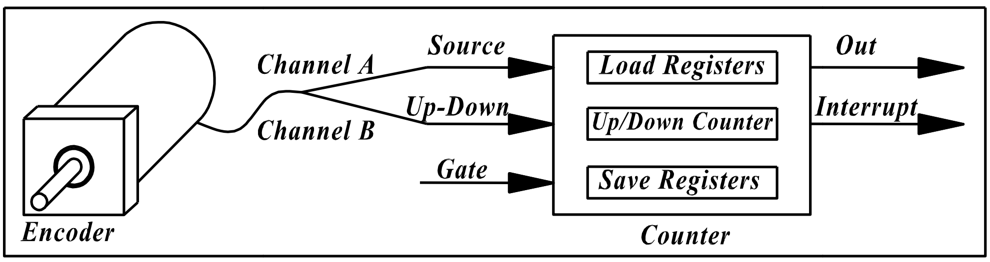


Figure (2.9): Connection of encoder to counter of DAQ card

c) Hall Effect Voltage Sensor

Hall-effect sensors can also provide a voltage signal, and like the inductive-type, can be mounted on the crankcase wall, or inside the housing of the distributor. The sensor has a permanent magnet, and a Hall switch, as part of its assembly, and an air-gap between the magnet's North and South poles. The switch is on 1 pole of the magnet, and an interrupter ring, with a number of square-shaped blades or segments, rotates through the gap formed by the poles.

When it's used in a distributor, this interrupter ring has the same number of blades as engine cylinders, and a corresponding number of windows, or gaps, between the blades. The magnetic field is strongest when the gap is aligned with the poles. This allows the switch to earth a low-current signal voltage that is applied to it.

When the interrupter ring rotates so that a blade is in line with the poles, the magnetic field is shielded, and the signal voltage is not earthed. With continuous rotation, the blades repeatedly move in and out of the air gap, and the signal voltage will appear to turn on and off repeatedly. The control unit uses this on-off signal to detect engine RPM, and to control ignition timing. If a sequential injection mode is used, the position of the camshaft also must be signalled to the control unit. This is done by making 1 blade of the interrupter ring shorter than the others. It is called a signature blade. It passes through the sensor, and alters the signal, so that injection commences at the correct time in the cycle. Since the distributor rotates at camshaft speed, the sensor in the distributor provides camshaft position readily.

When the sensors are on the crankshaft, a separate sensor is needed for camshaft position. It identifies when to commence injection for the number 1 cylinder. Injection for the other cylinders then occurs in the same sequence as the firing order [23].

2.2.2 Velocity Transducers

Angular velocity transducers are devices that give an output proportional to angular velocity. These sensors find wide application in motor-speed control systems. They are also used in position systems to improve their performance. Some of the most popular velocity transducers will be discussed in this section [24].

i) Velocity from position sensors

Velocity is the rate of change of position and can be expressed mathematically by equation (2.5)

$$\text{Velocity} = \frac{\Delta\theta}{\Delta t} = \frac{\theta_2 - \theta_1}{t_2 - t_1} \tag{2.5}$$

Where, $\Delta\theta$ is the change in angle, Δt is the change in time, θ_2, θ_1 is the position samples, and t_2, t_1 are times when samples are taken.

Because the only components of velocity are position and time, extracting velocity information from two sequential position data samples should be possible (if you know the time between them). The math could be done with hard-wired circuits or software. If the system already has a position sensor, such as a potentiometer, using this approach eliminates the need for an additional (velocity) sensor. Velocity data can be derived from an optical rotary encoder in two ways. The first would be the method just described for the potentiometer; the second method involves determining the time it takes for each slot in the disk to pass. The slower the velocity, the longer it takes for each slot to go by. The idea is to count the cycles of a known high-speed clock for the duration of one slot period [23, 24].

ii) Back induced electromotive force (e.m.f)

This method is used extensively with the built-in speed controller of DC drives. As the speed of a DC motor is proportional to back e.m.f. induced in the winding of the motor, the angular speed of the motor can be calculated as given by equation (2.6) [24].

$$\omega = \frac{V - I_a R_a}{K_e} \tag{2.6}$$

Where, ω is the angular velocity of the motor, V Is the input voltage to the motor, I_a is the armature current, and K_e is the back e.m.f. constant.

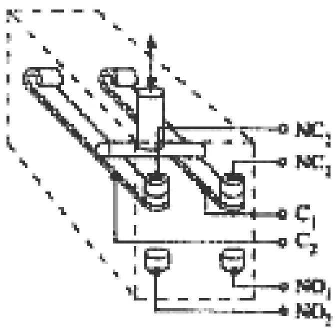
2.2.3 Presence Sensors

A special class of the position-related sensors are the sensors which sense the presence of an object with a sensing range and provides one of two discrete outputs: ON or OFF. Such sensors collectively called presence sensors or ON/OFF sensors.

i) Contact presence sensors (limit switches)

The most commonly-used sensor in industry is still the simple, inexpensive limit switch, shown in Figure (2.10). These switches are intended to be used as presence sensors. When an object pushes against them, lever action forces internal connections to be changed. Most switches can be wired as either normally open (NO) or normally closed (NC) or both. If a force is required to hold them at the other state, then they are momentary contact switches. Switches that hold their most recent state after the force is removed are called detent switches [24].

(a)



(b)

Figure (2.10): Limit switch (a) One group of contacts (b) Two group of contacts

ii) Non-contact presence sensor (proximity sensor)

Proximity sensor is one of the most common light-based presence sensors used in industry. Proximity sensors have two types, inductive and capacitive. All of these sensors are actually transducers, but they include control circuitry that allows them to be used as switches. The circuitry changes an internal switch when the transducer output reaches a certain value.

iii) Inductive proximity sensor

The inductive proximity sensor is the most widely used non-contact sensor due to its small size, robustness, and low cost. This type of sensor can detect only the presence of electrically conductive materials. Figure (2.11) demonstrates its operating principle [24].

An oscillator is used to generate AC in an internal coil, which in turn causes an alternating magnetic field. If no conductive materials are near the face of the sensor, the only impedance to the internal AC is due to the inductance of the coil. If, however, a conductive material enters the changing magnetic field, eddy currents are generated in that conductive material, and there is a resultant increase in the impedance to the AC in the proximity sensor. A current sensor, also built in the proximity sensor, detects when there is a drop in the internal AC current due to increased impedance. The current sensor controls a switch providing the output.

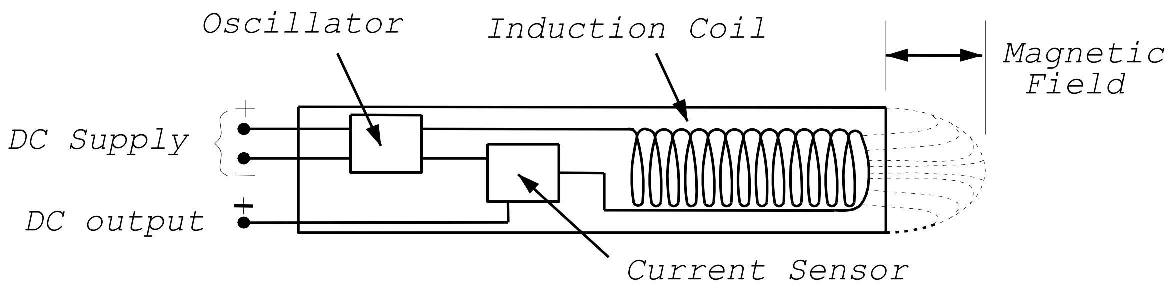


Figure (2.11): Inductive proximity sensor

iv) Capacitive proximity sensor

Inside the sensor as shown in figure (2.12) is a circuit that uses the supplied DC power to generate AC, to measure the current in the internal AC circuit, and to switch the output circuit when the amount of AC current changes. Unlike the inductive sensor, however, the AC does not drive a coil, but instead tries to charge a capacitor. Remember that capacitors can hold a charge because, when one plate is charged positively, negative charges are attracted into the other plate, thus allowing even more positive charges to be introduced into the first plate. Unless both plates are present and close to each other, it is very difficult to cause either plate to take on very much charge, where, only one of the required two capacitor plates is actually built into the capacitive sensor. The AC can move current into and out of this plate only if there is another plate nearby that can hold the opposite charge. The target being sensed acts as the other plate. If this object is near enough to the face of the capacitive sensor to be affected by the charge in the sensor's internal capacitor plate, it will respond by becoming oppositely charged near the sensor, and the sensor will then be able to move significant current into and out of its internal plate. The built in current sensor, senses the change in the value of the current and then controls a switch providing the output [24]

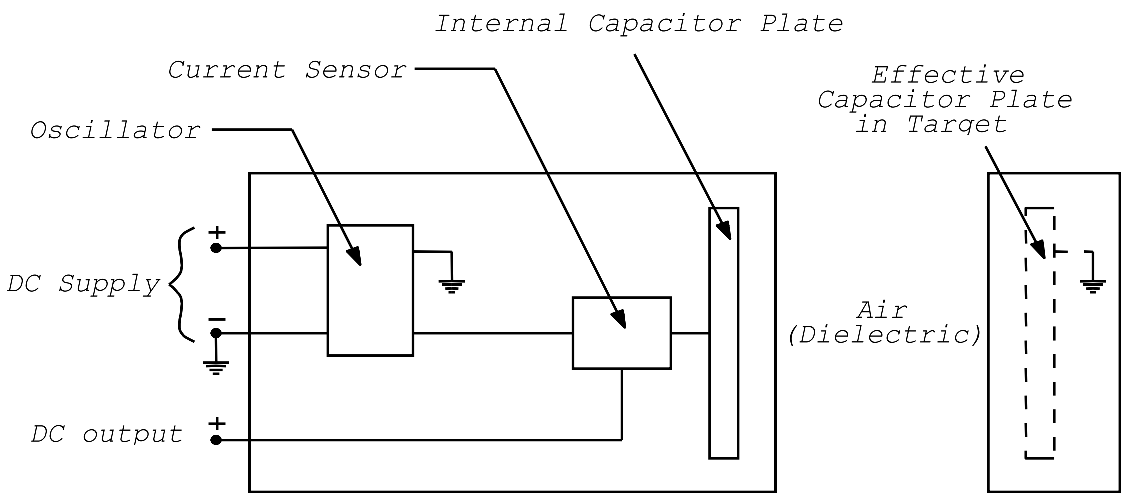


Figure (2.12): Capacitive proximity sensor

2.3 Actuators

An indispensable component of the control system is the actuator. The actuator is the first system component to actually move, converting electrical energy into mechanical motion. The most common type of actuators used in control systems are electric motors such as DC motors, AC motors, and stepper motors [23, 24].

The DC motors have been used extensively in motion control systems because DC motors have speed-control capability, which means that speed, torque, and even direction can be changed at any time to meet new condition.

The AC motors is commonly used in many applications, where, speed control is not necessary such as fans, pumps, mixers, machine tools, hydraulic power supplies, and household appliances due to their, constant speed-mechanical power, high efficiency, low cost, and low maintenance. For complete speed control of AC motors, both the input voltage and frequency must be adjusted, which requires a special electronic speed control circuitry such as the volts-per-hertz (*V/HZ*) [25].

A stepper motor is a unique type of motors that rotates in fixed steps of a certain number of degrees. The step size can range from 0.9 to 90°. Stepper motors are particularly useful in control applications because the controller can know the exact position of the motor shaft without the need of position sensors. This is done by simply counting the number of steps taken from a known reference position. In fact, most stepper motor systems operate open-loop, that is, the controller sends the motor a determined number of steps commands and assumes the motor goes to the right place [25].

From the above explanation of different types of motors, we can conclude that the convenient type of motors used in closed loop position control (servo systems) is DC motor which will be discussed in the following section.

2.4 DC Motors

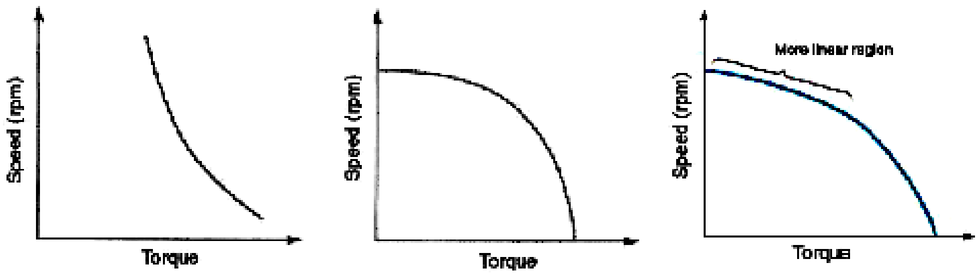
There are mainly two types of dc motors used in industry. The first one is the conventional dc motor where the flux is produced by the current through the field coil of the stationary pole structure. The second type is the Brushless DC (BLDC) motor where the permanent magnet provides the necessary air gap flux instead of the wire-wound field poles [25].

This kind of motor not only has the advantages of DC motor such as better velocity capability and no mechanical commutator but also has the advantage of AC motor such as simple structure, higher reliability and free maintenance. In addition, the BLDC motor has the following advantages: smaller volume, high force, and simple system structure. So it is widely applied in areas which need high performance drive [25].

The disadvantages of using a BLDC motor are the high cost and the more complex controller caused by the nonlinear characteristics. Another problem in a BLDC motor control is that the controller employed is usually simple in realization but it is difficult to obtain a sufficient high performance in the tracking application. It is, however, known that the tracking controller problem using a state variable feedback can be simply solved by the augmentation of the output error as a new state, even though this method is more complex than a PI controller. It is more efficient to obtain the control gain using the optimal control theory that has no problem compared to the classical controller [25].

2.4.1 Wound Field DC Motors

There are three types of wound field DC motors. The first, with speed/torque characteristics as shown in Figure (2.13), is the series wound DC motor. In this type of motor, the field windings and armature windings are connected in series. Current passing through the field windings must also pass through the commutator to the armature windings. Reducing the DC current to the field also reduces armature current. Since reducing armature current reduces speed, while reducing field strength increases speed, control of this type of motor is difficult. In fact, if the motor is allowed to run without a frictional load, it can accelerate all by itself until it self-destructs. It is also an interesting fact that the direction of rotation of a series wound DC motor cannot be changed by changing the polarity of the DC supply [26].



(a) (b) (c)

Figure (2.13) Speed/Torque characteristics for wound field DC motors for constant voltage (a) Series wound (b) Separately excited (c) Compound wound

Another type of wound field DC motor is the separately excited wound motor. In this motor, the field winding and the armature winding are brought out of the motor casing separately, and the user connects them to separate supplies so that the field strength and the armature current can be controlled independently. This type of motors can, depending on the type of control selected, have its speed reduced or increased from the nominal values. The direction of rotation of this type of motor can be changed by changing the polarity of either, but not both supplies [26].

If the field winding of the separately excited DC motors is connected in parallel with the armature winding, it is called shunt wound DC motor. The shunt wound DC motor has a torque-speed characteristic similar to that of separately excited one.

The compound motor has both shunt and series field windings, although they are not necessarily the same size. There are two configurations of the compound motor, the short shunt and the long shunt, typically, the series and shunt coils are wound in the same direction so that the field fluxes add. The main purpose of the series winding is to give the motor a higher starting torque. Once the motor is running, the counter E.M.F reduces the strength of the series field, leaving the shunt winding to be the primary source of field flux and thus providing some speed regulation. Also, the combination of both fields acting together tends to straighten out (linearize) a portion of the torque-speed curve .The motor discussed so far, where the fields add, is called a cumulative compound motor. Less common is the differential compound motor, where the field coils are wound in opposite directions. The differential compound motor has very low starting torque but excellent speed regulation. However, because it can be unstable at higher loads, it is rarely used. The compound motor direction of rotation is reversed by reversing the polarity of the armature windings.

2.4.2 Brushless DC (BLDC) Motor

(BLDC) or permanent-magnet (PM) motors use permanent magnets to provide the magnetic flux for the field. In conventional PM motors, the armature is similar to those in the wound-field motors discussed earlier. The fact that the field flux of a PM motor remains constant regardless of the speed. This is very desirable for control applications because it simplifies the control equations [27].

Brushless DC motors are referred to by many aliases: Brushless Permanent Magnet, and Permanent Magnet Synchronous Motors etc. The confusion arises because a brushless dc motor does not directly operate from a dc voltage source. However, the basic principle of operation is similar to a dc motor.

A brushless dc motor has a rotor with permanent magnets and a stator with windings. It is essentially a dc motor turned inside out. The brushes and commutator have been eliminated and the windings are connected to the control electronics. The control electronics replace the function of the commutator and energize the proper winding. The windings are energized in a pattern which rotates around the stator. The energized stator winding leads the rotor magnet, and switches just as the rotor aligns with the stator.

There are no sparks, which is one advantage of the brushless DC motor. The brushes of a dc motor have several limitations; brush life, brush residue, maximum speed, and electrical noise. BLDC motors are potentially cleaner, faster, more efficient, less noisy and more reliable. However, BLDC motors require electronic control. [26].

2.4.2.1 Construction and Operating Principle

BLDC motors are basically DC motors. In a DC motor the stator is a permanent magnet. The rotor has the windings, which are excited with a current. The current in the rotor is reversed to create a rotating or moving magnetic field by means of a split commutator and brushes. On the other hand, in a BLDC motor the windings are on the stator and the rotor is a permanent magnet. The Brushless DC motor does not operate directly from a DC voltage source. The Brushless DC motor has a rotor with permanent magnets, a stator with windings and commutation that is performed electronically. Typically three Hall sensors are used to detect the rotor position and commutation is performed based on Hall sensor inputs. [26, 27]

The motor is driven by rectangular or trapezoidal voltage strokes coupled with the given rotor position. The voltage strokes must be properly applied between the phases, so that the angle between the stator flux and the rotor flux is kept close to 90° to get the maximum generated torque. The position sensor required for the commutation can be very simple, since only six pulses per revolution (in a three-phase machine) are required. Typically, the position feedback is comprised using three Hall Effect sensors aligned with the back-EMF of the motor. In order to rotate the BLDC motor, the stator windings ought to be energized in an order. It is essential to understand the rotor position in order to know which winding will be energized following the energizing sequence. Rotor Position can be got by either a shaft encoder or, more often, by Hall Effect sensors that detect the rotor magnet position [28].

a) Stator:

The stator of a BLDC motor consists of stacked steel laminations with windings placed in the slots that are axially cut along the inner periphery as shown in figure (2.4). There are two types of stator windings variants: trapezoidal and sinusoidal motors. This differentiation is made on the basis of the interconnection of coils in the stator windings to give the different types of back Electromotive Force (EMF). As their names indicate, the trapezoidal motor gives a back EMF in trapezoidal fashion and the sinusoidal motor back EMF is sinusoidal, as shown in figure (2.14) and figure (2.15). In addition to the back EMF, the phase current also has trapezoidal and sinusoidal variations in their respective types of motor. This makes the torque output by a sinusoidal motor smoother than that of a trapezoidal motor. However, this comes with an extra cost, as the sinusoidal motors take extra winding interconnections because of the coils distribution on the stator periphery, thereby increasing the copper intake by the stator windings [28].

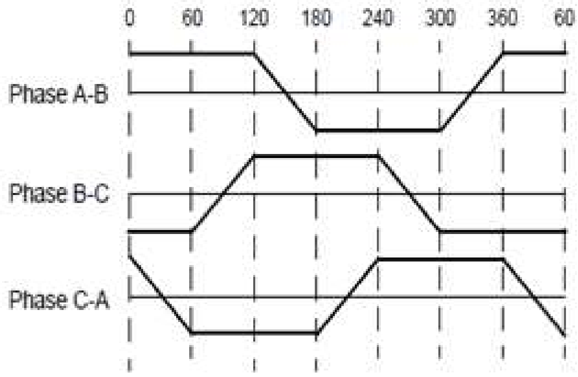


Figure (2.14): Trapezoidal back EMF

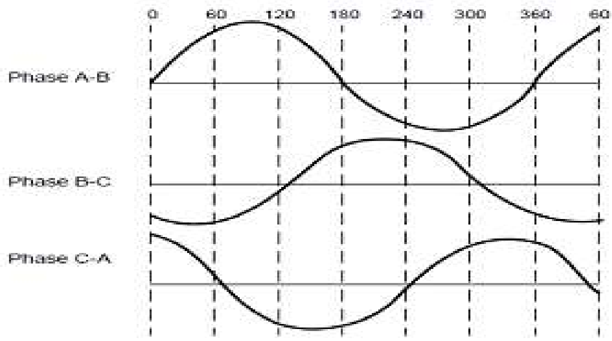


Figure (2.15): Sinusoidal back EMF

Figure (2.16): Stator of BLDC motor

b) Rotor

The rotor is made of permanent magnet and can vary from two to eight pole pairs with alternate North (N) and South (S) poles. Based on the required magnetic field density in the rotor, the proper magnetic material is chosen to make the rotor. Ferrite magnets are traditionally used to make permanent magnets. As the technology advances, rare earth alloy magnets are gaining popularity. The ferrite magnets are less expensive but they have the disadvantage of low flux density for a given volume. In contrast, the alloy material has high magnetic density per volume and enables the rotor to compress further for the same torque. Also, these alloy magnets improve the size-to-weight ratio and give higher torque for the same size motor using ferrite magnets. Neodymium (Nd), Samarium Cobalt (SmCo) and the alloy of Neodymium, Ferrite and Boron (NdFeB) are some examples of rare earth alloy magnets. Continuous research is going on to improve the flux density to compress the rotor further. Figure (2.17) shows cross sections of different arrangements of magnets in a rotor [28]

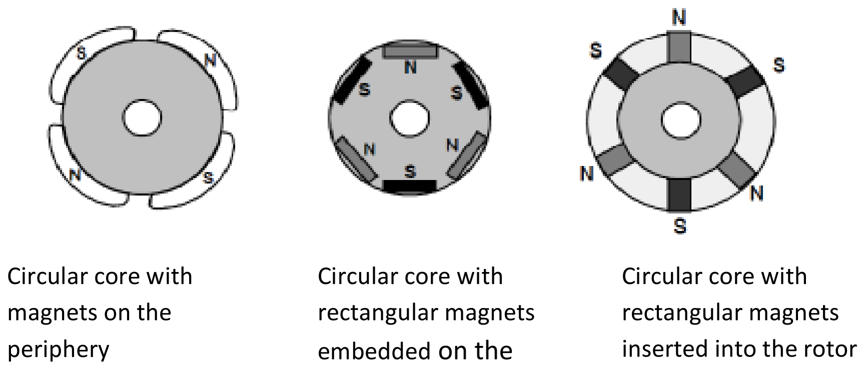


Figure (2.17): Rotor magnet cross section

2.4.2.2 Electronic Commutation

A BLDC motor is driven by voltage strokes coupled with the given rotor position. These voltage strokes must be properly applied to the active phases of the three-phase winding system so that the angle between the stator flux and the rotor flux is kept close to 90° to maximize torque. Therefore, the controller needs some means of determining the rotor orientation/position (relative to the stator coils).

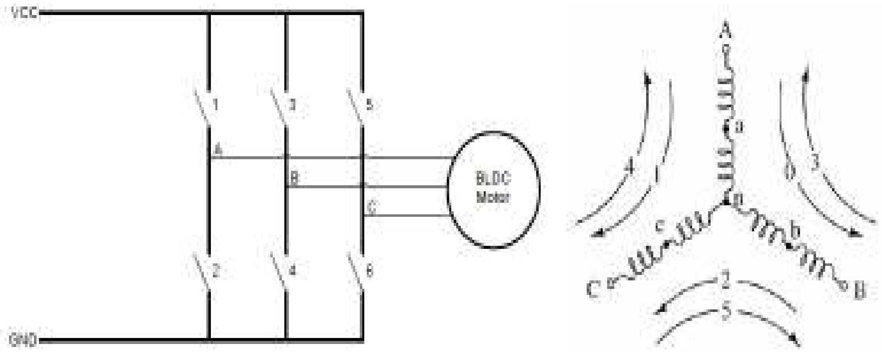


Figure (2.18): Three-phase Bridge and coil current direction

Figure (2.18) illustrate a systematic implementation on how to drive the motor coils for a correct motor rotation. The current direction through the coils determines the orientation of the stator flux. By sequentially driving or pulling the current though the coils the rotor will be either pulled or pushed. A BLDC motor is wound in such a way that the current direction in the stator coils will cause an electrical revolution by applying it in six steps. As also shown in figure (2.18) each phase driver is pushing or pulling current through its phase in two consecutive steps. These steps are shown in table (2.1). This is called trapezoidal commutation. Figure (2.19) shows the relation between the definitions six-step commutation (six Hall sensor edges H_1, H_2 and H_3), (i_a, i_b, i_c) and (e_a, e_b, e_c), figure (2.20) illustrates Three-phase full-bridge power circuit for BLDC motor drive.

Table (2.1): Switching sequence of BLDC motor

Sequence number	Switching interval	Phase current			Switch closed	
		A	B	C		
0	0° - 60°	+	-	OFF	1	4
1	60° - 120°	+	OFF	-	1	6
2	120° - 180°	OFF	+	-	3	6
3	180° - 240°	-	+	OFF	3	2
4	240° - 300°	-	OFF	+	5	2
5	300° - 360°	OFF	-	+	5	4

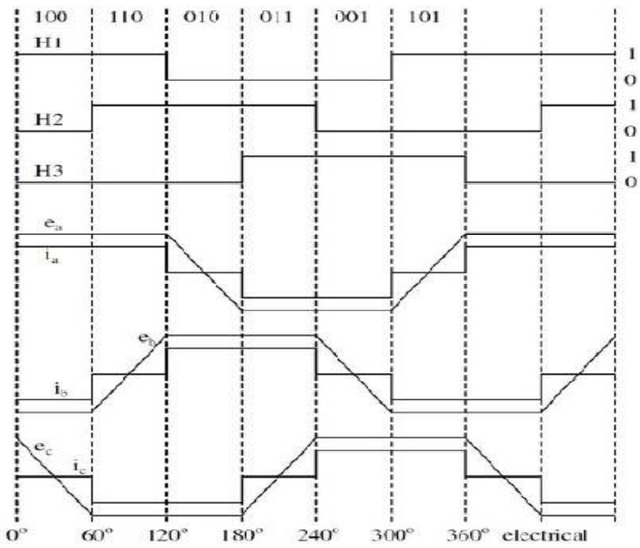


Figure (2.19): Trapezoidal control with Hall sensor feedback

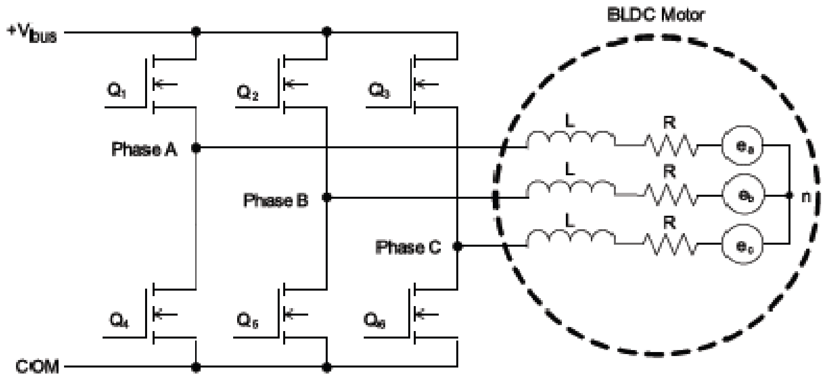


Figure (2.20): Three-phase full-bridge power circuit for BLDC motor drive

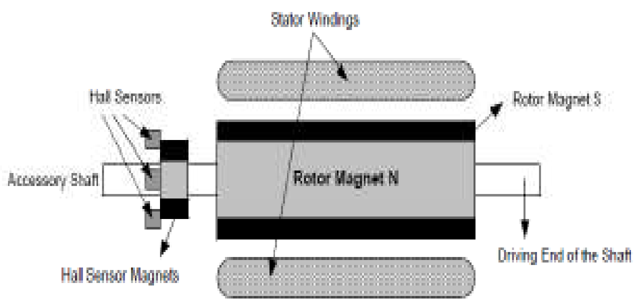


Figure (2.21): BLDC motor transverse section

There are different types of rotor position measuring device like potentiometer, linear variable differential transformer, optical encoder, resolver, and tachometer. The ones most commonly used for motors are encoders and resolver. Depending on the application and performance desired by the motor a position sensor with the required accuracy can be selected [28].

The hall sensors are placed such that they generate an edge at each switching interval. This makes it very easy to determine the current rotor orientation, and to activate each phase in the right sequence.

Most BLDC motors have three Hall sensors embedded into the stator on the non-driving end of the motor. Whenever the rotor magnetic poles pass near the Hall sensors, they give a high or low signal, indicating the N or S pole is passing near the sensors. Based on the combination of these three Hall sensor signals, the exact sequence of commutation can be determined. Figure (2.21) shows a transverse section of a BLDC motor with a rotor that has alternate N and S permanent magnets. Hall sensors are embedded into the stationary part of the motor [28].

2.4.2.3 Comparing BLDC Motor to Other Motor Types

BLDC motors have many advantages and few disadvantages. Table (2.2) summarizes the comparison between a BLDC motor and a brushed DC motor.

Table (2.2): Comparing a BLDC motor to a brushed DC motor

Feature	BLDC Motor	Brushed DC Motor
Commutation	Electronic commutation based on Hall position sensors.	Mechanical commutation.
Maintenance	Less due to absence of brushes.	Periodic maintenance is required.
Life	Longer.	Shorter.
Speed/Torque Characteristics	Flat – Enables operation at all speeds with rated load.	Moderately flat –At higher speeds, brush friction increases, thus

		reducing useful torque.
Efficiency	High – No voltage drop across brushes.	Moderate.
Output Power/ Frame Size	High – Reduced size due to superior thermal characteristics. Because BLDC has the windings on the stator, which is connected to the case, the heat dissipation is better.	Moderate/Low – The heat produced by the armature is dissipated in the air gap, thus increasing the temperature in the air gap and limiting specs on the output power/frame size.
Rotor Inertia	Low, because it has permanent magnets on the rotor. This improves the dynamic response.	Higher rotor inertia which limits the dynamic characteristics.
Speed Range	Higher – No mechanical limitation imposed by brushes/commutator.	Lower – Mechanical limitations by the brushes.
Electric Noise Generation	Low.	Arcs in the brushes will generate noise causing EMI in the equipment nearby.
Cost of Construction	Higher – Since it has permanent magnets, building costs are higher.	Low.
Control	Complex and expensive.	Simple and inexpensive.
Control Requirements	A controller is always required to keep the motor running. The same controller can be used for variable speed control.	No controller is required for fixed speed; a controller is required only if variable speed is desired.

Permanent magnet brushless motors can be divided into two subcategories. The first category uses continuous rotor-position feedback for supplying sinusoidal voltages and currents to the motor. The ideal motional EMF is sinusoidal, so that the interaction with sinusoidal currents produces constant torque with very low torque ripple. This called a Permanent Magnet Synchronous Motor (PMSM) drives, and is also called a PM AC drive, brushless AC drive, PM sinusoidal fed drive, and sinusoidal brushless DC drive.

The second category of PMBL motor drives is known as the brushless DC (BLDC) motor drive and it is also called a trapezoidal brushless DC drive, or rectangular fed drive. It is supplied by three-phase rectangular current blocks of 120° duration, in which the ideal motional EMF is trapezoidal, with the constant part of the waveform timed to coincide with the intervals of constant phase current. These machines need rotor-position information only at the commutation points, e.g., every 60°electrical in three-phase motors. A comparison between these two types is shown in table (2.3) [29].

Table (2.3): Difference between PMSM and BLDC

	PMSM	**BLDC**
Flux density (in space)	Sinusoidal distribution	Square distribution
Back EMF	Sinusoidal wave	Trapezoidal wave
Stator current	Sinusoidal wave	Square wave
Total power	Constant	Constant
Electromagnetic torque	Constant	Constant
Energized phases	3 phases ON at any time	2 phases ON at any time

2.5 Motor Drive Configuration

Adjustable motor speed drive is a device that controls speed, and direction of an AC or DC motor. Some high performance drives are able to run in torque regulation mode (current control mode).

The basic DC drive generally consists of firstly is a drive controller and secondly is a power converter. The schematic diagram of the built-in controllers of the DC drive is shown in figure (2.22) [29].

2.5.1 Drive controller

If the DC drive operates in the speed control mode, the input to the PI speed controller is the reference speed (set speed) and its output is the reference current which is the input to the PI current controller, the output of the current controller is the firing angle to the power converter.

In case of the DC drive operates in torque control mode, only the current controller is used.

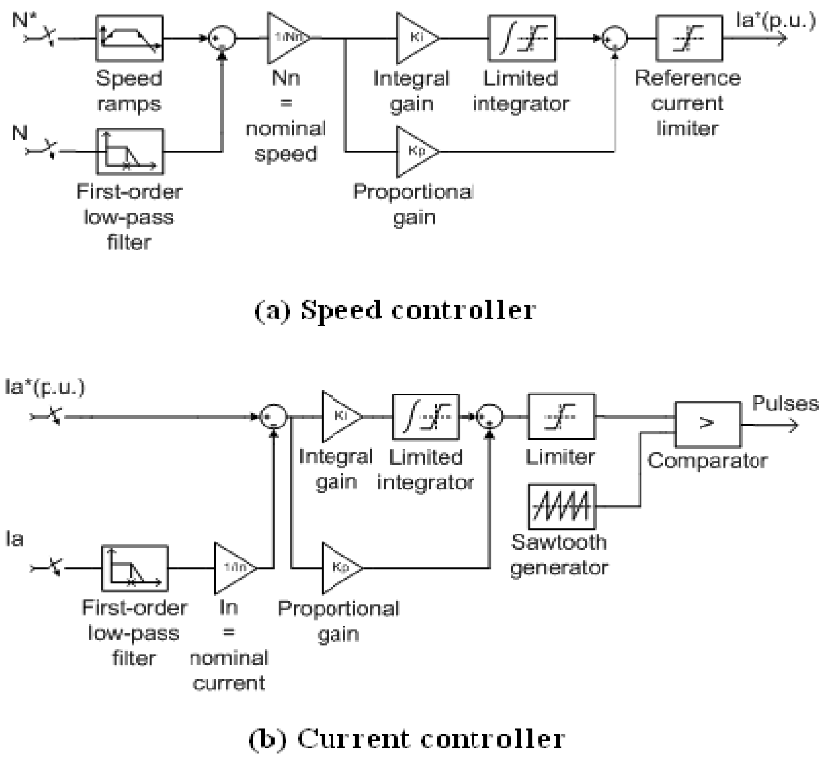


Figure (2.22) : DC drive built-in controllers (a) Speed controller, (b) Current Controller

2.5.2 Power converter

The second part of DC drive is the power converter which can be single phase (provide a variable DC output voltage, from a fixed single phase AC voltage), three phase (provide a variable DC output voltage, from a fixed three phase AC voltage), and chopper (provide a variable DC voltage from a fixed DC voltage), and in the following section, the single phase converters (rectifiers) are described.

2.5.2.1 Single Phase Converter

Rectification is the process of converting an alternating current or voltage into a direct current and voltage. This conversion can be achieved by a variety of circuits based on and using switching devices. The widely used switching devices are diodes, thyristors and power-transistors. The rectifier circuits can be classified into uncontrolled, half-controlled, and fully-controlled. An uncontrolled rectifier uses only diodes and the DC output voltage is fixed in amplitude by amplitude of the AC supply.

The fully-controlled rectifier uses thyristors as the rectifying element and the DC output voltage is a function of amplitude of the AC supply and the point on the wave at which thyristor is triggered (firing angle α). The half-controlled rectifiers contains a mixture of diodes and thristors, allowing more limited control over the DC output voltage than the fully controlled rectifiers [30].

Uncontrolled and half-controlled rectifiers will permit power to flow only from the AC supply to the DC load and, therefore, referred to as unidirectional converters. However with fully-controlled converters it is possible to allow power to be transferred from the DC side of the rectifier back into the AC supply. When this occurs, operation is said to be in inverting mode. The fully controlled converters may be refereed to as bidirectional converters and can be classified into three types as described in the following sections [26].

a) Single-phase half-wave controlled rectifier (one-quadrant converter)

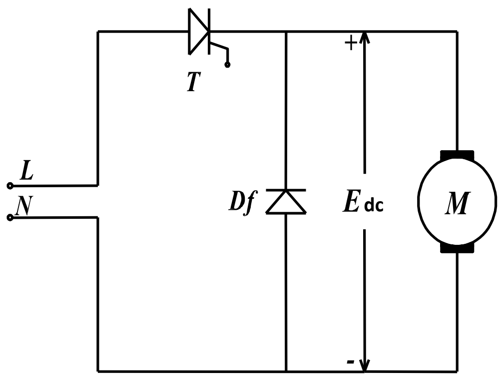
A single phase have-wave converters feeds a DC motor is shown in figure (2.23-a). The armature current is normally discontinuous unless a very large inductor is connected in the armature circuit. A freewheeling diode is always required for a DC motor load. The average armature voltage for a single phase half-wave converters is given by equation (2.7) [26].

$$E_{dc} = \frac{E_m}{2\pi}(1+\cos\alpha) \qquad \text{For } \ 0 \le \alpha \le \pi \tag{2.7}$$

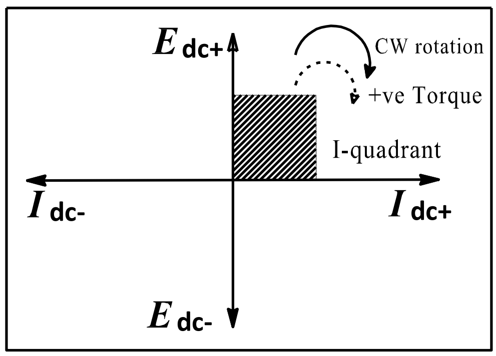
Where, E_m is the maximum voltage of the input AC supply, α is the firing angle.

Since this converter can provide only one polarity of voltage and current at DC terminal as shown in figure (2.23-b), so it is called one quadrant converter.

To stop the motor, the firing angle is adjusted so that converter voltage is equal to 0V. The motor will coast to a stop at a rate that depends on the mechanical load and the inertia of the revolving parts.



(a)



(b)

Figure (2.23): One quadrant converter (a) Circuit diagram (b) Voltage-current diagram

b) Single-phase full-wave controlled rectifier (two-quadrant converter)

A single-phase full-wave converter feeds DC motor is shown in figure (2.24-a) and the armature voltage given by equation (2.8) [30].

$$`\,E_{dc} = \frac{2E_m}{\pi}\cos\alpha \qquad \text{For } \; 0 \le \alpha \le \pi \tag{2.8}$$

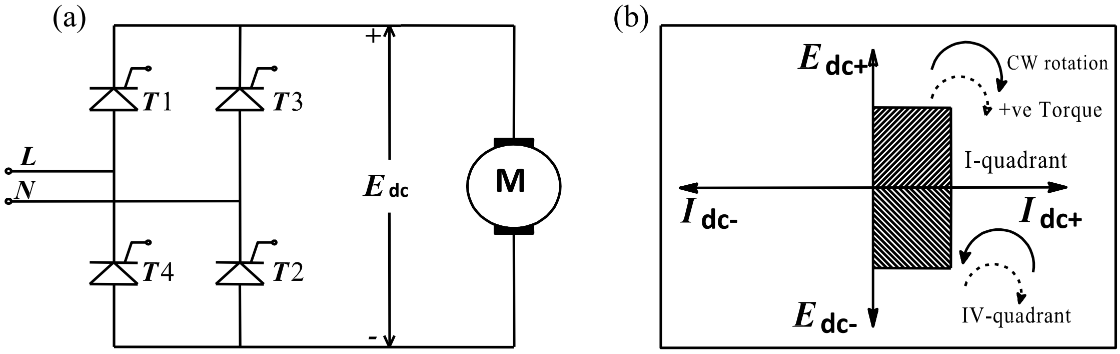


Figure (2.24): Two quadrant converter (a) Circuit diagram (b) Voltage-current diagram

It is clear from equation (2.8) that the converter can give positive output voltage (+E_{dc}) for $0 \le \alpha < \frac{\pi}{2}$ and negative output voltage (–E_{dc}) for $\frac{\pi}{2} \le \alpha \le \pi$. This allows operation in first and fourth quadrant as shown in figure (2.24-b). In the situation where a motor simply coast to a lower speed, the circuit has to be modified so that the motor acts temporally as a generator feeding power back into the supply which called regenerative braking. To achieve regenerative braking, we make the converter operates as an inverter ($90 \le \alpha \le \pi$) and must reverse the polarity of back e.m.f. E_0 of the motor by reversing the field connection or the armature connection as shown in figure (2.25). Finally the converter output E_{dc} must be adjusted to be less than E_0 to obtain the desired braking current [30].

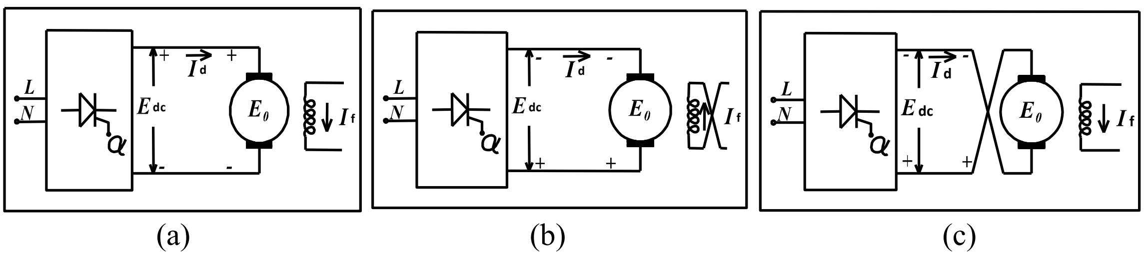


Figure (2.25): Two quadrant converter (a) Normal operation (b) Regenerative braking with field reversal (c) Regenerative braking with armature reversal

c) Single-phase full-wave dual converter (four-quadrant converter)

As described in the previous sections, the fully controlled converters can produces a reversible direct output voltage with output current in one direction, and in terms of a conventional voltage/current diagram (figure 2.26), it is said to be capable of operation in two quadrants, the first and fourth as in the case of the control of a torque motor which used to provide unidirectional torque with reversible rotation.

If four-quadrant operation of a DC motor is required, i.e. reversible rotation and reversible torque as discussed in table (2.4), a single converter needs the addition of either a change-over contactor to reverse the armature connection or means of reversing the field current in order to change the relationship between the converter voltage and the direction of rotation of the motor. Both of these are practicable in suitable circumstances but the best performance is obtained by connecting two fully-controlled converters back-to-back across the load circuit as shown in figure (2.27) [31].

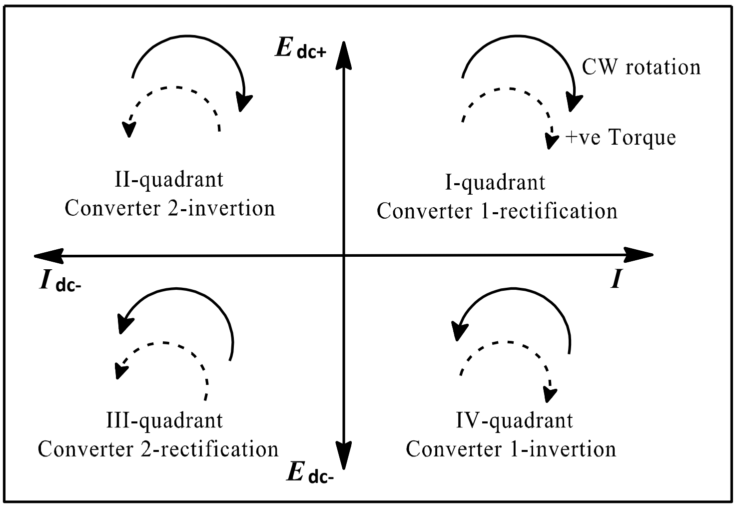


Figure (2.26): Voltage-current diagram

Table (2.4) Summary of control operation of DC motor

Quadrant	Type of Operation	Motor Rotation Direction	Motor Torque Direction	Applied Load Direction
I	Motoring	CW	CW	CCW
II	Regeneration	CCW	CW	CCW
III	Motoring	CCW	CCW	CW
IV	Regeneration	CW	CCW	CW

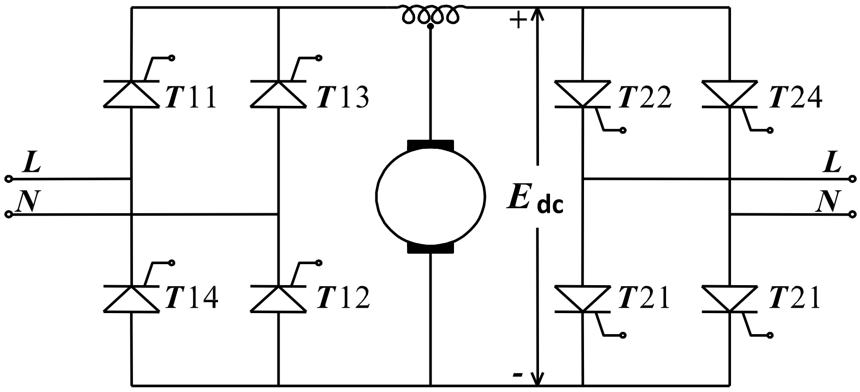


Figure (2.27): Single-phase dual converter

This system is known as a dual converter. Since, both voltage and current of either polarity are obtained with a dual converter; therefore the system will provide the four-quadrant operation.

A typical brushless drive system is shown in figure (2.28). It consists of a three phase ac motor fed from a three phase (pulse-width-modulation) PWM controlled power inverter. The drive control system has an outer motion loop, as in a brush dc servo system, which calculates the required torque to maintain the target velocity. The inner current control loop forces the appropriate winding currents, based on the machine model, so that the machine generates the desired torque.

Typically, the design of the outer motion loop is a function of mechanical system parameters and so it is independent of the drive type [31]. However, it is the way in which current is controlled in the motor windings to produce constant motor torque that differentiates the various drive system from one another.

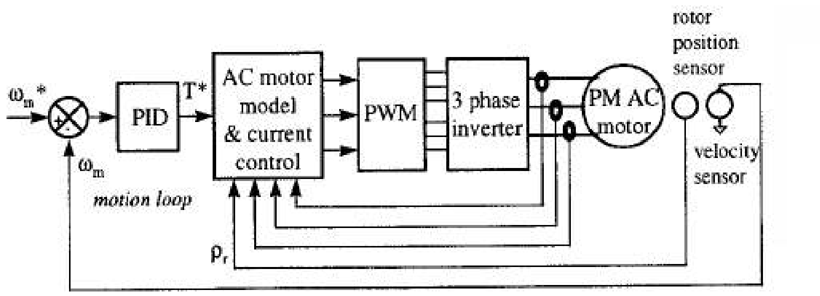


Figure (2.28): Typical brushless drive system

In order to drive the BLDC motor, an electronic commutation circuit is required. This deals with the position sensor-based commutation only. The widely used commutation methods for the BLDC motor are trapezoidal (or six-step), sinusoidal, and field oriented control (FOC) (or vector control). Each commutation method can be implemented in different ways, depending on control algorithms and hardware implementation to provide their own distinct advantages [30, 31].

2.6 PID Speed Control of BLDC Motor

The PID controller is commonly used to adjust speed. It receives signals from sensors and computes corrective action to the actuators from a computation based on the error (Proportion), the sum of all previous errors (Integral) and the rate of change of the error (Derivative).

A PID controller responds to an error signal in a closed control loop as shown in figure (2.29) and attempts to adjust the controlled quantity in order to achieve the desired system response. The controlled parameter can be any measurable system quantity, such as speed, voltage, current or stock price. The output of the PID controller can control one or more system parameters that will affect the controlled system quantity. The benefit of the PID controller is that it can be adjusted empirically by adjusting one or more gain values and observing the change in system response.

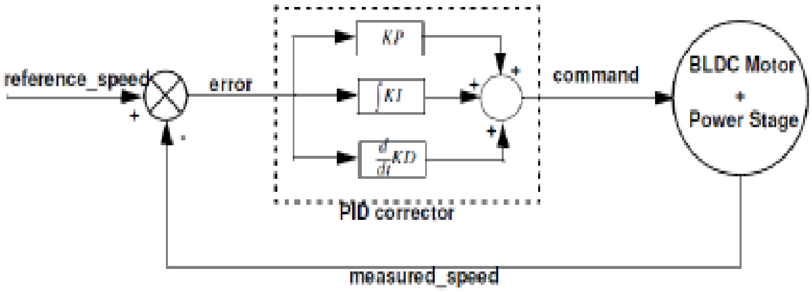


Figure (2.29): PID diagram

The mathematical model of the PID controller can be represented by:

$$u(t) = k_p[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d \frac{de(t)}{dt}] \tag{2. 9}$$

Where, *u(t)* is the output of PID controller, *e(t)* is the input of PID controller, which is the error between the desired input value and the actual output value, so called error signal, K_p is the proportional gain, T_i is the integral time, also called integral gain, and T_d Derivative time, also called derivative gain. Also the mathematical model of the PID controller can be represented by:

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(t) \cdot dt + K_d \frac{de(t)}{dt} \tag{2. 10}$$

Where, *u(t)* is the control signal, *e(t)* is the error signal, and K_p, K_i, and K_d denotes the proportional gain, integral gain and derivative gain respectively.

If different values of K_p, Ki and K_d are chosen, then it is obvious that various transient response of the plant will be obtained. The transient response of plant can be explained by four main parameters; rise time, settling time, maximum overshoot, and steady state error. The definition of each parameter is as follows:

a) Rise time (T_r)

Rise time is usually defined as the time taken for the controlled variable to go from 10 % to 90% of its final value.

b) Settling time (T_s)

Settling time refers to the time it takes for the response to settle down to within some small percentage (typically 2-5%) of its final value.

c) Maximum overshoot (M_p)

Maximum overshoot is the difference between the peak value of the response and the desired value of the controlled variable.

d) Steady state error ($e_{s.s}$)

Steady state error is the difference between where the controlled variable is and where it should be.

The mathematical model of the PID controller explained by equations (2.3) , (2.4) and (2.5) consists of Proportional Response, Integral Response and Derivative Response which are described as follows:

a) Proportional response

The proportional component can be expressed as:

$$K_p.e(t) \tag{2. 11}$$

In PID controller, the effect of controlling error depends on the proportional gain (K_p). In general, increasing the proportional gain will increase the speed of the control system response and reduce the steady-state error. However, if the proportional gain is too large, the system will begin to oscillate and become unstable. Thus, K_p must be suitably selected to keep the system stable and reduce the rise time and steady-state error.

b) Integral response

The integral component can be expressed as:

$$\frac{K_p}{T_i}\int_0^t e(t)dt \tag{2. 12}$$

From the expression shown above, it can be see that the integral component sums the error term over time. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the Steady-State error to zero. But the integral control will reduce the speed of the overall control system response and increase the overshoot. Increasing the integral gain (T_i) will cause the integral component to accumulate weakly and reduce the overshoot, thus it will make system not oscillate during the rising time, therefore improve its stability. However, it will slow the process of eliminating Steady-State error. Reducing T_i will strengthen the accumulation of integral component and shorten the time of eliminating error, but it will make the system oscillate. So *Ti* should be selected according to the practical needs.

c) Derivative response

The derivative component can be expressed as:

$$K_p T_d \frac{de(t)}{dt} \tag{2. 13}$$

The effect of derivative component depends on the derivative time constant (T_d). In general, the larger T_d, the better the effect to restrain the change of e(t) and vice versa. Thus, to select T_d properly can make the derivative component better meets the system requirement.

2.6.1 Adjusting the PID Gains

The P gain of a PID controller will set the overall system response. When first tuning a controller, the I and D gains should be set to zero. The P gain can then be increased until the system responds well to set-point changes without excessive overshoot or oscillations. Using lower values of P gain will 'loosely' control the system, while higher values will give tighter control. At this point, the system will probably not converge to the set-point.

After a reasonable P gain is selected, the I gain can be slowly increased to force the system error to zero. Only a small amount of I gain is required in most systems. Note that the effect of the I gain, if large enough, can overcome the action of the P term, slow the overall control response, and cause the system to oscillate around the set-point. If this occurs, reducing the I gain and increasing the P gain will usually solve the problem.

After the P and I gains are set, the D gain can be set. The D term will speed up the response of control changes, but it should be used sparingly because it can cause very rapid changes in the controller output. This behavior is called ‘set-point kick’. The set-point kick occurs because the difference in system error becomes instantaneously very large when the control set-point is changed. In some cases, damage to system hardware can occur. If the system response is acceptable with the D gain set to zero, you can probably omit the D term.

CHAPTER 3

Mathematical Model of Brushless DC Motor

Generally, a small horsepower BLDC motor used for position control is the same as a permanent magnet synchronous machine. The stator is constructed by three phase Y-connection without the neutral and the rotor is made by the permanent magnets. Since each phase has the phase angle difference of 120^0, the summation of all three phase currents becomes zero.

The term "brushless dc motor" is used to identify a particular type of self-synchronous permanent magnet motor in which the combination of ac machine, solid state inverter and rotor position sensor results in a drive system having linear torque-speed characteristics, as in conventional dc machine. The position sensors detect the position of the rotor poles and send control signals to switch on and off the devices in the dc - ac inverter at a frequency corresponding to the rotor speed.

For the implementation of field orientation, each three phase current control command must be generated separately. This command can be obtained by converting the controller current command based on the rotor reference frame to the stator reference frame. The three phase current command i_a, i_b and i_c are, then, tracked by the current regulated PWM (CRPWM) scheme. In this case, the current controller requires the absolute rotor position [28]. The brushless dc motor considered is a three phase permanent magnet synchronous motor. The stator windings are identical, displaced by 120^o and sinusoidal distributed. The voltage equations for the stator windings can be expressed as [26].

$$v_{as} = R.i_{as} + \frac{d}{dt}\left[L_a.i_{as} + L_{ba}.i_{bs} + L_{ca}.i_{cs}\right] + \omega_r.\lambda_m.\sin(\theta_r) \tag{3.1}$$

$$v_{bs} = R.i_{bs} + \frac{d}{dt}\left[L_{ba}.i_{as} + L_b.i_{bs} + L_{cb}.i_{cs}\right] + \omega_r.\lambda_m.\sin(\theta_r - 2\Pi/3) \tag{3.2}$$

$$v_{cs} = R.i_{cs} + \frac{d}{dt}\left[L_{ca}.i_{as} + L_{cb}.i_{bs} + L_{c}.i_{cs}\right] + \omega_r.\lambda_m.\sin\left(\theta_r + 2\Pi/3\right) \tag{3.3}$$

These equations can be rewritten in matrix form as:

$$\begin{pmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{pmatrix} = \begin{pmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{pmatrix} \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} L_a & L_{ba} & L_{ca} \\ L_{ba} & L_b & L_{cb} \\ L_{ca} & L_{cb} & L_c \end{pmatrix} \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix} + \omega_r.\lambda_m. \begin{pmatrix} \sin(\theta_r) \\ \sin(\theta_r - 2\Pi/3) \\ \sin(\theta_r + 2\Pi/3) \end{pmatrix} \tag{3.4}$$

Where:

v_{as}, v_{bs}, v_{cs} : The applied stator voltages

i_{as}, i_{bs}, i_{cs} : The applied stator currents

R : The stator resistance per phase

L_a, L_b, L_c : The self inductance of the stator windings

ω_r : The electrical rotor angular velocity

θ_r : The electrical rotor angular displacement

λ_m : The amplitude of the flux linkage established by the permanent magnet synchronous machine are uniform.

$$L_a = L_b = L_c = L \tag{3.5}$$

$$L_{ba} = L_{ca} = L_{cb} = M \tag{3.6}$$

Thus, equation no. (3.4) can be written by:

$$\begin{pmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{pmatrix} = \begin{pmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{pmatrix} \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} L & M & M \\ M & L & M \\ M & M & L \end{pmatrix} \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix} + \omega_r . \lambda_m . \begin{pmatrix} \sin(\theta_r) \\ \sin(\theta_r - 2\Pi/3) \\ \sin(\theta_r + 2\Pi/3) \end{pmatrix} \quad (3.7)$$

By Substituting of $i_{as} + i_{bs} + i_{cs} = 0$ into equation (3.7) yields:

$$\begin{pmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{pmatrix} = \begin{pmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{pmatrix} \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{pmatrix} \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix} + \omega_r . \lambda_m . \begin{pmatrix} \sin(\theta_r) \\ \sin(\theta_r - 2\Pi/3) \\ \sin(\theta_r + 2\Pi/3) \end{pmatrix} \quad (3.8)$$

The electromagnetic torque can be expressed as:

$$T_e = \frac{1}{\omega_r}\left[i_{as}.e_a + i_{bs}.e_b + i_{cs}.e_c\right] \quad (3.9)$$

The torque, velocity and displacement may be related by:

$$T_e = J.\left(\frac{2}{p}\right)\frac{d\omega_r}{dt} + B_m\left(\frac{2}{p}\right).\omega_r + T_L \quad (3.10)$$

$$\theta_r = \int \omega_r \, dt \quad (3.11)$$

$$\theta_m = \theta_r . \left(\frac{2}{p}\right) \quad (3.12)$$

Where:

J is the inertia of the motor, B_m is the friction coefficient, T_L is the load torque

θ_m is the mechanical angular displacement of rotor, *p* is the no. of poles

The mathematical model of BLDC motor described by Eq[s] (3.1) to (3.12) is used to obtain the block diagram of this system. Eq. (3.8) may be rewritten as:

$$v_{as} = R.i_{as} + \frac{d}{dt}(L - M)i_{as} + \omega_r.\lambda_m.\sin(\theta_r) \tag{3.13}$$

$$v_{bs} = R.i_{bs} + \frac{d}{dt}(L - M)i_{bs} + \omega_r.\lambda_m.\sin(\theta_r - 2\Pi/3) \tag{3.14}$$

$$v_{cs} = R.i_{cs} + \frac{d}{dt}(L - M)i_{cs} + \omega_r.\lambda_m.\sin(\theta_r + 2\Pi/3) \tag{3.15}$$

Equations (3.13) to (3.15) can be expressed in S-domain as:

$$v_{as} = R.i_{as} + S(L - M)i_{as} + \omega_r.\lambda_m.\sin(\theta_r) \tag{3.16}$$

$$v_{as} = [R + S(L - M)] \quad i_{as} + \omega_r.\lambda_m.\sin(\theta_r) \tag{3.17}$$

$$v_{as} - \omega_r.\lambda_m.\sin(\theta_r) = [R + S(L - M)] \quad i_{as} \tag{3.18}$$

Then,

$$e_a = \omega_r.\lambda_m.\sin(\theta_r) \tag{3.19}$$

$$v_{as} - e_a = [R + S(L - M)] \quad i_{as} \tag{3.20}$$

$$i_{as} = \frac{v_{as} - e_a}{R + S(L - M)} \tag{3.21}$$

By the same way we can get another two equations (3.14) to (3.15) for phases b and c respectively as:

$$e_b = \omega_r.\lambda_m.\sin(\theta_r - 2\Pi/3) \tag{3.22}$$

$$v_{bs} - e_b = [R + S(L - M)] \quad i_{bs} \tag{3.23}$$

$$i_{bs} = \frac{v_{bs} - e_b}{R + S(L - M)} \tag{3.24}$$

$$e_c = \omega_r . \lambda_m . \sin(\theta_r + 2\Pi / 3) \tag{3.25}$$

$$v_{cs} - e_c = [R + S(L - M)] \quad i_{cs} \tag{3.26}$$

$$i_{cs} = \frac{v_{cs} - e_c}{R + S(L - M)} \tag{3.27}$$

From equation (3.10), it is clear that:

$$\omega_r = (T_e - T_L)\left[\frac{p}{2}\frac{1}{JS + B_m}\right] \tag{3.28}$$

From equations (3.9) to (3.28) the block diagram is constructed for the model under consideration. This model is transferred to a simulink model by substituting with the motor parameters (Appendix B). By calculate the general constants such that (2π/3 = 2*3.14/3 = 2.09 Rad.) then, T_L is replaced by a constant torque. Eq. (3.9) can be written as:

$$T_e = 0.105\left[i_{as}.\sin(\theta_r) + i_{bs}.\sin(\theta_r - 2.09) + i_{cs}.\sin(\theta_r + 2.09)\right] \tag{3.29}$$

3.1 Build up the Complete System Simulink Model

The complete simulink system is obtained from the subsystem models. This model comprises the following subsystems.

3.1.1 BLDC Motor Model

The block diagram of BLDC motor used in simulink (Matlab) by using above equations is illustrated in figure (3.1) in the next page. The model parameter is shown in (Appendix B).

3.1.2 Three-Phase Inverter

Three-phase inverters are normally used for high power applications [20]. A three-phase output can be obtained from configuration of six transistors and six diodes as shown in figure (3.2). The gating signals of single phase inverters should be advanced or delayed by 120^{o} with respect to each other in order to obtain three-phase balanced voltages.

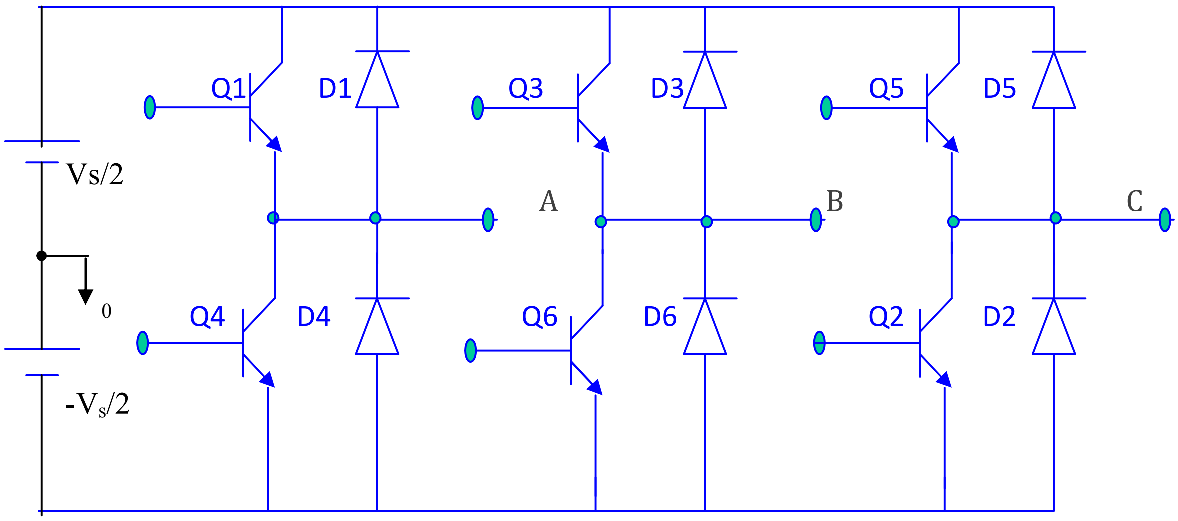


Figure (3.2): Three- phase inverter

Each transistor conducts for 180^{o} and three transistors remain ON at any instant of time. When transistor Q1 is switched ON, terminal A is connected to the positive terminal of the DC input voltage. When transistor Q4 is switched ON, terminal A is brought to the negative terminal of the DC source. There are six modes of operation in any cycle and the duration of each mode is 60^{o}. The transistors are numbered in the sequence of gating the transistors (eg.123, 234, 345, 456, 561, 612). The gating signals are shown in figure (3.3) and are shifted from each other by 60^{o} to obtain three-phase balanced voltages [20].

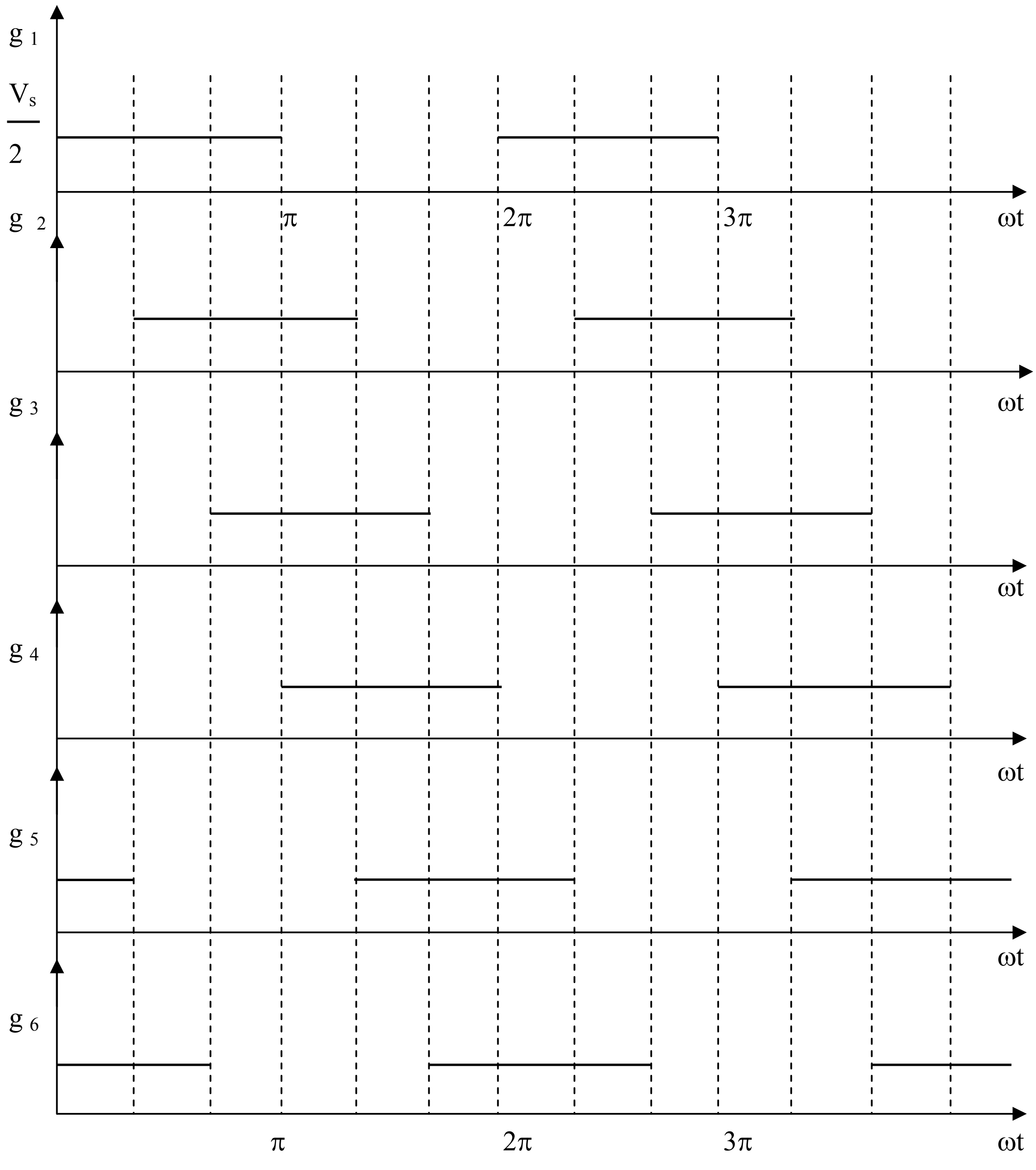


Figure (3.3): Gating signals with phase shift 60^0

The simulink model of the three phase shown in figure (3.4) is obtained using the toolbox as:

i) **IGBT inverter:** Is a three phase inverter that consists of six transistor, each of them is connected in parallel with free wheeling diode as given in figure(3.2). This inverter unit has two input teminals for voltage source, one input terminal for 6-gate signals, and three output terminals.

ii) **Voltage source:** Is split into two series connected constant voltage sources each of them is $V_s/2$.

iii) **Output voltage reference:** Is the mid-point of the sourcee voltage is considered as the reference point for the output phase voltages. This is shown by the summing block in figure (3.4).

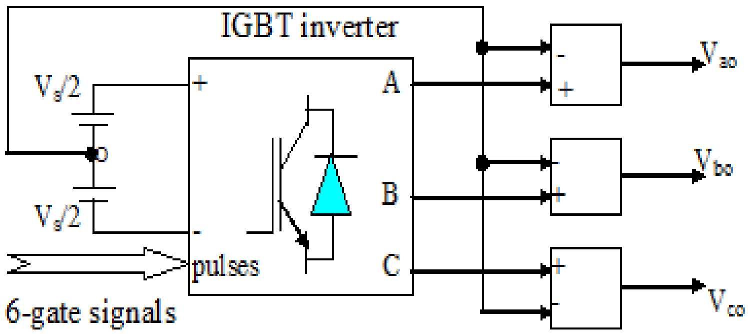


Figure (3.4): Simulink inverter model

iv) **Gate Signal Generation in Simulink:** In this subsystem a simulink model as shown in figure (3.5) to generate the 6-gate signals. The detailed gate signal simulink units are as follows:

Thee sinusoidal signals sinθ, $\sin(\theta_r\text{-}2\pi/3)$, and $\sin(\theta_r\text{+}2\pi/3)$ are generated using (sin) function blocks. These 3 functions are multiplied by 2 and then limited to generate trapezoidal emf as disscused in next section. These functions are multiplied by the controller output in order to obtain trapezoidal functions with variable amplitudes according to the controller output. The generated trapizoid waves are then compared with triangle signals with constant amplitudes and frequency in order to obtain pulse width modulated signals. The frequency of the trapezoidal signals determines the number of pulses in each half cycle. This is called trapezoidal modulation. Each sine wave generates two complementary gate signals for the two transistors in the same arm in order to prevent their switching on at the same time. In the simulink model, the sawtooth frequency = 10000Hz =10KHz, and amplitude is 1 volt. The sawtooth generation model is predesigned by MATLAB toolbox, just determine the frequency and amplitude of the signal.

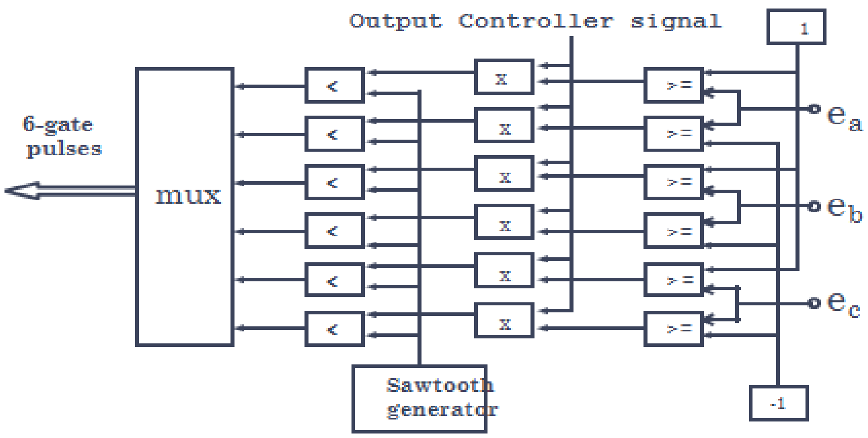


Figure (3.5): Simulink gate signal generation model

v) Trapezoidal EMF generation

The trapezoidal function is generated using sinusoidal function with amplitude =2 together with limiter with limit value =1 as shown in figure (3.6).

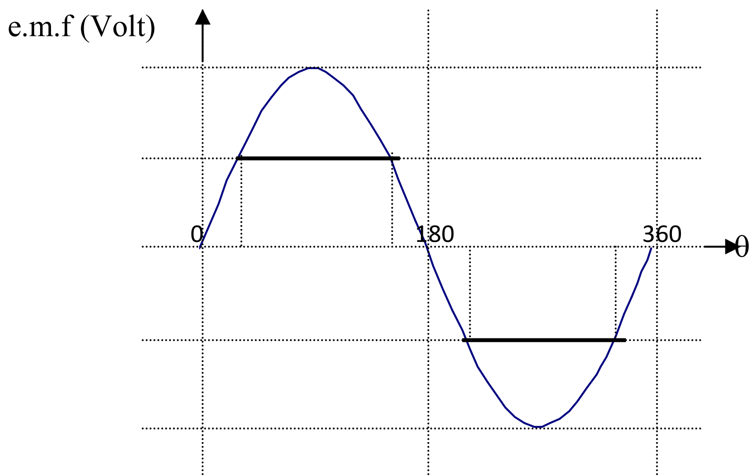


Figure (3.6) Trapezoidal back emf function

3.1.3 PID Controller Simulink Model

The simulink model of PID controller is shown in figure(3.7). The input to this model is the speed error signal obtained by subtracting the reference speed (ω_r) from the actual speed (ω) and the output is the manupulated signal. Three terms are added together, the first represents the integral term (1/S) with gain K_i (initial value 0.6), the second is the proportional term with gain K_p (initial value 0.8), and the third is the derivative term (du/dt) with gain K_d (initial value 0). The summing of the three terms gives the controller output. The complete BLDCM system model is shown in figure (3.8).

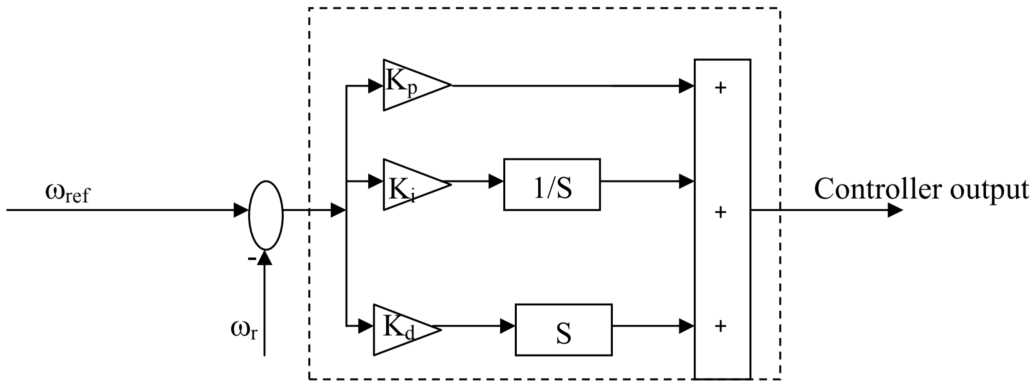


Figure (3.7): Simulink PID controller model

CHAPTER 4

Optimization Techniques

4.1 Introduction

Optimization is usually defined as the process of finding the conditions that produce a maximum or a minimum value to a function. Without loss of generality, optimization can be taken to mean minimization since the maximum of the function can also be found by seeking the minimum of the negative of the function [5].

The parameter tuning of PID controller can be considered by selecting the three parameters K_p, K_i, and K_d such that the response of the plant will be as desired. The tuning of the parameters of PID controller has been quite difficult because many industrial plants are often burdened with problems such as high orders; time delays; and nonlinearities. Ziegler-Nichols tuning formula is perhaps the most well-known tuning method, some other methods exist for the PID tuning, but in many industrial plants, it is often hard to determine optimal or near optimal PID parameters [22].

In this chapter, the classification of different optimization problems is presented along with a brief discussion of their selection factors, The general mathematical formulation of multi-objective optimization is described and followed by three solution techniques (Genetic algorithm, Particle Swarm Optimization and Bacterial Foraging Algorithm) and hybrid technique(Particle Swarm - Bacterial Foraging). The flexibility of the four techniques is demonstrated using different fitness functions.

4.2 Optimization Problem Classification

It is important for the designer to be able to correctly categorize the type of optimization problem. Incorrect identification usually leads to local or infeasible solutions. Optimization problems can be classified in several ways [32]:

i) Number Of Variables

-Univariate optimization: Functions with a single independent variable.

-***Multivariate optimization:*** Functions have 2 or more independent variables.

ii) Constraints

-***Constrained optimization:*** The solution must satisfy the constraints to be a feasible solution.

-***Unconstrained optimization:*** There are no constraints and the solution is always accepted.

iii) Type Of Objective And Constraint Functions

-***Linear:*** Objective and constraint functions are linear functions of the independent variables.

-***Non-Linear:*** At least one of the objective or constraint functions must be nonlinear.

iv) Determinism

-***Deterministic methods:*** The majority of optimization methods can be categorized as deterministic so that if they are repeatedly started from the same point they will always converge to the same value via the same route.

- ***Stochastic Methods:*** Stochastic optimization methods are increasingly being used to solve problems where convergence to a global minimum is required. These methods overcome the tendency of the descent techniques to converge to the closest local minimum by allowing some "uphill" movements to be accepted in the process. This hill climbing provides a method of escaping from local minima so that the possibility of convergence to the global minimum increases. These methods include: Simulated annealing and Genetic Algorithms. The main disadvantage of stochastic techniques is they need a large number of function evaluations to obtain convergence.

v) Number of objectives

-***Single objective optimization (SOP):*** The solution can be interpreted as a minimum or maximum of the objective-function.

-***Multi-objective optimization (MOP):*** The solution is usually a compromise of various conflicting objectives.

4.3 Selection Factors

The suitability of an algorithm for solving an optimization problem depends largely on the types of functions involved and on the ease with which the first and/or second derivatives of the function can be computed. All of the optimization techniques face the following critical factors when applied [33].

-Computation time.

-Local optima.

-Computation of derivatives.

- Search and decision maker.

4.4 General Formulation of Optimization Problem

Multi-objective optimization (MOP) also called multi-criteria or vector optimization can be defined as the problem of finding a vector of decisions or variables X which satisfies constraints and optimize a vector function F(X) whose elements represent the objective functions. The general programming (GP) problem can be stated as [33]:

general programming (GP) problem can be stated as [31]:

$$Minimize\, F(X) \qquad (X \;\varepsilon\; R^n) \tag{4. 1}$$

Subject to

$$h_i(X) = 0 \quad (i = 1, \ldots\ldots\ldots, m_e) \tag{4. 2}$$

$$g_i(X) \leq 0 \quad (j = 1, ...,m_i) \tag{4. 3}$$

and

$$F(X) = [F_1(X), F_2(2), ..., F_k(X)] \tag{4. 4}$$

Where F, h_i and g_j are real-valued functions of the n-variable vector X, m_e is the number of equality constraints, m_i is the number of inequality constraints and k is the number of objectives.

The number of variables n and the number of constraints (m_e + m_i) need not to be related in any way. A problem in which the total number of constraints equals zero is called an unconstrained optimization problem.

It is worth noting that single objective optimization (SOP) can be formulated as a special case of multi-objective optimization in which the number of objectives k is equal to one. In multi-objective optimization, the problem could be:

- Minimization of all the objective functions.

- Maximization of all the objective functions.

- Minimization of some and maximization of the reminders.

For simplicity reasons, all objectives are converted to a minimization form as:

$$\max F_i(X) = -\min(-F_i(X)) \tag{4. 5}$$

Similarly, the inequality constraints of the form

$$g_i(X) \geq 0 \quad (j = 1, ...,m_i) \tag{4. 6}$$

Can be converted to

$$-g_i(X) \leq 0 \quad (j = 1, ...,m_i) \tag{4. 7}$$

The feasible solution X_f is defined as the set of variables that satisfy the equality and inequality constraints:

$$X_f = \{X \ \varepsilon \ R^n \ \mid h_i(X) = 0, g_j(X) \le 0\} \tag{4. 8}$$

The corresponding feasible region Y_f in the objective function space can be defined as:

$$Y_f = F(X_f) \tag{4. 9}$$

4.5 Compromise Solution Methods for Multi-Objective Problems

In these methods, the original MOP is converted to an SOP and then solved by any of the deterministic or stochastic techniques. Traditional methods are usually designed to give the best compromise solution rather than to give a Pareto optimal set. There are a large number of compromise solution methods so for brevity only the most popular methods will be described here. These methods include: Aggregating Functions, Constraint Methods, Sequential Methods, Goal Attainment Method and Error Criterion Methods (IAE, ISE and ITAE) [31, 32]. Aggregating Functions include many techniques such as Weighted Sum Method and Global Criterion Method.

4.5.1 Weighted Sum Method

This method consists of adding all objective functions together using different weighting factors. Mathematically, the new function $F^{/}$ can be expressed as [34]:

$$F^{/} = \sum_{i=1}^{k} w_i f_i(X) \tag{4. 10}$$

Subject to

$$0 \le w_i \le 1 \tag{4. 11}$$

and

$$\sum_{i=1}^{k} w_i = 1.0 \tag{4. 12}$$

Where, w_i is the i th weighting factor

Weighting factors are usually assigned according to the importance of objectives. However, in general objectives may have different ranges of values and in this case they must be normalized.

- Advantages of Weighted Sum Method:

- It reduces an MOP to an SOP and hence fast SOP deterministic methods can be used to solve the problem.

- Varying the weighting factors locates different points in the Pareto set.

- It does not need an Ideal vector of optimized objectives.

- Disadvantages of Weighted Sum Method:

- The difficulty in choosing the best set of weights for the problem.

- The difficulty of dealing with different quantities that are measured on different scales (normalization problem).

- For non-convex problems, certain non-inferior solutions may be inaccessible.

4.5.2 Global Criterion Method

Global criterion is usually defined as a measure of how close the designer can get to the ideal vector.

The Scalar objective function for this method is usually written as:

$$F^{/} = \sum_{i=1}^{k} \left(\frac{f_i^k - f_i(X)}{f_i^*} \right)^P \qquad (4.\ 13)$$

Where, f_i^* is the ideal value of the i th objective.

In this formula *P* is usually taken as 1.0 or 2.0, however other values can also be used [34]. The results will differ greatly depending on the value of *P* selected.

-Advantages of the Global Criterion Method:

- Like the weighted sum method, it has the ability to reduce the MOP to an SOP.

- Disadvantages of Global Criterion Method:

- The difficulty of selecting suitable values of *P* to ensure a feasible solution.

- The method needs the optimizer to have a good idea about the ideal or at least the satisfied demand level for each objective function and this is sometimes not available especially for preliminary design problems.

4.5.3 Constraint (Trade-Off) Method

The Constraint method involves optimizing the main or primary objective function *Fp* and expressing the other objectives in the form of constraints [35].

$$\underset{x}{Minimize}\ F_b(X) \qquad (X \in R^n) \tag{4. 14}$$

Subject to

$$F_i(X) \leq \varepsilon_i \qquad (i = 1, ...k, and\ i \neq p) \tag{4. 15}$$

Where, ϵ_i are the parameters that are assumed by the optimizer to find the best compromise solution.

Figure (4.1) shows a two-dimensional representation of the constraint method for a two-objective problem.

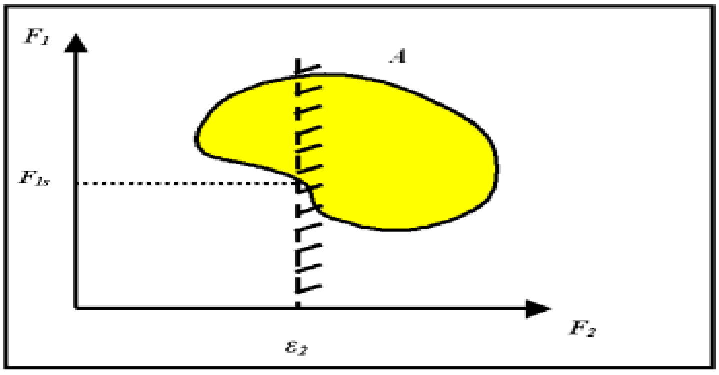


Figure (4.1): Graphical representation of Constraint Method

Advantages of the Constraint Method:

- Like the weighted sum method, it has the ability to reduce the MOP to an SOP.
- The ability to get the complete Pareto set of optimal points, but under the condition that all possible values of ϵ_i are used.
- This approach is able to identify a number of non-inferior solutions on a non-convex boundary that are not obtainable using the weighted sum technique. For example, at the solution point $F_1=F_{1s}$ *and* $F2=\epsilon_2$.

Disadvantages of the Constraint Method:

- The difficulty of selecting suitable values of ϵ_i to ensure a feasible solution.
- If the bounds ϵi are too hard, there is no solution and at least one of these bounds must be released or relaxed.
- Although it does not need the ideal vector of optimized functions, it does need the optimizer to have a good idea about the satisfied demand level of each objective function and this is sometimes unavailable especially for preliminary designs.

4.5.4 Sequential (Lexicographic) Method

This method can be applied only if a preference order can be assigned to the k objectives starting from the most important function for the decision maker to the least important one. Each objective is then minimized individually subject to a constraint that it does not allow the optimum of the new function to exceed a minimum of the previous function as follows [36]:

$$\underset{X}{Minimize}\ F_1(X) \qquad (X \in R^n) \tag{4. 16}$$

Subject to the same inequality and equality constraints of equations (4.2) and (4.3)

If f_1^* denotes the optimal value of F_1, then in the second step the problem can be solved:

$$\underset{X}{Minimize}\ F_2(X) \qquad (X \in R^n) \tag{4. 17}$$

Subject to all constraints plus one new constraint:

$$F_1(X) = f_1^* \tag{4. 18}$$

If f_2^* denotes the optimal value of F_2, then in the third step the problem can be solved as:

$$\underset{X}{Minimize}\ F_3(X) \qquad (X \in R^n) \tag{4. 19}$$

Subject to all equality and inequality constraints plus one new constraint:

$$F_i(X) = f_i^* \qquad (i = 1,2) \tag{4. 20}$$

And so on.

Advantages of the Sequential Method:

-It has the ability to reduce the MOP to a multi-step SOP and hence we can use classical SOP techniques to get the solution of each step.

-It is easy to implement and no ideal vector is needed for this approach.

Disadvantages of the Sequential Method:

-This method is not suitable for engineering problems where unique solutions of objectives are common.

-The method is not suitable for multi objective problems if all objectives are equally important.

-The method assumes that the relative importance of each objective is known in advance.

4.5.5 Min-Max (Goal Attainment) Method

The min-max optimum is the set of points which will give the smallest values of the relative deviation from the ideal (or goal) objective functions. This method assumes that each objective is equally important and allows under- or over-achievement, enabling the designer to be relatively imprecise about the initial design goals [35, 36].

If f_i^* is the ideal or the goal solution of the objective function F_i, then:

$$Z_h = \max\left[\left(\frac{\left|F_i(X) - f_i^*\right|}{f_i^*}, \frac{\left|F_i(X) - f_i^*\right|}{F_i}\right) w_i\right] \qquad (4.21)$$

Then the optimization problem can be formulated as follows:

$$\underset{X}{Min}\left(\max(Z_i(x))\right) \qquad X \in R^n \qquad (4.22)$$

If the weighting w_i= 1, all objectives are equally treated and the approach is called the Min-Max method. Otherwise the approach is called the goal attainment method.

Advantages of Min-Max Method:

-The main strength of this method is their efficiency (computationally) and eases of implementation.

Disadvantages of Min-Max Method:

-The main drawback of this method is the need for an ideal or goal vector and weight for each objective (goal attainment method).

4.5.6 Integrated Absolute Error (IAE)

Integrated absolute error (IAE) is one of the most common criterions that can be evaluated analytically in the frequency domain. The IAE performance criterion formulas is given as follow [37]:

$$IAE = \int_0^\infty |r(t) - y(t)|dt = \int_0^\infty |e(t)|dt \tag{4. 23}$$

where, *r(t)* is the desired output, *y(t)* is the plant output, and *e(t)* is the error signal

Then the fitness function (*F*) to be maximized using *IAE* is as follow:

$$F = \frac{1}{IAE} \tag{4. 24}$$

Advantages of the Integrated Absolute Error Method:

-The main advantage of this method is that it is simple in implementation [37].

Disadvantages of Integrated Absolute Error Method:

-The main disadvantage of the IAE criterion is that its minimization can result in a response with relatively small overshoot but a long settling time because the IAE performance criterion weights all errors equally independent of time.

4.5.7 Integrated Square Error (ISE)

The ISE performance criterion formulas are as follow [37]:

$$ISE = \int_0^\infty [e(t)]^2 \, dt \tag{4. 25}$$

Then the fitness function (F) to be maximized using ISE is as follow:

$$F = \frac{1}{ISE} \tag{4. 26}$$

<u>Advantages and Disadvantages of this method</u> are the same as Integrated Absolute Error Method.

4.5.8 Integrated of Time-Weighted-Absolute-Error (ITAE)

The ITAE performance criterion formulas are as follow [37]:

$$ITAE = \int_0^\infty t\left|e(t)\right| dt \tag{4. 27}$$

Then the fitness function (F) to be maximized using ITAE is as follow:

$$F = \frac{1}{ITAE} \tag{4. 28}$$

<u>Advantages of the Integrated Absolute Error Method:</u>

-The ITAE performance criterion can overcome the disadvantage of the IAE and ISE criteria.

<u>Disadvantages of Integrated Absolute Error Method:</u>

-The derivation processes of the analytical formula are complex and time-consuming.

4.6 Stochastic Optimization Techniques

Several new methods from an artificial intelligent approach, such as PSO and BFA, the applications of PSO and BFA have expanded into various fields. With the abilities for global optimization and good robustness, and without knowing anything about the underlying mathematics, PSO and BFA are expected to overcome the weakness of traditional PID tuning techniques and to be more acceptable for industrial practice [38].

4.6.1 Particle Swarm Optimization (PSO) Technique

Particle Swarm Optimization (PSO) is one of the EC algorithms. It is also a form of swarm intelligence that comprises artificial intelligence technique based on the study of collective behavior in decentralized, self-organized systems [38].

Inspiring from the swarms in nature, such as; birds, fish, etc., Kennedy and Eberhart proposed a population based algorithm called Particle Swarm Optimization (PSO). PSO combines cognition only model that values solely the self-experience and social only model that values solely the experience of neighbors. A particle encodes a candidate solution to a problem at hand. The algorithm uses a set of particles flying over a search space and moving towards a promising area to locate a global optimum [39].

The particles evaluate their positions relative to a goal (fitness function) every iteration, and particles in local neighborhood share memories to adjust their own velocities and thus subsequent positions. PSO is basically developed through simulation of bird flocking in two-dimension space. The position of each agent is represented by XY-axis position and also, the velocity is expressed by V_x (the velocity of x-axis) and V_y (the velocity of y-axis). Modification of the agent position is realized by the position and velocity information. Bird flocking optimizes a certain object function. Each agent knows its best value so far (p best) and its XY position. This information is analogy of personal experiences of each agent. Moreover, each agent knows the best value so far in the group (g best) among bests. This information is analogy of knowledge of how the other agents around them have performed [38, 39].

4.6.1.1 Background of Particle Swarm Optimization

Natural creatures sometimes behave as a swarm. One of the main streams of artificial life researches is to examine how natural creatures behave as a swarm and reconfigure the swarm models inside a computer. Swarm behavior can be modeled with a few simple rules. School of fishes and swarm of birds can be modeled with such simple models. Namely, even if the behavior rules of each individual (agent) are simple, the behavior of the swarm can be complicated. Reynolds called this kind of agent as boil and generated complicated swarm behavior [40]. He utilized the following three vectors as simple rules.

- to step away from the nearest agent
- to go toward the destination
- to go to the center of the swarm

Namely, behavior of each agent inside the swarm can be modeled with simple vectors. This characteristic is one of the basic concepts of PSO. Boyd and Richerson examine the decision process of human being and developed the concept of individual learning and cultural transmission. According to their examination, people utilize two important kinds of information in decision process. The first one is their own experience; that is, they have tried the choices and know which state has been better so far, and they know how good it was. The second one is other people's experiences; that is, they have knowledge of how the other agents around them have performed. Namely, they know which choices their neighbors have found are most positive so far and how positive the best pattern of choices was. Namely each agent decides his decision using his own experiences and other peoples' experiences. This characteristic is another basic concept of PSO [40].

According to the background of PSO and simulation of swarm of bird, Kennedy and Eberhart developed a PSO concept. Each agent tries to modify its position using the following information [16]:

- the current positions (x, y),
- the current velocities (v_x, v_y),
- the distance between the current position and p best
- the distance between the current position and g best

This modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation:

$$v_i^{k+1} = w.v^k + c_1 randx\ (pbest_i - s_i^k) + c_2 rand_2 x(gbest - s_i^k) \tag{4. 29}$$

where, v_i^k is the velocity of agent *i* at iteration *k*, *w* is the weighting function, c_j is the correction factor, rand is the random number between 0 and 1, s_i^k is the current position of agent *i* at iteration *k*, $pbest_i$ is the p best of agent *i*, g best is the g best of the group.

The following weighting function is usually utilized in (4.29):

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} x\ iter \tag{4. 30}$$

Where, $w_{\max}$ is the final weight, $w_{\min}$ is the initial weight, $iter_{\max}$ is the maximum iteration number, $iter$ is the current iteration number.

Using the above equation, a certain velocity, which gradually gets close to p best and g best can be calculated. The current position (searching point in the solution space) can be modified by equation (4.31).

$$s_i^{k+1} = s_i^k + v_i^{k+1} \tag{4. 31}$$

Figure (4.2) shows a concept of modification of a searching point by PSO.

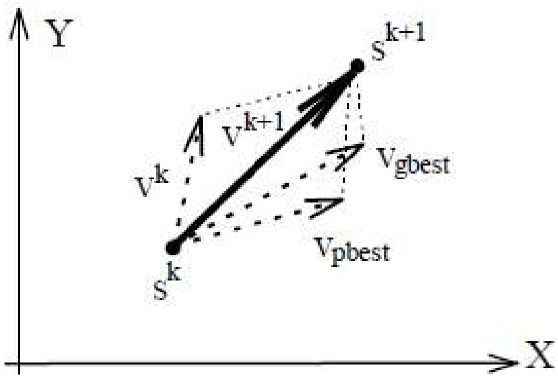


Figure (0.2): Concept of modification of a searching point by PSO.

s^k : Current searching point,

s^{k+1} : Modified searching point,

v^k : Current velocity,

v^{k+1} : Modified velocity,

v_{pbest} : Velocity based on p best

v_{gbest} : Velocity based on g best

The general flow chart of PSO can be described as follows.

Step 1: Generation of initial condition of each agent

Initial searching points (s_i^0) and velocities (v_i^0) of each agent are usually generated randomly within the allowable range. The current searching point is set to p best for each agent. The best-evaluated value of p best is set to g best and the agent number with the best value is stored.

Step 2: Evaluation of searching point of each agent

The objective function value is calculated for each agent. If the value is better than the current p best of the agent, the p best value is replaced by the current value. If the best value of p best is better than the current g best, g best is replaced by the best value and the agent number with the best value is stored.

Step 3: Modification of each searching point

The current searching point of each agent is changed using equations (4.29), (4.30) and (4.31).

Step 4: Checking the exit condition

The current iteration number reaches the predetermined maximum iteration number, then exit. Otherwise, go to step 2. Figure (4.3) shows the general flow chart of PSO.

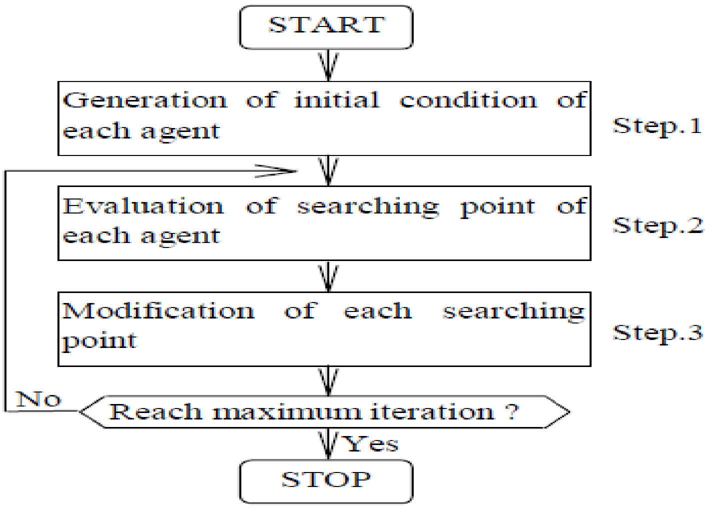


Figure (4.3): Flow chart of PSO

4.6.1.2 Parameter Selection

PSO has several explicit parameters whose values can be adjusted to produce variations in the way the algorithm searches the solution space. The parameters in equations (4.29) and (4.30) are as follows:

C_j is the weighting factor, w_{max} is the initial weight of the weight function and w_{min} is the final weight of the weight function. Shi and Eberhart tried to examine the parameter selection of the above parameters. According to their examination, the following parameters are appropriate and the values do not depend on problems: $C_j = 2.0$, w_{max}=0.9, w_{min}=0.4.

4.6.1.3 PSO Advantages

Particle swarm optimization has been found to be extremely effective in solving a wide range of engineering problems. It is very simple to implement (the algorithm comprises two lines of computer code) and solves problems very quickly [41].

Particle swarm optimizations have been found extremely effective in tuning controller parameters robustly and have proved their excellence in giving better results by improving the steady state characteristics and performance indices [42].

It was reported that the PSO algorithm can generally achieve a higher performance than genetic algorithm.

Another reason for utilizing PSO algorithm is that there is no need of gradient information or calculation of gradient.

PSO has a unique ability for optimization of non-linear and multidimensional function.

4.6.1.4 Tuning of PID Controller with PSO (PSO-PID)

Tuning procedures of PID controller parameters using PSO can be summarized in the following points [43].

a) The range of parameters K_p, K_i and K_d should be investigated from the stability point of view, this means applying Routh-stability test or any other stability check criteria into the system to be controlled with the PID controller and find the range of controller parameters in which the system is stable.
b) Initialize a population of particles with random positions and velocities on D dimensions in the problem space.
c) The system (model of the plant and the controller) is simulated. For each particle, evaluate the desired optimization fitness function in D variables. The fitness function can be calculated using equations (4.24), (4.26) and (4.28).

d) Compare particle's fitness evaluation with its p best. If current value is better than p best, then set p best equal to the current value, and p_i equals to the current location x_i in D-dimensional space.
e) Identify the particle in the neighborhood with the best success so far, and assign its index to the variable g.
f) Change the velocity and position of the particle according to equation (4.29) and (4.31).
g) Loop to step (c) until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations.

4.6.2 Bacterial Foraging Technique (BF)

Recently, search and optimal foraging of bacteria have been used for solving optimization problems. To perform social foraging, an animal needs communication capabilities and over a period of time it gains advantages that can exploit the sensing capabilities of the group. This helps the group to predate on a larger prey, or alternatively, individuals could obtain better protection from predators while in a group [18].

The common type of bacteria is Escherichia coli (E.coli) [19]. Its behavior and movement comes f rom a set of six rigid spinning (100–200 r.p.s) flagella, each driven as a biological motor. An E.coli bacterium alternates through running and tumbling. But they cannot swim straight. The chemotactic actions of the bacteria are modeled as follows:

- In a neutral medium, if the bacterium alternatively tumbles and runs, its action could be similar to search.
- If swimming up a nutrient gradient (or out of noxious substances) or if the bacterium swims longer (climb up nutrient gradient or down noxious gradient), its behavior seeks increasingly favorable environments.
- If swimming down a nutrient gradient (or up noxious substance gradient), then search action is like avoiding unfavorable environments.

Therefore, it follows that the bacterium can climb up nutrient hills and at the same time avoids noxious substances. The sensors it needs for optimal resolution are receptor proteins which are very sensitive and possess high gain. That is, a small change in the concentration of nutrients can cause a significant change in behavior. This is probably the best-understood sensory and decision-making system in biology [18].

Mutations in E.coli affect the reproductive efficiency at different temperatures, and occur at a rate of about 10^{-7} per gene per generation. E. coli occasionally engages in a conjugation that affects the characteristics of the population. There are many types of taxis that are used in bacteria such as, aerotaxis (attracted to oxygen), phototaxis (light), thermotaxis (temperature), magnetotaxis (magnetic lines of flux) and some bacteria can change their shape and number of flagella (based on the medium) to reconfigure in order to ensure efficient foraging in a variety of media. Bacteria could form intricate stable spatio-temporal patterns in certain semisolid nutrient substances and they can survive through a medium if placed together initially at its center. Moreover, under certain conditions, they will secrete cell-to-cell attractant signals so that they will group and protect each other.

The bacterial foraging system consists of four principal mechanisms, namely chemotaxis, swarming, reproduction, and elimination dispersal [18, 19]. Below each of these processes will be described and followed by a flowchart.

A. <u>Chemotaxis</u>

This process simulates the movement of an E.coli cell through swimming and tumbling via flagella. Biologically, an E.coli bacterium can move in two different ways. It can swim for a period of time in the same direction, or it may tumble, and alternate between these two modes of operation for the entire lifetime. Suppose $\theta^i(j,k,l)$ represents i^{th} bacterium at j^{th} chemotactic, k^{th} reproductive and l^{th} elimination dispersal step. *C(i)* is the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemotaxis the movement of the bacterium may be represented by equation (4.34).

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + c(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{4. 32}$$

Where Δ indicates a vector in the random direction whose elements lie in [–1, 1]

B. Swarming

Interesting group behavior has been observed for several motile species of bacteria including E.coli and S. typhimurium, where intricate and stable spatiotemporal patterns (swarms) are formed in a semisolid nutrient medium [21]. A group of E.coli cells arrange themselves in a travelling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemoeffecter. The cells, when stimulated by a high level of succinate, release an attractant aspirate, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density. The cell-to-cell signaling in E. coli swarm may be represented by the following function:

$$\begin{aligned} j_{cc}(\theta, p(j,k,l)) &= \sum_{i=1}^{s} j_{cc}(\theta, \theta^i(j,k,l)) = \sum_{i=1}^{s}[-d_{attractant}\exp(-\omega_{attractant}\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2)] \\ &+ \sum_{i=1}^{s}[h_{reppelant}\exp(-\omega_{reppelant}\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2)] \end{aligned} \tag{4. 33}$$

where $j_{cc}(\theta, p(j,k,l))$ is the objective function value to be added to the actual objective function (to be minimized) to present a time-varying objective function, S is the total number of bacteria, p is the number of variables to be optimized that are present in each bacterium, and $\theta = [\theta_1, \theta_2, ..., \theta_p]^T$ is a point in the p-dimensional search domain. $d_{attractant}$, $\omega_{attractant}$, $h_{repellant}$, $\omega_{repellant}$ are different coefficients that should be chosen properly.

C. Reproduction

The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.

D. <u>Elimination and Dispersal</u>

Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons: e.g., a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. Events can take place in such a fashion that all the bacteria in a region are killed or a group is dispersed into a new location. To simulate this phenomenon in BF technique, some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space.

4.6.3 Applications of BF Technique

BF technique has successfully been applied to real-world problems such as [18, 19]:

- optimal controller design
- harmonic estimation
- transmission loss reduction
- active power filter synthesis
- learning of artificial neural networks

CHAPTER 5

Simulation Results

The optimum PID controller tuned by PSO and BFO techniques given in chapter (4) is applied to closed-loop control of BLDC motor and its drive circuit. The design of PID controller requires specification of three parameters: proportional gain (K_p), Integral gain (K_i), Derivative gain (K_d), Integral time constant (T_i) and Derivative time constant (T_d). So far, great effort has been devoted to develop methods to reduce the time spent on optimization of the choice for controller parameter. The limitations of fixed parameters PID controller has lead to the requirement of developing more advanced control techniques such as artificial intelligence (AI)

5.1 Measurements of (BLDC) Motor Voltage and Current

Figure (5.1) shows back e.m.f from the motor, figure (5.2) shows the increasing of three-phase currents of the stator windings due to torque variation (50%full load torque) at (t=0.2 Sec.), figure (5.3) shows the increasing of three-phase currents of the stator windings due to torque variation (75%full load torque) at (t=0.2 Sec.). Figure (5.4) shows phase voltage V_a of the BLDC motor, figure (5.5) shows Torque / Speed characteristics.

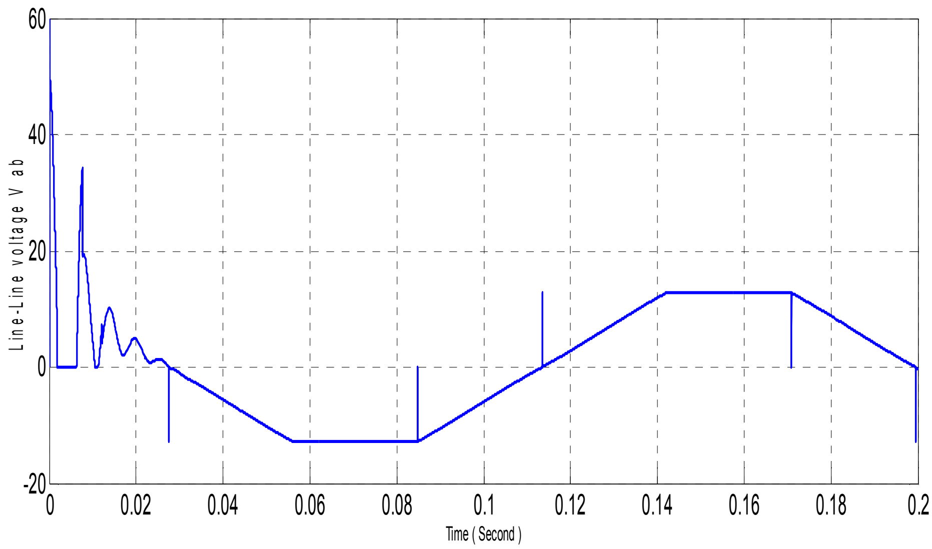


Figure (5.1): line – line voltage V_{ab}

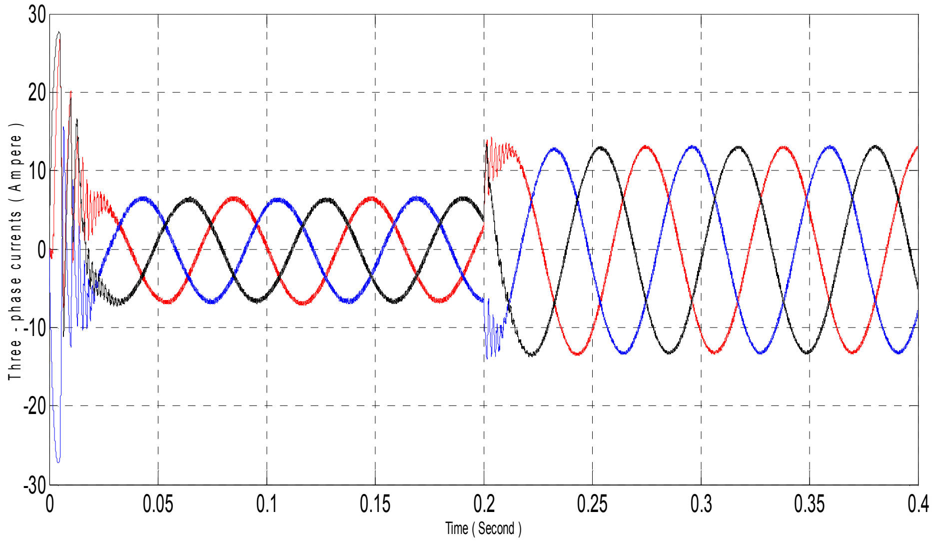


Figure (5.2): Effect the torque variation (50% full load torque) on three – phase currents of the stator

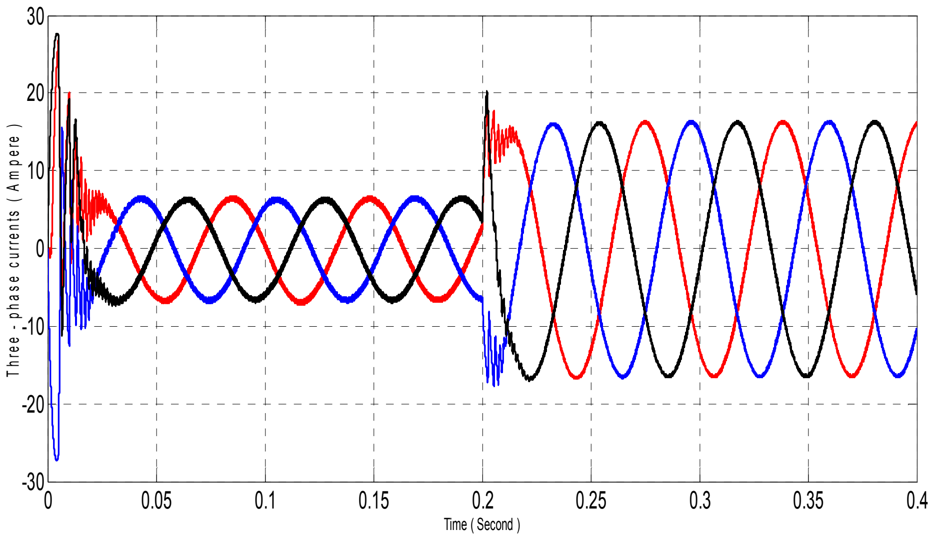


Figure (5.3): Effect the torque variations (75% full load torque) on three – phase currents of the stator

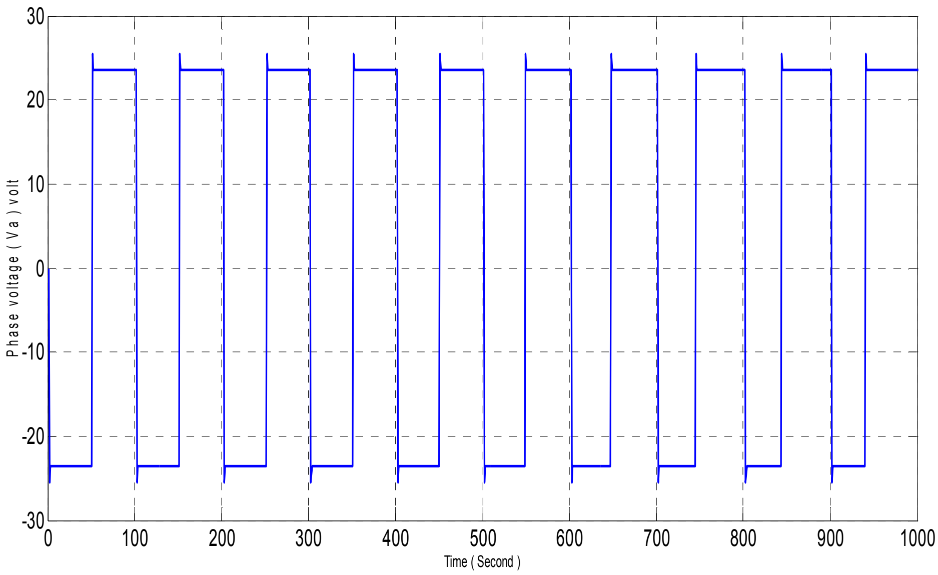


Figure (5.4): Phase voltage V_a

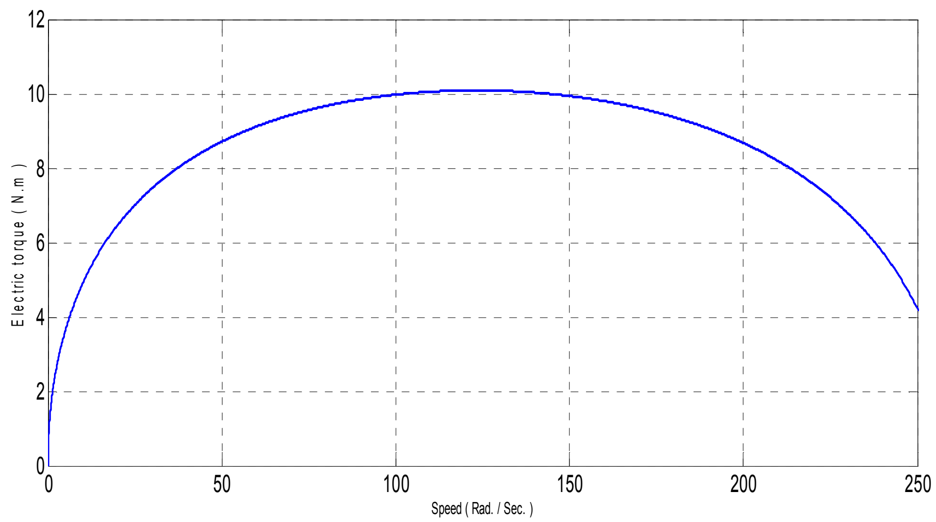


Figure (5.5): Torque / Speed characteristics

5.2 Step Response with (PSO-PID)

Table (5.1) and figure (5.6) shows step response parameters of the closed loop system with PSO-PID controller using ITSE based fitness function and without PSO-PID controller. Figure (5.7) shows the effect of torque variation (75% from full load torque) on the speed response at (t=0.05) Sec. with recovery time = 3.35 m Sec., figure (5.8) illustrates variation of external load torque (75% from full load torque).

Table (5.1): PSO-PID controller parameters

PSO (PID) Controller	K_p	K_i	K_d	Peak time (t_p)	Rise time (t_r)	Settling time (t_s)	Max. over shoot M_p %	Steady state error $e_{s.s}$
Without	1	0	0	0.0055	0.0030	0.0163	13.956	0.5248
With	6.6397	1	0.0028	0.0051	0.0030	0.0047	0.5698	0.0368

Where, no. of iterations = 100, Swarm size = 50, Correction factor = 2,

Weighting factor = 1 , ITSE (objective fitness function used with PSO) = $\int t\, e^2\, dt$

Where: ITSE = 6.431

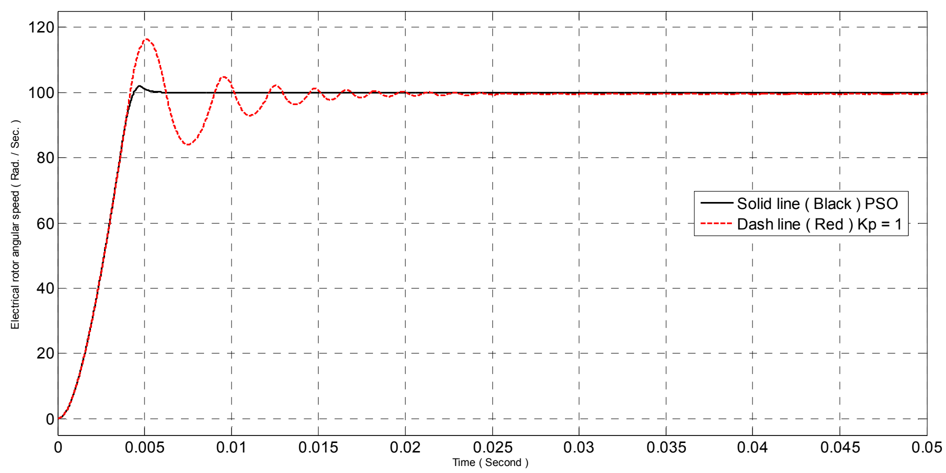


Figure (5.6): Step response of the closed loop system with PSO-PID controller using ITSE based fitness function and without PSO-PID controller

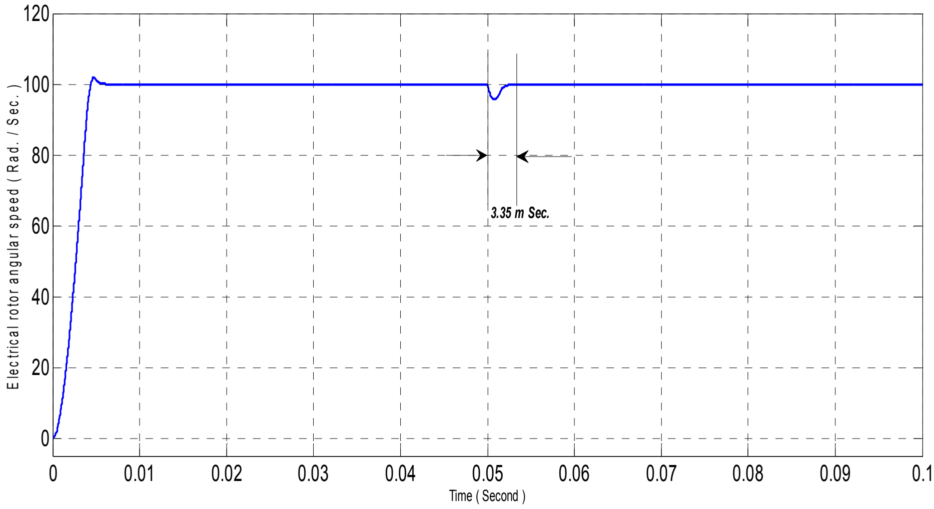


Figure (5.7): Effect of torque variation (75% from full load Torque) on the speed response using PSO

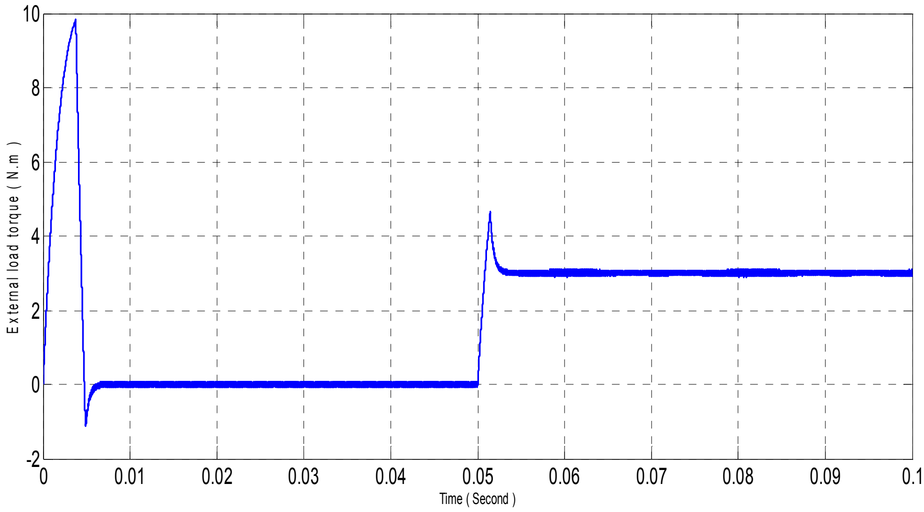


Figure (5.8): Variation of external load torque (75% from full load torque)

5.3 Step Response with (BF-PID)

Table (5.2) and figure (5.9) shows step response parameters of the closed loop system with BF-PID controller using ITSE based fitness function and without BF-PID controller. Figure (5.10) shows the effect of torque variation (50% from full load Torque) on the speed response using BFO at (t = 0.04) Sec. with recovery time = 3.52 m Sec., figure (5.11) illustrates variation of external load torque (50% from full load torque).

Table (5.2): BF-PID controller parameters

BF (PID) Controller	K_p	K_i	K_d	Peak time (t_p)	Rise time (t_r)	Settling time (t_s)	Max. over shoot M_p %	Steady state error $e_{s.s}$
Without	1	0	0	0.0055	0.0030	0.0163	13.956	0.5248
With	5.4936	0.9833	0.0034	0.0769	0.0030	0.0053	0	0.0561

Where:

- p (Dimension of the search space) = 3 , S (Total number of bacteria) = 50,
- N_c (Number of chemotactic steps) = 5 , N_s (Swimming length) = 4 ,
- N_{re} (Number of reproduction steps) = 10 , i (No. of iterations) = 100
- $c(i)$ Size of the step = 8.0e-007 , ITSE = 7.259
- N_{ed} (Number of elimination-dispersal events) = 2
- p_{ed} (Elimination –Dispersal probability) = 0.25

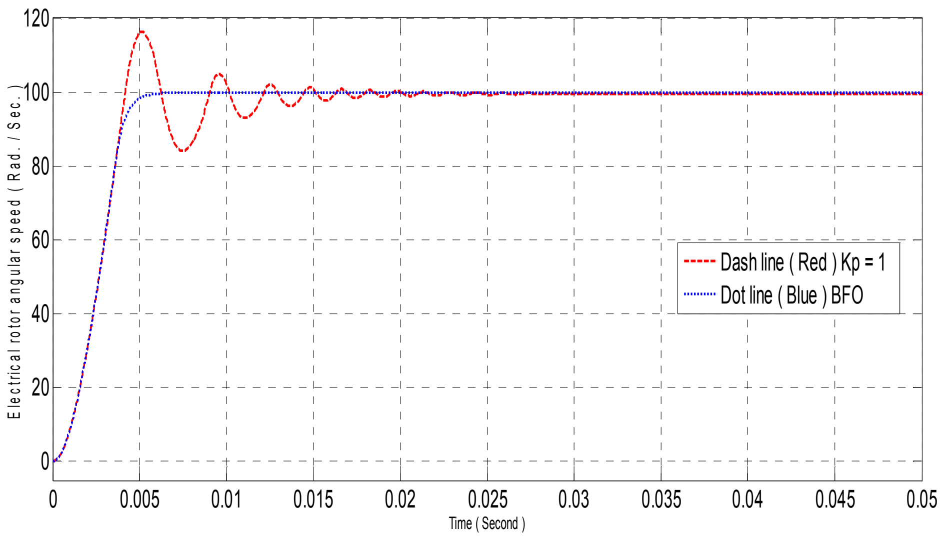


Figure (5.9): Step response of the closed loop system with BF-PID controller using ITSE based fitness function and without BF-PID controller

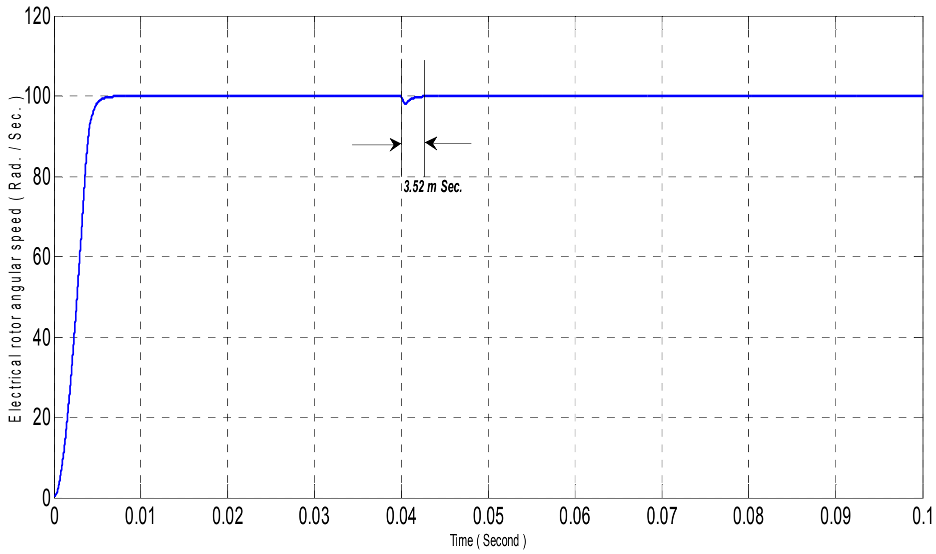


Figure (5.10): Effect of torque variation (50% from full load Torque) on the speed response using BF

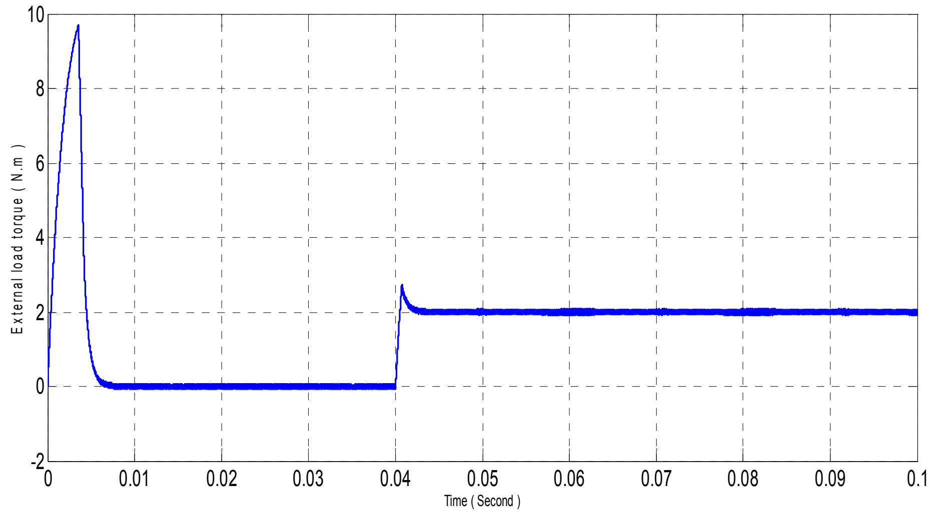


Figure (5.11): Variation of external load torque (50% from full load torque)

Figure (5.12) shows step response parameters of the closed loop system with (PSO, BF) PID controller using ITSE based fitness function and without (PSO, BF) PID controller.

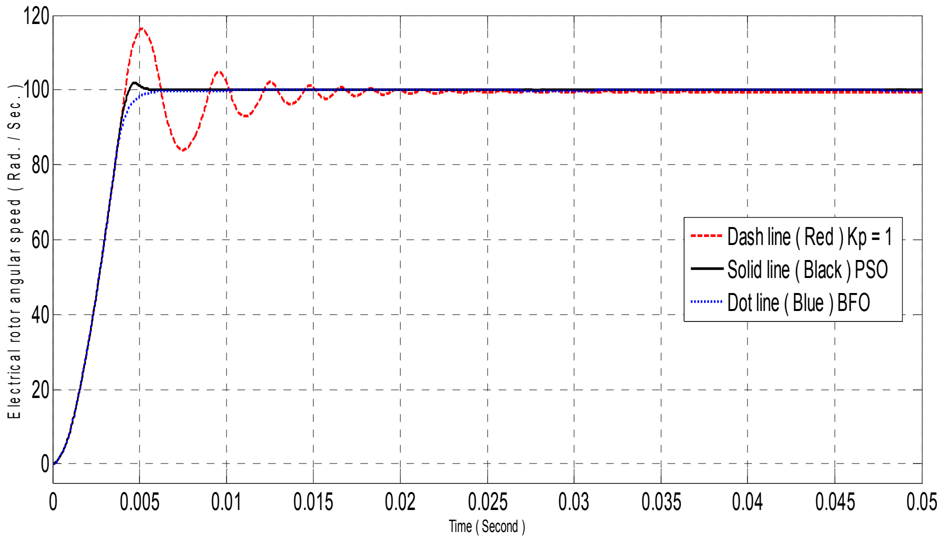


Figure (5.12): Step response of the closed loop system with (PSO, BF) PID controller using ITSE based fitness function and without (PSO, BF) PID controller

5.4 Speed Tracking Control

Figure (5.13) shows the variations in reference speed from (50-100-50) Rad. / Sec., figure (5.14) shows output speed follows up the variable reference without using controller.

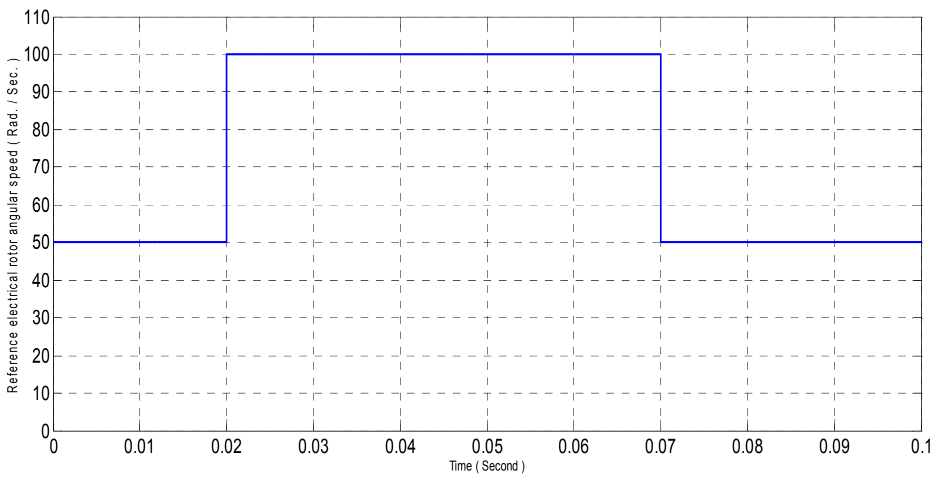


Figure (5.13): Variable reference speed (Rad./Sec.)

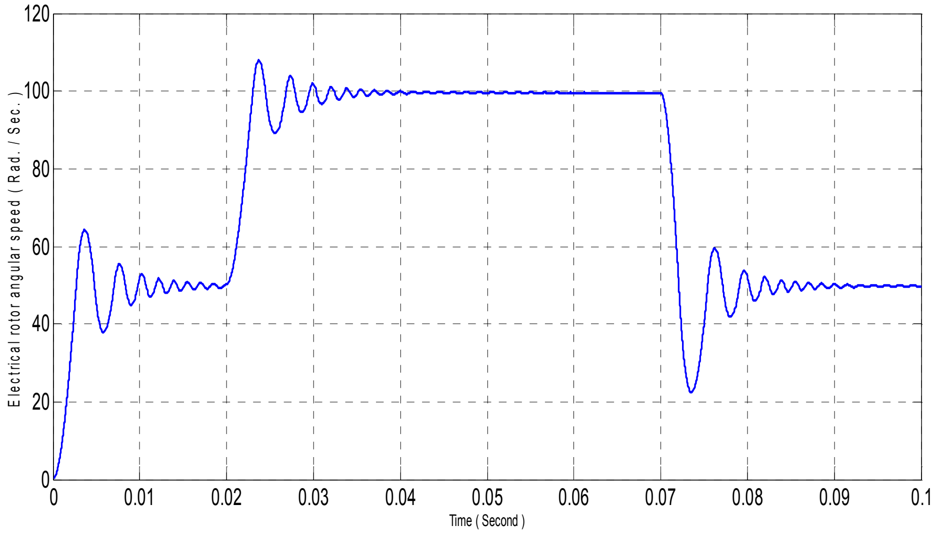


Figure (5.14): Variation of electrical rotor angular speed (Rad./Sec.) without using controller

Figure (5.15) shows output speed follows up the variable reference using (PID-PSO) controller, figure (5.16) shows output speed follows up the variable reference (PID-BFO) using controller.

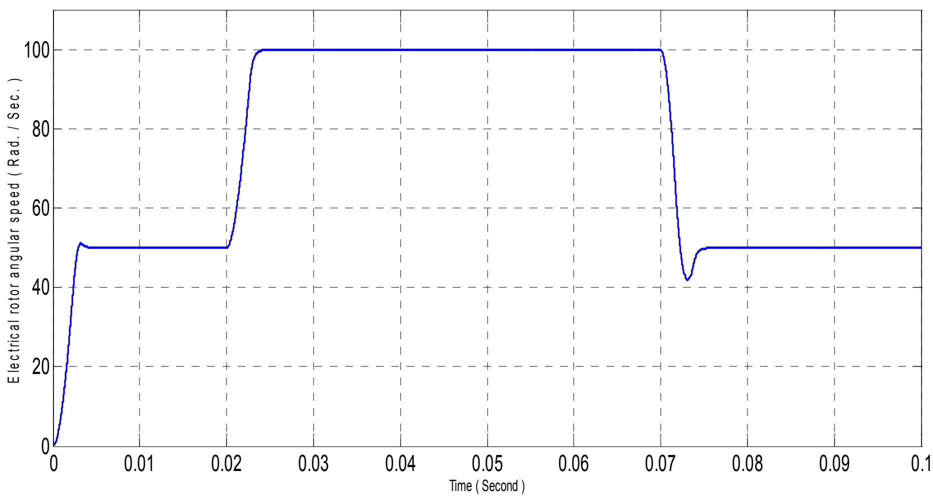


Figure (5.15): Variation of Electrical rotor angular speed (Rad. / Sec.) according to variable reference by using (PSO-PID) controller

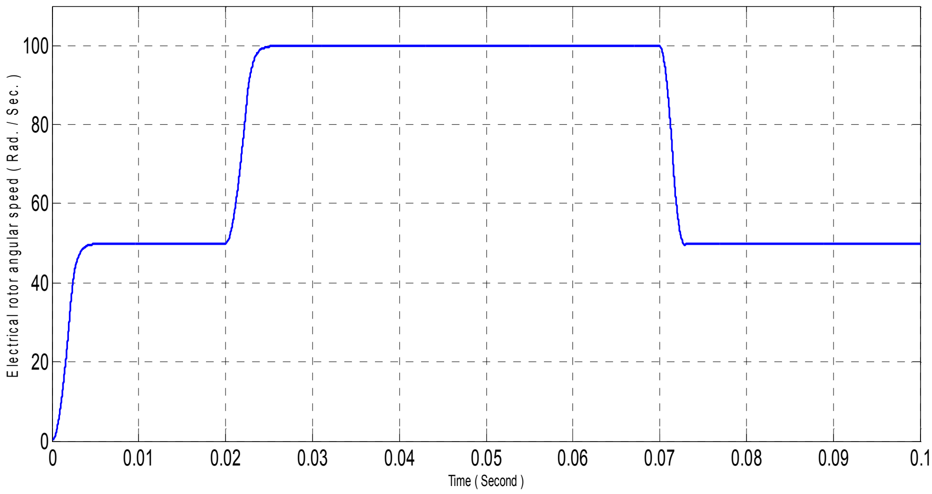


Figure (5.16): Variation of Electrical rotor angular speed (Rad. / Sec.) according to variable reference by using (BF-PID) controller

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

From the results presented in this thesis, it is believed that all of the project objectives have been fulfilled. The main conclusions and thesis contributions are detailed in the following points:

- Permanent magnet brushless (PMBL) motors are the latest choice of researchers due to their high efficiency, silent operation, compact size, high reliability and low maintenance requirements. These motors are preferred for numerous applications.

- PMBL motors find applications in diverse fields such as domestic appliances, automobiles, transportation, aerospace equipment, power tools, toys, vision and sound equipment and healthcare equipment ranging from microwatt to megawatts. Advanced control algorithms and ultra fast processors have made PMBLDC motors suitable for position control in machine tools, robotics and high precision servos, speed control and torque control in various industrial drives and process control applications.

- The research work presented in this thesis can be divided into two main parts. The first part is the comparative study and mathematical modeling of the brushless DC (BLDC) motor and its drive circuit, PID speed controller, also Traditional and Intelligent Optimization algorithms while the second part comprises the simulation results of the PID controller optimized by PSO and BFO using ITSE as objective function.

- This thesis introduced PSO and BFO as intelligent tools for tuning of PID controller, which eliminates the need for an experienced control system designer and always provides feasible optimum solution. In this work, a comparison study of using

PSO and BFO methods for the tuning of PID – controller for speed control of a BLDC motor. Obtained through simulation of BLDC motor, the simulation results show that the proposed controller can perform an efficient search for the optimal gains of PID controller. By comparing between PSO method and BF technique, it shows that, from point of view decreasing the steady state-error, settling time and peak time, the PSO is better than BFO while from point of view decreasing the max. overshoot and good performance of speed tracking control, the BFO can improve the dynamic performance of the system. However, the key equations of the PSO algorithm are still easier to implement in programming and also the seeking time for the same iterations is shorter.

6.2 Future work

The following points are candidates for investigation in the near future

- Implementation of an adaptive fuzzy logic control technique for brushless motor control.
- Applying the Ant Colony optimization approach in the tuning of PID controller.
- Adabting sophisticated control strategies such as neural network and neuro-fuzzy control techniques.

References

[1] Mehdi Nasri, Hossein Nezamabadi-pour, and Malihe Maghfoori. (20 April 2007)" A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor '". Proceeding of World Academy of Science on Engineering and Technology, pp. 211- 215.

[2] M.V.Ramesh, J.Amarnath, S.Kamakshaiah, B.Jawaharlad and G.S.Rao. (2011)"Speed Torque Characteristics Of Brushless DC Motor In Either Direction On Load Using ARM Controller", Vol.2 No.1, pp.217-222.

[3] A. Elwer and S. Wahsh. (15-18 Dec. 2007)"Improved Performance of Permanent Magnet Synchronous Motor by Using Particle Swarm Optimization Techniques", IEEE International Conference on Robotics and Biomimetics, pp. 2095 – 2100 .

[4] Dereje Shibeshi. (October 2007)"DSP Based Field Weakening Control Of PMSM ", Addis Ababa University, School of Graduate Study, Faculaty of Technology.

[5] P. Cominos and N.Munro. (January 2002)"PID Controllers: Recent Tuning Methods and Design to Specification", IEE Proceeding on Control Theory Applicationss, Vol. 149, Issues 1, pp. 46-53

[6] Zwe-Lee Gaing. (2004)"A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System", IEEE Transactions on Energy Conversion, Vol. 19, Issues 2, pp. 384-391

[7] J. G. Ziegler, N. B. Nichols. (1942)"Optimum Settings for Automatic Controllers",Transactions of ASME, Vol. 64, pp. 759-768.

[8] C. C. Hang, K.J. Astrom and, Q. G. Wang. (2002)"Relay Feedback Auto-tuning of Process Controllers", A Tutorial Review, Journal of Process Control, Vol. 12, pp.143-162.

[9] K. J. Astrom and T. Hagglund. (1984)"Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins",Automatica, Vol. 20, pp. 645-651.

[10] Katsuhiko Ogata. Modern Control Engineering." University of Minnesota , prentice Hall,Upper Saddlle River,New Jersey 07458 ,Fourth Edition.

[11] A. Moreno, J. Jullve, S. Silvestre and L. Castaner, (2000) " A Fuzzy Logic Controller for Stand Alone PV system ", Photovoltaic Speacialists Conference. Confernce record of the twenty-eighth IEEE.

[12] M.Massoum and B.A.Meroufel. (2009)“Model Reference Adaptive Fuzzy for Permanent Magnet Synchronous Motor”, Midiamira science Publisher, Actaelectrotechnica,Volume 50, No.1

[13] G. Tzafestas, and N. O. Papanikolopoulos. (1990)“Incremental Fuzzy Expert PID Control”. In Proceeding of IEEE Transactions on Industrial Electronics, Vol. 31, Issues 5, pp. 365-371.

[14] B. Porter and A. H.Jones. (1992)“Genetic Tuning of Digital PID Controllers”, Electronics Letters, Vol. 28, Issues 9, pp.843-844

[15] Rong-Fong Fung and Chih-Cheng Kao,. "Design of Self-Tuning PID Control in a Mechanisms System", Department of Mechanical and Automation Engineering National Kaohsiung ,Department of Electrical Engineering, Kao Yuan Institute of Technology, pp. 227-234.

[16] J. Kennedy and R. A.Eberhart. (1995)"Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks,IEEE, pp. 1942-1948.

[17] Abido, M.A., (2001) “Particle Swarm Optimization for Multi-Machine Power System Stabilizer Design “. In: Proceeding of Power Engineering Society Summer Meeting, Vol. 3, pp. 346-349

[18] Sambarta Dasgupta, Swagatam Das, Ajith Abraham , and Arijit Biswas. (2009)"Adaptive Computational Chemotaxis in BacterialForaging Optimization: An Analysis", IEEE Transactions on Evalutionary Computation, Vol. 13, No.4, pp. 919- 941

[19] Dong Hwa Kim, Ajith Abraham and Jae Hoon Cho. (2007)" A Hybrid Genetic Algorithm And Bacterial Foraging Approach For Global Optimization", pp. 3918– 3937

[20] C. C. Chan and K. T. Chau. (February 1997) “An Overview of Power Electronics in Electric Vehicles”, IEEE Trans. onIndustrial Electronics, Vol. 44, No. 1, pp. 3-13

[21] Yasser Said Mahmoud. (2009)“Modern Control Strategies of Electric Machines”, Military Technical College Chair of Electrical Power & Energy, Cairo

[22] Haibing Hu, Qingbo Hu, Zhengyu Lu and Dehong Xu. (2005) ‘’Optimal PID Controller Design in PMSM Servo System Via Particle Swarm Optimization”, Annual conference of IEEE on Industrial Electronic Society, pp. 79 – 83.

[23] Shiyoung Lee. “ A Comparison Study Of The Commutation Methods For The Three-Phase Permanent Mafnet Brushless DC Motor. Pennsylvania State University

Berks Campus, Reading, PA 19610-6009.

[24] K. Y. Lee and M.A.El-Sharkawi. "Modern Heuristic Optimization Techniques with Applications to Power Systems", New Intelligent Systems Technologies Working Group, Intelligent System Applications Subcommittee, Power System Analysis, Computing and Economics, IEEE Power Engineering Socity. Department of Electrical Engineering, university of Washinghton, Seattle, WA 98195-2500

[25] M.V.Ramesh, J.Amarnath, S.Kamakshaiah, B.Jawaharlad and G.S.Rao. (2011)"Speed Torque Characteristics Of Brushless DC Motor In Either Direction On Load Using ARM Controller", Vol.2 No.1, pp.217-222.

[26] Singh, Bhim Singh and Sanjeev. (June 2008)" State of the Art on Permanent Magnet Brushless DC Motor Drives". Electrical Engineering Department, Indian Institute of Technology, Delhi, New Delhi, India.

[27] S.Jong, H. L.Jung, C.Sekyo and J. G.Myung. (October 1993)"A robust digital position control of brushless DC motor with dead beat load torque observe",IEEE transactions on industrial electronics,Vol. 40, No. 5, pp. 512-520.

[28] C.C. Chan, Fellow, J.Z.Jiang, W. Xia, and K.T. Chau. (September 1995) ''Novel Wide Range Speed Control of Permanent Magnet Brushless Motor Drives" IEEE Transaction On Power Electronic, Vol. 10 ,No.5, pp. 539-546.

[29] Chun-Yu Du and Gwo-Ruey Yu. (5-7 Sept. 2007)"Optimal PI Control of a Permanent Magnet Synchronous Motor Using Particle Swarm Optimization", Second International Conference on Innovative Computing, Information and Control, pp.255-258

[30] Christopher T Kilian. (2000)"Modern Control Technology: Components and Systems", Novato, CA: Delmar Thomson Learning.

[31] Bhim Singh Fellow, B P Singh Fellow and (Ms) K Jain. (June 2003)'' Implementation of DSP Based Digital Speed Controller for Permanent Magnet Brushless dc Motor", IE(I) Journal-EL, Vol. 84, pp. 22-27.

[32] E. Polak and D. Q.Mayne. (1979)"AI Algorithm for Optimization Problems with Functional Inequality Constrains", IEEE Transactions on Automatic Control, Vol. 21, Issues 2, pp.184-193.

[33] W. Gesing and E. J.Gavison. (1979)"An Exact Penalty Function Algorithm for Solving General Constrained Parameter Optimization Problem", Automatica, Vol.15, pp.175-188.

[34] Bijay Kuma and Rohtash Dhiman. (November 2011)"Optimization of PID Controller for liquid level tank system using Intelligent Techniques", Canadian Journal on Electrical and Electronics Engineering Vol. 2, No. 11, pp 531-535.

[35] B.Nagaraj, S.Subha and B.Rampriya. (April 2008)"Tuning Algorithms for PID Controller UsingSoft Computing Techniques",Kamaraj College of engg and technology, IJCSNS International Journal,Virudhunagar, India,Vol.8, No.4, pp.278- 281.

[36] N.G.Pavlidis, K.E. Pasopouos, M. N. Vrahatis,(2005),"Computing Nash Equilibria through Computional Inteeligence Methods", pp. 113-136

[37] Parsopoulos KE, Varahatis MN, (2002b), " Particale Swarm Optmization Method in Multi-objaective Problems ". In: Proceeding of the ACM Symposium on Applied Computing (SAC), pp. 603-607

[38] R.C.Eberhart, Y.Shi., (2000) "Comparing Inertia Weights and Constriction Factors in Particlw Swarm Optmization ". In, Proceeding of IEEE International Congress on Evolutionary Computation , pp 84 -88

[39] H. Hjalmarsson and T. Birkeland. (1998) " Iterative feedback tuning of linear time-invariant mimo systems " In 37th IEEE Conference on Decision and Control, pages 3893–3898

[40] Bergh, F. and Engelbrecht, (2002), " Anew Locally Convergent Particle SwarmOptimizer", Conference on systems, Man and cybernetics. pp. 96-101

[41] Y. Shi and R. C.Eberhart. (1999)"Empirical Study of Particle Swarm Optimization", IEEE International Conference on Evolutionary Computation, pp.1945–1950.

[42] Xu-zhou Li, Fei Yu and You-bo Wang. (15-19 Dec. 2007)" PSO Algorithm Based Online Self-Tuning of PID Controller", International Conference on Computational Intelligence and Security, pp. 128 – 132.

[43] Shih-Feng Chen. (19-22 Aug. 2007)" Particle Swarm Optimization for PID Controllers with Robust Testing", International Conference on Machine Learning and Cybernetics, pp. 956 – 961

Appendix (A)

Implemented Algorithm (PSO)

```
clear

clc
iterations = 20;
inertia = 1;
correction_factor = 2;
swarm_size = 10;

% ---- initial swarm position -----
index = 1;
for i = 1 : 1
   for n = 1 : 5
      for j = 1 : 4
      swarm(index, 1, 1) = n;
      swarm(index, 1, 2) = i;
      swarm(index, 1, 3) = j;
      index = index + 1;
      end
   end
end

swarm(:, 4, 1) = 10000;          % best value so far
swarm(:, 2, :) = 0;             % initial velocity

%% Iterations
for iter = 1 : iterations

   for i = 1 : swarm_size
         %-- evaluating position
      swarm(i, 1, 1) = swarm(i, 1, 1) + swarm(i, 2, 1)/1.3;     %update x position
      swarm(i, 1, 2) = swarm(i, 1, 2) + swarm(i, 2, 2)/1.3;     %update y position
      swarm(i, 1, 3) = swarm(i, 1, 3) + swarm(i, 2, 3)/1.3;     %update z position


if(swarm(i, 1, 1)<0)
   swarm(i, 1, 1)=0.1;
end

if(swarm(i, 1, 2)<0)
   swarm(i, 1, 2)=0.5;
end
if(swarm(i,1,3)<0)
```

```
    swarm(i,1,3)=0.4;
end


        kp = swarm(i, 1, 1);
        ki = swarm(i, 1, 2);
        kd = swarm(i, 1, 3);

    sim BLDC.mdl;
iter
i
        val = min(ISTE);      % fitness evaluation (you may replace this objective function

                                                          with any function having a global minima)

        if val < swarm(i, 4, 1)                  % if new position is better
            swarm(i, 3, 1) = swarm(i, 1, 1);     % update best x,
            swarm(i, 3, 2) = swarm(i, 1, 2);     % best y postion
            swarm(i, 3, 3) = swarm(i, 1, 3);     % best z position
            swarm(i, 4, 1) = val;                % and best value
        end
    end

    [temp, gbest] = min (swarm(:, 4, 1));        % global best position

    %--- updating velocity vectors
    for i = 1 : swarm_size

        swarm(i, 2, 1) =rand*inertia*swarm(i, 2, 1) + correction_factor*rand*(swarm(i, 3,
1) - swarm(i, 1, 1)) + correction_factor*rand*(swarm(gbest, 3, 1) - swarm(i, 1, 1));   %x
velocity component

        swarm(i, 2, 2) = rand*inertia*swarm(i, 2, 2) + correction_factor*rand*(swarm(i, 3,
2) - swarm(i, 1, 2)) + correction_factor*rand*(swarm(gbest, 3, 2) - swarm(i, 1, 2));   %y
velocity component

        swarm(i, 2, 3) = rand*inertia*swarm(i, 2, 3) + correction_factor*rand*(swarm(i, 3,
3) - swarm(i, 1, 3)) + correction_factor*rand*(swarm(gbest, 3, 3) - swarm(i, 1, 3));   %z
velocity component

    end
  % t(iter)=swarm(gbest, 3);
  t(iter)=temp;
    tx(iter)=iter;
      %% Plotting the swarm
    clf
```

```
 %plot(iter,ISTE)
 grid on
  plot(swarm(:, 1, 1), swarm(:, 1, 2),'x')   % drawing swarm movements
   axis([-5 5 -5 5]);
 pause(.1)
end

x(1,1)=swarm(gbest, 3, 1);
x(1,2)=swarm(gbest, 3, 2);
x(1,3)=swarm(gbest, 3, 3);

kp=swarm(gbest, 3, 1);
ki=swarm(gbest, 3, 2);
kd=swarm(gbest, 3, 3);

kp
ki
kd
```

Implemented Algorithm (BFO)

```
tic
%Initialization
clear all
clc
p=3;                     % dimension of search space
s=50 ;          %20            % The number of bacteria
Nc=4;           %5          % Number of chemotactic steps
Ns=2;           %4          % Limits the length of a swim
Nre=10;                    % The number of reproduction steps
Ned=2;                     % The number of elimination-dispersal events
Sr=5;                    % The number of bacteria reproductions (splits) per generation
Ped=0.25;                  % The probabilty that each bacteria will be eliminated/dispersed
i=100;                    % number of itration
c(:,1)=0.000000008*ones(s,1);       % the run length
for i=1:100
for m=1:s                  % the initial posistions
     P(1,:,1,1,1)= 30*rand(s,1)';
%     P(2,:,1,1,1)= (0.005*rand(s,1)+0.007)';
     P(2,:,1,1,1)= 2*rand(s,1)';
     P(3,:,1,1,1)= 0.1*rand(s,1)';
end

%%
%Main loop
```

```
%Elimination and dispersal loop
for ell=1:Ned


%Reproduction loop


    for K=1:Nre

%  swim/tumble(chemotaxis)loop

      for j=1:Nc

          for i=1:s
              J(i,j,K,ell)=track_BG(P(:,i,j,K,ell));

% Tumble


              Jlast=J(i,j,K,ell);
              Delta(:,i)=(2*round(rand(p,1))-1).*rand(p,1);
              P(:,i,j+1,K,ell)=P(:,i,j,K,ell)+c(i,K)*Delta(:,i)/sqrt(Delta(:,i)'*Delta(:,i)); %
This adds a unit vector in the random direction

% Swim (for bacteria that seem to be headed in the right direction)

              J(i,j+1,K,ell)=track_BG(P(:,i,j+1,K,ell));
              m=0;        % Initialize counter for swim length
                 while m<Ns
                      m=m+1;
                      if J(i,j+1,K,ell)<Jlast
                        Jlast=J(i,j+1,K,ell);
P(:,i,j+1,K,ell)=P(:,i,j+1,K,ell)+c(i,K)*Delta(:,i)/sqrt(Delta(:,i)'*Delta(:,i)) ;
                        J(i,j+1,K,ell)=track_BG(P(:,i,j+1,K,ell));
                      else
                        m=Ns ;
                      end

                 end
              J(i,j,K,ell)=Jlast;
              sprintf('The value of interation i %3.0f ,j = %3.0f  , K= %3.0f, ell= %3.0f' , i, j,
K ,ell );

          end % Go to next bacterium
```

```
        end  % Go to the next chemotactic

%Reprodution
        Jhealth=sum(J(:,:,K,ell),2);            % Set the health of each of the S bacteria
        [Jhealth,sortind]=sort(Jhealth);         % Sorts the nutrient concentration in order of
ascending
        P(:,:,1,K+1,ell)=P(:,sortind,Nc+1,K,ell);
        c(:,K+1)=c(sortind,K);                  % And keeps the chemotaxis parameters with
each bacterium at the next generation

%Split the bacteria (reproduction)
            for i=1:Sr
                P(:,i+Sr,1,K+1,ell)=P(:,i,1,K+1,ell); % The least fit do not reproduce, the most
fit ones split into two identical copies
                c(i+Sr,K+1)=c(i,K+1);
            end
      end %  Go to next reproduction


%Eliminatoin and dispersal
        for m=1:s
            if  Ped>rand % % Generate random number
                P(1,:,1,1,1)= 20*rand(s,1)';
%                   P(2,:,1,1,1)= (0.005*rand(s,1)+0.007)';
                P(2,:,1,1,1)= 5*rand(s,1)';
               P(3,:,1,1,1)= 1*rand(s,1)';
            else
                P(:,m,1,1,ell+1)=P(:,m,1,Nre+1,ell); % Bacteria that are not dispersed
            end
        end
end % Go to next elimination and disperstal

%Report
           reproduction = J(:,1:Nc,Nre,Ned);
           [jlastreproduction,O] = min(reproduction,[],2);  % min cost function for each
bacterial
           [Y,I] = min(jlastreproduction);
           pbest=P(:,I,O(I,:),K,ell);
           Kp=pbest(1,:)
           Ki=pbest(2,:)
           Kd=pbest(3,:)
toc
         end
```

Appendix (B)

Simulated (BLDC) Parameters

- Volt = 24 V (DC)
- Power = 52 Watts
- Nominal speed = 200 rpm
- Nominal current = 4.2 A
- No. of poles (p) = 4
- R (Phase stator resistance) = 0.6 Ω
- λ_m (The amplitude of flux linkage) = 0.105 Wb. Turns
- L (Self inductance)
- M (Mutual inductance)
- ($L - M$) = 1.5 m H
- D (Viscous coefficient) = zero
- J (Moment of Inertia) = 1.42 * 10^{-4} Kg.m^2
- T (Full Load torque) = 4 N.m

وفى هذه الرسالة تم إستخدام المتحكم التفاضلى التناسبى التكاملى للتحكم فى سرعة محرك بمغناطيس دائم وبدون فرش كربونية والذى يحرك الحمل. وقد تم استخدام طريقة تجمهر الجزيئات وتقنية البكتريا الباحثة عن الطعام وذلك لضبط معاملات المتحكم فى السرعة للوصول إلى استجابة محددة. وتستند هذه التقنيات على تقليل ما يسمى بدالة التقييم أو دالة الهدف . و تحتوي الرسالة علي ستة فصول و فيما يلي ملخصا لمحتويات كل فصل.

الفصل الأول:

الهدف من الرسالة مع شرح بعض التفصيلات عن المحرك ذو المغناطيس الدائم عديم الفرش الكربونية و كذلك نبذة عن المتحكمات من نوع التناسبي/التكاملي/التفاضلي و بعض التعديلات التي أدخلت عليه و طرق ضبطه بالأضافة إلي موجز لمحتويات فصول الرسالة.

الفصل الثاني:

أنواع عناصر الإستشعار المستخدمة في أنظمة التحكم للمحركات عديمة الفرش الكربونية و منها أجهزة قياس الموضع و السرعة و كذلك مشغلات نظم التحكم و تفصيلات إضافية عن دوائر التشغيل و التحكم المختلفة.

الفصل الثالث:

إستنباط النموذج الرياضي للمحرك و دراسة خصائصه المختلفة.

الفصل الرابع:

الطرق المستخدمة في الضبط الأمثل للمتحكمات مع إستعراض طريقتي PSO & BFO و كيفية إستخدامهما مع المتحكم السابق الذكر.

الفصل الخامس:

عمل محاكاة لنظام التحكم في المحرك بإستخدام طريقتي حشد الجزيئات و البكتيريا و تحليلها مع عمل مقارنات للإستجابة الزمنية تحت ظروف تشغيل مختلفة.

الفصل السادس:

الإستنتاجات من الرسالة و كذلك التوصيات للأعمال البحثية المستقبلية.

ملخص الرسالة

نظرا لتزايد الطلب على المحركات المدمجة والموثوق بها وتطور مفاتيح أشباه الموصلات الكهربية ومواد المغناطيس الدائم الأقل تكلفة ، أصبحت المحركات بدون الفرش الكربونية أكثر أهمية لإستخدامها فى الأجهزة والسيارات والمنتجات الازمة للخدمات الطبية والعسكرية وصناعة الطيران والفضاء. وعلى عكس محركات التيار المستمر ذات الفرش الكربونية،فان كل محرك بدون فرش كربونية يحتاج الى "دائرة تشغيل" لتغذية التيارات الكهربية الى ملفات المحرك والتى تكون متزامنة مع دوران العضو الدوار. وبعبارة اخرى، فان نوعا ما من أجهزة استشعار وضع العضو الدوار يكون ضرورى لتشغيل المحركات بدون الفرش الكربونية. وتضم محركات المغناطيس الدائم بدون فرش كربونية إصدارات موجية جيبية أو موجية مربعة. ويسمى أيضا محرك المغناطيس الدائم بدون الفرش الكربونية ذات الموجة الجيبية بالمحرك المتزامن ذات المغناطيس الدائم حيث يغذى بتيار موجة جيبية ويستخدم إشارات رجوع مستمرة لمواضع العضو المتحرك للتحكم فى التغذية. ومن ناحية اخرى، فإن محرك المغناطيس الدائم بدون فرش كربونية ذات الموجة المربعة يسمى ايضا بمحرك التيار المستمر ذات المغناطيس الدائم بدون فرش كربونية، ويغذى بتيار موجة مربعة ويستخدم إشارات رجوع متقطعة لمواضع العضو المتحرك للتحكم فى التغذية.

وفى هذه الرسالة يتم تقديم بعض التفاصيل حول محركات المغناطيس الدائم بدون فرش كربونية من وجهة نظر التصنيفات، والمزايا، والبناء والتشغيل ومقارنتها مع المحركات ذات الفرش الكربونية والمحركات الغير متزامنة. إن هذا النوع من المحركات لا يحتوي فقط على مزايا محرك التيار المتردد مثل القدرة على سرعة أفضل وعدم وجود عاكس التيار الميكانيكى ولكن أيضا لديه ميزة من محرك تيار متردد مثل بنية بسيطة ودرجة أعلى من الموثوقية وخلوه من الصيانة. وبالإضافة إلى ذلك، يحمل محرك التيار المستمر بدون الفرش الكربونية بعض المزايا حيث أنه أصغر حجماً ذو قوة عالية وبنية أساسية بسيطة ليتم إستخدامها على نطاق واسع في التطبيقات التي تحتاج إلى محرك عالي الأداء.

ومن وجهة نظر التحكم فى الآونة الأخيرة، فقد تم اقتراح العديد من منهجيات التحكم الحديثة مثل التحكم غير الخطى والتحكم الأمثل والتحكم الهيكلى المتغيروالتحكم التكيفى وذلك على نطاق واسع بالنسبة للمحركات المغناطيسية الدائمة بدون فرش كربونية. ومع ذلك فإن هذه الأساليب معقدة سواء في الأسس النظرية أو صعوبة تنفيذها.ويغطى التحكم التفاضلى التناسبى التكاملى بثوابته الوظيفية الثلاثة حلولا لكل من الاستجابة العابرة والحالة المستقرة حيث يقدم الحل الأبسط والأكثر فعالية حتى الآن لكثير من مشاكل التحكم العملية الحقيقية. ورغم البنية البسيطة ومتانة الأسلوب فإن الضبط المثالى لثوابت المتحكمات التفاضلية التناسبية التكاملية كانت صعبة للغاية.