Parameter Identification and Control of DC Motor Using Firefly Algorithm

Dissertation submitted in partial fulfilment of the requirement for the award of degree

of

Master of Engineering

in

Power Systems and Electric Drives

By:

Ankit Vashistha

(Roll No. 801141004)

Under Supervision of:

Mr. Souvik Ganguli

Assistant Professor, EIED

ELECTRICAL AND INSTRUMENTATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA-147004

JULY, 2013

CERTIFICATE

I hereby certify that the work which is being presented in this dissertation entitled **"Parameter Identification and Control of DC Motor using Firefly Algorithm"** in partial fulfillment of the requirement for the award of the degree of Master of Engineering in Power Systems & Electric Drives submitted in Electrical & Instrumentation Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under supervision of Mr. Souvik Ganguli, Assistant Professor, EIED.

The matter presented in this report has not been submitted for the award of any other degree of this or any other University.

Ankit Vashistha

Ankit Vashistha

Roll No. 801141004

It is certified that the above statement made by the student is correct to the best of my knowledge and brief.

Ganguli 12/07/13

Mr. Souvik Ganguli

Assistant Professor,

EIED, Thapar University

S. Ghosh

Dr. Smarajit Ghosh
Professor & Head, EIED
Thapar University
Patiala

Dr. S.K. Mohapatra
Sr. Professor & Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

I would like to express my deep gratitude to my project guide **Mr. Souvik Ganguli,** Assistant Professor, Electrical and Instrumentation Engineering Department (EIED), who has always been source of motivation and firm support for carrying out the work. I express my gratitude to **Dr. S. Ghosh**, Professor and Head, Electrical and Instrumentation Engineering Department (EIED) and **Ms. Manbir Kaur**, Associate Professor & PG coordinator for their invaluable suggestions and constant encouragement all through the work and also express my deep regards for **Mr. Shyam Lal Verma**, Lab Superintendent, EIED for carrying out the experimentation in EMEC Lab. I would also like to convey my sincerest gratitude and indebtedness to all other faculty members and staff of Electrical and Instrumentation Department (EIED), Thapar University, Patiala, who bestowed their great effort and guidance at appropriate times without which it would have been very difficult on our project work. Further, I would like to express my feeling towards my parents and God who directly or indirectly encouraged and motivated me during this work.

Ankit Vashistha

Ankit Vashistha

Roll No. 801141004

ABSTRACT

In this work, we have applied the firefly algorithm (FFA) for some non-linear benchmark functions to highlight the efficacy of the algorithm. There are two commonly used approaches to relate the output with its input: transfer function, state space representation. Then, a dc motor is modeled using armature resistance control and field control method by transfer function, state space representation approach. We have used the transfer function approach to identify the parameters of armature controlled dc shunt motor viz. armature resistance, armature inductance, moment of inertia, viscous friction, back emf/torque constant by proposed algorithm. Then, we validate our results from the experimental approach. Thereafter, we have controlled the dc motor using PI controller because of its better responsiveness and improvement of steady state response. There is a need to control its parameters according to our requirement. There are different classical tuning methods to tune the parameters of the controller viz. Ziegler-Nichols, Tyreus-Luyben, Astrom-Hagglund, etc. Then, we have applied proposed algorithm to tune the parameters of controller and compared its performance with the classical tuning methods.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background of Work

DC motor is generally used for the loads requiring adjustable speed, good speed regulations and frequent starting, braking and reversing. This inherent characteristic makes it useful in industries. Some important applications are rolling mills, hoists, machine tools, traction, printing presses, textile mills and cranes etc. Fractional horsepower DC motors are widely used as servo motors for positioning and tracking. Although, it is being predicted that AC drives will replace DC drives, however, even today the variable speed applications are dominated by DC drives because of lower cost, reliability and simple control of its speed. Moreover, the speed/torque characteristics of DC motor is much superior to AC motor. Mathematical modeling is one of the most important and often the most difficult step towards understanding a physical system. In modeling of DC motor, the aim is to find differential equation that relates the input data to output data. There are few parameters of the DC motor that are complicated to determine using practical measurements. Thus, we need parameter estimation techniques to estimate those parameters.

The term identification was first introduced by Zadeh (1956) as a generic expression for the problem of "determining the input-output relationships of a black box by experimental means." System identification is the art and science of building mathematical models of dynamic system from observed input-output data. It can be seen as the interconnection between the real world of applications and mathematical world of control theory to abstract their models. System identification can be elaborated with different techniques which depend on the character of the models to be estimated either it may be linear, nonlinear, hybrid or nonparametric model etc. System identification is, in one sense, a comparatively new concept, yet in another it is ancient. For example, if two persons are talking to each other, then we can predict the relationship between them just by observing their discussion and expressions. In other words, from their discussion and expressions we build a model to identify the relation between them. Similarly, from input-output data of a system we can build a model to identify the system. Model is extraction of mathematical description of the process. System identification has four steps to identify the model of the system: Data recording, Model set selection, Identification Criteria and Model validation. In recent years, system identification technique is used in many areas like engineering, economics, medicine and space technology etc. There are number of methods that have been applied earlier for DC motor parameter identification viz. Recursive least square estimation techniques [7, 9], Least square [9, 10], moment method [17], pasek method [18], algebraic parameter identification technique [19] and steepest descent gradient method etc. Due to some limitation in classical methods, the recent trend is to use intelligent meta-heuristic algorithm for DC motor identification viz. Bacterial Foraging Optimization Algorithm (BFOA) [28], Adaptive tabu search method [29], Radial Basis Function Neural Network (RBFNN) [30], Genetic Algorithm (GA) [38,40,41,43].

The two major components in modern meta-heuristic algorithms are: intensification and diversification or exploitation and exploration. For an algorithm to be efficient and effective, it must be able to generate diverse solutions including the potentially optimal solutions so as to explore the whole search space effectively, while intensification means to focus on the search in a local region by exploiting the information that a current good solution is found in this region. Diversification is often in the form of randomization term attached with a deterministic component in order to explore the search space effectively and efficiently, while intensification is the exploitation of previous solutions so as to select the potentially good solutions via elitism or use of memory or both. Diversification avoids the solution to be trapped at local minima. The effectiveness of any algorithm will depend upon the balancing of these two major factors. The Meta-heuristic algorithms are used everywhere to solve the optimization problems in diverse field viz. engineering, science, finance etc. because it can handle both continuous and discrete time variables and effectively deal with large complicated problems.

A new meta-heuristic algorithm, based on flashing light of fireflies, Firefly algorithm was developed by Xin-She Yang in 2007. Although the real purpose and the details of this complex biochemical process of producing the flashing light is still a debating issue in the scientific community, many researchers believe that it helps fireflies for finding their mates, protecting themselves from their predators and attracting their potential prey. In the firefly algorithm, the objective function of a given optimization problem is associated with its flashing light or light intensity which helps the swarm of fireflies to move to brighter and more attractive locations in order to obtain efficient optimal solutions. In the Firefly algorithm, the diversification is represented by the randomization component, while the intensification is implicitly controlled by the attraction of different fireflies, this might be an important factor for its success in solving multi-objective and multimodal optimization problems. Thus, we have applied this algorithm to optimize the parameters of the DC Motor. The next step is to control the performance of the motor for desired purpose.

Proportional-Integral-Derivative (PID) controller is the most commonly used controller in industrial applications. In process control, more than 95% of the control loops are PID, most of them are PI controller because of its relatively simple structure and implementation. Thus, we have used the PI controller for the control of DC motor.

1.2 Objective of Work

The dissertation fulfills the following objectives:

i. Firefly algorithm is discussed in details and used to optimize the value of different un-constraint, non-linear benchmark functions like Sphere function, Griewank function, Hyper-Ellipsoid function and Ackley function.

ii. Parameter Identification of DC motor is been carried out using proposed algorithm.

iii. PI controller tuning for DC motor is been carried out using Harmony search algorithm and compared its performance with classical tuning techniques.

1.3 Organization of the Dissertation

Chapter 2 discuss the Firefly Algorithm and its implementation on various unconstraint, non-linear standard functions like Sphere function, Ackley function, Hyper-Ellipsoid function and Greiwank function.

Chapter 3 covers the basic block diagram, mathematical model and overall transfer function of the separately excited DC motor and their servo and regulatory response.

Chapter 4 elaborates the parameter estimation of DC motor using the Firefly Algorithm (FFA).

Chapter 5 shows the comparison of DC motor controller tuning using conventional methods and Firefly Algorithm.

Chapter 6 concludes the dissertation and also discusses its future perspectives.

CHAPTER 2

FIREFLY ALGORITHM

2.1. Introduction

Optimization techniques are used to find the optimal values of objective function when all its associated constraints are satisfied. It can be classified into two main categories: deterministic and stochastic algorithms. Deterministic algorithms such as hill-climbing approach will produce the same set of solutions if the iterations start with the same initial guess. On the other hand, stochastic algorithms sometimes produce different solutions even with the same initial starting point. However, the final results, though slightly different, will usually converge to the same optimal solutions within a given accuracy.

Deterministic algorithms are almost all local search algorithms, and they are quite efficient in finding local optima. However, there is a risk for it to be trapped at local optima, while the global optima are out of reach. A common practice is to introduce some stochastic component to an algorithm so that it becomes possible to jump out of such locality and find the global optima. In this case, algorithms become stochastic.

Stochastic algorithms have a deterministic component and a random component associated with it. The stochastic component can take many forms such as simple randomization by randomly sampling the search space or by random walks [1]. Most stochastic algorithms can be considered as Meta heuristic and good examples are Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Firefly Algorithm (FFA), Harmony Search (HS) etc.

Bio-inspired meta-heuristic algorithms have recently become the forefront of the current research as an efficient way to deal with many optimization problems and non-linear optimization constrained problems in general. These algorithms are based on a particular successful mechanism of a biological phenomenon of nature in order to achieve global optimal solution. It works as follows: a population of individuals is randomly initialized where each individual represents a potential solution to the problem. The quality of each solution is then evaluated using a fitness function. Thereafter, a selection process is applied in order to form a new population. The searching process is biased towards the better individuals to increase their chances of being included in the new population. This procedure is repeated until convergence rules are met.

A new nature inspired algorithms is Firefly Algorithm (FFA) developed by Xin-She Yang in 2007, which is based on the flashing light of fireflies. The primary purpose of this flashing light is to act as a signal to attract other fireflies. In the firefly algorithm, the objective function of a given optimization problem is associated with its flashing light or light intensity which helps the fireflies to move to brighter and more attractive locations in order to obtain optimal solution [2].

2.2. Firefly algorithm

2.2.1. Firefly in Nature

The flashing light of fireflies is an amazing sight in the summer sky in the tropical and temperate regions. There are about two thousand firefly species, and most fireflies produce short and rhythmic flashes. The pattern of flashes is often unique for a particular species. The flashing light is produced by a process of bioluminescence, and the true functions of such signaling systems are still debating. However, two fundamental functions of such flashes are to attract mating partners (communication), and to attract potential prey. In addition, flashing may also serve as a protective warning mechanism [5].

We know that the light intensity at a particular distance r from the light source obeys the inverse square law. That is to say, the light intensity (I) decreases as the distance r increases ($I \propto 1/r^2$). Furthermore, the air absorbs light which becomes weaker and weaker as the distance increases. These two combined factors make most fireflies visible only to a limited distance, usually several hundred meters at night, which is usually good enough for fireflies to communicate. The flashing behavior can be formulated in such a way that it is associated with the objective function to be optimized. This makes it possible to formulate new optimization algorithms.

2.2.2. Firefly Algorithm

The Firefly Algorithm (FFA) is a meta-heuristic algorithm, inspired by the flashing behavior of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. Now this can idealize some of the flashing characteristics of fireflies so as to develop firefly-inspired algorithm. For simplicity in describing our new Firefly Algorithm (FFA), there are the following three idealized rules [3].

i. All fireflies are unisex, and they will move towards more attractive and brighter ones regardless their sex.
ii. The degree of attractiveness of a firefly is proportional to its brightness which decreases as the distance from the other firefly increases due to the fact that the air absorbs light. If there is not a brighter or more attractive firefly than a particular one, it will then move randomly.
iii. The brightness or light intensity of a firefly is determined by the value of the objective function of a given problem. For maximization problems, the light intensity is proportional to the value of the objective function.

In the firefly algorithm there are two important issues of the variation of light intensity and the formulation of the attractiveness. For simplicity, it is assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function of the optimization problems. The attractiveness function β(r) can be any monotonically decreasing functions such as the following generalized form of

$$\beta(\mathrm{r}) = \beta_0 e^{-\gamma {r_{ij}}^2}$$

Where r_{ij} is the distance between firefly i and j, β_0 is the attractiveness at r = 0 and γ is the light absorption coefficient. The distance between any two fireflies i and j at x_i and x_j respectively, is the Cartesian distance

$$r_{ij} = \sqrt{\sum_{k=1}^{dim} (x_{i,k} - x_{j,k})^2}$$

where, $x_{i,k}$ is the kth component of the spatial coordinate x_i of ith firefly.

The movement of a firefly i to another more attractive (brighter) firefly j is determined by

$$x_i = x_i + \beta_0 e^{-\gamma {r_{ij}}^2} (x_j - x_i) + \alpha \left(rand - \frac{1}{2}\right)$$

Where, the second term is due to the attraction while the third term is randomization with α being the randomization parameter. rand is a random number generator uniformly distributed in [0, 1]. For most cases in our implementation, we can take β_0 = 1 and $\alpha \epsilon$ [0, 1]. As we can see in basic firefly algorithm, the solutions are still changing as the optima are approaching. Thus, it is possible to improve the solution by reducing the randomness by introducing the variable δ.

2.2.3. Firefly Algorithm Variants

The basic firefly algorithm is very efficient, but we can see that the solutions are still changing as the optima are approaching. It is possible to improve the solution quality by reducing the randomness [3].

A further improvement on the convergence of the algorithm is to vary the randomization parameter α so that it decreases gradually as the optima are approaching. For example, we can use approaching. For example, we can use

$$\alpha = \alpha_\infty + (\alpha_0 - \alpha_\infty)e^{-t} \quad \text{... (2.1)}$$

where $t \in [0, t_{max}]$ is the pseudo time for simulations and t_{max} is the maximum number of generations. α_0 is the initial randomization parameter while α_∞ is the final value. We can also use the similar function to the geometrical annealing schedule. That is

$$\alpha = \alpha_0 \theta^t \quad \text{… (2.2)}$$

where, $\theta \in [0, 1]$ is the randomness reduction constant.

Algorithm

Step 1. Define objective function f(x).

Step 2. Assume the parameters of algorithm- α(randomness), β(attaractiveness), γ(light absorption coefficient), δ (randomness reduction parameter), number of fireflies, maximum iteration, stopping criteria.

Step 3. Generate initial population of fireflies x_i ,(i=1, 2,…n) randomly.

Step 4. Determine Light intensity I_i or function value at x_i by value of $f(x_i)$.

Step 5. Arrange the fireflies according to their light intensity.

Step 6. Set iteration count, iter = 1

Step 7. If I_i<I_j where i, j=1: n

$$x_i = x_i + \beta_0 e^{-\gamma {r_{ij}}^2}\left(x_j - x_i\right) + \alpha\left(rand - \frac{1}{2}\right)$$

Step 8. Find the value of function at new values of x_i.

Step 9. Check stopping criteria and iter < maximum iteration, if it is satisfied GOTO step 12.

Step 10. Modify randomness parameter α as

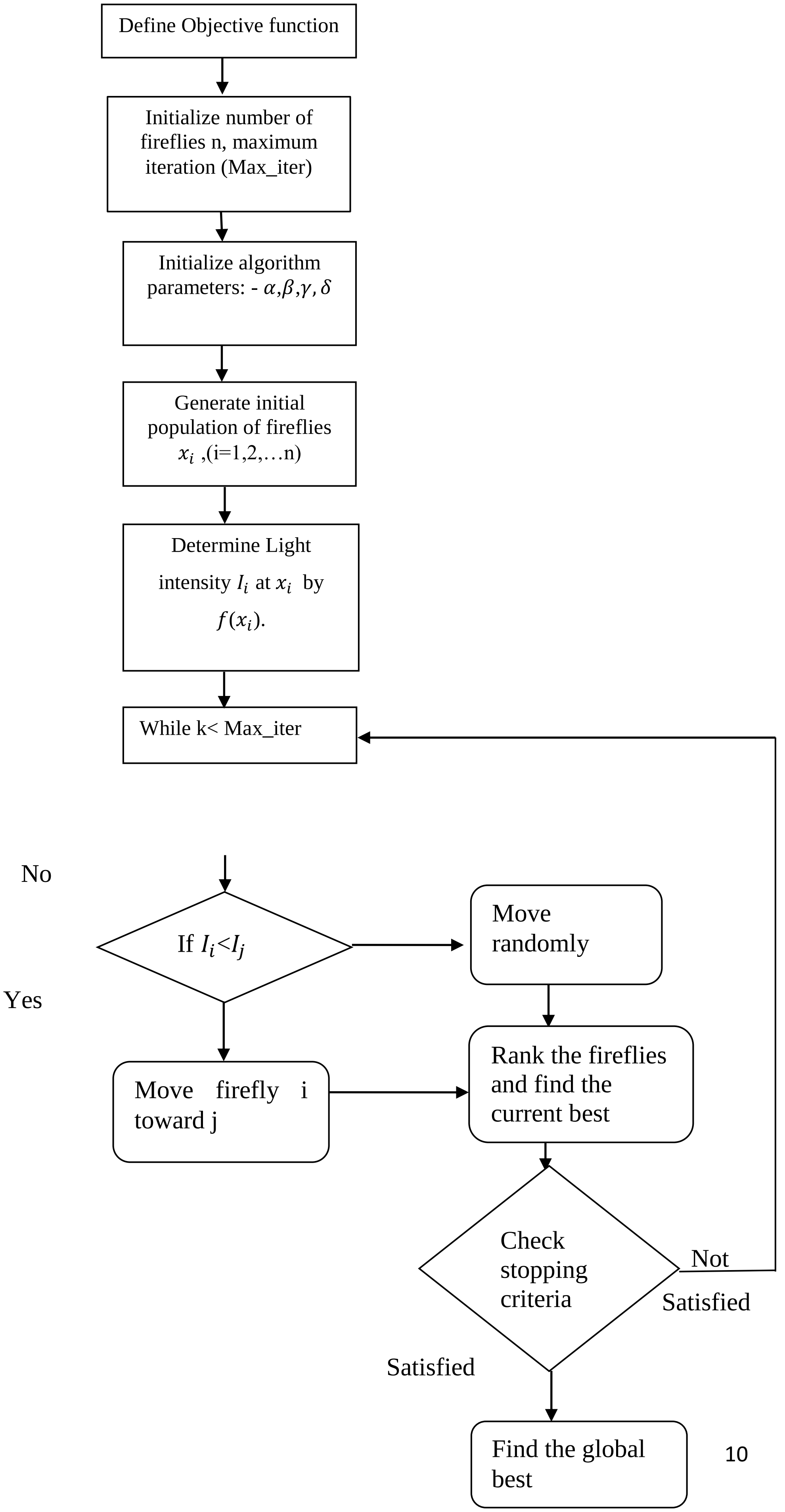
$$\alpha = \delta * \alpha.$$

Step 11. Advance the iteration count, iter = iter +1 and GOTO step 5.

Step 12. Find the optimal value of the function.

Step 13. Stop.

2.2.4. Flow chart of Firefly Algorithm



2.2.5. Advantages

Firefly Algorithm has following advantages:

1. It has high convergence rate.

2. Firefly works individually and finds a better position for itself in consideration with its current position as well as the position of other fireflies. Thus, it escapes from the local optima and finds a global optimum in less number of iterations.

3. It is used for convergence and cost minimization problems.

4. It is a robust algorithm.

2.2.6. Applications

Firefly algorithm (FFA) has following applications area:

1. Data Clustering.

2. Economic load dispatch.

3. Scheduling and travelling salesman problems.

4. Chemical phase equilibrium.

5. Nonlinear engineering optimization problems.

2.3. Standard functions

2.3.1. Sphere function

Sphere function is also known as De Jong's function since it is the first function of De Jong. It is one of the simplest benchmark functions. This function is continuous, unimodal and convex. It has following general definition [4].

$$f(x) = \sum_{i=1}^{N} {x_i}^2$$

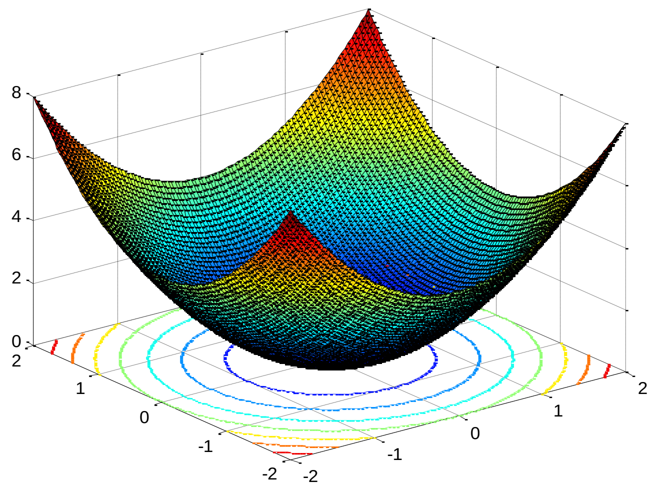


Figure 2.2 Two dimension plot for Sphere function

2.3.2. Greiwank's function

Greiwank function has many regularly distributed widespread local minima. Function has following definition [4].

$$f(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

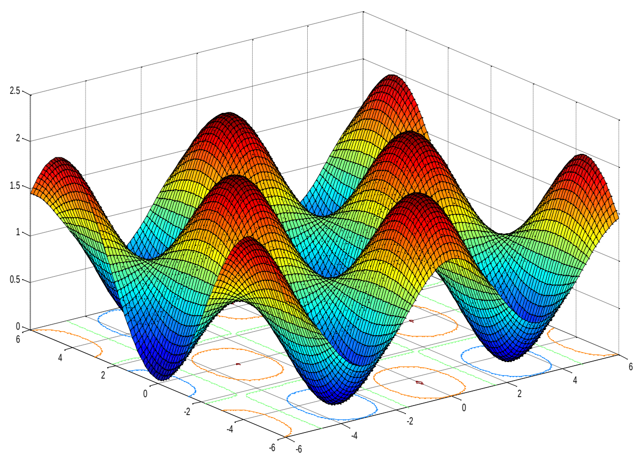


Figure 2.3 Two dimension plot for Griewank function

2.3.3. Rotated Hyper-Ellipsoid function

This function produces rotated hyper-ellipsoids. This function is continuous, convex and unimodal in nature. Function has the following definition [4].

$$\mathrm{f}\,(x) = \sum_{i=1}^{N} \left(\sum_{j=1}^{i} x_j\right)^2$$

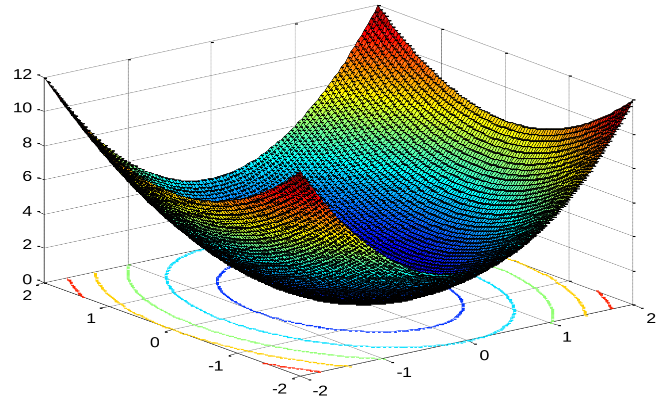


Figure 2.4 Two dimension plot for Hyper-Ellipsoid function

2.3.4. Ackley function

Ackley is widely used test function. It is most difficult function to optimize. The definition of Ackley function is as follows [4].

$$\mathrm{f}\,(x, y) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} {x_i}^2}\right) - \exp\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)}\right) + 20 + \exp(1)$$

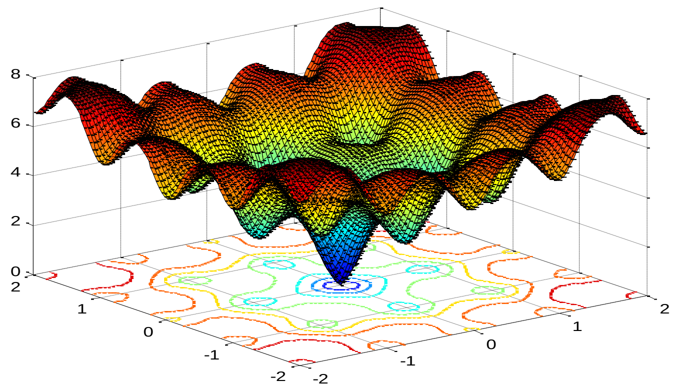


Figure 2.5 Two dimension plot for Ackley function

2.4. Results

1. Sphere function

The variation of function values for two dimensions with iterations is shown in Figure 2.6.

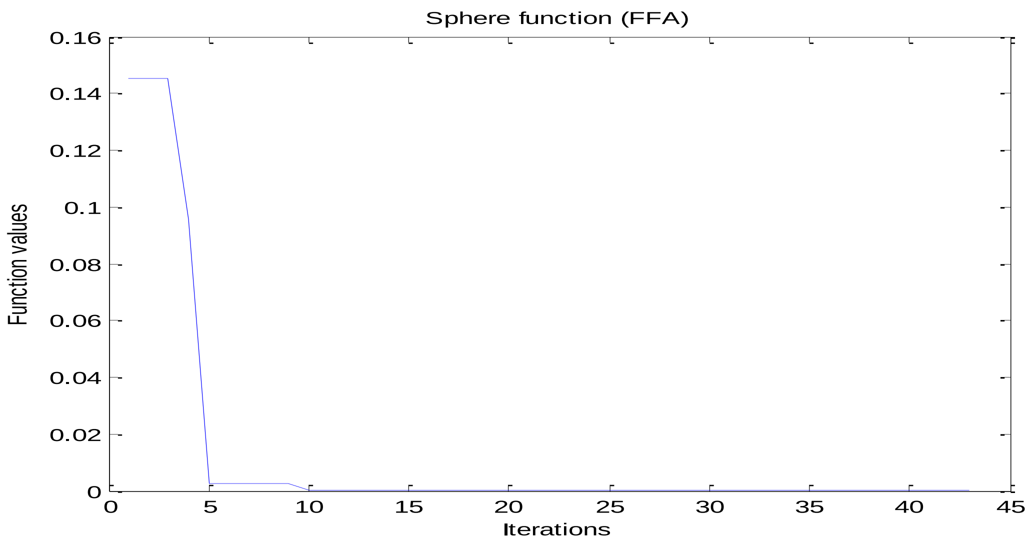


Figure 2.6 Variation of Sphere function values with iterations using FFA

2. Greiwank function

The variation of function values for two dimensions with iterations is shown in Figure 2.7.

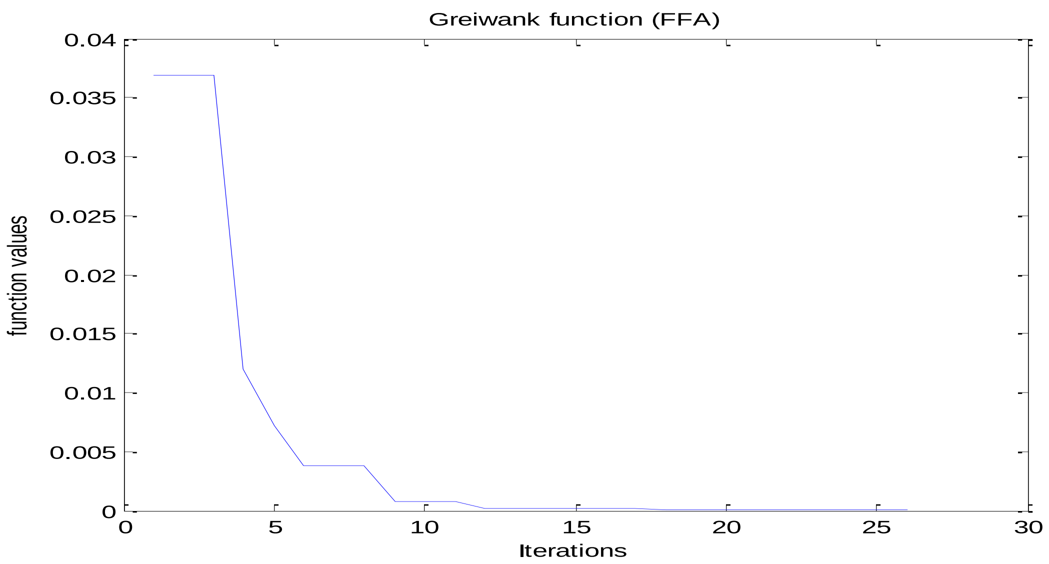


Figure 2.7 Variation of Griewank function values with iterations using FFA

3. Rotated Hyper-Ellipsoid function

The variation of function values for two dimensions with iterations is shown in Figure 2.8.

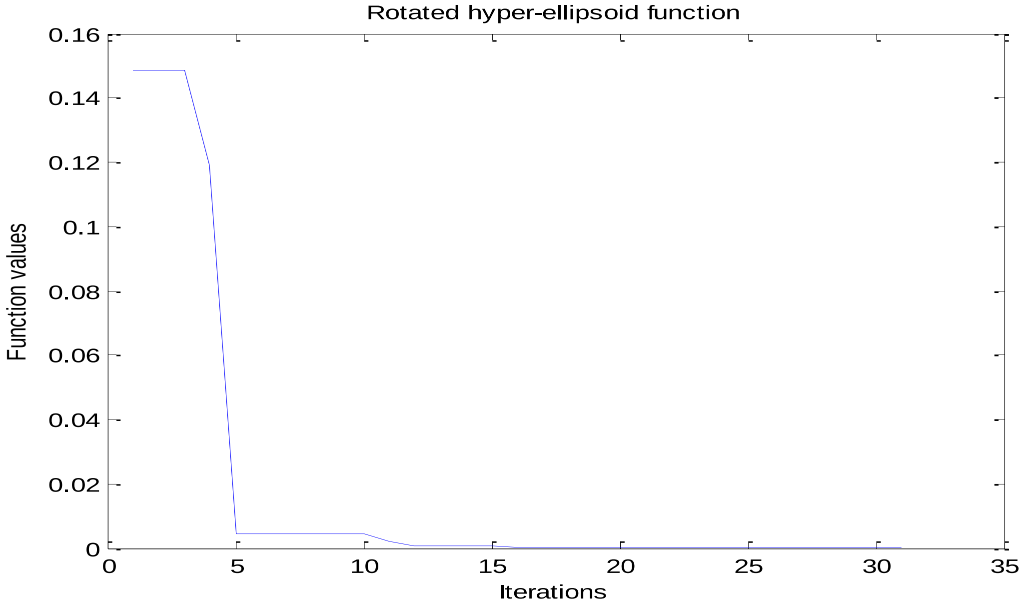


Figure 2.8 Variation of Rotated Hyper-Ellipsoid function values with iterations using FFA

4. Ackley function

The variation of function values for two dimensions with iterations is shown in Figure 2.9.

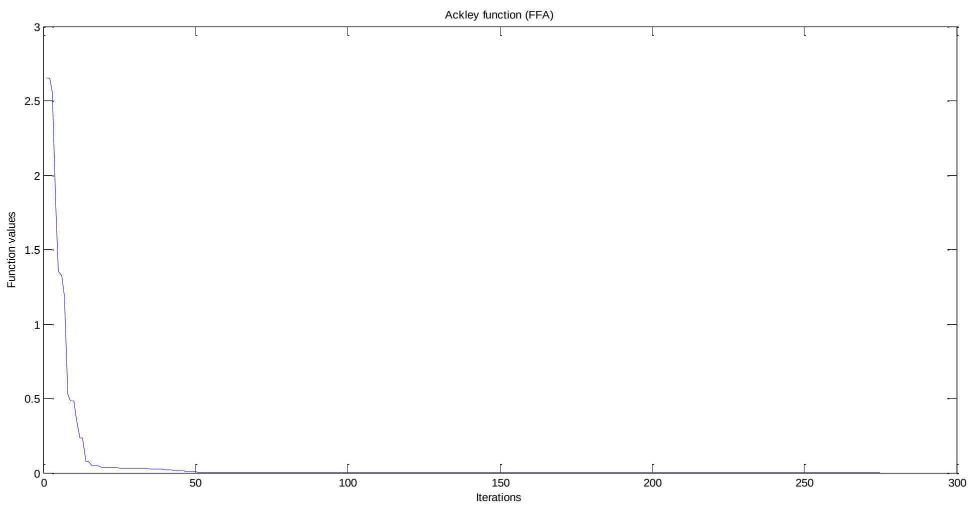


Figure 2.9 Variation of Ackley function values with Iterations using FFA

Table 2.1 Results for Firefly Algorithm (FFA) for different standard functions

Function	Dimension (N)	α	β	γ	δ	X_1	X_2	No. of fireflies	f_{min}	count	Time elapsed (sec.)
Sphere	2	0.2	1.0	1.0	0.97	-0.0015	0.0018	12	5.6903×10^{-6}	43	0.207728
Greiwank	2	0.2	1.0	1.0	0.97	-0.0016	-5.1923×10^{-4}	12	1.3595×10^{-6}	26	0.253399
Rotated Hyper-Ellipsoid	2	0.2	1.0	1.0	0.97	-7.1838×10^{-4}	-0.0029	12	9.3147×10^{-6}	31	0.264208
Ackley	2	0.2	1.0	1.0	0.97	1.9675×10^{-6}	-7.9613×10^{-8}	12	5.5697×10^{-6}	275	0.584791

2.5. Conclusion

In this chapter, we have studied the firefly algorithm (FFA) and applied it to optimize some non-linear benchmark functions viz. Sphere, Greiwank, Rotated Hyper-Ellipsoid and Ackley function. Thus, we can conclude that this algorithm provides optimum value or near optimum value of the given function in less number of iterations and consumes less amount of time. We have applied this algorithm to identify the parameters of DC Motor in forthcoming chapter.

CHAPTER 3

DC MOTOR MODELING

3.1 Introduction

DC motor drives are widely used in applications requiring adjustable speed, good speed regulation and frequent starting, braking and reversing operations. Some important applications are rolling mills, paper mills, mine winders, hoists, machine tools, traction, printing presses, textile mills and cranes. Fractional horsepower DC motors are widely used as servo motors for positioning and tracking. Although, it is being predicted that AC drives will replace DC drives, however, even today the variable speed applications are dominated by DC drives because of its reliability and ease of control. In this chapter, the basic model of a separately excited DC motor drive is discussed. Transfer functions and state space models are derived for position and speed control applications.

3.2 DC Motor Modeling

System model represents a certain property of the system in graphical, pictorial, analytical, or mathematical form. Block diagram and signal flow graph are the graphical models. Transfer function and state space representations are mathematical models. The speed control of DC motor are mainly governed by armature control and field control method and are discussed in the subsequent sections.

3.2.1 Armature Control method

In armature control of DC motor, it is assumed that demagnetizing effect of armature reaction is neglected, magnetic circuit is assumed to be linear and the field voltage is constant. This method is used to obtain the speed below the rated speed. Figure 3.1 shows the equivalent model of armature controlled DC motor.

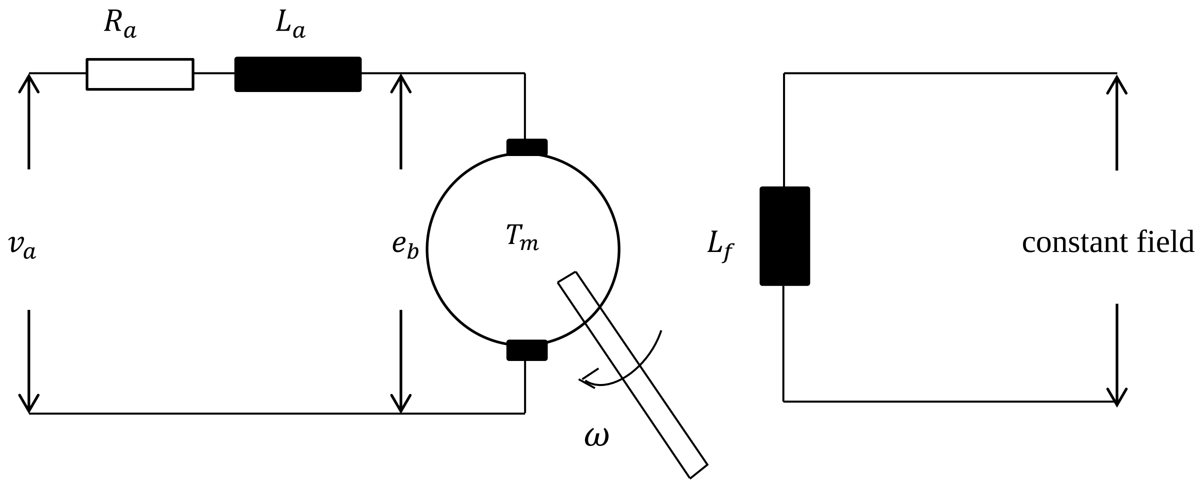


Figure 3.1: Armature Controlled DC Motor Model

$v_a(t)$ =Armature Voltage (Volts)

R_a=Armature Resistance (Ohm)

L_a= Armature Inductance (H)

K_b=Back emf constant

K_t=Torque Constant

J_m=Rotor Inertia

B_m=Viscous Friction

$E_b(t)$=Back e.m.f (Volts)

$i_a(t)$=Armature Current

$T_m(t)$=Motor Torque (Nm)

$T_L(t)$=Load Torque (Nm)

$\omega(t)$=Speed of the motor

The governing equations of a dc motor are given as follows:

$$v_a(t) = R_a i_a(t) + L_a \frac{di_a}{dt} + e_b(t) \quad \dots(3.1)$$

$$e_b(t) = K_b \omega(t) \quad \dots(3.2)$$

$$T_m(t) = K_t i_a(t) \quad \dots(3.3)$$

$$T_m(t) - T_L(t) = J_m \frac{d\omega(t)}{dt} + B_m \omega(t) \quad \dots(3.4)$$

On taking Laplace transform of eq (3.1) and eq (3.4)

$$V_a(s) = R_a I_a(s) + sL_a I_a(s) + K_b \omega(s) \quad \dots(3.5)$$

$$K_t I_a(s) = sJ_m\omega(s) + B_m\omega(s) + T_L(s) \qquad \ldots(3.6)$$

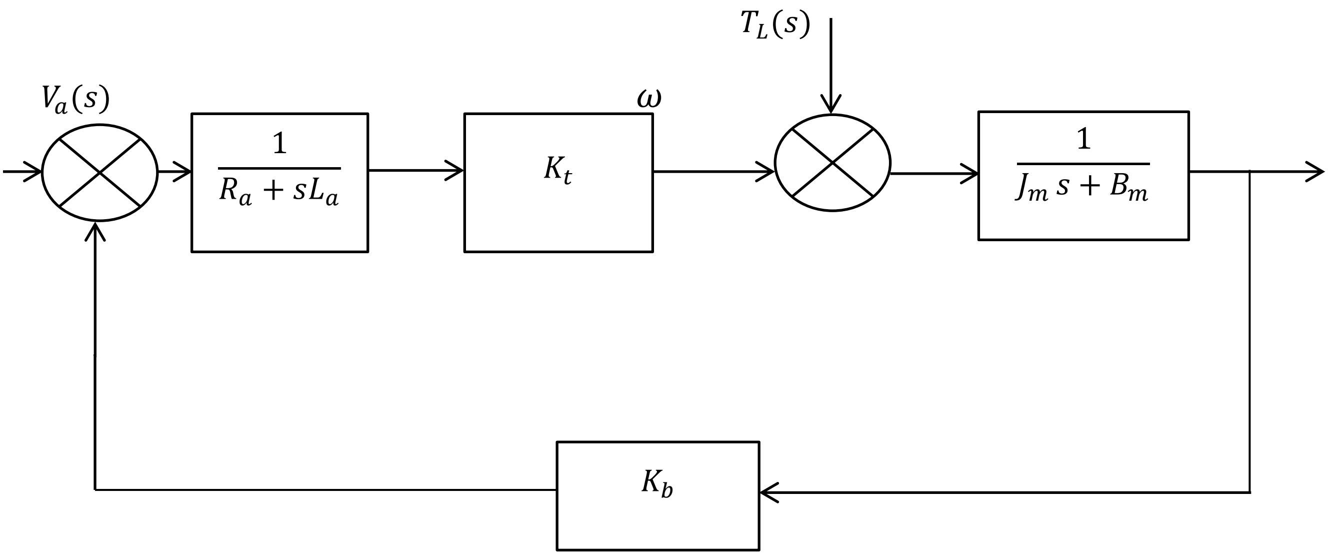


Figure 3.2: Basic block diagram of Armature Controlled DC Motor

There are two cases:

(1). When $T_L = 0$, the output follows the input signal and the corresponding response so obtained known as **Servo response** of the system. Figure 3.3 shows the block diagram of servo model of the DC motor drive.

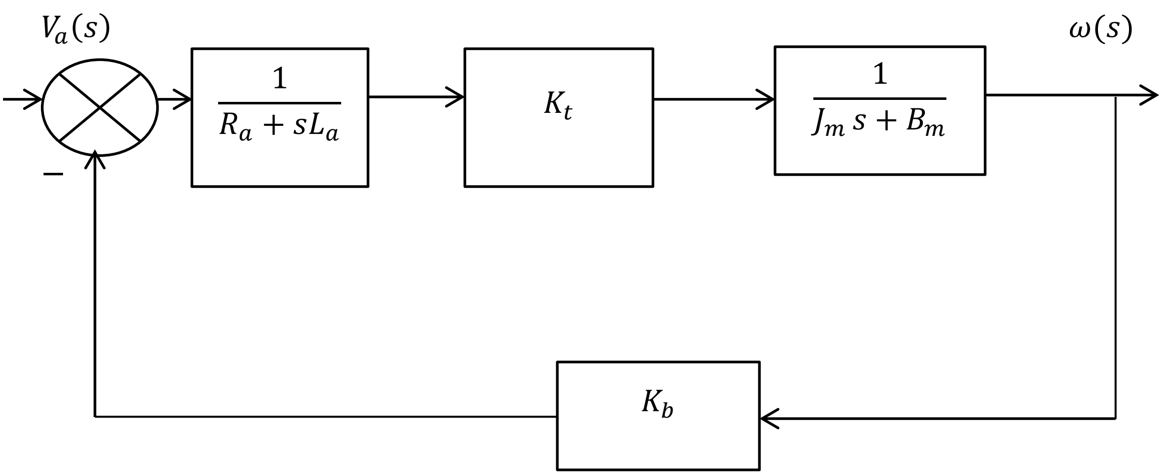


Figure 3.3: Servo Model

$$\frac{\omega(s)}{V_a(s)} = \frac{K_t}{L_a J_m s^2 + (R_a J_m + L_a B_m)s + (R_a B_m + K_b K_t)} \qquad \ldots(3.7)$$

(2). When $V_a = 0$, the response is known as **Regulatory response** of the system. Figure 3.4 shows the block diagram of regulatory model of the DC motor drive.

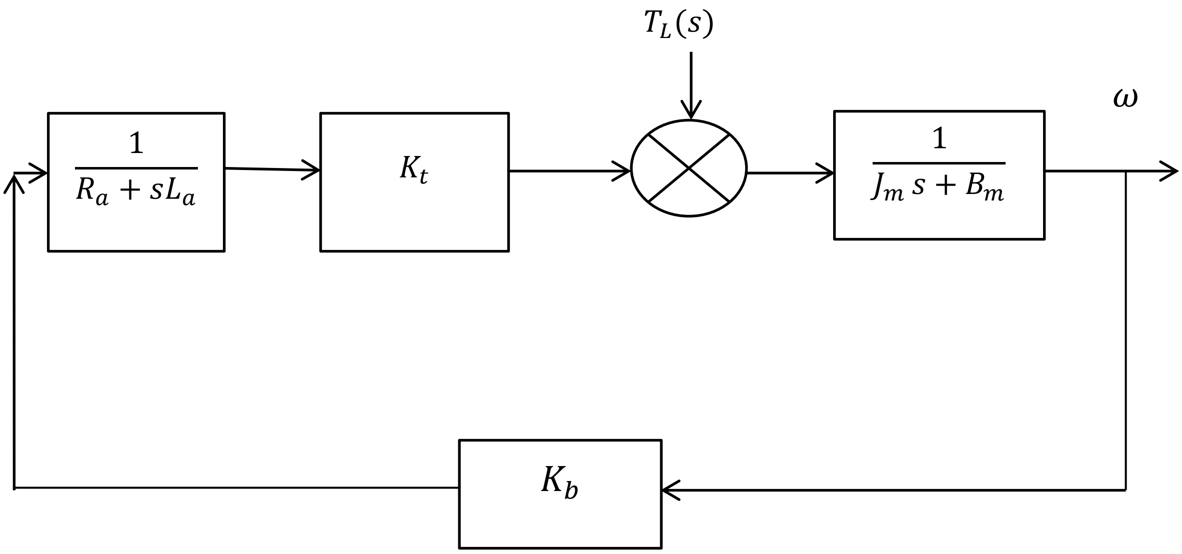


Figure 3.4: Regulatory Model

$$\frac{\omega(s)}{T_L(s)} = \frac{-(R_a+sL_a)}{L_aJ_ms^2+(R_aJ_m+L_aB_m)s+(R_aB_m+K_bK_t)} \qquad \ldots(3.8)$$

State –Space modeling

The dynamic equations are cast in state space form and are given by

$$\begin{bmatrix} \dot{i_a} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{-R_a}{L_a} & \frac{-K_t}{L_a} \\ \frac{K_t}{J_m} & \frac{-B_m}{J_m} \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & \frac{-1}{J_m} \end{bmatrix} \begin{bmatrix} v_a \\ T_L \end{bmatrix} \qquad \ldots(3.9)$$

Equation 3.9 can be expressed in the form given by

$$\dot{X} = AX + BU \qquad \ldots(3.10)$$

Where, $X = [i_a \ \omega], U = [v_a \ T_L], X$ the state variable vector and U is the input vector.

Where

A= $\begin{bmatrix} \frac{-R_a}{L_a} & \frac{-K_t}{L_a} \\ \frac{K_t}{J_m} & \frac{-B_m}{J_m} \end{bmatrix}$, B= $\begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & \frac{-1}{J_m} \end{bmatrix}$

The roots of the system are evaluated from the A matrix and given by

$$\beta 1 = \frac{-\left(\frac{R_a}{L_a}+\frac{B_m}{J_m}\right)+\sqrt{\left(\frac{R_a}{L_a}+\frac{B_m}{J_m}\right)^2-4\left(\frac{R_a B_m}{J_m L_a}+\frac{K_t{}^2}{J_m}\right)}}{2}$$

$$\beta 2 = \frac{-\left(\frac{R_a}{L_a}+\frac{B_m}{J_m}\right)-\sqrt{\left(\frac{R_a}{L_a}+\frac{B_m}{J_m}\right)^2-4\left(\frac{R_a B_m}{J_m L_a}+\frac{K_t{}^2}{J_m}\right)}}{2}$$

It is interesting to observe that these roots will always have a negative real part whatsoever be the values of the parameters, indicating that the motor is stable on open loop operation.

3.2.2 Field Control method

We can also control the speed of DC motor by varying the field flux. This method of control is generally used when the motor has to run above its rated speed. In this method, a constant current i_a is fed to the armature and flux is proportional to the field current. Figure 3.5 shows the equivalent model of field controlled DC motor drive.

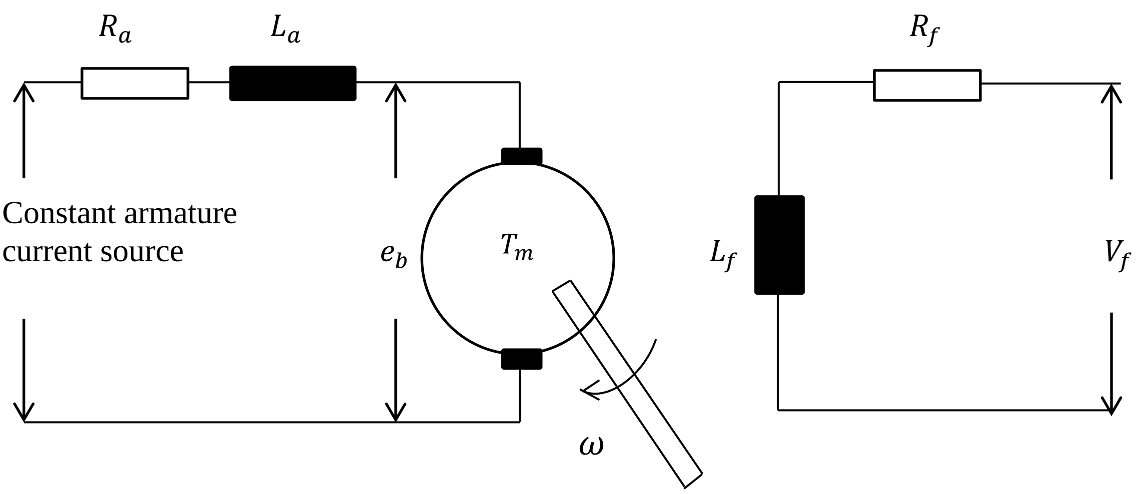


Figure 3.5: Field Controlled DC Motor Model

The governing equations of a dc motor are given as follows:

$$v_f(t) = R_f i_f(t) + L_f \frac{di_f}{dt} \quad \text{...(3.11)}$$

$$\emptyset = K_f i_f(t) \quad \text{...(3.12)}$$

$$T_m(t) = K\emptyset i_a(t) \quad \text{...(3.13)}$$

$$T_m(t) = KK_f i_f(t) \quad \text{...(3.14)}$$

$$T_m(t) - T_L(t) = J_m \frac{d\omega(t)}{dt} + B_m\omega(t) \quad \text{...(3.15)}$$

Where,

$v_f(t)$ =Field voltage (Volts)

R_f=Field Resistance (Ohm)

L_f= Field Inductance (H)

K_f, K= constant

J_m=Rotor Inertia

B_m=Viscous Friction

$T_m(t)$=Motor Torque (N-m)

$T_L(t)$=Load Torque (N-m)

$\omega(t)$=Speed of the motor

On taking Laplace transform of eq (3.11), eq (3.14) and eq (3.15)

$$V_f(s) = R_f I_f(s) + sL_f I_f(s) \quad \text{...(3.16)}$$

$$T_m(s) = KK_f i_f(s) \quad \text{...(3.17)}$$

$$T_m(s) - T_L(s) = sJ_m\omega(s) + B_m\omega(s) \quad \text{...(3.18)}$$

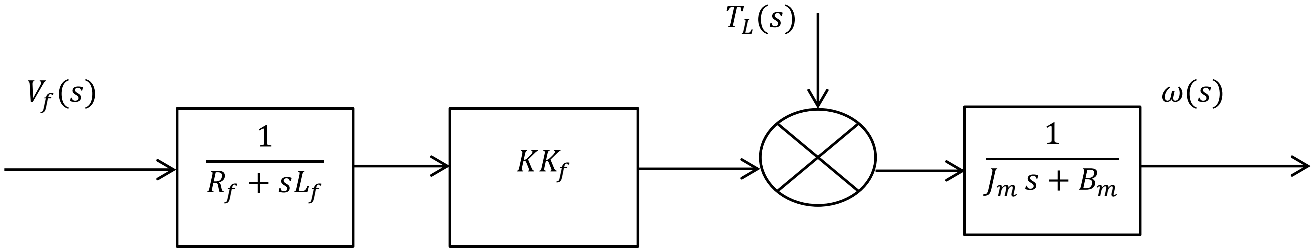


Figure 3.6: Basic block diagram of field controlled DC Motor

$$\frac{\omega(s)}{V_f(s)} = \frac{KK_f}{(L_f s + R_f)(J_m s + B_m)} \qquad \text{…(3.19)}$$

State –Space modeling

The dynamic equations can be cast into state space form and can be given by

$$\begin{bmatrix} \dot{i_f} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{-R_f}{L_f} & 0 \\ \frac{KK_f}{J_m} & \frac{-B_m}{J_m} \end{bmatrix} \begin{bmatrix} i_f \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L_f} & 0 \\ 0 & \frac{-1}{J_m} \end{bmatrix} \begin{bmatrix} v_f \\ T_L \end{bmatrix} \qquad \text{…(3.20)}$$

Equation 3.20 can be expressed in the form given by

$$\dot{X} = AX + BU \qquad \text{…(3.21)}$$

Where, $X = [i_a\ \omega], U = [v_a\ T_L], X$ the state variable vector and U is the input vector.

$$A = \begin{bmatrix} \frac{-R_a}{L_a} & \frac{-K_t}{L_a} \\ \frac{K_t}{J_m} & \frac{-B_m}{J_m} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & \frac{1}{J_m} \end{bmatrix}$$

The roots of the system are evaluated from the A matrix and given by

$$\beta 1 = \frac{-\left(\frac{R_f}{L_f} + \frac{B_m}{J_m}\right) + \sqrt{\left(\frac{R_f}{L_f} + \frac{B_m}{J_m}\right)^2 - 4\left(\frac{R_f B_m}{J_m L_f}\right)}}{2}$$

$$\beta 2 = \frac{-\left(\frac{R_f}{L_f} + \frac{B_m}{J_m}\right) - \sqrt{\left(\frac{R_f}{L_f} + \frac{B_m}{J_m}\right)^2 - 4\left(\frac{R_f B_m}{J_m L_f}\right)}}{2}$$

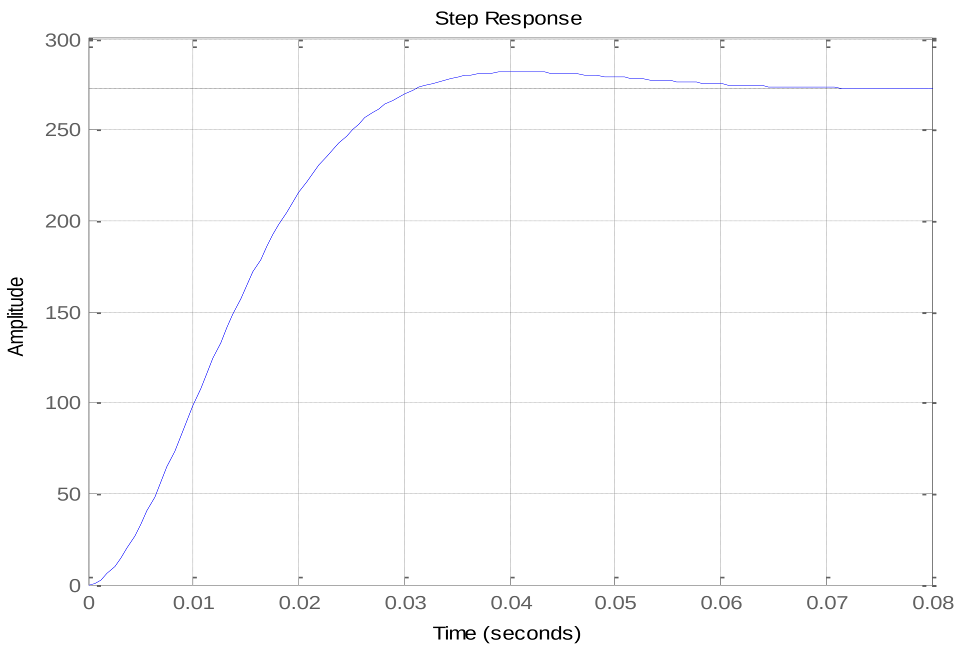
It is interesting to observe that these roots will always have negative real part irrespective of parameter values, indicating that the motor is stable on open loop operation.

3.3 Results

The DC Motor specifications are given below [5]

DC Supply Voltage	220 Volt
Armature Resistance (R_a)	0.5 Ohm
Armature Inductance (L_a)	0.003 Henry
Viscous Friction (B_m)	0.01 Nm/rad/sec
Moment of Inertia (J_m)	0.0167 Kg-m^2
Back Emf constant (K_b) = Torque constant (K_T)	0.8 Volt/rad/sec
Load Torque (T_L)	100 N-m

3.3.1. Servo Response



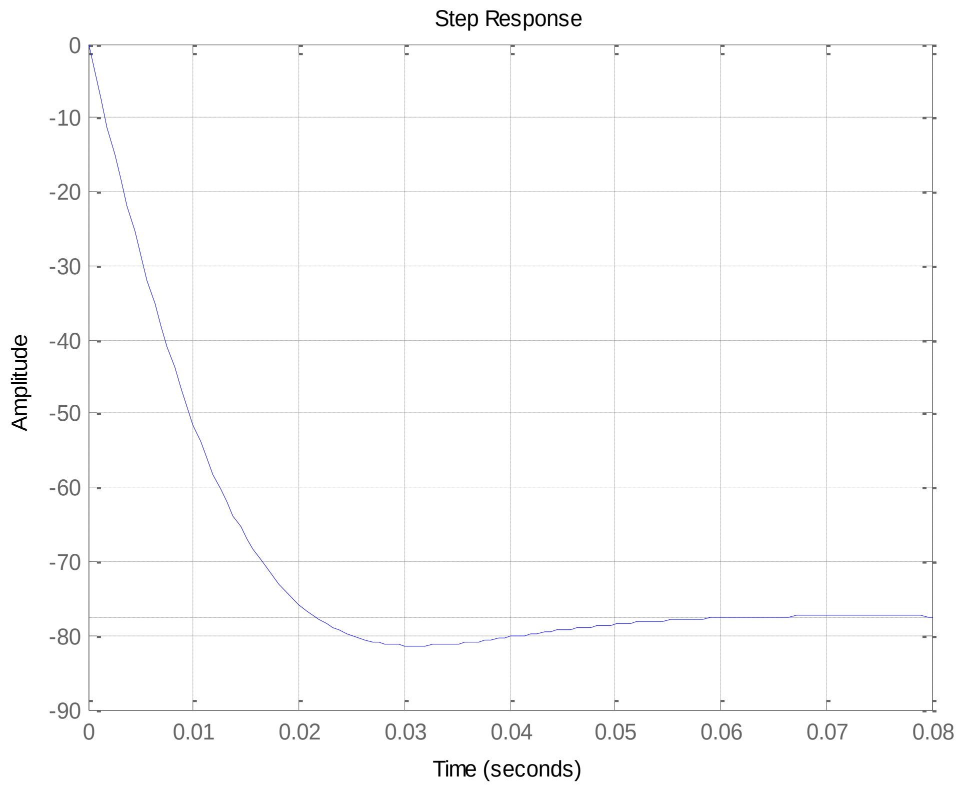
Rise time = 0.0198;

Overshoot = 3.2494;

Peak time = 0.0408;

Settling time = 0.0515

3.3.2. Regulatory Response



Rise time = 0.0150;

Overshoot = 4.9703;

Peak time = 0.0314;

Settling time = 0.0462

3.4. Conclusion

In this chapter, we have modeled a dc motor by armature control and field control method. The time constant of armature controlled dc motor is generally small compared to the field control hence the time response of the former is usually faster. Further in armature controlled dc motor, the back emf developed by the armature adds to the damping. That is why we have used the armature controlled dc motor model or the identification of dc motor drive in the forthcoming chapter.

CHAPTER 4

PARAMETER IDENTIFICATION OF DC MOTOR USING FIREFLY ALGORITHM

4.1 Introduction

Understanding why a system behaves in a certain way and predicting its future behavior is a major field of research in control engineering. The system in question could be anything from bacteria growth and stock markets to global warming and galaxy movements. Although some of these systems, such as the stock market are a man-made system yet its exact behavior can't be determined. This uncertainty in such systems is due to the complexity of the system. In other situations, some simple man-made systems deviate from its designed behavior due to aging or wearing out, which was assumed to be static. For example, the performance of car brakes changes with time due to aging or wearing out. The behavior of natural systems such as galaxy movements and global warming is far more complex and not easier to understand.

System identification involves creating a model for the system in question, that the model will produce an output that matches the original system output to a certain degree of accuracy. The input or excitation given to both system and the model, and their corresponding output are used to create and tune that model until a satisfactory degree of model accuracy is reached. As shown in Figure 4.1, the input $u(t)$ is fed to both the system and the model M, then their corresponding outputs $y(t)$ and $y1(t)$ are produced and matched to produce the error. The error $e\ (t)$ reflects how much the model matches the system; the lower the error, the more the model resembles the system.

System identification of practical systems is not an easy task to be accomplished by traditional techniques. Most real-world systems contain dynamic components. Due to this dynamic nature of the system, the output of the system not only depends on the current input but also on the past inputs and outputs of the system. As the number of old data affecting the system increases, the number of terms used in the classic system models increases substantially. Another difficulty encountered in identifying real-world system is its nonlinearity. In order to simplify the problem, the discrete time non-linear systems can be represented by Non-linear Auto-Regressive Moving Average model with exogenous inputs (NARMAX) models.

System identification is essential for many fields of applications which may be anything from planetary movement in astronomy to algae growth in Biology. System identification consists of two subtasks (a). Structural Identification, (b). Parameter Identification. It can be formulated as an optimization technique where objective is to find a model and its parameters that minimize the error between the system and model M. The sum of squared error (SSE) is a commonly used measure of the prediction error.

$$\text{SSE}=\sum[y(t)-y1(t)]^2$$

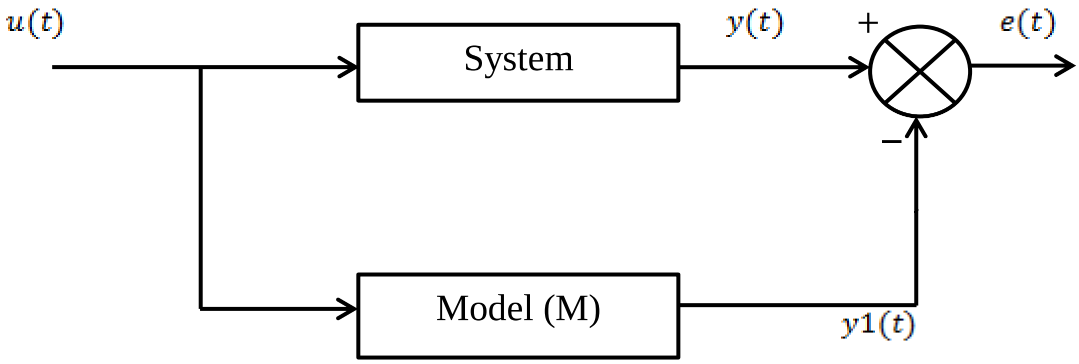


Figure 4.1 Process of System Identification

4.2 Identification Procedure

The system identification process of constructing a system model can be described by the following procedure [6].

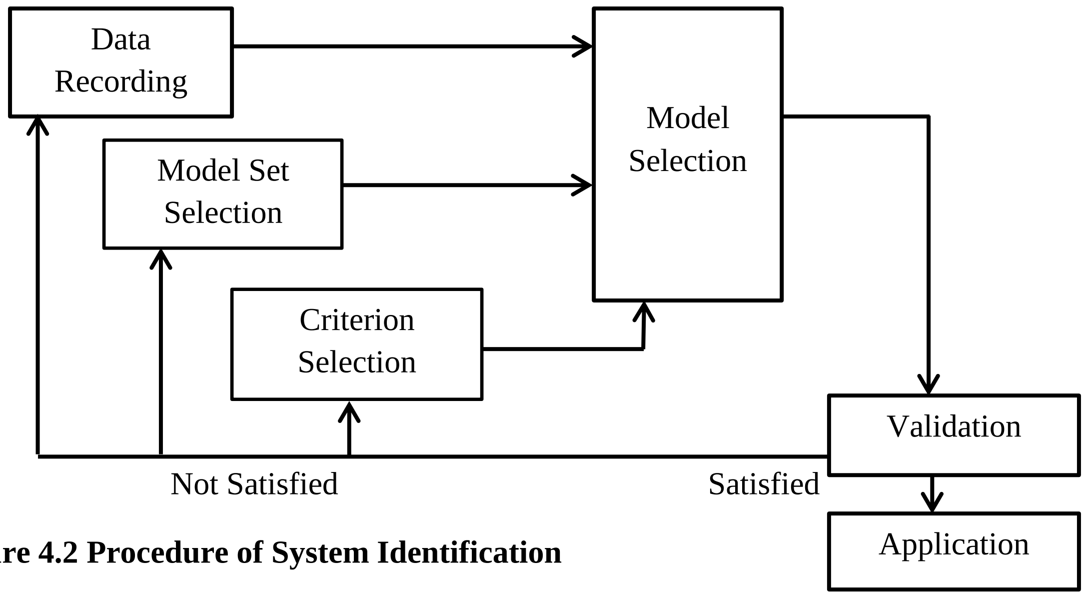


Figure 4.2 Procedure of System Identification

i. Data Recording

Sometimes the input and output of system are recorded during identification experiment where user may choose when and what input and output is to be recorded. The objective of data recording is to make the data more informative.

ii. Model Set Selection

The next step is to choose a model set by specifying what types of models we are looking for. The model set selection is the most important and at the same time, the most difficult step of identification process. To describe a process or a system we need a model of it. We can develop the model of the system from experiences for example, when we drive a car and approaching a road bump, we slow down because we feel intuitively that when the speed is too high, we will hit the head in the roof and another ways to describe the systems by white box modeling, black box modeling and grey box modeling. When the model is completely constructed from mathematical scientific relations, such as differential equation, difference equation and algebraic relation, the resulting model is known as white box model. When a model is formed by means of identification without knowing the inside of system, the model is known as black box model. There is another model in which some but incomplete knowledge about the system is known as grey box model.

iii. Best Model Selection

After the model set is selected, the best model in this set is selected using some of the input-output data recorded previously. In this step, the model parameters are tuned so that the model output would fit the system output as much as possible.

iv. Model Validation

The next step is to verify the quality of the developed model by comparing the model output to the original system output when both are fed with samples of the input-output data recorded previously (in this case, the validation data are different from the data used for model selection), or with the system in real operation situation (not experimental mode). If the model meets the chosen criteria, the model is accepted, otherwise, it is rejected and another model is created. This procedure is repeated until a satisfactory model is created. It is to be noted that a system model will only imitate the original system in certain aspects of interest to the model designer. A model can never be accepted as a final and true description of the system.

4.3 Literature Review

Krneta *et al.* [7] presented procedure of parameters identification of DC motor model using a method of recursive least squares. To identify the system an experiment, measuring of signals was carried out at input-supply of voltage and output of the system for identification - motor angle speed. This paper investigated the issues involved in applying Recursive Least Squares method in parameters identification of DC motor model. The issues were considered both theoretically and experimentally.

Saab *et al.* [8] implemented parameter identification of dc motor employing least square algorithm without the use of D/A converter and a power amplifier. In addition to improve the overall identification performance the dc parameters were first estimated by decoupling the ac parameters using a dc input signal. Subsequently the estimated dc parameters are then used to identify the ac parameters.

Arshad *et al.* [9] presented the working and efficiency of the two basic algorithms used for parameter estimation: Least Square (LS) and Recursive Least Square (RLS). A simple DC motor was taken as an example of a SISO system. The input voltage and the output in the form of rotations of the motor were given to the parameter estimator. The value of error for LS was far less than that achieved using RLS. Also, LS, due to its uniterative nature, was fast in estimating the unknown parameters. All these points made LS the better option between the two when it comes to estimating the parameters of a simple DC motor.

Nassef *et al.* [10] demonstrated a procedure using least squares algorithm for the identification of a feed drive system coefficients in time domain using a reduced model based on windowed input and output data. The command and response of the axis were first measured in the first 4 ms, and then least squares were applied to predict the transfer function coefficients for this displacement segment. The obtained results revealed a considerable potential of least squares method to identify the system’s time-based coefficients and predict accurately the command response as compared to measurements.

Eker *et al.* [11] demonstrated discrete-time on-line identification of a permanent-magnet DC motor in open-loop conditions. Studies were carried out by formulating the mathematical model using differential equations, and digital identification using plant input-output data. A real-time implementation of the RLS estimator was presented on the DC motor. The open-loop experimental tests was conducted successfully and clearly justified the ease of the computer based parameter identification method.

Thananchai [12] demonstrated theoretical analysis and experimental simulation of micro DC motor in real time environment using simulink/xpc target toolbox. The micro dc motor dynamics identified in this paper was experimentally verified in the modeling and parameter identification of a motion control system.

Wu [13] presented a convenient and effective system identification approach to estimate the DC motor torque constant, mechanical time constant, electrical time constant and friction coefficients using step response. This approach was implemented on two Mabuchi motors, and the test results were presented.

Kara *et al.* [14] presented nonlinear modeling and identification of a bidirectional DC motor with real time experiments. Linear and nonlinear models for the system were obtained for identification purpose, and the major nonlinearities in the system, such as Coulomb friction and dead zone were investigated and integrated in nonlinear model. The paper also revealed that the performance of the non-linear approach was superior as compared to the low speed operation, where the non-linearities in the system proved effective.

Tutunji *et al.* [15] modeled Auto Regressive Moving Average (ARMA) with steepest descent gradient algorithm to minimize error between the original and modeled velocities and then transfer function realized. After that, the input voltage was varied and the identified model results were compared with the original system.

Hadef, *et al.* [16] developed inverse problem methodology for parameter identification of separately excited dc motor. Conjugate gradient method was used to determine the unknown parameters while Tikhonov’s regularization method was used to replace the original ill-posed problem with a well-posed problem.

Bourouina *et al.* [17] showed parameter identification of DC Motor via moments Method. The basic idea was general application in identification, model order reduction and controller design and comparison between direct test and Moment method was discussed based on real measurements taken in a laboratory on a separately excited dc motor.

Hadef *et al.* [18] proposed two identification algorithms developed based on the moments and Pasek's methods and applied them for the parameter identification of a DC motor. The first model was based on Pasek's method, and the second based on the moments method, both of which were used to identify all the motor parameters. The second method makes the model closer to reality, especially in a transient regime.

Becedas *et al.* [19] proposed a fast, non-asymptotic, algebraic parameter identification method applied to a DC motor to estimate its uncertain parameters: viz, viscous friction coefficient and inertia. The methodology was developed and analyzed. A comparative study between the traditional recursive least square method and the algebraic identification method was carried out. Some of advantages of this method were that it did not require any statistical knowledge of the noises corrupting the data. The methodology was robust with regard to the constant perturbation input, the Coulomb friction and also robust with regard to zero mean high frequency noises.

Kundsen, *et al*. [20] developed a non-linear model structure for a permanent magnet DC motor, appropriate for simulation and controller design. Experimental result demonstrated that linear models, with the parameter determined from traditional static measurements fitted dynamic measurements poorly while non-linear models with the parameter estimated from dynamic measurements however, fitted the measurements very well.

Saab, *et al.* [21] demonstrated to estimate the parameters of a DC motor experimentally employing discrete measurements of an integrated dynamometer. The parameters under consideration were the motor armature-winding resistance and inductance, back emf constant, motor torque constant, moment of inertia and viscous friction. A Kalman filter was also implemented, as a state observer, to estimate the angular acceleration and the derivative of the armature current.

Tjahjowidodo *et al.* [22] proposed identification and control of friction in a high load torque DC motor to the end of achieving accurate tracking. Model-based friction compensation in the feed-forward part of the controller was considered. For this purpose, friction model structures ranging from the simple Coulomb model to the recently developed Generalized Maxwell Slip (GMS) model were employed. For motions with high velocities, the classical models such as Coulomb friction and the Stribeck friction model gave satisfactory results comparable to the advanced models.

Mamani *et al.* [23] demonstrated adaptive position control scheme for a DC motor based on online closed loop continuous time identification. A fast, non-asymptotic, algebraic identification method was used to estimate the unknown system parameters and to update the controller. The method was suitable for simultaneously identifying both the viscous friction coefficient and the inertia of the motor. The methodology proved robust with respect to the Coulomb friction torque, considered as a constant perturbation input and was also robust with respect to zero mean high-frequency noises as seen from digital computer-based simulations. The estimation was obtained in a very short period of time, and good results were achieved.

Dub *et al.* [24] described mathematical model of separately excited dc motor featured by transfer function and applied Nelder-Mead's Simplex method to identify its parameters of dc motor and result was that the model simulated response practically matched the real system output. It was clearly visible that we can search parameters of the mathematical model from parameter offline identification of both rational speed response and armature current response.

Mohammed [25] presented a new system identification algorithm for obtaining an optimal set of mathematical models for system with perturbed coefficients. Then this algorithm was further applied practically by an "On Line System Identification Circuit", based on real time speed response data of a permanent magnet DC motor.

Kiran Raj *et al.* [26] presented a paper to control the speed of the PMDC motor and find the model of the closed loop system. Experimentally interface the PMDC motor to PC using NI USB-6008 DAQ card. For designing the PID controller, Lab View control system toolkit was used. The parameters were tuned to meet the requirements of the quarter amplitude decay ratio and the state space model of the closed loop system was found by sub space or black box method in Lab View.

Ganesh *et al.* [27] presented a method of determining mechanical parameters viz. moment of inertia and friction coefficient of motor and load. This paper also stressed that load parameters have appreciable effect on the dynamic response of systems and have to be determined. Effect of load on the system dynamics was emphasized by considering the PID controller tuning. Proposed method was used for estimation of moment of inertia and friction of DC motor and load under dynamic load variations.

Bhushan *et al.* [28] presented Bacterial foraging algorithm (BFA) implementation for indirect adaptive control of two nonlinear systems. The nonlinear systems considered for analysis were liquid level control of surge tank and armature controlled DC motor speed control system**.** Simulations for both the nonlinear systems were also performed by GA (Genetic Algorithm). In DC motor error in speed trajectory was less fluctuating in BFA adaptive control compare to GA based adaptive control, while performance of BFA was nearby same as GA based adaptive control in liquid level system. Elapsed time of simulation is less in BFA adaptive control as compared to GA based control.

Udomsuk *et al.* [29] developed that the mathematical model of a dc motor with Adaptive Tabu Search (ATS) method could obtain a speed response (simulation) nearly the same as those of the testing (experiment). Hence, the good agreement of the speed responses was to confirm that the parameters from the proposed method were correct.

Yassin *et al*. [30] presented a Radial Basis Function Neural Network (RBFNN) based Nonlinear Auto-Regressive Model with Exegeneous Inputs (NARX) model to identify a DC motor drive model. The OSA tests performed good model fit (indicated by the residuals); while tests on the residuals showed that the NARX model had sufficiently covered all underlying dynamics present in the DC motor.

Rahim *et al*. [31] presented a study on non-linear autoregressive moving average with exogenous input (NARMAX) model using multilayer perceptron (MLP) neural networks for DC motor modeling. The results showed that DC motor drive system could be successfully modeled using the NARMAX model with parameter fitting employed by MLP neural network.

Feilat *et al.* [32] proposed neural network approach for the identification and control of a separately excited dc motor (SEDCM) loaded with a centrifugal pump. Radial basis function neural networks (RBFNN) were used as RBFNN identifier and RBFNN controller, which were trained to make the motor speed follow a selected reference signal. RBFNN was found to be effective in designing robust neuro-controllers of SEDCM with excellent dynamic behaviors.

Cong *et al.* [33] used compound evolution algorithms for parameter identification of a non-linear dc motor. The Genetic Algorithm (GA) with global optimization character and the simplex method were combined and used into the application of the parameter identification.

Using the Global search ability of GA and fast convergence of simplex method, the efficiency of parameter identification were increased significantly.

Arif *et al.* [34] demonstrated Elman Neural Network (ENN) based identification of dc motor. Moreover, ENN learned by Genetic algorithms were found to be more representative to system order in terms of its structural complexity in comparison to those learned by back propagation algorithm. This method was utilized efficiently to find the minimum ENN structure that represents the discrete time state space model of the DC motor. Linear ENN based Genetic algorithm was not only used to find the exact state-space order of the linear system but also to identify its unknown physical parameters.

Rubaai *et al.* [35] proposed a multilayered feed forward artificial neural-network-based adaptive control structure for unknown motor/load dynamics. The random training for the neural networks was accomplished online, which enabled better absorption of system uncertainties into the neural controller. The two controller topologies considered had shown to yield satisfactory tracking performance. The ability of a neural controller to perform in the presence of noisy environment was investigated. Satisfactory performance was observed for reference track.

Weerasooriya *et al.* [36] demonstrated an artificial neural network based high performance speed control system for a dc motor. The objective was to achieve accurate trajectory control of the speed, especially when motor and load parameters were unknown. The unknown time invariant, non-linear operating characteristics of the dc motor and its load were successfully calculated by ANN. The concepts of model reference adaptive control were used in conjunction with the trained ANN to achieve trajectory control of the rotor speed.

Narendra *et al* [37] demonstrated that neural network can be used effectively for identification and control of nonlinear dynamical systems. Emphasis was based on both identification and control. Multilayer and Recurrent networks were connected in novel configurations to provide optimum result.

Lankarany *et al.* [38] proposed the application of Genetic Algorithm optimization in estimating the parameters of dc motors and compared GA with least square estimation method and found that GA estimation could be used for any LTI systems which were not linear due to parameters and was also applicable in offline parameter estimation.

Liu *et al.* [39] demonstrated fault detection and diagnosis of a permanent-magnet dc motor based on block-pulse function series to estimate the continuous-time model of the motor and showed that neural network has a good ability for pattern classification. An MLPN (Multilayer perceptron network) had been adopted to isolate the faults of the permanent-magnet dc motor.

Tzes *et al.* [40] presented a fuzzy-logic-based model describing the friction present in a dc-motor system. The fine-tuning of the parameters was accomplished using genetic algorithm which minimized a system modeling relevant function. Experimental results were offered to validate the performance of the proposed fuzzy modeling and control technique and concluded that the proposed scheme reduced the system's dead-band and compensated any hysteresis and/or disturbance related effects.

Rezazade [41] proposed the application of Genetic Algorithm in estimation of the parameters of servo electrical drives and compared GA based parameter estimation with least square estimation method and showed that the GA method of estimation produced better results in the startup of the system where there was a lack of persistent excitation.

Maboud *et al.* [42] demonstrated application of Genetic Algorithm (GA) Optimization to estimate the parameters of dynamical and electrical state of a DC Motor. The problem was organized in two sections. The first problem was about dynamical treatment of a DC motor. After consideration the parametric model of a DC motor, three unknown parameters was identified by assistance of collecting data and using proposed method. The second problem was about estimation of electrical parameters of a DC motor. Finally comparison between LSE and GA optimization was presented to indicate robustness, resolution, accuracy and quicker response of GA identification method in parameter estimation.

Kowalska *et al.* [43] presented the application of genetic algorithm to the parameter identification of DC motor drive based on easy measurable motor variables. Three identification methods of the electro-magnetic as well as mechanical motor parameters were discussed. The first one was based on current and simple speed sensor (used to measure the steady-state speed). The second and the third methods based on only current sensor. The genetic algorithm was very effective in the application for the parameter identification problem of DC motor drive. The only disadvantage of this solution was a relatively long computational time.

4.4 Methodology

The initial values of parameters for DC motor like armature resistance (R_a), armature inductance (L_a), back emf/torque constant (K) are set from experimental measurement and the values of moment of inertia (J_m), viscous friction (B_m) are taken randomly because it is complicated to determine these parameters from the practical measurement. Then, we optimize these parameters using firefly algorithm and validate the results from the experiment.

4.5 Algorithm for the Parameter Identification Process

Step 1. Define the number of parameters to be identified for a DC motor.

Step 2. Initialize the parameters of firefly algorithm (FFA) viz. α (randomness), β (attractive-ness), γ (light absorption coefficient), δ(randomness reduction parameter), number of fireflies n, maximum iteration, stopping criteria.

Step 3. Define the objective function f(x).

Step 4. Select the range for the objective function parameters.

Step 5. Generate the initial value x_i , where (i=1: number of model parameters) of the model parameters randomly.

Step 6. Set iteration count, iter = 1

Step 7. Determine function value I_i at x_i , where (i=1: number of model parameters) by the value of $f(x_i)$.

Step 8. Arrange the model parameters according to their function values.

Step 9. If I_i<I_j where i, j =1: n, the parameter values will change according to the equation as follows:

$$x_i = x_i + \beta_0 e^{-\gamma {r_{ij}}^2}\left(x_j - x_i\right) + \alpha\left(rand - \frac{1}{2}\right)$$

Step 10. Find the value of function at new values of x_i.

Step 11. Check stopping criteria and iter > maximum iteration, if it is satisfied GOTO step 14.

Step 12. Modify randomness parameter α according to the formula given below:

$$\alpha = \delta \times \alpha.$$

Step 13. Advance the iteration count, iter = iter +1 and GOTO step 7.

Step 14. Find the optimal value of the function and associated parameters.

Step 15. Stop.

4.6 Experimental setup

The rating of the DC motor for the identification process is given below in Table 4.1.

Table 4.1 Specification of the DC Motor used for the experiment

Type of motor	DC shunt motor
Rated speed (N)	1250 rpm
Armature Current (I_a)	4 Amp
Supply Voltage (V)	220 V DC
Rated Power (P)	0.9 kW

Experimentally variation of speed of the motor against different armature voltages is given below in Table 4.2.

Table 4.2 Variation of speed corresponding to different armature voltages

Armature voltage (volt)	Speed (N) (rpm)
220	1296
214	1254
204	1204

200	1174

Figure 4.3 Experimental Setup

4.7 Results

The initial and optimum values of parameters are given below in Table 4.3 when user defined parameters of Firefly Algorithm (FFA) are taken as α=0.95, β=1.0, γ=1.0, δ=0.97.

Table 4.3 Comparison between measured and optimized value of the parameters

Parameters	Initial values	Optimized values
Armature resistance(R_a)	8.9 ohm	8.6472 ohm
Armature inductance(L_a)	0.0994 H	0.1015 H
Moment of Inertia(J_m)	0.03 kg-m^2	0.0345 kg-m^2
Viscous friction(B_m)	0.048 N-m-s/rad	0.0549 N-m-s/rad
Back emf constant/Torque constant(K)	1.45	1.2363

Step response of speed with initial and optimized values of DC motor parameters is shown in Figure 4.4. There is a mismatch between the two due to the initial guess of the parameters viz. moment of inertia (J_m) and viscous friction (B_m).

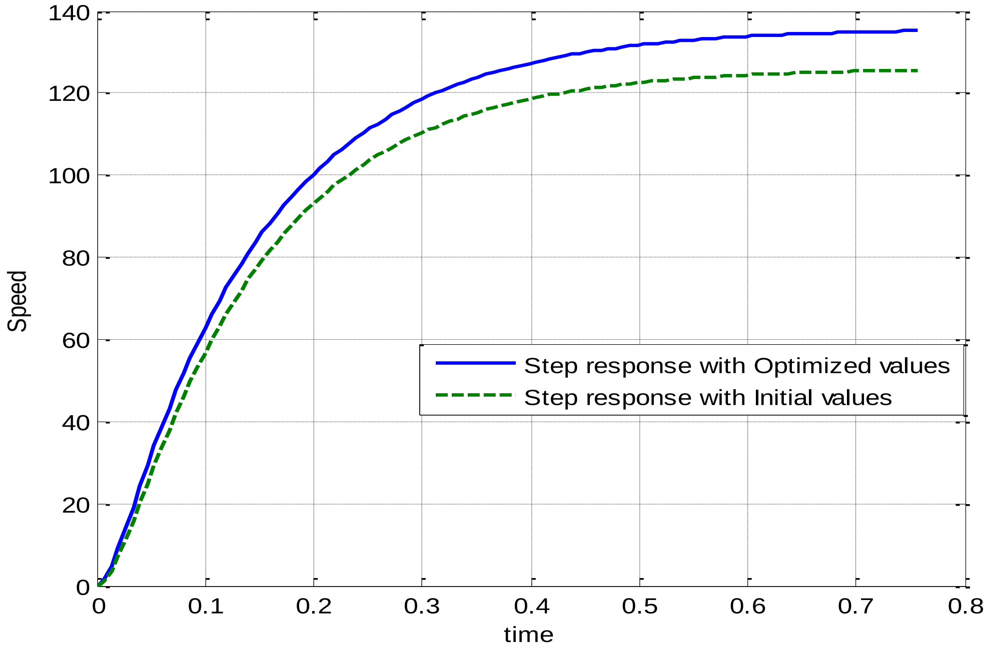


Figure 4.4 Step response of Speed (rad/sec) with initial and optimized values of DC motor parameters (continuous time system)

Step response of speed (speed simulation) at sampling time T_s=0.01 second is shown in Figure 4.5.

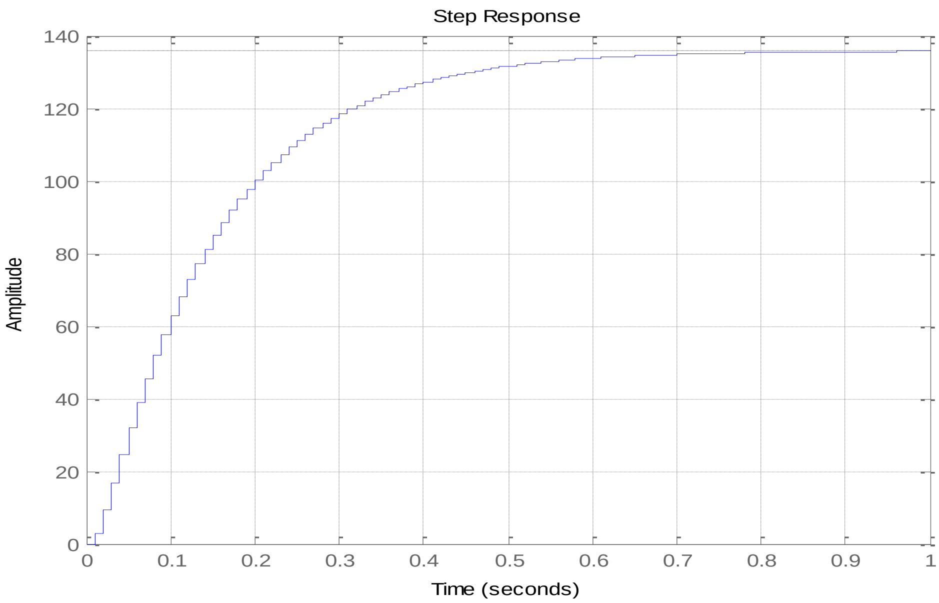


Figure 4.5 Step response of Speed (rad/sec) (simulated) for T_s = 0.01 second (discrete time system)

The variation of square error between simulated speed and experimental speed with iterations is shown in Figure 4.6.

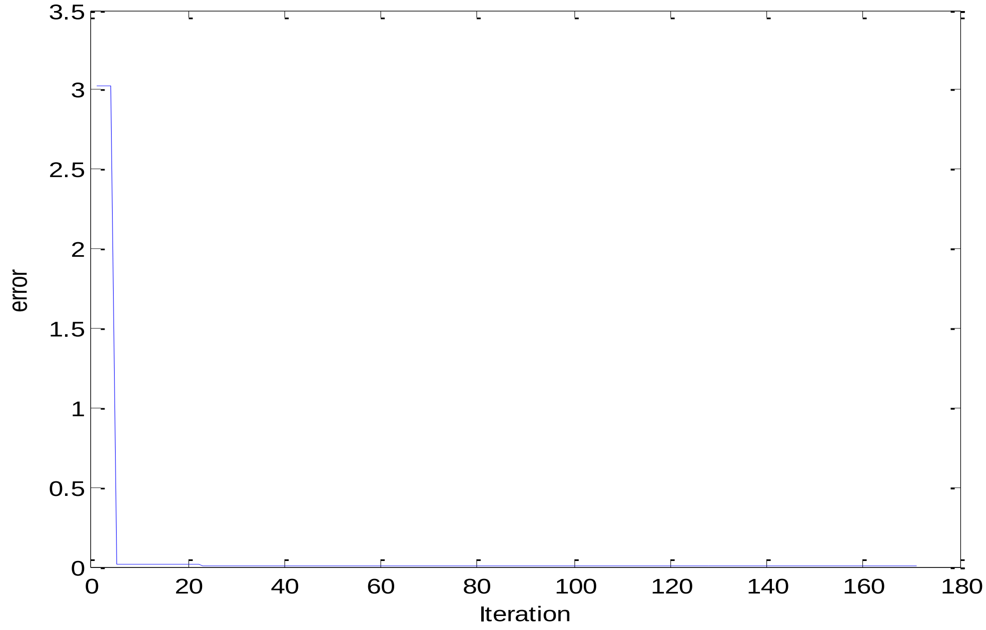


Figure 4.6 Variation of square error with iterations

The variation of armature resistance (R_a) with iterations is shown in Figure 4.7.

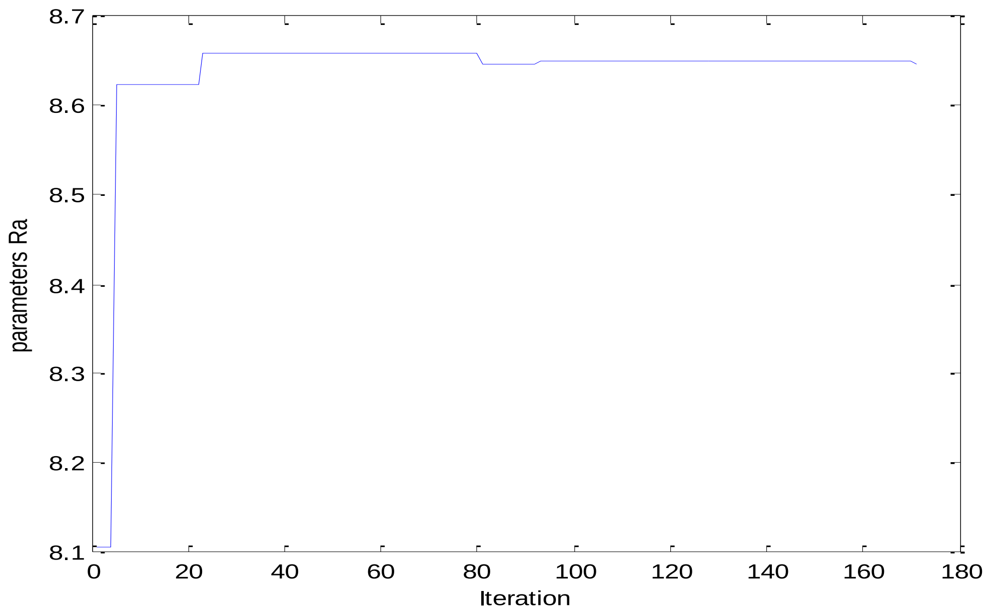


Figure 4.7 Variations of armature resistance with iterations

The variation of armature inductance (L_a) with iterations is shown in Figure 4.8.

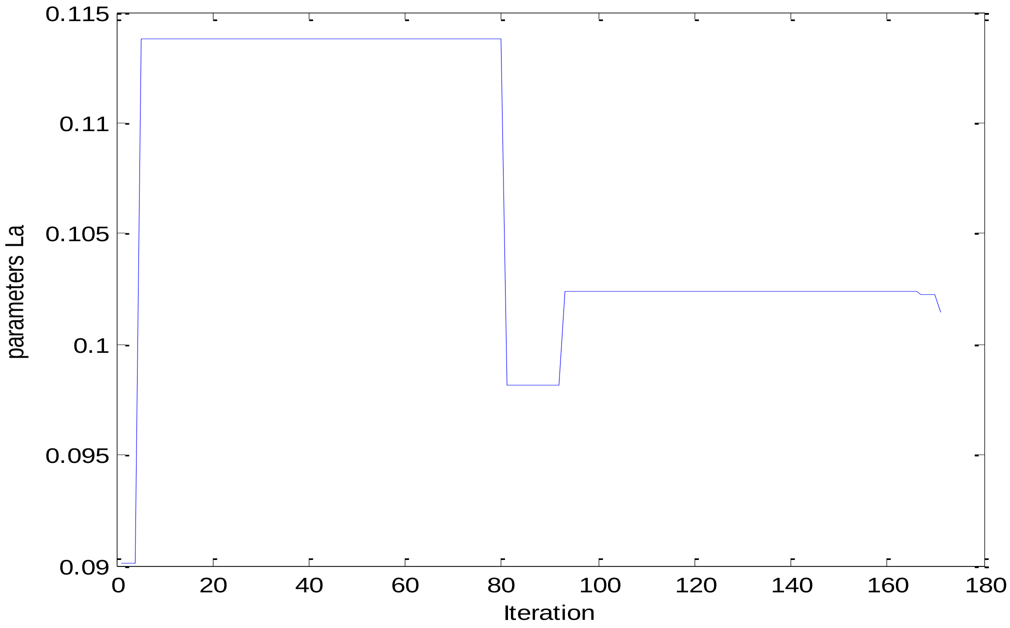


Figure 4.8 Variation of armature inductance with iterations

The variation of moment of inertia (J_m) against iterations is shown in Figure 4.9.

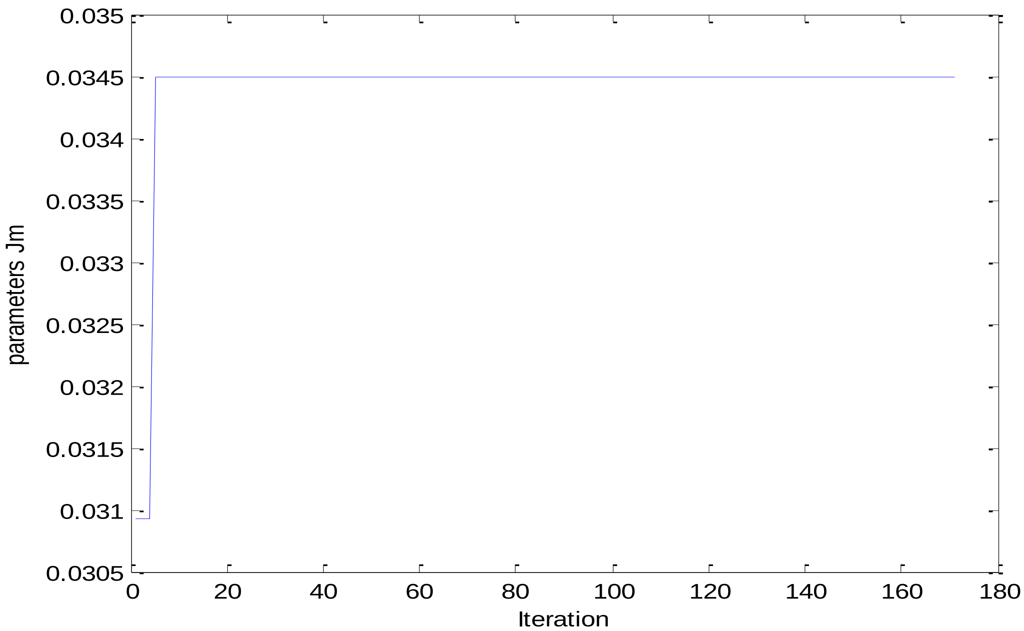


Figure 4.9 Variation of moment of inertia with iterations

The variation of viscous friction (B_m) with iterations is shown in Figure 4.10.

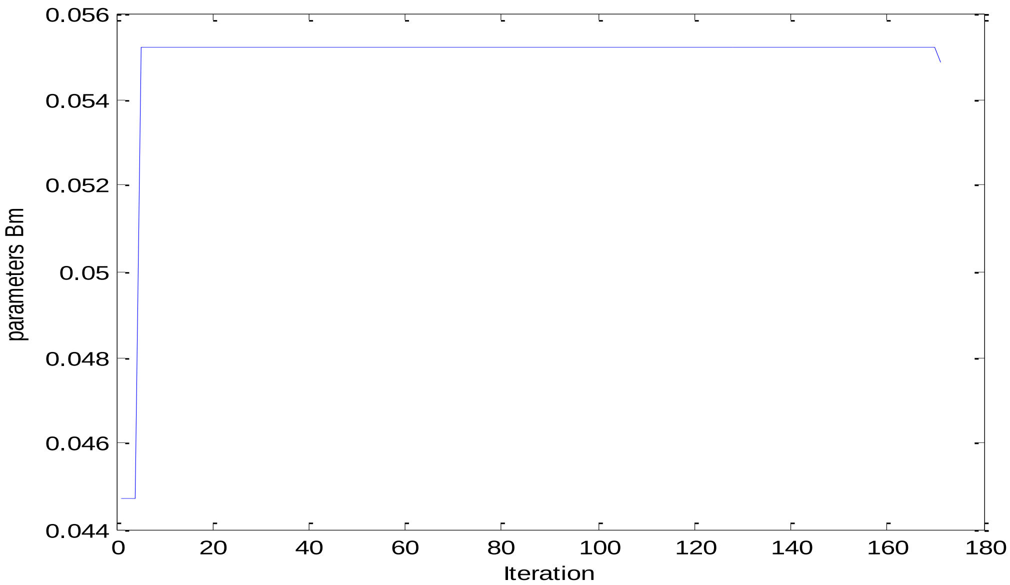


Figure 4.10 Variation of viscous friction with iterations

The variation of back emf constant/ torque constant (K) with iterations is shown in Figure 4.11.

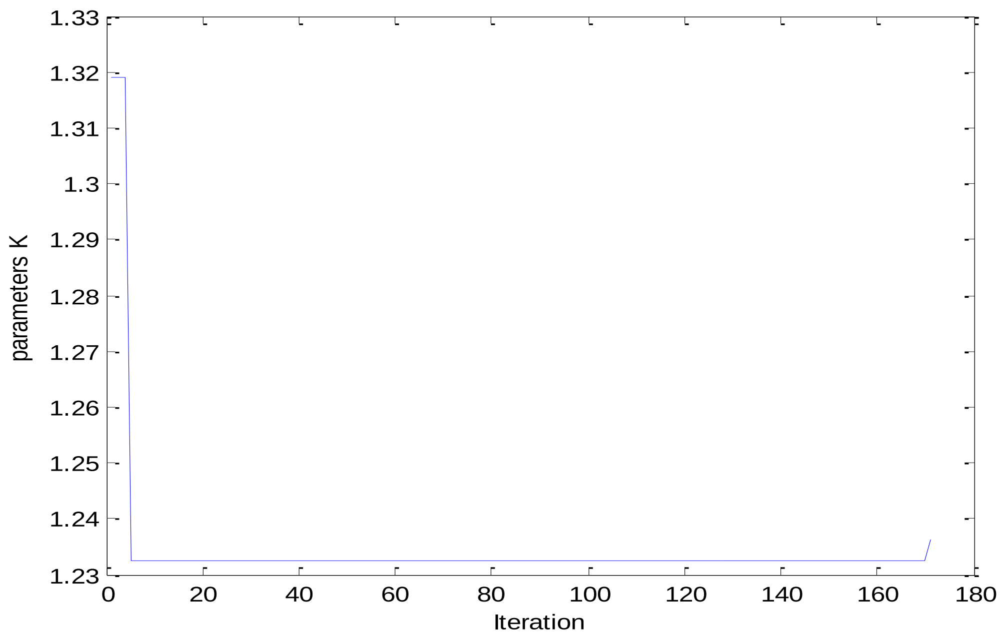


Figure 4.11 Variation of back emf/torque constant with iterations

4.8 Conclusion

Normally, the electrical parameters of DC motor can be determined from the experimental setup. However, it is difficult to determine some of the parameters viz. moment of inertia (J_m), viscous friction (B_m) from the experimental setup. Therefore, in this chapter we have presented the Firefly Algorithm (FFA) to identify the motor parameters. The motor speeds at different armature voltage are determined from the experimental setup and are used in the identification process. We have concluded that the speed response obtained from the simulation nearly the same as those of the testing. Thus, from the speed responses of the motor we confirmed that the parameters obtained from the proposed algorithm are correct. In the next chapter, we will control the motor model developed using PI controller and tune the controller parameters using firefly algorithm.

CHAPTER 5

CONTROLLER DESIGN FOR DC MOTOR USING FIREFLY ALGORITHM

5.1 Introduction

Proportional-Integral-Derivative (PID) controller is the most commonly used controller in industrial applications. In process control, more than 95% of the control loops are P-I-D, most of them are P-I controller. The widespread popularity of PID controllers is because of their robust performance in a wide range of operating conditions and functional simplicity. As the name suggests, a PID algorithm consists of three basic coefficients: proportional, integral and derivative. These coefficients are varied to achieve a desired system response. Practically all P-I-D controller made today on microprocessor.

In this chapter, we have discussed different tuning methods of P-I-D controller and compares their performance with firefly algorithm based tuning method.

5.2 PID Controller

PID controller consists of three types of control i.e. Proportional, Integral and Derivative control. The basic structure of PID is shown in the figure 5.1. The output is measured by the sensor and compares it with the reference input and produces the error. The error is sent to the controller and controller will perform integral and derivative action on it. Thereafter, the signal which is the output of the controller is equal to the sum of [the product of proportional gain and magnitude of error], [the product of integral gain and integral of error], [the product of derivative gain and derivative of error]. The output of the controller is sent to the input of the system and it changes some aspect of the system. For example, if you are controlling a motor, the controller would provide more or less current. The system output is then measured again and the whole process repeats. The output of the controller is defined as follows:

$$u = (Kp * error + \mathrm{Ki} \times \int Error \; dt + K_d \times \frac{d(Error)}{dt})$$

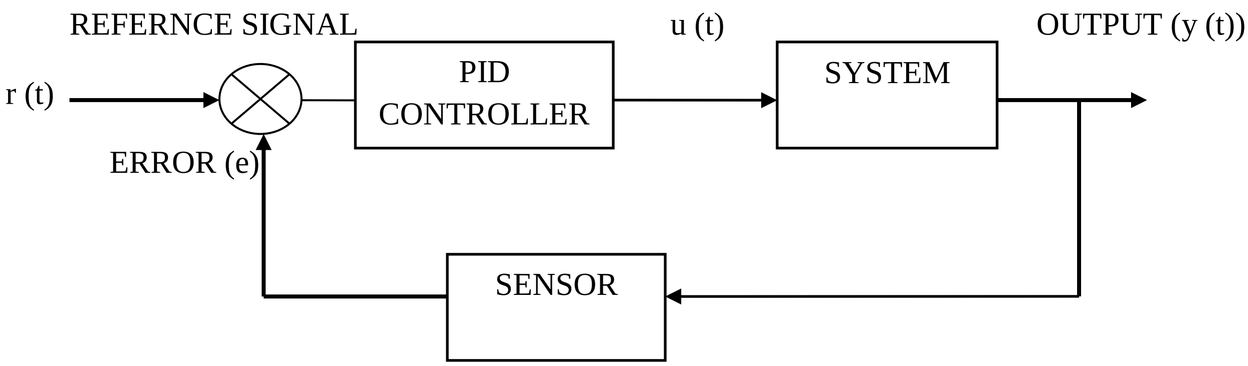


Figure 5.1 Basic structure of PID controller

5.2.1 Proportional Controller (P)

The proportional controller output (u) is proportional to the system error (e = r (t)-y (t)) to control the system. It exerts a positive command to the system when r (t) > y (t) and negative command when r (t) < y (t). Primarily, P controller is only responsible for the responsiveness for the PID controller.

$$u = K_p \times Error$$

where,

K_p= Proportionality constant

5.2.2 Integral Controller (I)

The integral controller accumulates the errors, and generates its output according to the values of the errors at that particular instant. Therefore, this controller produces the output even when the r (t) = y (t) if the error accumulated at that time is not zero. If the accumulated errors are positive, the controller drives the system response to overshoot and if the accumulated errors are negative, the controller will drive the response to undershoot. These two of the conditions are undesirable as these make the system unstable. However, Integral controller reduces the steady state error under the condition of disturbances in the system.

$$u = K_i \times \int Error \; dt$$

5.2.3 Derivative Controller (D)

The derivative controller output is proportional to the rate of change of the error (e = r (t) – y (t)). If the reference input signal is constant, $e = [-y(t)]$, and the output of the controller will be $u\ (t) = -K_D \dot{y}\ (t)$. Therefore, the D controller will act opposite to the motion, similar to the damper used in mass-spring-damper system. Thus, D controller is used to reduce or eliminate overshoot and improves the transient stability of the system.

$$u\ = K_D \times \frac{d(Error)}{dt}$$

5.2.4. Standard Transfer Function for PID Controller

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

For $P = K_p$,

$I = {}^{Ki}/_{S}$, and $D = \ K_D s$

So, $G_c(s) = \ P + I + D = Kp + {}^{K_i}/_{S} + K_D\ s$

$$G_c\ (s) = \ K_p(1 + \frac{1}{T_i s} + T_d\ s)$$

where, K_p is the proportional gain, K_i is the integration coefficient and K_D is the derivative coefficient. T_d is known as the integral action time or reset time and T_d is the derivative action time or rate time.

5.3 Characteristics of PID Controller

The proportional gain (K_p) reduces the rise time and might reduce or eliminate the steady state error. The effect of (K_i) to eliminate the steady state error but it makes the transient response worse. The derivative gain (K_D) improves the transient response of the system in terms of maximum percentage overshoot, stability. The effect of K_p, K_i, K_D on closed loop system is shown in Table 5.1. Thus, the type of controller is selected on the basis of the desired performance of the system.

Table 5.1 Effect of controller parameters on system performance

Controller parameters	**Rise time**	**Overshoot**	**Settling time**	**Steady state error**
K_p	Decrease	Increase	Small change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_D	Small change	Decrease	Decrease	Small change

5.4 Closed Loop Tuning Methods

Tuning of the controller is to determine the parameters of the controller K_p ,T_i , and T_d according to the system requirements. The control system performs poor in characteristics and even it becomes unstable, if improper values of the controller tuning parameters are used. So, it becomes necessary to tune the controller parameters to achieve good control performance with the proper choice of tuning constants. There are following techniques for tuning of the controller parameters:

5.4.1 Ziegler-Nichols Tuning Method

The Z-N method was introduced by John G. Ziegler and Nathaniel B. Nichols in the 1940s. According to the method, initially set the controller to P mode only. Next, set the gain of the controller (K_p) to a small value. Make a small set point (or load) change and observe the response of the controlled variable. If K_p is low the response would be sluggish. Increase K_p by a factor of two and make another small change in the set point or the load. Keep increasing K_p (by a factor of two) until the response becomes oscillatory. Finally, adjust K_p until a response is obtained that produces continuous oscillations. This is known as the ultimate gain (K_u). Also note the period of the oscillations (P_u). The K_u is the gain margin of the system and;

$$P_u = \frac{2.\pi}{wcg}$$

where, wcg the is the gain crossover frequency.

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

Table 5.2 Ziegler-Nichols Closed Loop Method

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

5.4.2 Tyreus-Luyben Tuning Method

The Tyreus-Luyben tuning method is quite similar to the Ziegler–Nichols method but the final controller settings are different. Also this method only proposes settings for PI and PID controllers. These settings that are based on ultimate gain and period of oscillation is shown in the Table 5.3.

Table 5.3 Tyreus-Luyben Closed Loop Method

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

5.4.3 Astrom-Hagglund Tuning Method

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

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

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

Table 5.4 Astrom-Hagglund Closed Loop Method

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

5.5 Methodology

The initial value of the parameters of the PI controller (K_p, T_i) are taken as randomly and used the transfer function of the DC motor determined from the identification process. Thereafter, we applied the firefly algorithm to tune the parameters of PI controller according to our desired response criteria and also compared the results with the classical tuning methods.

5.6 Algorithm for Controller Design of DC Motor using Firefly Algorithm

Step 1. Initialize the parameters of firefly algorithm (FFA) viz. α (randomness), β (attractiveness), γ (light absorption coefficient), δ(randomness reduction parameter), number of fireflies n, maximum iteration, stopping criteria.

Step 2. Define the objective function f(x).

Step 3. Select the range for the objective function parameters.

Step 4. Generate the initial value x_i , where (i=1: number of model parameters) of the model parameters randomly.

Step 5. Determine function value I_i at x_i , where (i=1: number of model parameters) by the value of $f(x_i)$.

Step 6. Arrange the model parameters according to their function values.

Step 7. Set iteration count, iter = 1

Step 8. If I_i<I_j where i, j =1: n, the parameter values will change according to the equation as follows:

$$x_i = x_i + \beta_0 e^{-\gamma {r_{ij}}^2}(x_j - x_i)+\alpha(rand - 1/2)$$

Step 9. Find the value of function at new values of x_i

Step 10. Check stopping criteria and iter > maximum iteration, if it is satisfied GOTO step 14.

Step 11. Modify randomness parameter α according to the formula given below:

$$\alpha = \delta * \alpha.$$

Step 12. Advance the iteration count, iter = iter +1 and GOTO step 9.

Step 13. Find the optimal value of the function and its associated parameters.

5.7. Results

The comparison of the results of the system using firefly algorithm and classical tuning method for PI controller is given below in Table 5.5 when user defined parameters are taken as α=0.95, β=1.0, γ=0.1, δ=1.0.

Table 5.5 Comparison of classical and firefly algorithm tuning method

Tuning methods	Rise time (seconds)	Settling time (seconds)	Overshoot (percentage)
Ziegler-Nichols Method (Z-N)	0.0258	0.1185	28.5926
Tyreus-Luyben Method	0.0457	0.1289	3.3052
Astrom-Hagglund Method	0.5058	2.3065	0
Firefly algorithm	0.0699	0.1116	0.8533

The comparison of closed loop system response of DC motor using classical tuning methods viz. Ziegler-Nichols, Tyreus-Luyben, Astrom- Hagglund and Firefly algorithm is shown in the Figure 5.2.

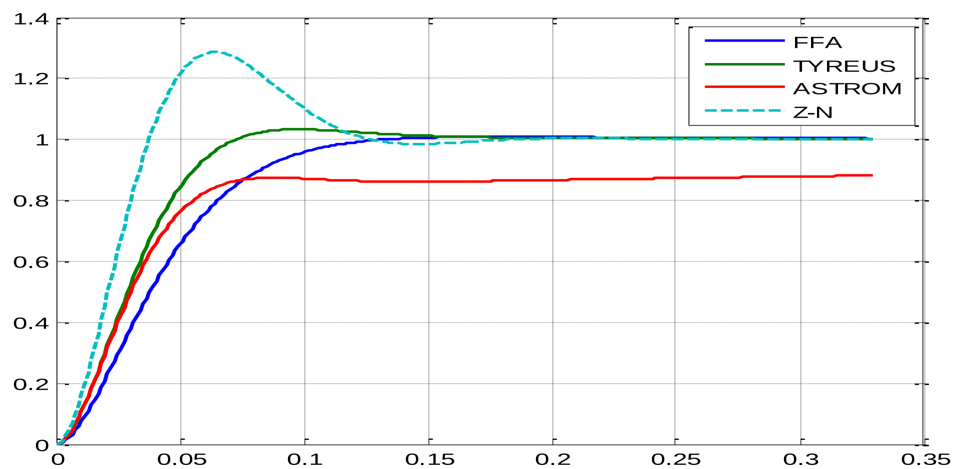


Figure 5.2 Comparison of Closed loop response of DC motor using classical and FFA methods

5.8 Conclusion

In this chapter, we have devised a PI controller for the identified transfer function of the DC motor model using classical and firefly algorithm. Thus, the result shows that after using the proposed algorithm, closed loop response of DC motor are far better than the classical tuning techniques in terms of rise time, settling time and maximum overshoot.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The proposed algorithm is applied to different standard benchmark function viz. Sphere, Griewank, Rotated hyper ellipsoid and Ackley function to find the efficacy of the algorithm. Thereafter, we build the model of the armature controlled DC Motor from its input and output relationship using transfer function approach and then apply firefly algorithm (FFA) on it to optimize its parameters (viz. armature resistance, armature inductance, moment of inertia, viscous friction, emf/torque constant). Finally, we tuned the response according to our design requirement using FFA and then compared its performance with different classical tuning techniques and concluded that FFA works better than existing tuning techniques.

6.2 Future Scope of the Work

Future considerations are depicted as follows:

i. Further we can consider the regulatory response of the DC motor to optimize the model parameters.
ii. We can also incorporate nonlinearity in the system by introducing load torque as a function of speed.
iii. We can also compare the result of parameter identification of dc motor with the existing algorithm like genetic algorithm (GA), bacterial foraging optimization algorithm (BFOA) etc.
iv. Modified or improved approach (like variant of Firefly Algorithm) can also be applied for the DC motor parameter identification purpose.
v. Hybridization with different algorithm like GA, BFOA can also be used for parameter identification of DC motor to improve its performance.

REFERENCES

[1] Yang, X.S., “Firefly algorithm, Stochastic test functions and design optimization,” International Journal of Bio-Inspired Computation, vol.2, No.2, pp.78-84, 2010.

[2] Chai-ead, N., Aungkulanon, P., and Luangpaiboon, P., “Bees and Firefly Algorithms for Noisy Nonlinear Optimisation Problems,” In Proceeding of the International Multi Conference of Engineers and Computer Scientists, vol.2, pp.1449-1454, March 2011.

[3] Yang, X.S., “Engineering Optimization: An Introduction with Metaheuristic Applications,” ISBN: 978-0-470-58246-6, Wiley 2010.

[4] Molga, M., and Smutnicki, C., “Test functions for optimization needs,” *http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf* , 2005.

[5] Krishnan, R., “Electric Motor Drives: Modeling, Analysis, and Control,” ISBN 0-13-091014-7, 2001.

[6] Ljung, L., “System Identification: Theory for the User,” ISBN: 0-13-881640-9, 1987.

[7] Krneta, R., Antic, S., and Stojanovic, D., “Recursive least squares method in parameters identification of DC motors models,” *Facta Universitatis Series: Electronics and Energetics*, vol.18, No. 3, pp. 467-478, Dec. 2005.

[8] Saab, S. S., and Kaed-Bey, R. A., “Parameter Identification of a DC Motor: An Experimental Approach,” *IEEE International Conference on Electric Circuit and Systems*, vol.4, pp. 981-984, 2001.

[9] Saher A., Saima Q., Tayba J., Ahsan M., “Parameter Estimation of a DC Motor using Ordinary Least Squares and Recursive Least Squares Algorithms,” *proceeding of 8th International Conference on Frontiers of Information technology,* Article no. 31, ISBN: 978-1-4503-0342-2, Dec. 2010.

[10] Nassef, M. G. A., Li, L., Schenck, C., and Kuhfuss, B., "Short Time Identification of Feed Drive Systems using Nonlinear Least Squares Method," World Academy of Science, Engineering and Technology, vol.59, pp.1643-1648, 2011.

[11] Eker İ., Vural A., and Susluoglu B., “Experimental identification of an electromechanical system running in open-loop conditions,” *International Conference on Electrical and Electronics Engineering, ELECO,* pp.284-288, 2003.

[12] Thananchai, L., “Model-based Analysis for Experimental Parameter Identification of Micro DC Motor,” *Thammasat International Journal of Science and Technology*, vol.13, No.1, pp.38-46, March 2008.

[13] Wu, W., “DC Motor Parameter Identification Using Speed Step Responses,” *Hindawi Publishing Corporation Modeling and Simulation in Engineering,* vol.2012, doi: 10.1155/2012/189757, pp.1-5, Article ID 189757, 2012.

[14] Kara, T., and Eker, I., “Nonlinear modeling and identification of a dc motor for bidirectional operation with real time experiments,” *Energy Conversion and Management, vol.45*, pp. 1087–1106, 2004.

[15] Tutunji, T., “DC motor identification using impulse response data,” In Computer as a Tool, 2005, *EUROCON*, vol.2, pp.1734–1736, Nov. 2005.

[16] Hadef M., and Mekideche, M. R., “Parameter Identification of a separately excited dc motor via inverse problem methodology,” *Turk Journal of Electrical and Computer Science*, vol.17, No.2, pp. 99-106, 2009.

[17] Hadef, M., Bourouina, A., and Mekideche, M. R., “Parameter Identification of a DC Motor via Moments Method,” *Iranian Journal of Electrical & Computer Engg*, vol. 7, No.2, pp.159-163 Summer-Fall 2008.

[18] Mounir, H., and Mekideche, M. R., “Moments and Pasek’s methods for parameter identification of a DC motor,” *Journal of Zhejiang University SCIENCE* C, vol. 12, No. 2, pp.124-131, 2011.

[19] Becedas, J., Mamani, G., and Feliu V., “Algebraic parameter identification of dc motors: Methodology and analysis,” *International Journal of Systems Science.*, vol. 41, No.10, pp. 1241–1255, 2010.

[20] Knudsen, M., and Jensen, J. G., “Estimation of nonlinear DC-motor models using a sensitivity approach,” *In Proceeding of 3rd European Control Conference*, vol. 95, pp.319–325, 1995.

[21] Saab, S. S., and Kaed-Bey, R. A., “Parameter Identification of a DC Motor: An Experimental Approach,” *Proceeding at the 8th IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, pp. 981-984, 2001.

[22] Tjahjowidodo, T., Al-Bender, F., and Van Brussel, H., “Friction identification and compensation in a DC motor,” *Proceeding of 16th IFAC World Congress, Prague,* Paper ID: 4017, Jul. 2005.

[23] Mamani, G., Becedas, J., Feliu-Batlle, V., and Sira-Ramírez, H., “Open- and closed-loop algebraic identification method for adaptive control of DC motors,” *International Journal of Adaptive Control and Signal Processing*, vol. 23, No. 12, pp. 1097–1103, 2009.

[24] Dub, M., and Jalovecky, R., “DC motor experimental parameter identification using the Nelder-Mead simplex method,” *In Proceedings of 14th Power Electronics and Motion Control Conference (EPE-PEMC),* pp.S4-9, Sep. 2010.

[25] Basil, M., M., “System Identification Algorithm for Systems with Interval Coefficients,” *Journal of Engineering*, vol. 18, pp-239-246, Feb. 2012.

[26] Kiran Raj, A. S., Manoj, K., Mansi, S. S., Prem K., Suchithra, R., and Varun K., “LabVIEW Based PID Speed Control and System Identification of a PMDC Motor,” *International Conference on Computing and Control Engineering,* ISBN-978-1-4675-2248-9, April, 2012.

[27] Ganesh, C., Abhi, B., Anand, V. P., Aravind, S., Nandhini, R., and Patnaik, S. K., “DC Position Control System–Determination of Parameters and Significance on System Dynamics,*” ACEEE International Journal on Electrical and Power Engineering*, vol. 03, No. 01, pp.1-5, Feb 2012.

[28] Bhushan, B., and Singh, M., “Adaptive Control of Nonlinear Systems Using Bacterial Foraging Algorithm,” *International Journal of Computer and Electrical Engineering,* vol.3, No.3, pp.335-342, June 2011.

[29] Udomsuk, S., Areerak, K. L., Areerak, K. N., and Srikaew, A., “Parameter Identification of Separately Excited DC Motor using Adaptive Tabu Search Technique,” *In Advances in Energy Engineering*, IEEE, pp.48-51, June 2010.

[30] Yassin, I. M., Taib, M. N., Abdul Aziz, M. Z., Abdul Rahim, N., Tahir, N. M., and Johari, A., "Identification of DC motor drive system model using Radial Basis Function (RBF) Neural Network," *IEEE Symposium on Industrial Electronics and Applications* pp.13-18, Sep. 2011.

[31] Rahim, N. A., Taib, M. N., Adom, A. H., and Mashor, M. Y., “The NARMAX model for a DC motor using MLP Neural Network,” *Proceeding of the First International Conference on MAN- MACHINE SYSTEMS*, pp.61-65, 2006.

[32] Feilat, E. A., and Maaitah, E. K., “Identification and Control of DC Motor Using RBF Neural Network Approach,” *Proceeding of International Conference on Communication, Computer and Power (ICCCP 09), IEEE Press*, Muscat, pp.258-264, Feb.2009.

[33] Cong, S., Li, G., and Feng, X., “Parameter identification of nonlinear DC motor model using compound evolution algorithm,” *Proceeding of the World Congress on Engineering*, vol.1, pp.15-20, July 2010.

[34] Al-qassar, A. A., and Othman, M. Z., “Experimental determination of electrical and mechanical parameters of DC motor using Genetic Elman neural network,” *Journal of Engineering Science and Technology*, vol.3, No.2, pp.190-196, 2008.

[35] Rubaai, A., and Kotaru, R., “Online Identification and Control of DC motor using Learning Adaptation of Neural Networks,” *IEEE Transaction on Industry application,* vol. 36, No.3, pp.935-942, June 2000.

[36] Weerasooriya, S., and El- Sharkawi, M. A., “Identification and control of a dc motor using back propagation Neural Networks” *IEEE Transaction on Energy Conversion*, vol.6, pp.663-669, Dec.1991.

[37] Narendra, K. S., and Parthasarathy, K., “Identification and Control of Dynamical Systems Using Neural Networks,” *IEEE Transaction on Neural Networks*, vol. 1, pp.4-27, March 1990.

[38] Lankarany, M., and Rezazade, A., “Parameter Estimation Optimization Based on Genetic Algorithm Applied to DC Motor,” *International Conference on Electrical Engineering, IEEE*, pp.1-6, 2007.

[39] Liu, X. Q., Zhang, H. Y., Liu, J., and Yang, J., “Fault detection and diagnosis of permanent magnet DC motor based on parameter estimation and Neural Network,” *IEEE Transactions on Industrial Electronics*, vol.47, No. 5, pp.1021-1030, 2000.

[40] Tzes, A., Peng, P. Y., and Guthy, J., “Genetic based fuzzy clustering for DC motor friction identification and compensation” *IEEE Transaction on Control System Technology*, vol.6, No.4, pp.462-472, 1988.

[41] Rezazadeh, A., “Genetic Algorithm based Servo System Parameter Estimation during Transients,” *Advances in Electrical and Computer Engineering*, vol.10, pp.77—81, Nov.2010.

[42] Maboud, Arash, “Dynamical and Electrical parameter identification of DC motor using Genetic Algorithms,” *The International Conference on Electrical Engineering*, No.P-009, Japan, July 2008.

[43] Orlowska- Kowalska, T., Szabat, K., “Application of Genetic Algorithms to parameter identification of DC motor drives,” *Prace Naukowe Instytutu Maszyn, Napędów i Pomiarów Elektrycznych Politechniki Wrocławskiej. Studia i Materiały*, vol.56, No.24, pp.159-170, 2004.

[44] Astrom, K., and Hagglund, T., “PID Controllers: Theory, Design, and Tuning Rules,” 2nd Edition, Instrument Society of America, ISBN: 155-617-516-7, 1995.