CHAPTER 5
Optimization of PID Controller and Velocity Control of DC Motor Using Modified Interactive Evolutionary Computing (MIEC) Algorithm

This chapter concentrates on algorithm known as Modified Interactive Evolutionary computing (MIEC), optimization of controller parameter for speed control of a DC Motor. The same mathematical model developed in the previous chapter for GA implementation is again used. Also brief description about the various steps of design involved for implementation of MIEC is presented. The development of MIEC and its implementation for velocity control is presented. The comparison of GA optimized results with MIEC for five different error models are presented.

5.1 Introduction

Every industrial segment needs accurate and precise system functionalities with high accuracy and better performance. Similarly in mechanical operations or devices, whose functions are based on electric motors, needs high speed control with optimum accuracy. The control of motor drives becomes very critical in order to perform the desired function and hence to achieve the better plant output.

The effectiveness and efficiency of a DC Motor and hence the electro-mechanical operation depends on the effectiveness of rotating part and hence the motor. Therefore, it has been a prime issue for the control engineering to develop an optimal way to control the velocity of motor and control in its operational orientation. Many algorithms have been deployed, tried, analyzed and reengineered to get better performance. The previous chapter deals with implementation of GA for obtaining the optimization in the speed control, but still there is certain possibility for getting better performance. For achieving this, implementation is required to design the DC Motor with its functional parameters. A
large number of research processes are going on to achieve the optimized speed/velocity control.

In the previous chapter successful implementation of Genetic Algorithm (GA) for optimizing the system control and up to a limit of some satisfactory results have been obtained. Still there are many situations where the GA implementation may not be effective and there is an intention to increase our vision towards some more effective algorithms, and thus it has paved the way to think about the dominant technology called MIEC.

The designing of a motor consists of many parameter optimization and thus to optimize the system performance by adopting number of modern control methodologies. Few prominent methodologies are like adaptive control [2], Non-Linear control [3], and optimal control [4] which has been discussed earlier and then they have been tried to get better optimized measures for system control. Although it has been studied but practically these approaches are very complex and complicated for implementation. The motor control based on proportional, integral and derivative (PID) controllers has been largely studied and implemented with success. Here, in the present scenario it has been emphasized for getting a better way for controlling the time domain performance parameters of a DC Motor to achieve better speed control. The present work deals with the analysis of the time domain performance parameters to control and achieve the speed control of a DC Motor. The control gain is a key parameter which determines the minimized performance or the maximized performance. Therefore, in order to get the better optimized results, here the optimization techniques to get the desired output are employed.

In order to achieve better performance, many optimization algorithms like GA have been employed but further study has revealed that there are some more options through which the control system can be further optimized. There are certain limitations of the GA implemented system architecture because the performance factors are not optimized for every situation, therefore it has motivated us for using some other better functional algorithms like Interactive Evolutionary Computing (IEC). Here we are employing the
Modified form of Interactive Evolutionary Computing algorithm to get the better performance and to minimize the errors in time domain performance s-parameters of DC Motor.

The modeling and the system simulation has been carried out on the MATLAB and SIMULINK software tool. The present work represents a case for system optimization for DC Motor speed control. Here, the objective is to emphasize on the deployment of the MIEC Algorithm for achieving the optimized system control and better performance. Since the mathematical modeling, analysis and the data or result interpretation is one of the dominant key factors for optimization. Hence here also, initially the mathematical model for the DC Motor has been derived and then based on the parameters achieved further implementation on the MATLAB software has been done. The mathematical model developed in the previous Chapter 4 for the implementation of GA is considered again.

5.2 Implementation of Modified Interactive Evolutionary Computing (MIEC) for Optimization

5.2.1 General Introduction: Interactive Evolutionary Computing

Interactive Evolutionary Computation (IEC) or aesthetic selection is a general term used for method of Evolutionary Computation (EC) that uses human evaluation. Usually human evaluation is necessary when the form of fitness function is not known (for example, visual appeal or attractiveness; as in Dawkins, 1986) or the result of optimization should fit a particular user preference (for example, taste of coffee or color set of the user interface).

The interactive EC is a technology that optimizes the system based on human subjective evaluation. Simply stated, the EC fitness function is replaced by a human. Humans have two aspects: knowledge and KANSEI. Conventional Artificial Intelligence (AI) has mainly focused on the former. KANSEI is the total concept of intuition, preference, subjectivity, sensation, perception, cognition, and other psychological
processing functions. The interactive EC is a technology that embeds the KANSEI into system optimization.

For example, suppose we wish to tune a music synthesizer to create a timber between a violin and clarinet or we wish to create graphic art that matches the emotion of our living room. Since these tasks can be seen as the optimization parameter of the music synthesizer and computer graphics (CG), we can apply numerical optimization techniques to the tasks. For these cases, there is no measure for evaluation of the optimization techniques except that could be measured in the human mind. Interactive EC is the optimization technique based on the subjective scale. Humans evaluate the distance between the goal and a system output in psychological space. On the other hand, EC searches in a parameter space. The interactive EC is a system in which both human and EC cooperatively optimize a target system based on mapping relationship between the two spaces.

Interactive evolutionary computing is a technique from the class of evolutionary algorithms (EA) whose fitness function is replaced by a human. A human plays the role of fitness function and selects one or more individual(s) which survive(s) and reproduces to constitute a new generation. IEC takes the advantage of EA and human knowledge and intuition in selecting the individual(s).

**A. Evolution Strategy**

Search procedures that mimic with natural evolution of the species in a natural system are coined as Evolutionary strategies [6]. Like GA, they require data based on objective function and constraints, and not the derivatives or other auxiliary knowledge. Evolution Strategies (ESs) were developed by [7], with selection, mutation and a population of size one.
**B. Selection**

For a given generation, parents and off springs are generated by recombination and mutation. They are sorted according to user defined criteria and the user selects the best to become parents for the next generation.

**C. Reproduction**

Next set of population for the successive generation is generated by a process called reproduction and this involves crossover (recombination) and mutation. This result in a new set of population derived from the selected solutions of the previous population. Generally, the average fitness of the population is increased as compared to the previous population.

**D. Termination**

The process of optimization is stopped once a required criterion is achieved. The termination condition can be based on either the number of generations or the solution satisfying an optimum criterion.

**E. Modification to IEC**

As in automated Evolutionary optimization algorithms, IEC does not use many individuals and is not required to be iterated over many generations. The limitation arises because the decision maker is human and due to human fatigue, the process cannot be iterated over many generations. Hence, MIEC is developed which efficiently searches with a few individuals and a few generations. Moreover, selection of individuals as done by a human is also incorporated in the software in an automatic way by selecting the best four individuals from the population to participate in next generation. The fitness function results are sorted out from a generation and the four individuals with least value of the fitness function are allowed to participate in the next generation. This overcomes the limitation of the human fatigue and iterations over the concept where a few generations can be performed and much better optimizations can be made.
5.3 Implementation of MIEC in Motor Control

In order to implement the Interactive Evolutionary model for the motor control mechanism, the system development and its implementation has been done similar to that of GA but with few of the Interface and the interactivity will be different.

The System implementation based on the MIEC can be summarized as follows:

(i) System Equation development.

(ii) Proposed algorithm implementation.

(iii) Parameter optimization using various error models.

(iv) Choice of a model and velocity control of a DC Motor using MATLAB / SIMULINK implementation.

(v) Harmonic Estimation.

In order to achieve the required goal initially we have to design the motor model where the implementation with the MIEC algorithm has to be performed.

5.3.1 Motor Model

Motor model is developed using MATLAB / SIMULINK. The parameters for PID Controller are the individuals in MIEC population. For each individual, the SIMULINK model is run and the error is computed. Four individuals with minimum errors are selected for generating the next generation population. Different error models are used for the evaluation. The feedback loop consists of a current sensor in the current loop (inner loop) and the tacho-generator in the velocity loop (outer loop). Generally the reference voltage to the motor $V_a$ is fed through a PWM controller. The PWM controller consists of a set of comparators that compares a reference signal with the tacho-feedback signal and generates PWM pulses for an H-bridge converter. The H-bridge converter has been modeled as a single pole function as shown in Fig. 5.1 mentioned below and the desired voltage has been applied to the armature of a motor.
As mentioned earlier for the motor design the requirements are Armature resistance, armature inductance, moment of inertia for motor, system gain parameter, DC gain and the functional frequency for PWM implementation. The parameters being used for modeling the system are mentioned in Chapter four.

5.3.2 PID Controller Design

Fig. 5.2: PID Controller Block architecture
The PID Controller can be represented in terms of its transfer function. The Transfer Function for the PID Controller can be presented as below:

\[
\frac{K_ds^2 + K_ps + K_i}{s(s + \omega_d)}
\]  

............... (5.1)

PID Controller takes error input and computes the proportional, derivative and integral error. Each of these components in the PID Controller is having special characteristics and hence has the responsibility, like the derivative component is responsible for improving the system response, whereas integral and proportional components shape the steady state performance. Generally in MIEC there are four key parameters which are used for the initializing the population. The parameters being used for the initial population are given below in the Table 5.1:

Table 5.1: MIEC Parameter

<table>
<thead>
<tr>
<th>Initial Population Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>K_p</td>
</tr>
<tr>
<td>K_d</td>
</tr>
<tr>
<td>K_i</td>
</tr>
<tr>
<td>\omega_d</td>
</tr>
<tr>
<td>Maximum Generations</td>
</tr>
<tr>
<td>Fitness function</td>
</tr>
</tbody>
</table>

There are many error functions which play a vital role for certifying the system for new generation, i.e. these parameters are significant for analyzing the overall system functionalities and its versatilities. These important error functions are ITSE (Integral Time Squared Error), ITAE (Integral Time Absolute Error), Peak Overshoot, Absolute
Error and Squared Error. The significant minimum and the maximum value presents the robustness of the developed system. These functions are computed for every generation and four individuals with the minimum value of the error are selected to participate for forming the next generation. The definitions of different error functions used are given in Chapter 4.

5.3.3 Development of Simulation Model

The development of simulation model for various components associated are similar to that explained in the modeling of GA implementation chapter 4. The following Table 5.2 represents the transfer function of the associated components.

<table>
<thead>
<tr>
<th>Table 5.2: Transfer Function parameters for the sub-system</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Motor</strong></td>
</tr>
<tr>
<td>G(s) = ( \frac{1}{Js (1+sL_u / R_u)} ); ( H(s) = K )</td>
</tr>
<tr>
<td><strong>Current Sensor</strong></td>
</tr>
<tr>
<td>( (1 + 0.8 \times 10^{-3} s) )</td>
</tr>
<tr>
<td><strong>Tacho Generator</strong></td>
</tr>
<tr>
<td>( (1 + 25 \times 10^{-3} s) )</td>
</tr>
<tr>
<td><strong>H-Bridge Controller</strong></td>
</tr>
<tr>
<td>( (1 + 0.5 \times 10^{-3} s) )</td>
</tr>
<tr>
<td><strong>Current Controller</strong></td>
</tr>
<tr>
<td><strong>PID Controller</strong></td>
</tr>
</tbody>
</table>
5.3.4 Implementation of the Develop Function on the SIMULINK

![SIMULINK model for MIEC Implementation](image)

Fig. 5.3: SIMULINK model for MIEC Implementation

Fig. 5.3 presents the model developed for control optimization through MIEC. Here, input is the angular output itself which has to be optimized for the predefined parameters. The input is fed to PID Controller which has numerator and denominator. The PID Controller calculates an “error” value based on the difference between a measured process variable and a desired set point. And then it attempts to minimize the error by adjusting the process control inputs by choosing one best fitness function among the four best randomly chosen fitness functions.

The output of PID Controller is fed to the current controller which then transfers the output to H-Bridge converter. The H-Bridge converter develops the voltage to be applied across a load in either direction. Enabling of the converter output is forwarded to the DC Motor model which regulates the armature current and its angular velocity. The armature current is then fed back to the current controller through current sensor. And thus the regulated output can be visualized on the output terminal.
5.4 Optimization Results Obtained from Modified Interactive Evolutionary Computing:

5.4.1 Results of Optimization for 500 rpm

The following section presents the results obtained from the MIEC implementation for the optimization purpose of the DC Motor speed control. Here, the graphical results are illustrating the variation of different PID parameters for the different error functions like ITSE, ITAE, absolute error, peak overshoot and the squared error.

![Graph showing motor speed optimized for ITSE](image)

**Fig. 5.4: Motor Speed optimized for ITSE**

Fig. 5.4 shows the plot for the motor speed with respect to time with speed optimization for ITSE error model. The set command speed is 500 rpm and we find that in case of ITSE, initially the motor accelerates from rest and builds up the speed which is slightly higher than required thereafter it stabilizes and establishes the control stabilization within 2 sec. The number of generations required for termination is 29 with small angular variation as compared to other models.
Fig. 5.5: Motor Speed optimized for ITAE

Fig. 5.5 shows the plot for the motor speed with respect to time with speed optimization for ITAE error model. The set command speed is 500 rpm and we find that in case of ITAE, initially the motor accelerates from rest and builds up the speed which is slightly higher than required thereafter it stabilizes and establishes the control stabilization within 5 sec. The number of generations required for termination is 16 with small angular variation as compared to other models.

Fig. 5.6: Motor Speed optimized for Peak Overshoot
Fig. 5.6 shows the plot for the motor speed with respect to time with speed optimization for Peak Overshoot Error model. The set command speed is 500 rpm and we find that in case of Peak Overshoot Error model, initially the motor accelerates from rest and builds up the speed which is slightly higher than required thereafter it stabilizes and establishes the control stabilization within 2 sec. The number of generations required for termination is 8 with small angular variation as compared to Squared Error Model, whereas it is larger when compared to ITSE, ITAE and Squared Error models.

Fig. 5.7: Motor Speed optimized for Absolute Error

Fig. 5.7 shows the plot for the motor speed with respect to time with speed optimization for Absolute Error model. The set command speed is 500 rpm and we find that in case of Absolute Error model, initially the motor accelerates from rest and builds up the speed which is slightly higher than required thereafter it stabilizes and establishes the control stabilization within 3 sec. The number of generations required for termination is 10 with larger angular variation as compared to other types of models.
Fig. 5.8: Motor Speed optimized for Squared Error

Fig. 5.8 shows the plot for the motor speed with respect to time with speed optimization for squared error model. The set command speed is 500 rpm and in case of squared error model, initially the motor accelerates from rest and builds up the speed to the desired value and stabilizes within 2 sec. The number of generation required for termination is 3 with small angular variation as compared to other models. The different optimization functions and the resulting PID Controller transfer function are listed below for various types of error models:

1. **ITSE (Integral Time Square Error)**: It penalizes large errors more than small. This is the cumulative sum of the error which is represented in the form of transfer function as follows:

\[
\text{PID} = \frac{4.195s^2 + 4.576s + 0.397}{s(s+1)}\\
\text{............... (5.2)}
\]
2. **ITAE (Integral Time Absolute Error):** When there is sum of areas above and below the set-point, this penalizes all errors equally regardless of direction. The ITAE value for the PID can be represented as the following transfer function:

\[
PID = \frac{9.206s^2 + 9.811s + 5.943}{s(s+1)}
\]

\[\text{…………… (5.3)}\]

3. **Absolute Error:** It can be represented as the following transfer function:

\[
PID = \frac{2.823}{s(s + 3.147)}
\]

\[\text{…………… (5.4)}\]

4. **Squared Error:** It can be represented as the following transfer function:

\[
PID = \frac{6.519s + 7.180}{s(s + 1)}
\]

\[\text{…………… (5.5)}\]

5. **Peak Overshoot:**

\[
PID = \frac{1.712s^2 + 2.654s + 1.002}{s(s + 1.905)}
\]

\[\text{…………… (5.6)}\]

<table>
<thead>
<tr>
<th>Table 5.3: Optimized PID parameters of MIEC for 500 rpm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>ITSE</td>
</tr>
<tr>
<td>ITAE</td>
</tr>
<tr>
<td>Absolute Error</td>
</tr>
<tr>
<td>Squared Error</td>
</tr>
<tr>
<td>Peak Overshoot</td>
</tr>
</tbody>
</table>
Table 5.3 shows the different values for the different error functions. These values correspond to the control gain parameter $K$ for each component proportional, integral and derivative. The corresponding generation value has been considered for each error function and finally the squared error value with its overall PID parameters and generation number have been given.

5.4.2 Optimization Results for 1000 rpm

The following sections presents the results obtained from the MIEC implementation for the optimization purpose of the DC Motor speed control. Here, the graphical results are illustrating the variation of different PID parameters for the different error functions like ITSE, ITAE, peak overshoot, absolute error and the squared error. Here the set command speed is 1000 rpm.

![Graph of Motor Speed optimized for ITSE](image)

**Fig. 5.9: Motor Speed optimized for ITSE**

Fig. 5.9 shows the plot for the motor speed with respect to time with speed optimization for ITSE error model. The set command speed is 1000 rpm and in case of ITSE, initially the motor accelerates from rest and builds up the speed which is slightly higher than required thereafter it stabilizes and establishes the control stabilization within
12 sec. The number of generations required for termination is 4 with small angular variation as compared to other models.

Fig. 5.10: Motor Speed optimized for ITAE

Fig. 5.10 shows the plot for the motor speed with respect to time with speed optimization for ITAE error model. The set command speed is 1000 rpm and in case of ITAE, initially the motor accelerates from rest and builds up the speed which is slightly higher than required thereafter it stabilizes and establishes the control stabilization within 8 sec. The number of generations required for termination is 26 with small angular variation as compared to other models.

Fig. 5.11: Motor Speed optimized for Peak Overshoot
Fig. 5.11 shows the plot for the motor speed with respect to time with speed optimization for Peak Overshoot Error model. The set command speed is 1000 rpm and we find that in case of Peak Overshoot Error model, initially the motor accelerates from rest and builds up the speed which is slightly higher than required thereafter it stabilizes and establishes the control stabilization within 4 sec. The number of generations required for termination is 5 with small angular variation as compared to other models.

Fig. 5.12: Motor Speed optimized for Absolute Error

Fig. 5.12 shows the plot for the motor speed with respect to time with speed optimization for Absolute Error model. The set command speed is 1000 rpm and in case of Absolute Error model, initially the motor accelerates from rest and builds up the speed without any overshoot and it stabilizes within 3 sec. The number of generations required for termination is 2 with larger angular variation as compared to other types of models.
Fig. 5.13: Motor Speed optimized for Squared Error

Fig. 5.13 shows the plot for the motor speed with respect to time with speed optimization for squared error model. The set command speed is 1000 rpm and in case of squared error model, initially the motor accelerates from rest and builds up the speed to the desired value and stabilizes within 3 sec. The number of generations required for termination is 5 with small angular variation as compared to other models. The study of various functions like ITSE, ITAE, overshoot value, absolute value and the squared value have been obtained with various numbers of generations and are given below:

1. **ITSE (Integral Time Square Error):** The transfer function for ITSE value is as given below:

   \[
   \text{PID} = \frac{1.1576s^2 + 1.5154s + 0.3201}{s(s+1)} \quad \text{................ (5.7)}
   \]

2. **ITAE (Integral Time Absolute Error):** The transfer function of IATE value is given below

   \[
   \text{PID} = \frac{2.4059s^2 + 3.1796s + 0.5360}{s(s + 1.7307)} \quad \text{............. (5.8)}
   \]
3. **Peak Overshoot**: The transfer function of peak overshoot value is given by

\[
\text{PID} = \frac{1.3452s^2 + 2.235s}{s(s+1)} \quad \text{…………………… (5.9)}
\]

4. **Absolute Error**: The transfer function of absolute error value is given by

\[
\text{PID} = \frac{0.7315s^2 + 2.1451s}{s(s + 4.6279)} \quad \text{…………… (5.10)}
\]

5. **Squared Error**: The transfer function of squared error is given by

\[
\text{PID} = \frac{1.3452s^2 + 2.2035s}{s(s+1)} \quad \text{…………… (5.11)}
\]

### Table 5.4: Optimized parameters of MIEC for 1000 rpm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(\omega_d)</th>
<th>(K_i)</th>
<th>(K_p)</th>
<th>(K_d)</th>
<th>Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITSE</td>
<td>1</td>
<td>0.3201</td>
<td>1.5154</td>
<td>1.1576</td>
<td>4</td>
</tr>
<tr>
<td>ITAE</td>
<td>1.7307</td>
<td>0.5360</td>
<td>3.1796</td>
<td>2.4059</td>
<td>26</td>
</tr>
<tr>
<td>Peak Overshoot</td>
<td>1</td>
<td>0</td>
<td>2.2035</td>
<td>1.3452</td>
<td>5</td>
</tr>
<tr>
<td>Absolute Error</td>
<td>4.6279</td>
<td>0</td>
<td>2.1451</td>
<td>0.7315</td>
<td>2</td>
</tr>
<tr>
<td>Squared error</td>
<td>1</td>
<td>0</td>
<td>2.2035</td>
<td>1.3452</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.4 shows the different values for the different error functions. These values correspond to the control gain parameter \(k\) for each component, Proportional, integral and derivative. The corresponding generation value has been considered for each error.
function and finally the squared error value with its overall PID parameters and generation number have been given. It shows that the number of generations required reduces drastically and expected control may be achieved faster. The controller parameters are also smaller along with their angular speed. Very few generations are required for squared error optimization than that of ITSE, ITAE, Absolute error and Peak Overshoot model Optimization. But with Absolute Error model based optimization $K_i$ & $K_d$ values becomes zero, only $K_p$ of smaller value is required and angular speed is larger than the other cases.

The following Table 5.5 presents the details about the number of generations required to stabilize the controller for the speed of 500 rpm and 1000 rpm.

<table>
<thead>
<tr>
<th>Type of Error model</th>
<th>No. of generations for 500 rpm</th>
<th>No. of generations for 1000 rpm</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITSE</td>
<td>29</td>
<td>4</td>
</tr>
<tr>
<td>ITAE</td>
<td>16</td>
<td>26</td>
</tr>
<tr>
<td>Peak Overshoot</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Absolute error</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Squared error</td>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>

5.4.3 Velocity Control of DC Motor using MIEC

The following figures show the various characteristic curves that are developed by MIEC based PID tuning to control the speed of a DC Motor. The optimization of controller parameter has been carried out using different types of models such as ITSE, ITAE, Peak overshoot, Absolute Error and Squared Error models. The results are
analyzed based on the controller gain parameters as well as number of generations. Even though less number of generations are required for peak overshoot error model optimization which can provides better performance in MIEC. For the comparison purpose the model chosen for GA implementation is Absolute error model and the same is used for MIEC implementation and also for velocity control of a motor.

**Trapezoidal Speed Profile**

![Trapezoidal Speed Profile](image)

**Fig. 5.14: Speed v/s Time**

Fig. 5.14 shows the motor speed response for a trapezoidal speed profile as a command speed using MIEC method for PID tuning. The speed response developed with controller adopted shows that it follows more closely with the command speed over the entire duration without any delay.

The time response of controller current for the trapezoidal speed command is shown in Fig. 5.15 for the MIEC based PID tuning. It is observed that current through the motor changes in proportion to speed depending on the torque requirement. When the motor accelerates it demands more current and where as decelerate it demands lower current in order to develop the desired value of torque for the drive to operate under specified operating conditions. For steady state operation current drawn is maintained constant as shown.
Motor voltage profile for trapezoidal speed command are presented in Fig. 5.16 for MIEC based PID tuning. There is a linear change in voltage when the motor is either accelerating or decelerating. For constant speed range voltage supplied also remains constant in order to follow the speed command profile considered.
The variation of voltage and current for trapezoidal profile speed command using MIEC based PID tuning is shown in Fig. 5.17. Both of these are varying in such a way that the desired torque is being generated to operate satisfactorily.

**Square wave speed profile (Low-High-Low Transition)**

Motor response for MIEC based tuning for another speed profile such as square waveform is shown in the following figures:

---

**Fig. 5.17 Controller Current, Voltage v/s Time**

**Fig. 5.18: Square Wave Speed Profile (Command Speed & Motor Speed v/s Time)**
Fig. 5.18 shows the motor speed response for a square wave speed profile as a command speed using MIEC method for PID tuning. The speed response developed with controller shows that it follows the command speed over the entire duration without much delay. There is a very higher and smaller values of speed at the point of transition is observed but it is negligible as compared to operating speed.

The time response of motor current for the square wave speed command is shown in Fig. 5.19 for MIEC based tuning. It is observed that current through the motor changes depending on the torque requirement as shown. When the motor accelerates it demands more current and whereas decelerates demands lower current in order to develop the desired value of torque for the drive to operate under specified operating conditions.

![Controller Current for different motor speed commands](image)

**Fig. 5.19 Motor Current & Command Speed v/s Time**
Motor voltage profile for square wave speed command is presented in Fig. 5.20 for MIEC based PID tuning. There is an instantaneous change in voltage when the motor is accelerating or decelerating. For constant speed range voltage demands also remain constant in order to follow the speed command profile.

Fig. 5.20 Motor Voltage & Command Speed v/s Time

Fig. 5.21 Controller Current & Voltage v/s Time
The variation of controller current and voltage are presented in Fig. 5.21. It is found that both are varying instantaneously or remains constant to meet the requirements of a load so that the desired torque is developed to maintain the desired speed as shown.

**Square Wave Speed Profile (High-Low-High Transition)**

Fig. 5.22 shows the motor speed response for a square wave speed profile as a command speed using IEC method for PID tuning. The speed response developed with controller shows that it follows the command speed over the entire duration without any delay, There is a slightly higher and lower values of speed is developed at the point of transition but it is negligible as compared to operating conditions.

![Graph of Motor Speed & Command Speed vs Time](image)

**Fig. 5.22: Motor Speed & Command Speed v/s Time**

The time response of motor current for the square wave speed command is shown in Fig. 5.23 for the MIEC based tuning. It is observed that current through the motor changes according to the torque requirement as shown. While the motor accelerates it demands more current and when decelerates it demands lower current in order to develop the desired value of torque for the drive to operate for the specified operating conditions.
Motor voltage profile for square wave speed command are presented in Fig. 5.24 for MIEC based PID tuning. There is an instantaneous change in voltage when the motor either accelerating or decelerating. For constant speed range voltage demands also remain constant in order to follow the speed command profile. The variation of controller current and voltage are presented in Fig. 5.25. It is found that both are varying instantaneously to meet the requirement of speed as shown in the figure.
Fig. 5.25 Controller Current & Voltage v/s Time

Multiple Transition speed profile

Fig. 5.26: Motor Speed & Command Speed v/s Time

Fig. 5.26 shows the motor speed response for multiple transition speed profile using MIEC method for PID tuning. The speed response developed with controller shows that it follows more closely with the command speed over the entire duration without any delay,
There is a very small change in speed is observed at the point of transition but it is negligible. The time response of motor current and voltages for the multiple transition speed profile command is shown in Fig. 5.27 and 5.28 for the MIEC based PID tuning. It is observed that both current and voltage change instantaneously depending on the torque requirement as shown. When the motor accelerates there is a demand for more current and voltage whereas when it decelerates it demands low current and voltage in order to develop the desired value of torque for the drive to operate under the specified operating conditions.

![Controller Current for different motor speed commands](image)

**Fig. 5.27 Motor Current & Command Speed v/s Time**
Fig. 5.28 Motor Voltage & Command Speed v/s Time

Fig. 5.29 Controller Current & Voltage v/s Time
The time response of motor voltage and currents for the multiple transition of speed is shown in Fig. 5.29. This shows that both are varying in proportion to maintain the desired torque while the drive either accelerates or decelerates.

5.4.4 Comparative studies between MIEC and GA

Table 5.6 presented below represents the comparative result analysis between the two evolutionary computational algorithms such as GA and MIEC. The various errors models considered for these purposes are ITSE, ITAE, Peak overshoot, absolute error and squared error. The results presented in the table below are for two different operating conditions i.e., for (i) 500 rpm and (ii) 1000 rpm. The number of generations that are required for convergence for both GA and MIEC are tabulated for different models. For a speed command of 500 rpm, GA requires 62 generations which is reduced to 29 in case of MIEC and for a speed command of 1000 rpm GA requires 51 generations which is reduced to 4 in case of MIEC for ITSE error model. Similarly for other error models also even for the same two different speeds it is observed that lesser number of generations are required in case of MIEC than that of GA.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Number of generation for 500 rpm</th>
<th>Number of generation for 1000 rpm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIEC</td>
<td>GA</td>
</tr>
<tr>
<td>ITSE</td>
<td>29</td>
<td>62</td>
</tr>
<tr>
<td>ITAE</td>
<td>16</td>
<td>67</td>
</tr>
<tr>
<td>Abs Error</td>
<td>10</td>
<td>24</td>
</tr>
<tr>
<td>Squared Error</td>
<td>3</td>
<td>35</td>
</tr>
<tr>
<td>Peak Overshoot</td>
<td>8</td>
<td>51</td>
</tr>
</tbody>
</table>
From the analysis of above data for the relative error function parameters tabulated confirms that the MIEC based implemented optimization technique likely to provide better performance than the GA implemented control system because it required lesser number generations to converge.

5.5 Conclusion:

IEC is an Evolutionary Algorithm whose fitness function is provided by human users. The main benefit of IEC is that it is an efficient and convenient method to incorporate the prior knowledge of the user into a user-guided optimization and identification problems. Interactive Evolutionary Computing approach is then used for optimizing the PID Controller parameters for the speed control loop of a DC Motor. Major limitation of IEC due to the human intervention can be overcome by automatic selection of individuals to form the population of next generation that is known as MIEC. Different error models used in [8] are considered for optimization and the PID Controller parameters are evaluated. It is found that the MIEC has the better capability in the minimization criteria comparable with GA, takes lesser number of generations of iteration for convergence. Hence MIEC can be used to achieve faster approach for optimization as compared to GA.