الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la Recherche scientifique

Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique
Option : Automatique et informatique industrielle

Réf : ....

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER

Thème

Meta-heuristic Design of fracional order PID controller

Présenté par : Serhani Mostafa Youcef

Soutenu le : Juillet 2019

Devant le jury compose de :

M.A A. MECHGOUG Raihane	**Président**
M.A.A. MEGHERBI Hassina	**Encadreur**
M.NEBBAR Hanan	**Examinateur**

Année universitaire : 2018 / 2019

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique

Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique
Option : Automatique et informatique industrielle

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER

Thème

Meta-heuristic Design of fracional order PID controller

Présenté par :Serhani Mostafa Youcef

Avis favorable de l'encadreur :

Avis favorable du Président du Jury

Cachet et signature

بسم الله الرحمن الرحيم

« يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ »

[المجادلة: 11]

Dédicaces

A ma chère mère, pour ses sacrifies depuis qu'elle mis au
monde,
A mon père m'a toujours soutenu et aide à affronter les difficultés, pour tous
ce qui ont fait pour que je puisse les honneur,
A mes très chères frères , à tout ma grande famille
A tous mes amis .

Je dédie ce modeste travail.

Remerciements

Je remercie Dieu le tout puissant qui ma donneé la volonté et la force pour realiser ce modeste travail.

Je tiens à remercier en premier lieu Madame MAGHERBI .H d'avoire accepté détre mon encadreur durant de ce travail, et pour la confiance qu'elle m'a donnée et ses précieux conseils .

Enfin , on remercie tout les personnes qui ont aidé de prés ou de loin Dans la réalisation de ce mémoire

Figures Table :

Section I

Section II

Section III

Fig. 20 Comparison of step responses of PID and FOPID controller

Tables list

Section III

INTRODUCTION

Fractional calculus is a more than 300 years old topic. The number of applications where fractional calculus has been used rapidly grows. These mathematical phenomena allow to describe a real object more accurately than the classical “integer-order” methods. The real objects are generally fractional ,however, for many of them the fractionality is very low. A typical example of a non-integer (fractional) order system is the voltage-current relation of a semi-infinite lossy transmission line or diffusion of the heat through a semi-infinite solid, where heat flow is equal to the half-derivative of the temperature .

The main reason for using the integer-order models was the absence of solution methods for fractional differential equations. At present time there are lots of methods for approximation of fractional derivative and integral and fractional calculus can be easily used in wide areas of applications (e.g.: control theory - new fractional controllers and system models, electrical circuits theory - fractances, capacitor theory, etc.).

For closed-loop control systems, there are four situations. They are :

1) IO (integer order) plant with IO controller

2) IO plant with FO (fractionalorder) controller

3) FO plant with IO controller

4) FO plant with FO controller

From control engineering point of view, doing something better is the major concern. Existing evidences have confirmed that the best fractional order controller can outperform the best integer order controller. It has also been answered in the literature why to consider fractional order control even when integer (high) order control works comparatively well .

Fractional order PID controller tuning has reached to a matured state of practical use. Since (integer order) PID control dominates the industry, we believe FO-PID will gain increasing impact and wide acceptance. Furthermore, we also believe that based on some real world examples, fractional order control is ubiquitous when the dynamic system is of distributed parameter nature.

In this project, a simple tutorial on fractional calculus in controls. Basic definitions of fractional calculus, fractional order dynamic systems and controls are presented first in Sec. I. Then, Optimisation algorithms are introduced in Sec. II. In Sec. III. A DC Motor simulation on MatLab with IOPID and FOPID to compare their performance.

Section I :

FRACTIONAL ORDER CALCULUS

And

FRACTIONAL ORDER SYSTEMS

I.1FRACTIONAL ORDER CALCULUS: MATHEMATICAL OVERVIEW :

Fractional order calculus is an area where the mathematicians deal with derivatives and integrals from noninteger orders.

Gamma function is simply the generalization of the factorial for all real numbers. The definition of the gamma function is given by :

$$\Gamma(x)= \int_0^{\infty} z^{x-1}\, e^{-z} dz \qquad (1)$$

$$\Gamma(x)=(x-1)!$$

${}_aD_t^{\alpha}$ is the combination of differentiation and integration operation commonly used in fractional calculus. Reimann- Liouville definition for ${}_aD_t^{\alpha}$ is

$${}_aD_t^{\alpha} \begin{cases} \frac{d^{\alpha}}{dt\alpha} & \alpha > 0 \\ 1 & \alpha = 0 \\ \int_0^t (d\tau)^{-\alpha} & \alpha < 0 \end{cases} \qquad (2)$$

Here α is the fractional order. a and t are the limits.

There are two commonly used definitions for general Differintegral ${}_aD_t^{\alpha}$.

1. Grunwald - Letnikov

2. Riemann- Liouville

Grunwald – Letnikov definition

$${}_aD_t^{\alpha} f(t) = \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{[\frac{t-a}{h}]} (-1)^j \binom{\alpha}{j} f(t-jh) \qquad (3)$$

Riemann-Liouville definition

$${}_aD_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \qquad \textbf{(4)}$$

Laplace Transform of Differintegral operator ${}_aD_t^{\alpha}$

$$L[{}_aD_t^{\alpha} f(t)] = \int_0^{\infty} e^{-st}\, {}_aD_t^{\alpha} f(t)\, dt \qquad (5)$$

$$L[{}_aD_t^{\alpha} f(t)] = s^{\alpha} F(s) - \sum_{m=0}^{n-1} s\, (-1)^j\, {}_0D_t^{\alpha-m-1} f(t) \qquad (6)$$

n lies in between n-1< α ≤ n.

Properties of Fractional Calculus :

The main properties of fractional derivatives and integrals are the following:

- If f(t) is an analytical function of t, its fractional derivative ${}_aD_t^{\alpha} f(t)$ is an analytical function of z and α.
- For $\alpha = n$, where n is an integer, the operation ${}_aD_t^{\alpha} f(t)$ gives the same result as classical differentiation of integer order n.
- For $\alpha = 0$ the operation ${}_aD_t^{\alpha} f(t)$ is the identity operator: ${}_aD_t^{\alpha} f(t) = f(t)$
- Fractional differentiation and fractional integration are linear operations:
  ${}_0D_t^{\alpha} f(t) + bg(t) = a\ {}_0D_t^{\alpha} f(t) + b\ {}_0D_t^{\alpha} g(t)$
- The additive index law (semigroup property)
  ${}_0D_t^{\alpha}\ {}_0D_t^{\beta} f(t) = {}_0D_t^{\alpha}\ {}_0D_t^{\beta} f(t) = {}_0D_t^{\beta+\alpha} f(t)$
  holds under some reasonable constraints on the function *f(t).*

The fractional-order derivative commutes with integer-order derivative

$$\frac{d^n}{dt^n}({}_aD_t^r f(t)) = {}_aD_t^r \left(\frac{d^n f(t)}{dt^n}\right) = {}_aD_t^{r+n} f(t)$$

under the condition $t = a$ we have $f^{(k)}(a) = 0$, $(k = 0,1,2,\ldots,n-1)$.

The relationship above says the operators $\frac{d^n}{d^n t}$ and ${}_aD_t^{\alpha}$ commute.[14]

I.2Fractional Order Dynamic Systems :

A fractional-order dynamic system can be described by a fractional differential equation of the following form :

$$a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \cdots + a_0 D^{\alpha_0} y(t) = b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \cdots + b_0 D^{\beta_0} u(t) \quad (7)$$

Where $D^{\gamma} \equiv {}_0D_t^{\gamma}$; a_k $(k = 0,\cdots n)$, b_k $(k = 0,\cdots m)$ are constants; and β_k (k = 0,⋯ n), α_k $(k = 0,\cdots m)$ are arbitrary real numbers.

Without loss of generality we can assume that $a_n > a_n - 1 > \cdots > a_0$, and $\beta_m > \beta_m - 1 > \cdots > \beta_0$.

For obtaining a discrete model of the fractional-order system (7), we have to use discrete approximations of the fractional-order integro-differential operators and then we obtain a general expression for the discrete transfer function of the controlled system [15].

$$G(z) = \frac{b_m \left(w\left(z^{-1}\right)\right)^{\beta_m} + \ldots + b_0 \left(w\left(z^{-1}\right)\right)^{\beta_0}}{a_n \left(w\left(z^{-1}\right)\right)^{\alpha_n} + \ldots + a_0 \left(w\left(z^{-1}\right)\right)^{\alpha_0}} \quad (8)$$

where $(w(z^{-1}))$ denotes the discrete equivalent of the Laplace operator s, expressed as a function of the complex variable z or the shift operator z^{-1}.

The fractional-order linear time-invariant system can also be represented by the following state-space model :

$$ {}_0D_t^q x(t) = Ax(t) + Bu(t) $$

$$ y(t) = Cx(t) \qquad (9) $$

where x ∈ R^n, u ∈ R^r and y ∈ R^p are the state, input and output vectors of the system and $A \in R^{nXn}$, $B \in R^{nXr}$, $C \in R^{pXn}$, q is the fractional commensurate order.

I.3Fractional Order Control Systems :

Classical PID Controller :

The classical PID controller can be considered as a particular form of lead-lag compensation in the frequency domain. Its transfer function can be expressed as :

$$ C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d . s \qquad (10) $$

$$ C(s) = k \frac{(s/\omega_c)^2 + \frac{2\delta_c s}{\omega_c} + 1}{s} \qquad (11) $$

With $\omega_c = \sqrt{K_i/K_d}$, $\delta = K_p/(2\sqrt{K_i K_p})$,$k = K_i$

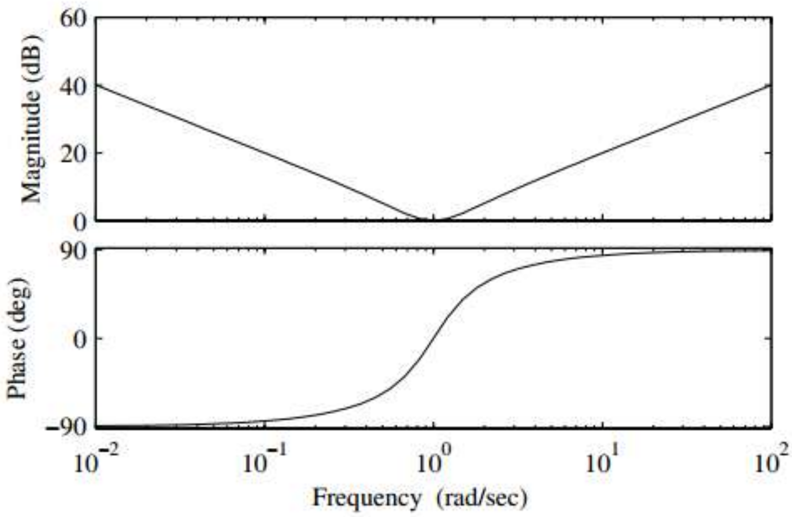
Another form can be :

$$ C(s) = k \frac{(s + a)(s + b)}{s} \qquad (12) $$

Therefore, the contributions of the controller depend on one of:

- Gains K_p, K_i, K_d
- Gain k and parameters ω_c, δ_c
- Gain k and location of zeros a and b. [14][11]

Fig.1-Frequency response of the classical PID controller with $K_p = K_i = K_d = 1$



Fractional-order PID Controller :

The integro-differential equation defining the control action of a fractionalorder PID controller is given by :

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^{\mu} e(t) \qquad (13)$$

$$C(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d . s^{\mu} = \ k \frac{(s/\omega_f)^{\lambda+\mu} + \frac{s\delta_f s^{\lambda}}{\omega_f} + 1}{s^{\lambda}} \qquad (14)$$

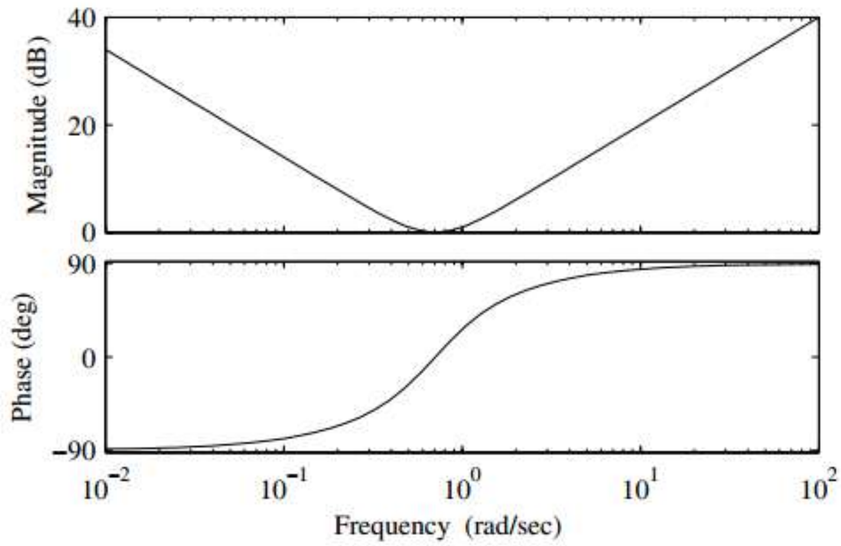


Fig.2 Frequency response of the classical PID controller with Kp = 1, Ki = 0.5,Kd =1

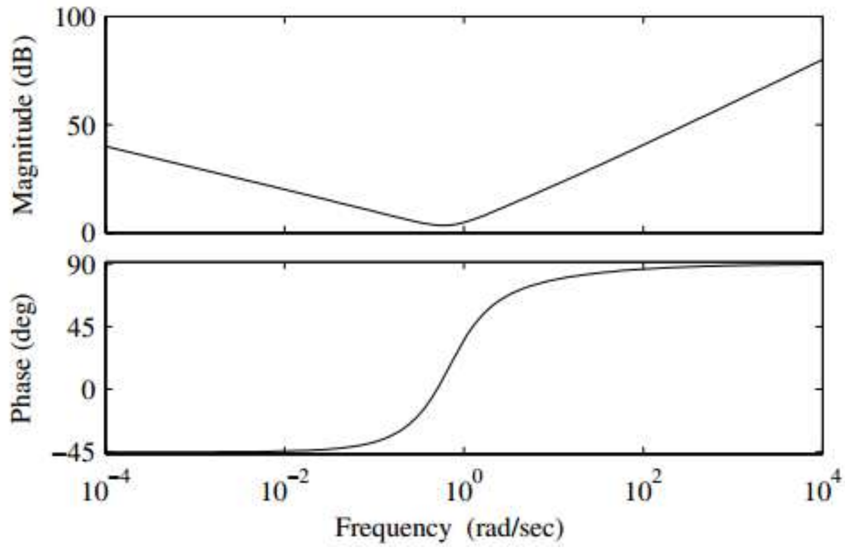


Fig.3 Frequency response of the fractional-order PID controller with k = 1, ωf = 1, δf = 1, and λ = μ = 0.5

As can be observed, this fractional-order controller allows us to select both the slope of the magnitude curve and the phase contributions at both high and low frequencies.

In a graphical way, the control possibilities using a fractional-order PID controller are shown in **Fig. 4**, extending the four control points of the classical PID to the range of control points of the quarter-plane defined by selecting the values of λ and μ

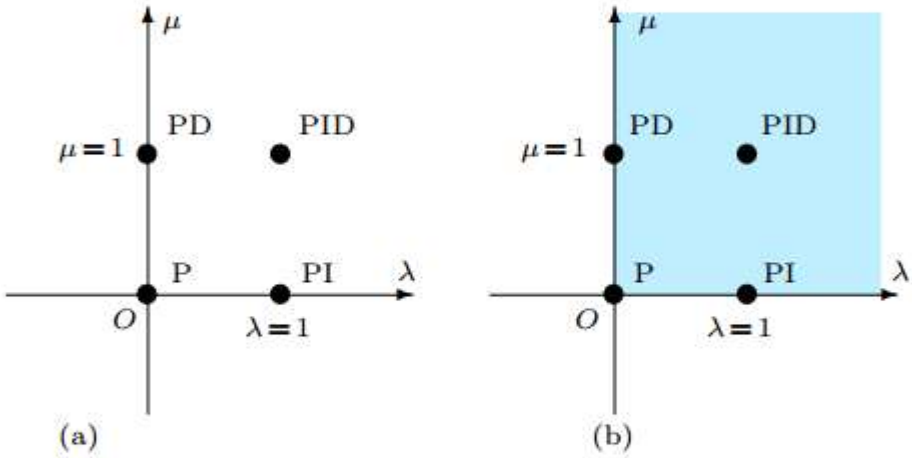


Fig.4 Fractional-order PID vs classical PID: from points to plane: (a) integer-order and (b) fractional-order

Section II :

METAHEURISTICS

II .1 INTRODUCTION

Optimization plays a vital role in many engineering applications. In design activity, an optimal design is achieved by comparing a few alternative design solutions created by using prior problem knowledge. In such activity the feasibility of each design solution is first investigated. Thereafter an estimate of the underlying objective (cost, profit, etc) of each design solution is computed and the best solution is adopted. Optimization algorithms provide systematic and efficient ways of creating and comparing new design solutions in order to achieve an optimal design. The optimization process must only be used in those problems where there is a specific need of accomplish a quality product or a competitive product. It is expected that the design solution obtained through an optimization method is better than other results in terms of the selected objective.

II .1 OPTIMIZATION

The optimization is from 'Optimum' which implies a point at which the conditions are best and most favorable. Optimization is finding better among different possible solutions with the measure of the quality of those solutions. The real problems are 'hard', it means that, it is not guaranteed to find the best solution in acceptable amounts of time.

For many engineers and researchers, optimization is an esoteric technique used in Mathematics and Operations Research related activities. With the advent of computers, optimization has become a part of computer aided activities.

An optimization algorithm is a procedure which is executed iteratively by comparing various solutions till the optimum or a satisfactory solution is found.

In many industrial design activities, optimization is achieved indirectly by comparing a few chosen design solutions and accepting the best solution. This simplistic approach never guarantees an optimal solution. On the contrary, optimization algorithms begin with one or more design solutions supplied by the user and then iteratively check new design solutions in order to achieve the trueoptimum solution.

There are two distinct types of optimization algorithms which are in use today. :

1.Algorithms which are deterministic, with specific rules for moving from the one solution to the other. These algorithms have been in use for quite some time and have been successfully applied to many engineering design problems.

2.Algorithms which are stochastic in nature, with probabilistic transition rules These algorithms are comparatively new and are gaining popularity due to certain properties which the deterministic algorithms do not have. These are mainly classified as, Traditional optimization algorithms and Evolutionary Algorithms. Traditional optimization algorithms some methods are shown in **fig.5**

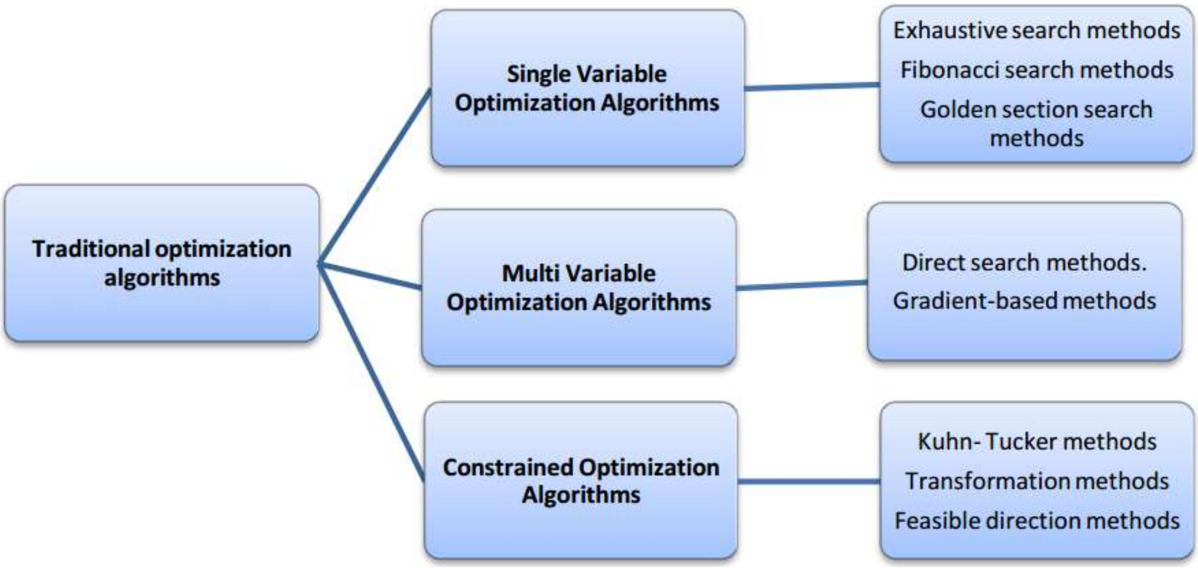


Fig5 Methods of Traditional optimization algorithms

II .3 METAHEURISTICS

A metaheuristic is anadvanced technique or heuristic designed to locate, create, or select a heuristic that may provide adequately superior result to an optimization problem, specially with partial or imperfect information. Metaheuristic sample a set of solutions which is too large to be completely sampled. Metaheuristic may make few assumptions about the optimization problem being solved, and so they may be usable for a variety of problems.

A heuristic is any approach to problem solving, learning, or discovery that employs a practical method not guaranteed to be optimal or perfect, but sufficient for the immediate goals. Where finding an optimal solution is impossible or impractical, heuristic methods can be used to speed up the process of finding a satisfactory solution.

MetaheuristicAlgorithms

These algorithms are found to be potential in search and optimization for complex engineering optimization problems. In these categories, important metaheuristic algorithms are :

- Genetic algorithm (GA)

- Artificial Immune System (AIS) Algorithm

- Particle Swarm Optimization (PSO) Algorithm

- Ant Colony Optimization (ACO) Algorithm

- Sheep Flocks Heredity Model Algorithm (SFHM)

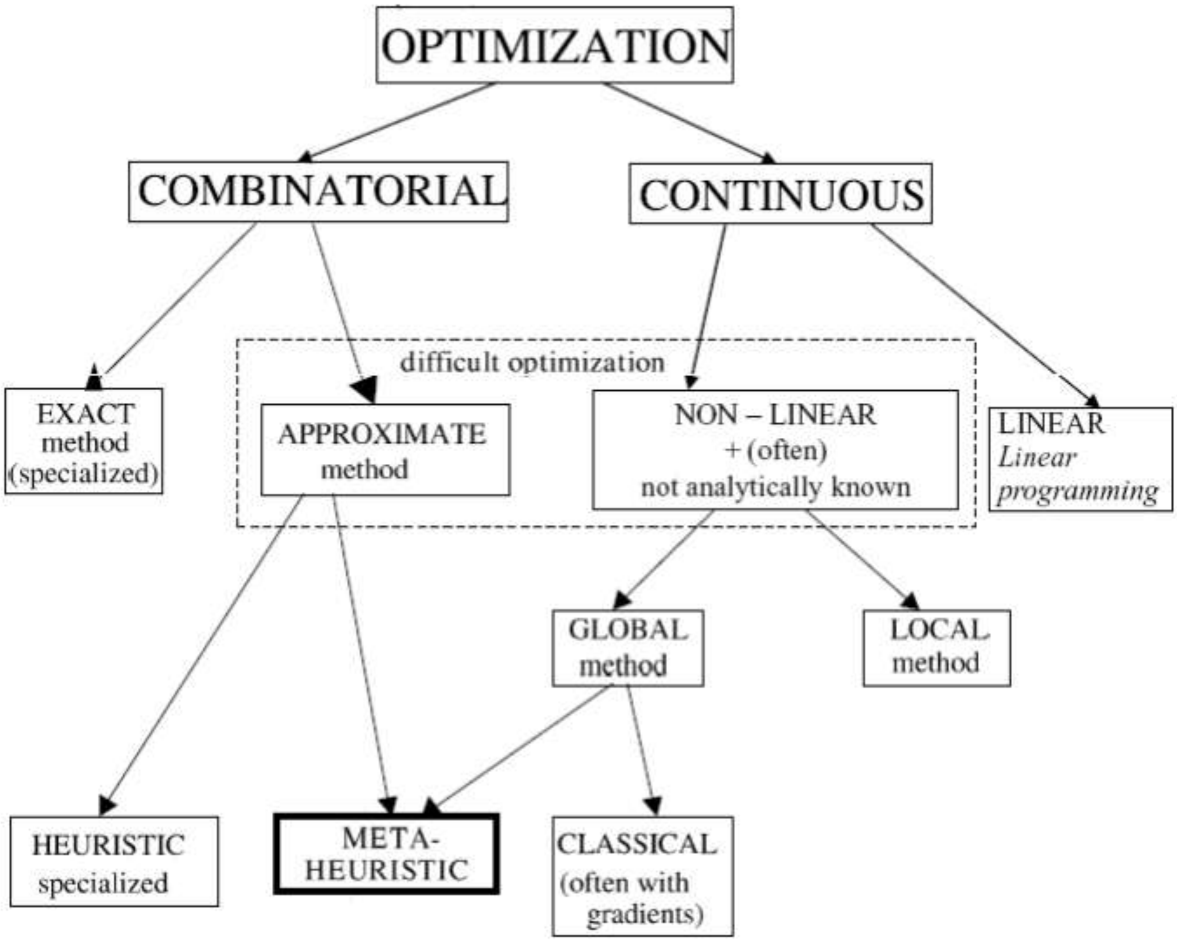


Fig6 Optimisation methodes

II .4Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. It has been applied successfully to wide variety of search and optimization problems. It can be applied to virtually any problem that can be expressed in terms of an objective function.

Suppose the following scenario: a group of birds are randomly searching for food in an area. There is only one piece of food in the area being searched. All of birds do not know where the food is. But they know how far the food is in each iteration. So what is the best strategy to find the food? The effective one is to follow the bird that is nearest to the food.

The PSO algorithm is iterative and involves initializing a number of vectors (called particles) randomly within the search space of the objective function. These particles are collectively known as the swarm. Each particle represents a potential solution to the problem expressed by the objective function. During each time the objective function is evaluated to establish the fitness of each particle using its position as input. Fitness values are used to determine which positions in the search space being attracted to both their personal best position as well as the best position found by the swarm so far.

The PSO algorithm has shown its robustness and efficacy in solving function value optimization problems in real number space. The attractiveness of the PSO algorithm is due to the features natural metaphor, stochastic move, adaptively, and positive feedback.

The features of PSO are,

1. PSO can be applied for non-linear, non-continuous optimization problem with continuous variables PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.

2. PSO is mainly solving continuous optimization tasks.

3. PSO gets better results in a faster, cheaper way compared with other methods.

4. PSO has robustness and efficiency in solving function value optimization problems in real number space.

5. PSO has good convergence speed and there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications.

6. Unlike the other heuristic techniques, PSO has a flexible and well-balanced mechanism to enhance and adapt to the global and local exploration abilities.

7. PSO is a zero order algorithm, for no derivative is necessary for its implementation.

8. PSO has natural metaphor, stochastic move, adaptivity, and positive feedback.

9. PSO has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

10.Unlike GA, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm.

11.In GA, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area. In PSO, only GBest (or IBest) gives out the information to others. It is a one-way information sharing mechanism. The evolution only looks for the best solution quickly even in the local version in most cases.

Figure 7 shows the flow diagram for PSO.

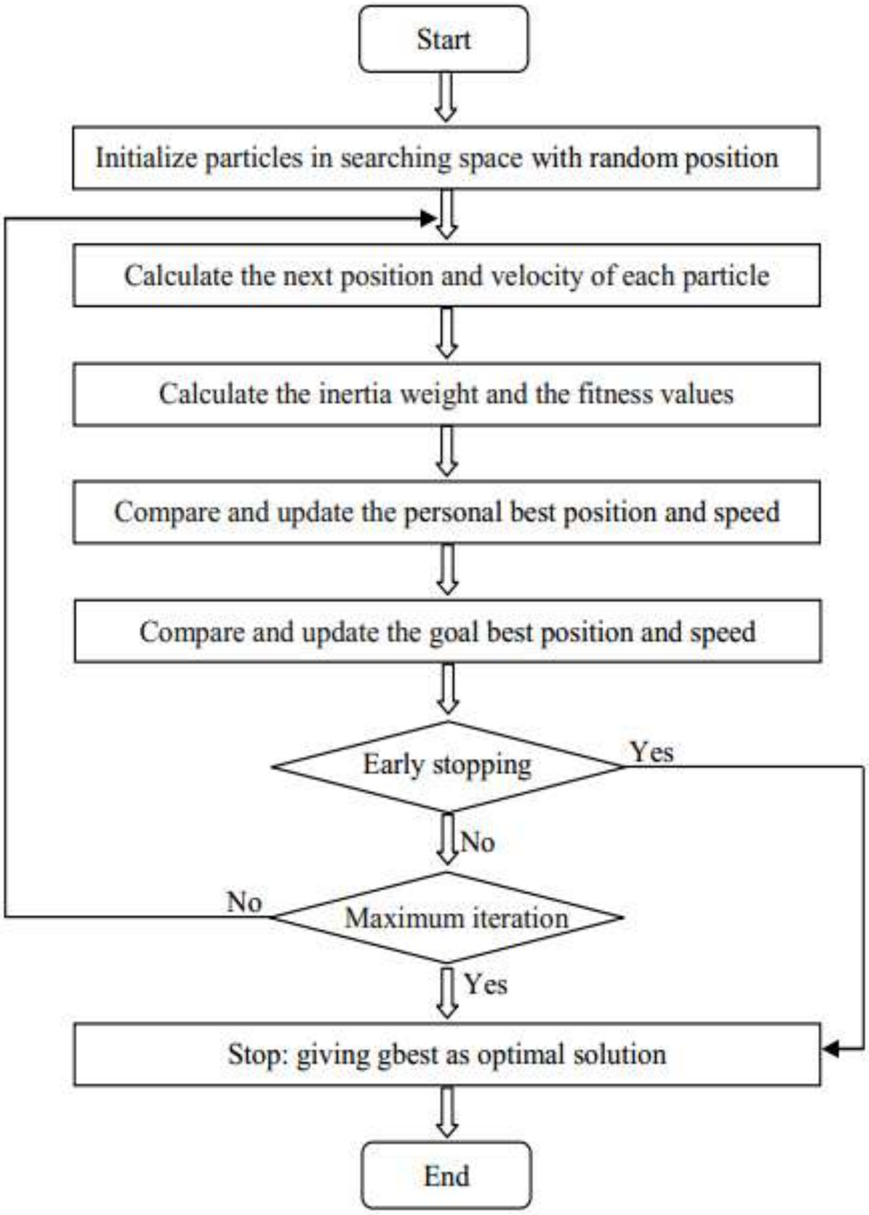


Fig7 the flow diagram for PSO.

Steps in PSO

Step 1 :The population has to be generated and number of iterations has also to be fixed.

Step 2 : Evaluate the objective functions along with the required design variables.

Step 3 : Assign Pbest[i] = initial solution where i = 1,2, ... N (N: no of particles). Store the objective value.

Step 4 :Find best among all particles and assign this to Gbest. Compare step 2 objective functions with step 3 objective functions and carryout swapping to obtain the step 3 objective function, store the swapped data.

Step 5 :Generate initial velocities randomly for all particles.

Step 6 :Add velocities to the corresponding particles, i.e., Present[i] (new) = Present[i] old) + V[i].

Step 7 :Update velocity according to V[i] = V[i] (present) + Cl * (Pbest[i] - present[i]) +c2 * (Gbest[i] - present [i]).

Step 8 : Evaluate the updated particles to get new ones Step 9 : If number of iterations is lesser for optimization Go to step6.

Step l0: Compare the Objective Function Value with the last solution obtained, go to step7 if they are not equal, else, end the loop.

Step 11: The algorithm has to be terminated after pertaining required numbers of iterations.

PSO

```
Initialize a population of particles with random positions and velocities on D dimensions in the search space
while Terminating condition is not reached do
    for each particle i do
        Adapt velocity of the particle using Equation (10)
        Update the position of the particle using Equation (11)
        Evaluate the fitness f(X_i)
        if (f(X_i) < f(P_i)) then
            P_i ← X_i
        end
        if (f(X_i) < f(P_g)) then
            P_g ← X_i
        end
    end
end
```

Section III :

SIMULATION

And

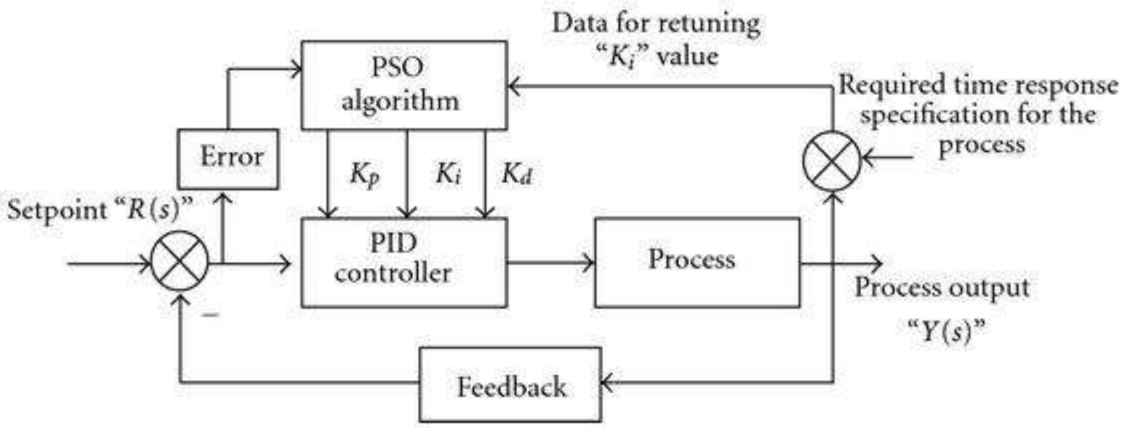
RESULTS

III.1 INTRODUCTION :

PID controllers have been used for several decades in industries for process control applications .The reason for their wide popularity lies in the simplicity of design and good performance including low percentage overshoot and small settling time for slow process plants . The performance of the PID controllers can be improved by making use of fractional order derivatives and integrals.

This greatest flexibility makes us possible to design more robust control system. In fractional order PID (FOPID) controller, the integral and derivative orders are usually fractional. In FOPID besides Kp, Ki, Kd we have two more parameters λ and μ , the integral and derivative orders respectively. If λ =1 and μ =1, then it becomes integer PID. The five parameters Kp, Ki, Kd,λ , μ are to be optimized in five-dimensional hyper-space to obtain an optimal solution that satisfies all the user specifications. It is necessary to understand the theory of fractional calculus in order to understand the significance of FOPID controller . This paper mainly focuses on the better way of tuning by comparing the results of PSO based tuning of FOPID controllers with the other conventional tuning methods.

Dynamic systems based on fractional order calculus have been a subject of extensive research in recent years since the proposition of the concept of the fractional order $\boldsymbol{PI^{\lambda}D^{\mu}}$ controllers and the demonstration of their effectiveness in actuating desired fractional order system responses .Classical optimization techniques cannot be used here because of the roughness of the objective function surface . We, therefore, use a derivative-free optimization technique — particle swarm optimization (PSO) originally devised by Kennedy and Eberhart , in this section were going to the system shown in fig .7,the compare the FOPID to the IOPID



.

Fig.8

III.3 DC MOTOR DESIGN :

The electrical equivalent diagram of an armature controlled DC motor is given in the fig.9

Where R = armature resistance (Ω), L = self inductance of armature (H),Ia = armature current (A), If = field current (A), Ea = applied armature voltage (V), Eb = back emf (V), Tm = torque produced by the motor (Nm), θ = angular displacement of motor shaft (rad), ω = angular speed of motor shaft (rad/sec), J = equivalent moment of inertia of motor and load referred to motor shaft (kg-m2), B = equivalent viscous friction coefficient of motor and load referred to motor shaft (Nm*s/rad).

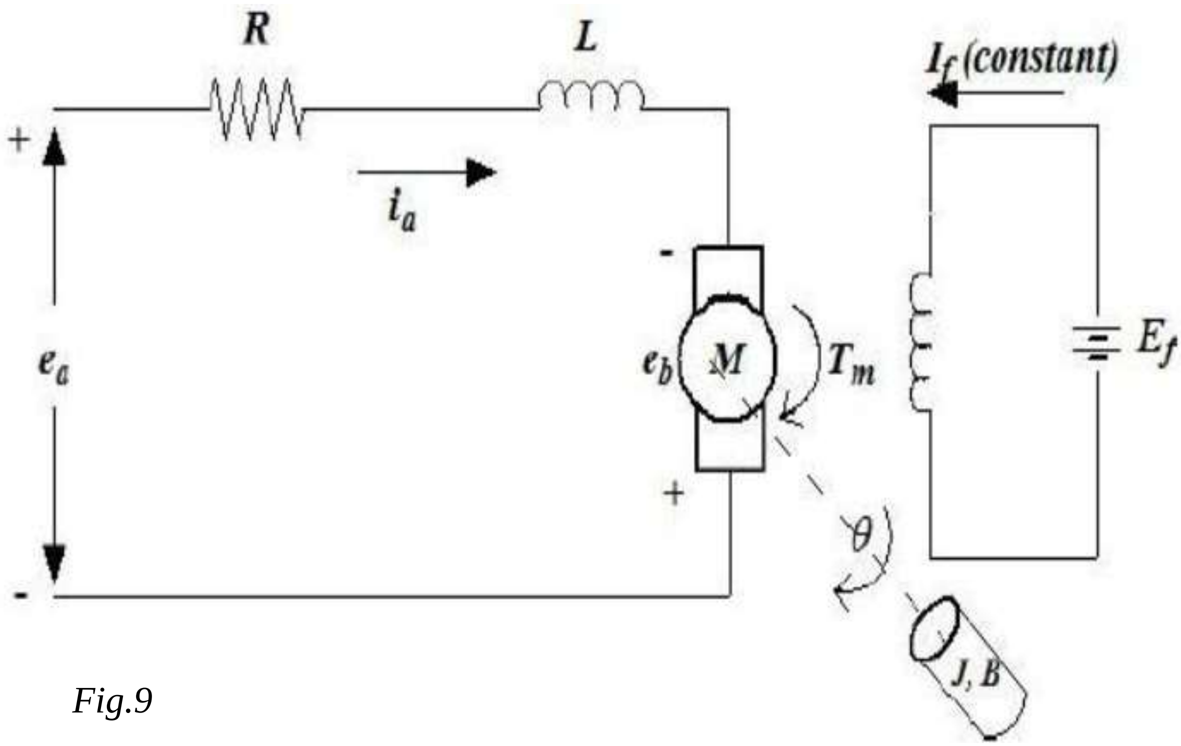


Fig.9

The transfer function of DC motor is given by: $G_p(s) = \frac{s \cdot \theta(s)}{E_a(s)} = \frac{\omega(s)}{E_a(s)} = \frac{k_T}{[(R+Ls)(Js+B)+k_T K_b]}$

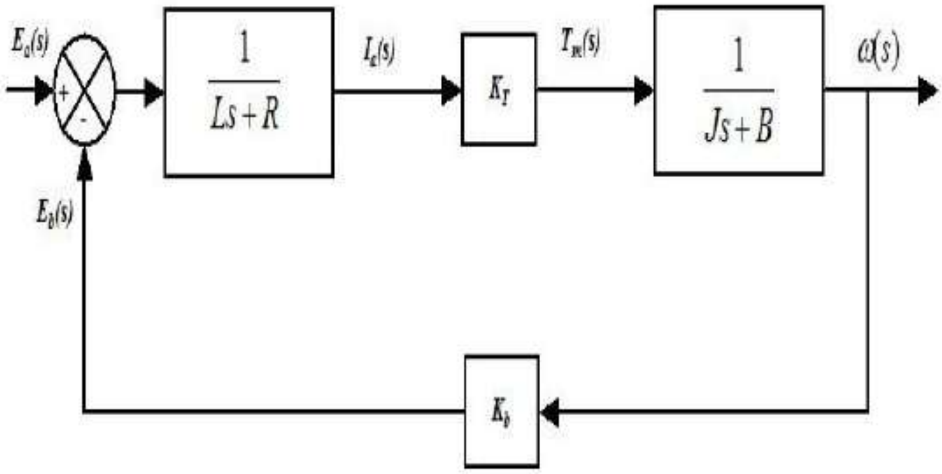


Fig.10 Block diagram of armature controlled DC Motor

Specification of DC motor	R = 1 Ω	L = 0.5 H	K = 0.01	J = 0.01 kg-m^2	B = 0.1 Nm*s/rad

Tab.1 Parameter values of DC motor

the final transfer function of DC motor becomes

$$G_p(s) = \frac{0.01}{0.005s^2 + 0.006s + 0.1001}$$

III.4 Simulation and results

Now we are going to see the unit step response of DC motor transfer function using classical and fractional PID controller and its performance parameters.Classical PID controller is tuned by Ziegler-Nicholas method and we obtained the proportional gain Kp = 6, integral gain Ki = 28.3 and derivative gain Kd = 0.318.The unit step response and performance parameters such as peak overshoot,peak time and settling time for PID control is shown below:

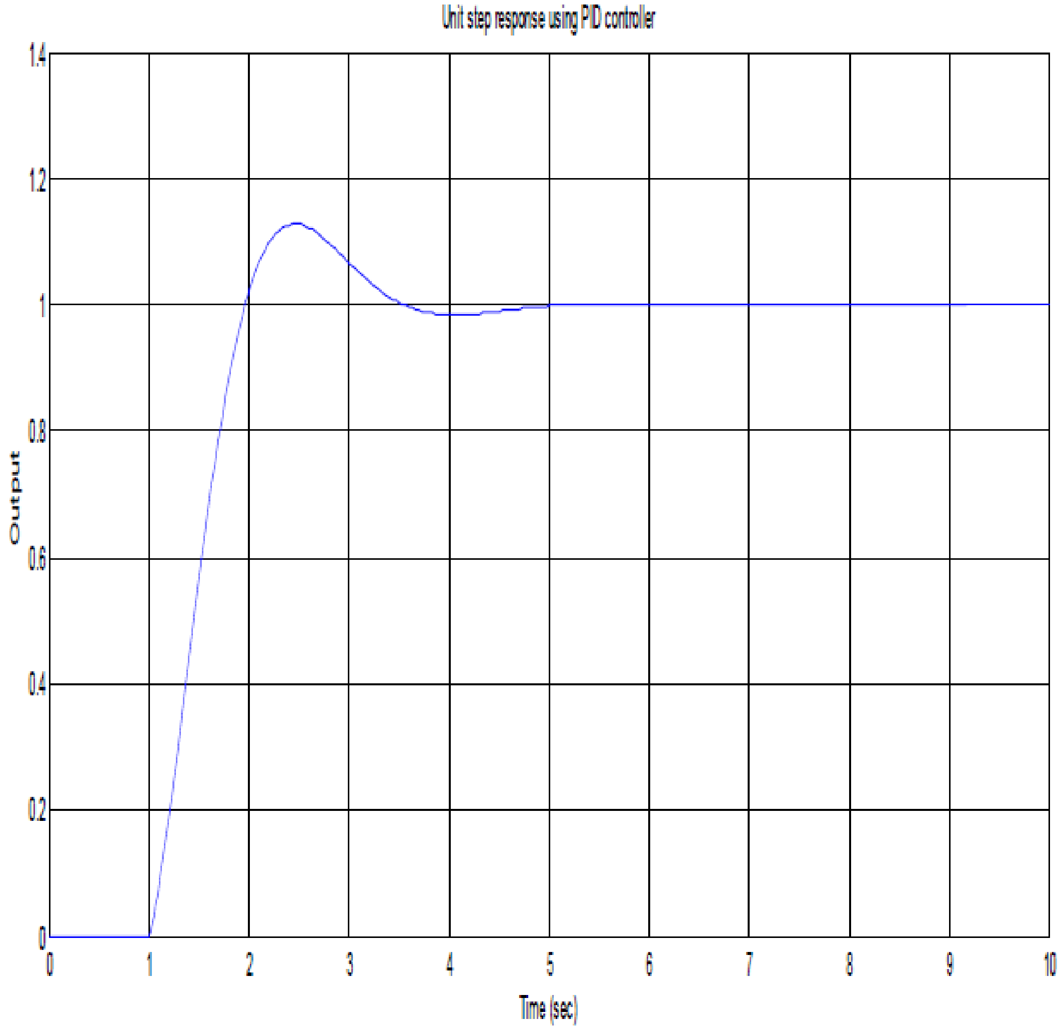


Fig .11 Unit step response of DC motor using PID controller

The unit step response using PID controller gives an overshoot (Mp) of 12.85%,peak time (Tp) of 2.47 sec and settling time (Ts) of 3.1 sec which is undesirable.To minimize these parameters, we use fractional order PID controller which can provide better performance.

The unit step response and control performance parameters for FOPID controller with different combinations of λ and μ is shown below.These graphs shows the step responses of system with fractional PID controller, where the derivative order μ and integral order λ are in fractions. The fractions can be less than or greater than 1.

a) With λ = 1 and varying values of μ < 1

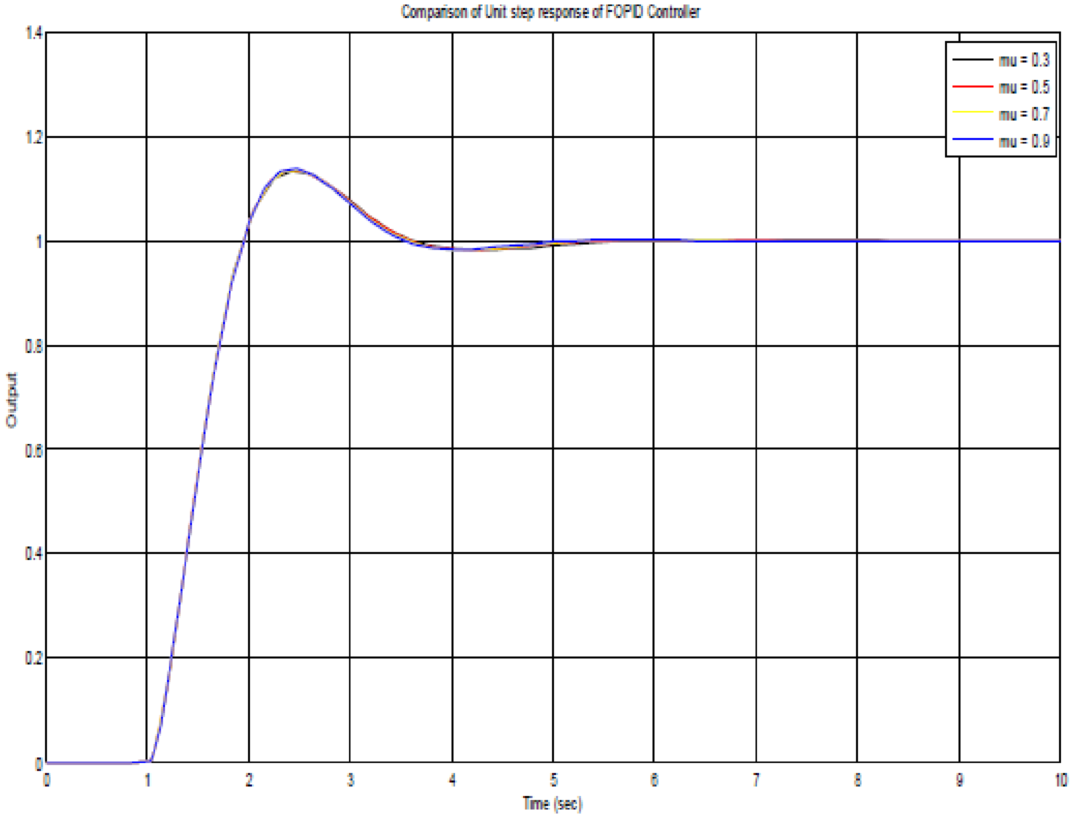


Fig.12 Unit step response of DC motor using FOPID controller for varying values of μ < 1

λ	μ	M_p	T_p	T_s	ISE
1	0.3	13.4387	2.4254	2.9745	0.3458
1	0.5	13.5116	2.3713	3.1023	0.3456
1	0.7	13.7049	2.4996	3.0076	0.3457
1	0.9	13.9560	2.4698	3.0152	0.3458

Tab.2 Comparison of Parameters for Different Combinations of λ=1 and μ<1

Changing the μ value here does no difference.

b) With varying values of λ<1 and μ= 1

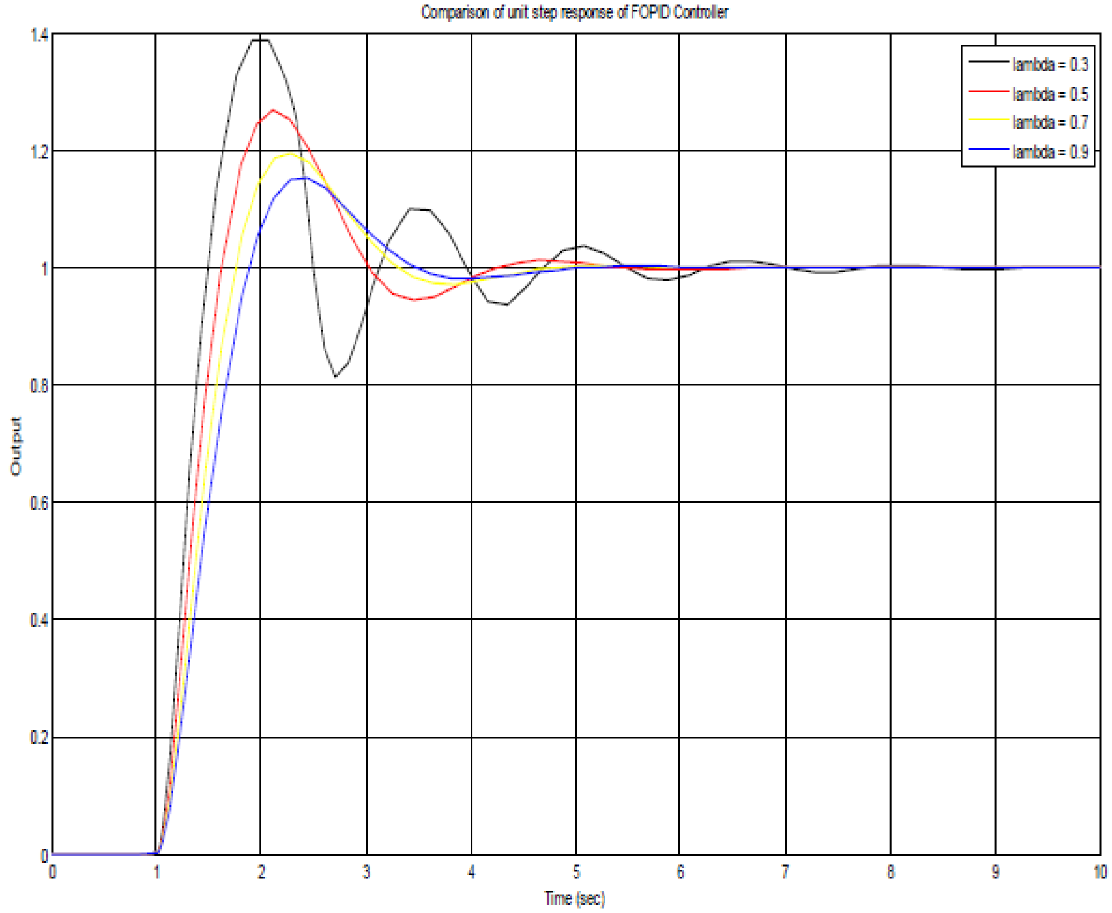


Fig.13 Unit step response of DC motor using FOPID controller for varying values of λ < 1

λ	μ	M_p	T_p	T_s	ISE
0.3	1	38.8953	1.9178	4.3480	0.2997
0.5	1	26.9260	2.1115	3.4573	0.2862
0.7	1	19.6730	2.2867	2.8444	0.3041
0.9	1	15.2586	2.4476	3.0131	0.3307

Tab.3 Comparison of Parameters for Different Combinations of λ <1 and μ=1

In this case ,it's clear that increasing the λ value improves the control parameters.

c) With varying values of λ<1 and μ<1

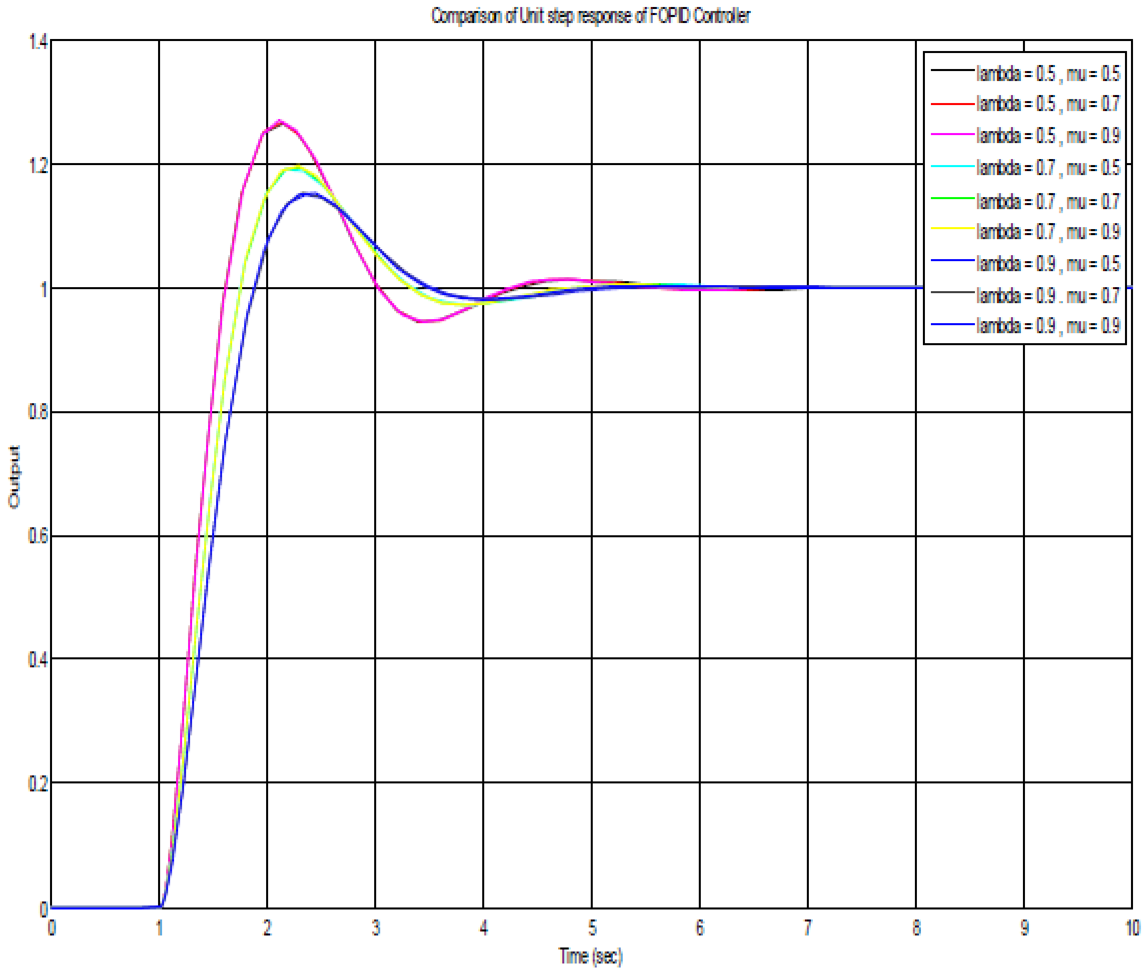


Fig.14 Unit step response of DC motor using FOPID controller for varying values of λ < 1 and μ < 1

λ	μ	M_p	T_p	T_s	ISE
0.5	0.5	26.5058	2.1536	3.5961	0.286
0.5	0.7	26.7243	2.1446	3.6187	0.2864
0.5	0.9	26.9232	2.1085	3.4517	0.2869
0.7	0.5	19.0836	2.1777	2.9069	0.3041
0.7	0.7	19.2968	2.3212	3.0267	0.3044
0.7	0.9	19.6501	2.2879	2.8258	0.3048
0.9	0.5	15.0713	2.3615	3.0890	0.3312
0.9	0.7	15.1803	2.3340	3.0644	0.3313
0.9	0.9	15.1968	2.4555	3.0036	0.3315

Tab.4 Comparison of Parameters for Different Combinations of λ<1 and μ<1

It can be seen from the above table that when λ=0.9 and μ=0.7, control parameters values are the smallest.

d) With λ=1 and varying values of μ>1

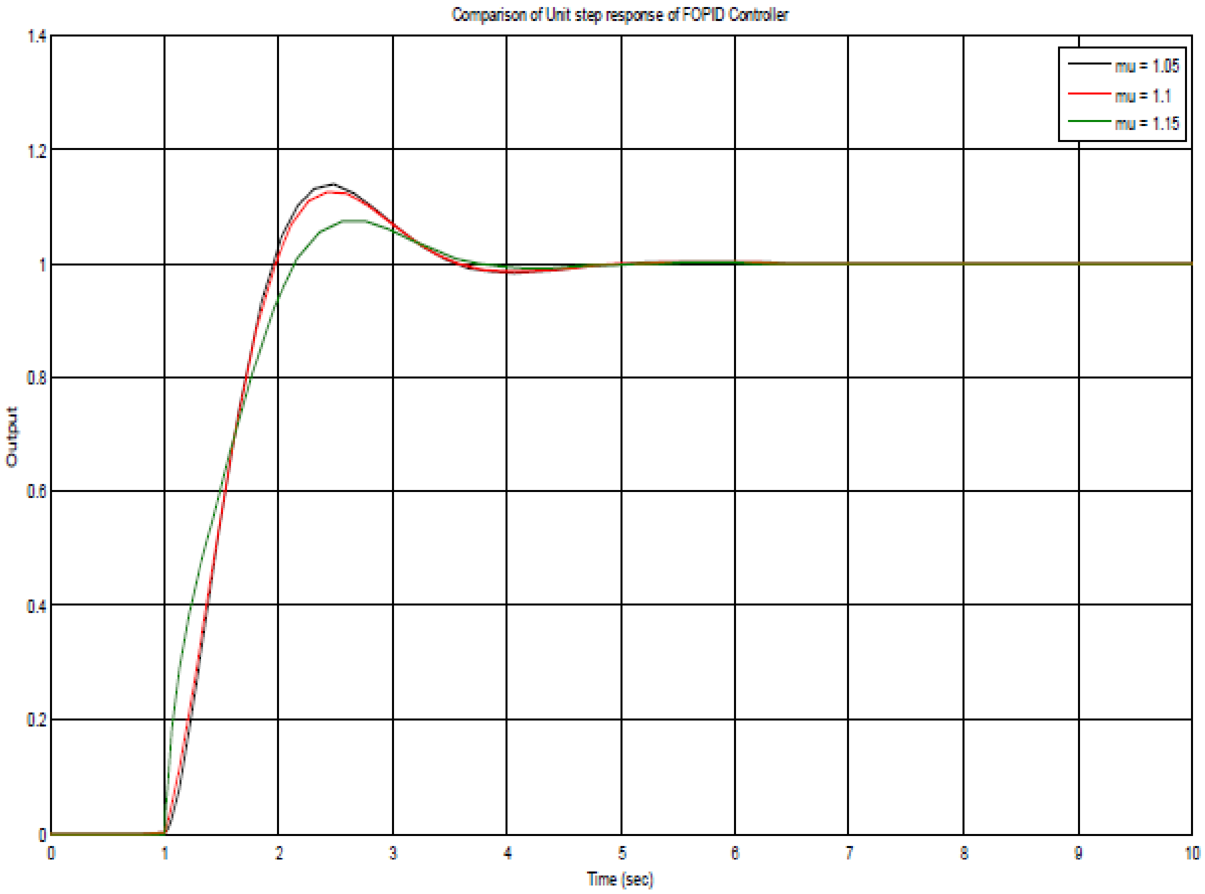


Fig.15 Unit step response of DC motor using FOPID controller for varying values of μ > 1

λ	**μ**	**M_p**	**T_p**	**T_s**	**ISE**
1	1.05	13.7526	2.4785	3.0580	0.3407
1	1.1	12.5777	2.4209	2.9814	0.3198
1	1.15	7.3682	2.5606	2.9606	0.2294

Tab.5 Comparison of Parameters for Different Combinations of λ=1 and μ>1

Slight improvement in increasing μ, but it ends after μ=1.15.

e) With varying values of λ>1 and μ=1

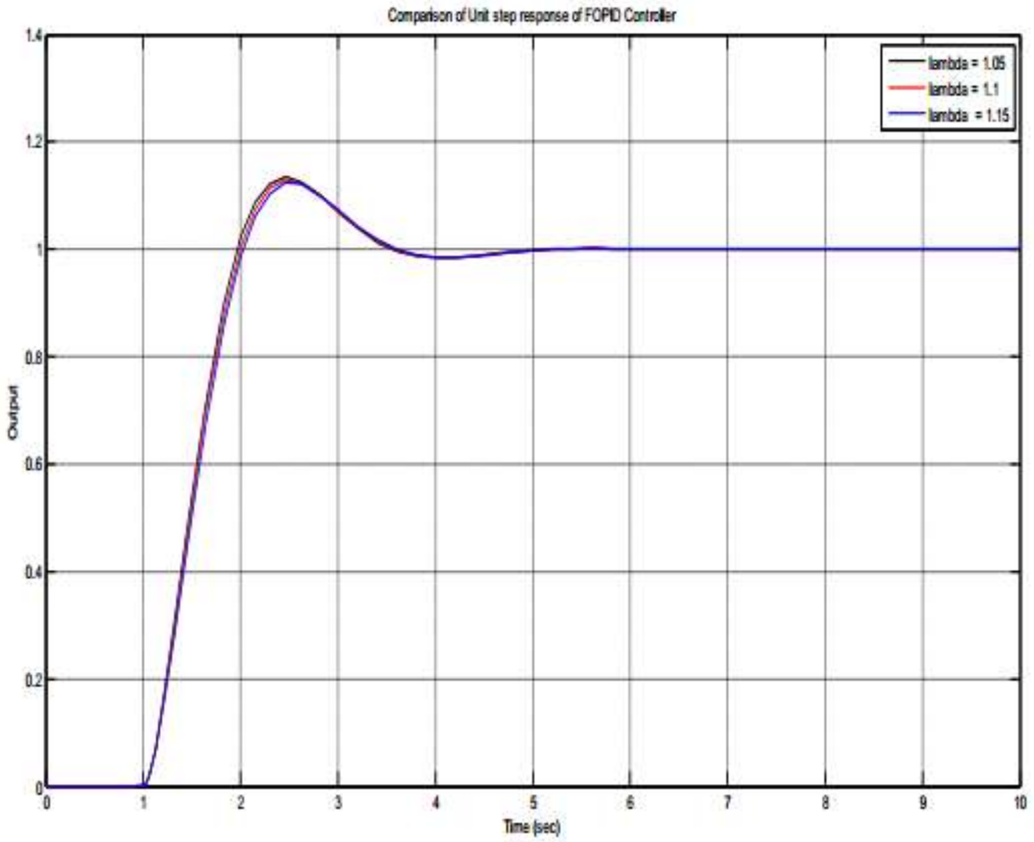


Fig.16 Unit step response of DC motor using FOPID controller for varying values of λ > 1

λ	μ	M_p	T_p	T_s	ISE
1.05	1	13.4255	2.4603	3.0204	0.3521
1.1	1	12.8878	2.4621	3.0136	0.3592
1.15	1	12.3550	2.4585	2.9987	0.3663

Tab.6 Comparison of Parameters for Different Combinations of λ>1 and μ=1

Increasing λ decreases the overshoot bu slows down the system a little.

f) With varying values of λ>1 and μ>1

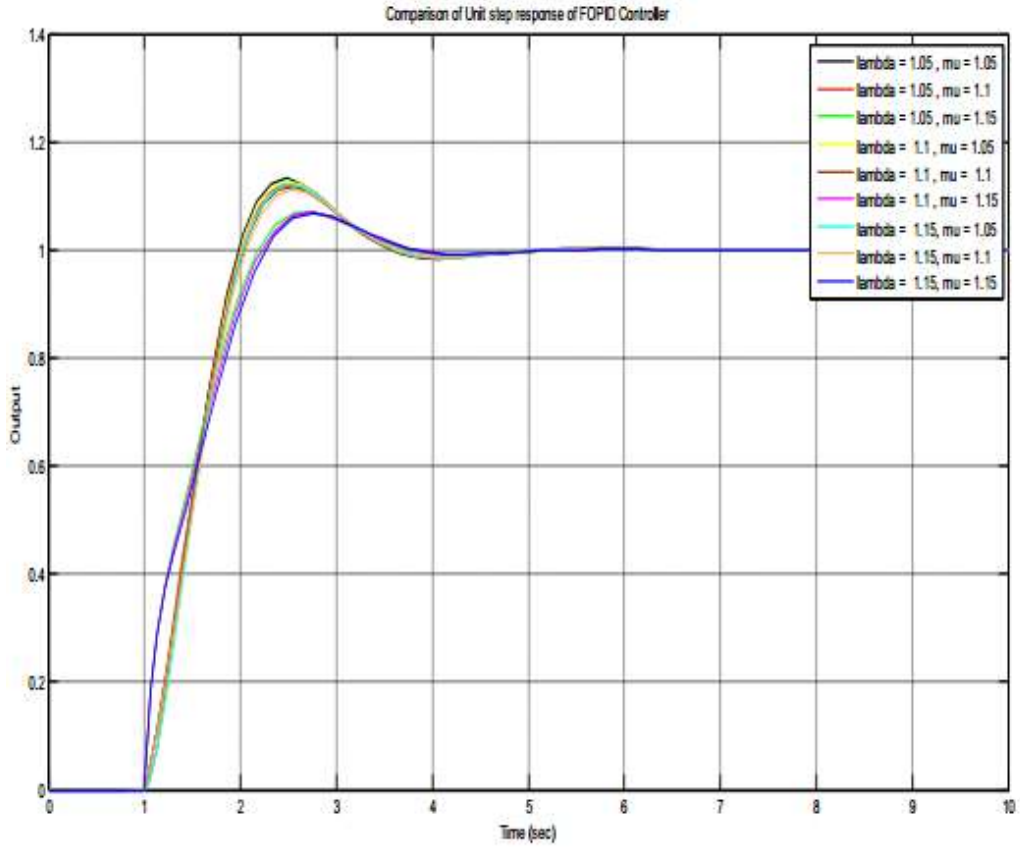


Fig.17 Unit step response of DC motor using FOPID controller for varying values of λ > 1 and μ > 1

λ	μ	M_p	T_p	T_s	ISE
1.05	1.05	13.2209	2.4806	3.0514	0.3479
1.05	1.1	11.8799	2.4159	2.9683	0.327
1.05	1.15	7.1071	2.7573	2.9573	0.237
1.1	1.05	12.7180	2.4831	3.0413	0.355
1.1	1.1	11.5889	2.5615	2.9388	0.3342
1.1	1.15	6.9922	2.7544	2.9544	0.2445
1.15	1.05	12.2015	2.4725	3.0215	0.3621
1.15	1.1	11.2334	2.5383	3.0984	0.3413
1.15	1.15	6.8675	2.7519	2.9519	0.252

Tab.7 Comparison of Parameters for Different Combinations of λ>1 and μ>1

λ=1.15 and μ=1.15 are the best values here

g) With varying values of λ<1 and μ>1

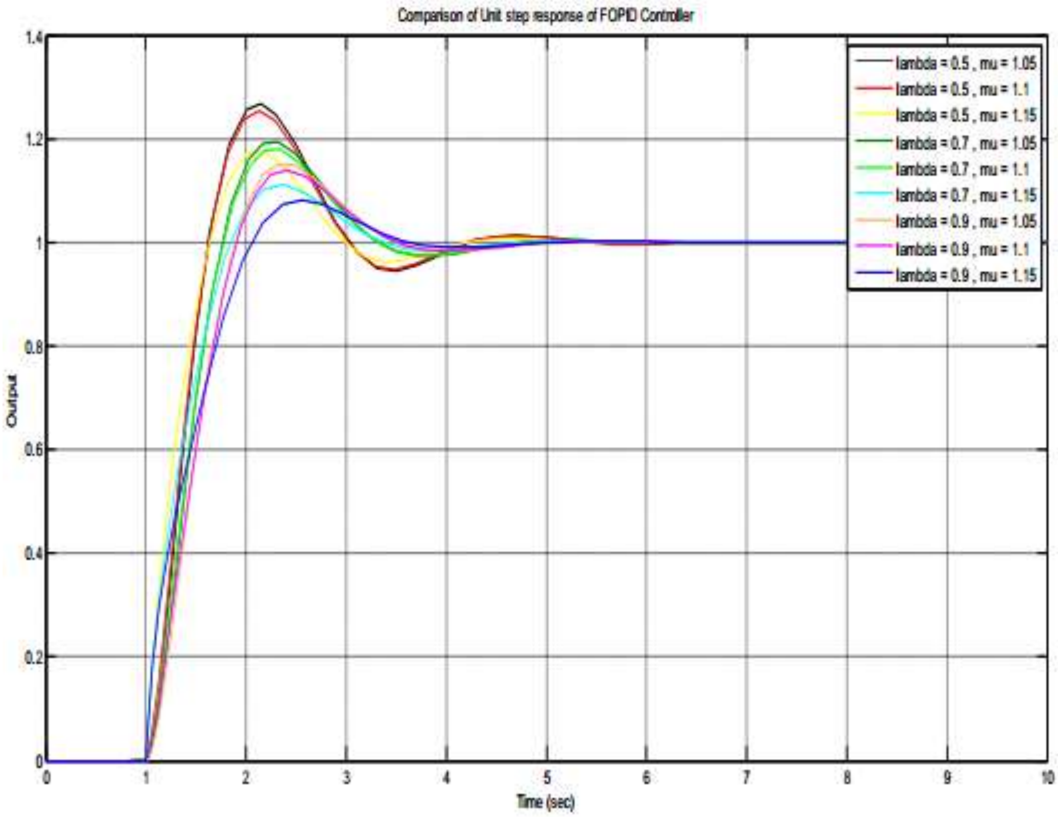


Fig.18 Unit step response of DC motor using FOPID controller for varying values of λ < 1 and μ > 1

λ	μ	M_p	T_p	T_s	ISE
0.5	1.05	26.6241	2.1518	3.5008	0.282
0.5	1.1	25.2814	2.1337	3.4911	0.2607
0.5	1.15	17.3934	2.1920	2.5868	0.167
0.7	1.05	19.2889	2.3197	2.8923	0.2998
0.7	1.1	17.9001	2.3375	2.9159	0.2787
0.7	1.15	11.1935	2.3890	2.7890	0.1862
0.9	1.05	15.0213	2.3179	3.0679	0.3265
0.9	1.1	13.9977	2.4005	2.9847	0.3054
0.9	1.15	8.2657	2.5689	2.9689	0.2143

Tab.8 Comparison of Parameters for Different Combinations of λ<1 and μ>1

Again ,that reduces the peak overshoot but slows down the system.

h) With varying values of λ>1 and μ<1

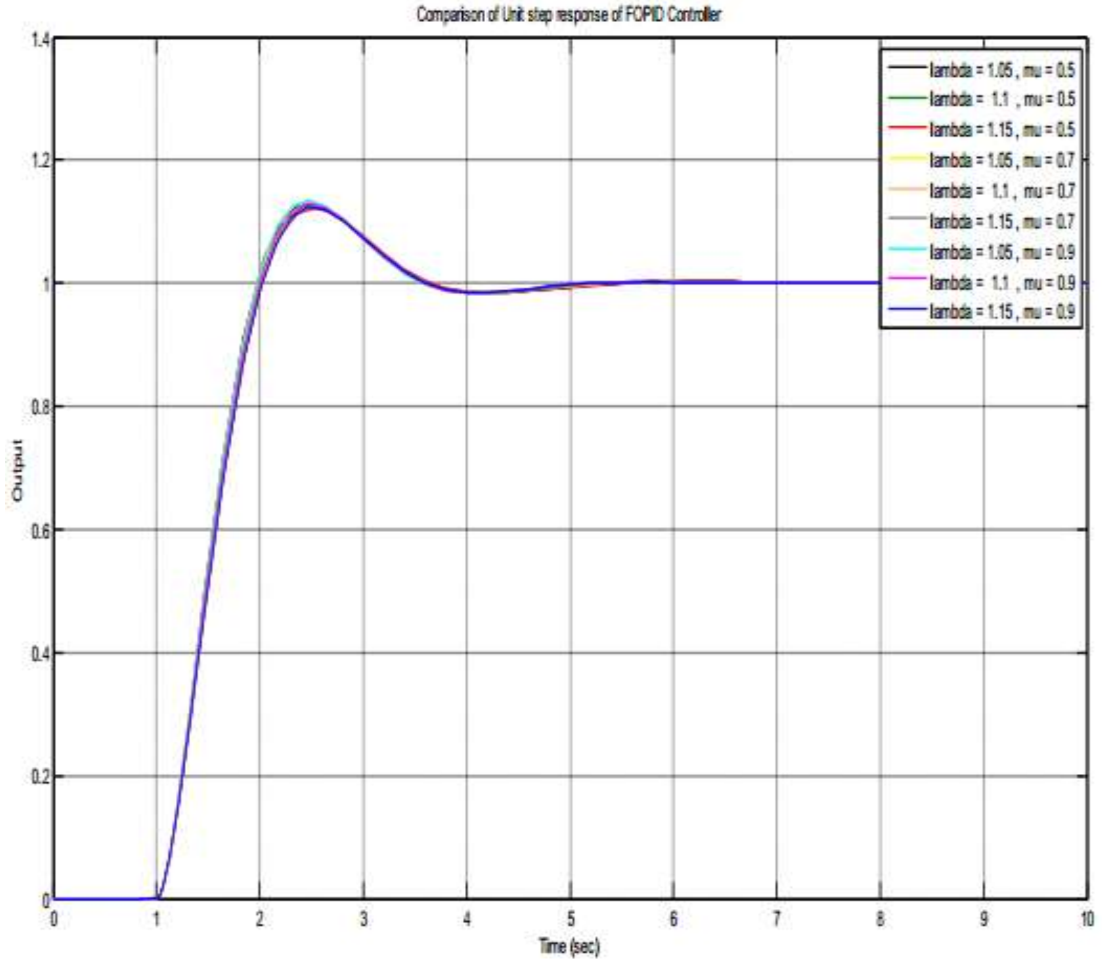


Fig.19 Unit step response of DC motor using FOPID controller for varying values of λ > 1 and μ < 1

λ	μ	M_p	T_p	T_s	ISE
1.05	0.5	12.8914	2.5407	3.0028	0.3529
1.1	0.5	12.5023	2.5422	2.9890	0.3601
1.15	0.5	12.1468	2.5382	3.1775	0.3673
1.05	0.7	13.2261	2.4994	2.9947	0.3529
1.1	0.7	12.7621	2.5056	2.9957	0.3601
1.15	0.7	12.3349	2.5086	3.0501	0.3673
1.05	0.9	13.4155	2.4676	3.0097	0.353
1.1	0.9	12.8932	2.4686	3.0158	0.3602
1.15	0.9	12.3787	2.4651	2.9955	0.3673

Tab.9 Comparison of Parameters for Different Combinations of λ>1 and μ<1

Now ,into using PSO to find the optimal FOPID parameters

The parameter values taken for running the PSO algorithm in MATLAB environment is given in table below:

Parameter	Values
Number of Particles	50
Maximum no. of Iterations	100
Cognitive Component C_1	2
Social Component C_2	2
Maximum Speed	10
Minimum Inertia Weight	0.4
Maximum Inertia Weight	0.9

Tab.10 PSO parameter values

we obtain the following solution set which gives the most optimal parameter values of the controller in the defined search space.

[1.15 1.15] = [6.87 2.75 2.95 0.252]

After getting the optimal values of λ and μ ,we compare the unit step response of optimal FOPID controller and classical PID controller ,we get the following results :

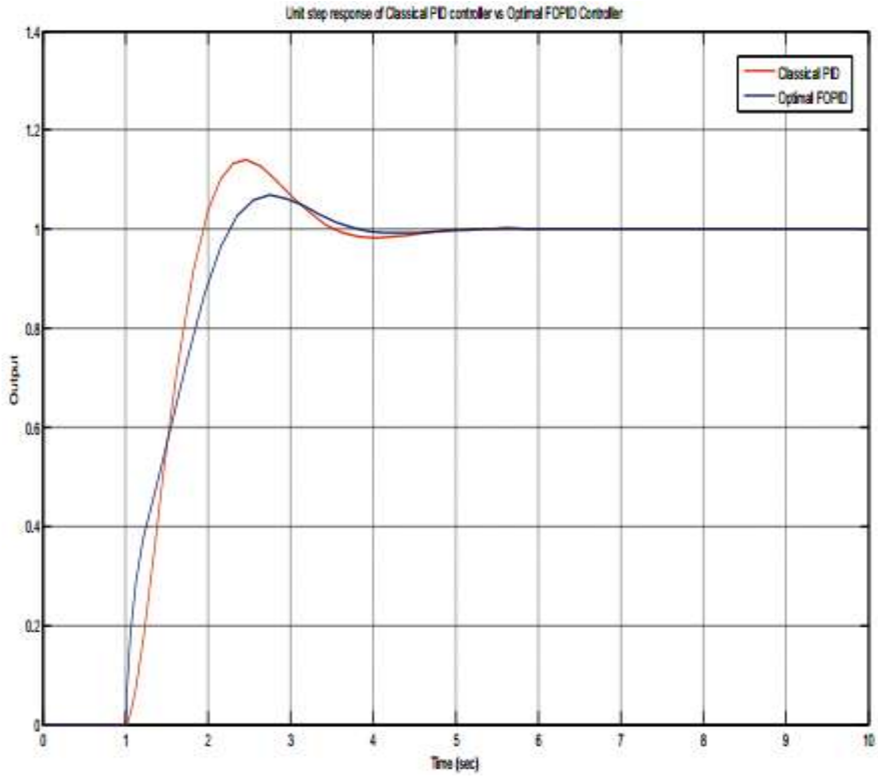


Fig. 20 Comparison of step responses of PID and FOPID controller

Controller	M_p	T_p	T_s	ISE
PID	12.87	2.47	3.1	0.3449
Optimal FOPID	6.87	2.75	2.95	0.252

Tab.11 Comparison of performance parameters of PID and FOPID controller

The optimal FOPID is performing better than the regular PID controller and it can be seen clearly thru the table and the graph above.

CONCLUSION

In this Project, a fractional order PID controller is. Using numerical optimization (PSO), numerous simulation comparisons presented in this paper indicate that, the fractional order PID controller, if properly designed and implemented, will outperform the conventional integer order PID controller. It was shown that the best FO PID works better than IO PID. For actually implementation, we introduced a modified approximation method to realize the designed fractional order PID controller.

We used simulation to illustrate that the order the approximation does not affect the control performance in any noticeable amount. With the rapid development of computer performances, the realization of fractional order control systems also became possible and much easier than before. Despite fractional order control's promising aspects in modeling and control design, fractional order control research is still at its primary stage. The notable future research is to develop tuning rules for FO PID and in particular on tuning the fractional orders.

REFERENCES :

[1] I. Petras, Fractional Order Non Linear Systems: Modeling, Analysis and Simulation, Higher Education Press, Beijing and SpringerVerlag Berlin Heidelberg, 2011.

[2]J. P. Jiang, S. C. and P. K. Sinha, “Optimal feedback control of direct current motors, ” IEEE Transactions on Industrial Electronics, vol. 31, no. 4, Aug. 1990, pp. 269-274.

[3]B. Allaoua, A. Abderrahmani, B. Gasbaoui and A. Nasri, “Intelligent controller design for DC motor speed control based on fuzzy logic genetic algorithms optimization,” Leonardo Journal of Sciences, Dec. 2008, pp. 90-102.

[4]G. Zeng, J. Chen, Y. Dai, L. Li, C. Zheng and M. Chen, "Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization", Neurocomputing, vol. 160, pp. 173-184, 2015.

[5]Y. Luo, Y.Q. Chen, Fractional-order [proportional derivative] controller for robust motion control: Tuning procedure and validation, American Control Conference, pp. 1412-1417, 2009.

[6]Mohamed Karim Bouafoura, Naceur Benhadj Braiek, “PID controller design for integer and fractional plants using piecewise orthogonal functions,” Commun Nonlinear Sci Number Simulat 15, 2010, pp. 1267- 1278

[7]Majid Zamani, Masoud Karimi-Ghartemani, Naseer Sadati, “FOPID controller design for robust performance using particle swarm optimization,” Fractional Calculus and Applied Analysis, vol. 10, no. 2, 2007, pp. 169-187

[8]Ivo Petras, Fractional-order nonlinear systems: Modeling, Analysis and Simulation, Springer, 2010

[9]Ricardo Enrique Gutierrez, Joao Mauricio Rosario, Jose Tenreio Machado, “Fractional order calculus: Basic concepts and engineering applications,” Mathematical Problems in Engineering, Hindwai Publishing Corporation, vol. 2010. 2010. pp. 1- 19

[10]Schlegel Milos , C Ech Martin, “The fractional order PID controller outperforms the classical one,” Proc. 7th Int. Sci. & Tech. Conf.- Process Control, 2006, pp. 1-7

[11]Concepcion A. Monje, Blas M. Vinagre, Vicente Feliu, YangQuan Chen, “Tuning and auto-tuning of fractional order controllers for industry applications,” Control Engineering Practice, 16, 2008, pp. 792 – 812

[12]A survey on optimization metaheuristics Ilhem Boussaïd a, Julien Lepagnot b, Patrick Siarry b,Université des sciences et de la technologie Houari Boumediene, Electrical Engineering and Computer Science Department, El-Alia BP 32 Bab-Ezzouar, 16111 Algiers, Algeria b Université Paris Est Créteil, LiSSi, 61 avenue du Général de Gaulle 94010 Créteil, France

[13]F. Glover, M. Laguna, R. Marti, Scatter search and path relinking: advances and applications, in: F. Glover, G. Kochenberger (Eds.), Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol. 57, Springer, New York, 2003, pp. 1–35.

[14]I. Podlubny: Fractional Differential Equations. Academic Press, San Diego, 1999.

[15]B. M. Vinagre, I. Podlubny, A. Hernandez, and V. Feliu: On realization of fractional-order controllers. Conference Internationale Francophone d’Automatique, Lille, Jule 5-8, 2000, pp. 945–950.