International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-3, Issue-2, May 2013

Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control Using Particle Swarm Optimization

Ankit Rastogi, Pratibha Tiwari

Abstract-PID controller is the most widely used controller in industry for control applications due to its simple structure and easy parameter adjusting.But increase in complexity of control systems has introduced many modified PID controllers.The recent advancement in fractional order calculus has introduced fractional order PID controller and it has recieved a great attention for researchers.Fractional order PID (FOPID) controller is an advancement of conventional PID controller in which the derivative and integral order are fractional rather than integer.Apart from the usual tuning parameters of PID, it has two more parameters λ (integer order) and μ (derivative order) which are in fractions.This increases the flexiblity and robustness of the system and gives a better performance than classical PID controller. In this research paper, FOPID has been applied to DC motor for speed control and optimal values of λ and μ has been obtained using particle swarm optimization technique.

Index Terms- DC motor, Fractional order PID controller, PID controller, Particle swarm optimization

I. INTRODUCTION

Proportional-Integral-Derivative (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 [1]. The performance of PID controllers can be further improved by making use of fractional order derivatives and integrals.

In fractional order controllers, integral and derivative operations are usually of fractional order, therefore besides tuning the proportional(k_p), derivative(k_d) and integral(k_i) constants we have two more parameters: the power of s in integral and derivative actions- λ and μ respectively.This adds flexibility and makes the system more robust,thus,enhancing its dynamic performance compared to its integer counterpart.Finding an optimal set of values for k_p, k_d, k_i, λ and μ to meet the specifications of user for a given process plant calls for real parameter optimization in five-dimensional hyperspace [3].

The paper is organised as follows: Section II gives a brief review of classical PID controller.Section III gives an introduction of fractional order controller.In section IV, we present a mathematical model of DC motor and Section V deals with particle swarm optimization technique for parameter optimization.Section VI presents the simulations and results. Finally, conclusions are drawn in Sec. VII.

II. CLASSICAL PID CONTROLLER

A proportional-integral-derivative controller (PID controller) is basically a generic control loop feedback mechanism widely used in industrial control systems [2]. A PID controller calculates an "error" value as the difference between a measured plant variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control inputs. Fig.1 shows a basic structure of a closed loop controller.

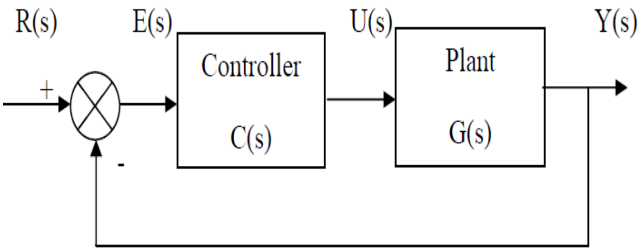

Fig. 1 Classical Unity Feedback Control System

The differential equation of a PID controller is given by:

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(t)dt + K_d \frac{d}{dt} e(t) \qquad (1)$$

and the transfer function is given by:

$$G_{PID}(s) = K_p + \frac{Ki}{s} + s \cdot K_d \qquad (2)$$

The tuning of PID controller parameters is mostly done by well known *Ziegler–Nichols* method in which the K_i and K_d gains are first set to zero.K_p is increased from 0 to some critical value $K_p = K_{cr}$,at which sustained oscillations occur.Then value of K_{cr} and the oscillation period P_{cr} is used to set the gains as follows:

Manuscript received May, 2013.
Ankit Rastogi, Department of Electrical Engineering, Sam Higginbottom Institute of Agriculture ,Technology & Sciences, Allahabad, India.
Pratibha Tiwari, Department of Electrical Engineering, Sam Higginbottom Institute of Agriculture, Technology & Sciences, Allahabad, India.

Table I: Ziegler-Nichols Method

Controller	K_p	K_i=Kp/T_i	K_d=KpT_d
PID	0.6K_{cr}	P_{cr}/2	P_{cr}/8

III. FRACTIONAL ORDER CONTROLLER

A. Fractional Order Calculus: An Overview

Fractional order calculus is an area of calculus which generalizes the derivative or integral of a function to non-integer (fractional) order. Fractional calculus evaluates (d^ny/dt^n), n-fold integrals where n is fractional, irrational or complex [13].These mathematical operations allow to describe a real object more accurately than the classical integer-order methods. The main reason for using the integer-order models was the absence of solution methods for fractional differential equations. At present there are a number of methods available for approximation of fractional derivative and integral and fractional order calculus can be applied in wide areas of applications such as in control theory for designing fractional order controllers and system models.

In Fractional order calculus, we use Differintegral operator which is denoted by ${}_aD_t^{\alpha}$ where a and t are the limits and α (α ∈ℜ) is the order of the operation.It is the combination of differentiation and integration operation commonly used in fractional calculus.It is defined as follows:

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

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

1. Grunwald – Letnikov
2. Riemann- Liouville

These definitions are required for realization of control algorithm.

Grunwald – Letnikov definition:

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

Riemann-Liouville definition:

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

The condition for above equation is n-1< α < n.Γ (.) is called gamma function.The definition of gamma function is given by $\Gamma(x)=\int_0^{\infty} z^{x-1}e^{-z}dz$ (5)

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

$$ L[{}_aD_t^{\alpha} f(t)] = \int_0^{\infty} e^{-st} {}_aD_t^{\alpha} f(t)\, dt \quad (6) $$

$$ L[{}_aD_t^{\alpha} f(t)] = s^{\alpha} F(s) - \sum_{m=0}^{n-1} s\,(-1)^j\, {}_0D_t^{\alpha-m-1} f(t) \quad (7) $$

Here n lies in between n-1< α ≤ n.

B. Fractional Order PID Controller:

Fractional Order PID controller denoted by $PI^{\lambda}D^{\mu}$ was proposed by Igor Podlubny [5] in 1997.It is an extension of Conventional PID Controller where λ and μ have fractional values.Figure 2 shows the block diagram of fractional order PID controller.

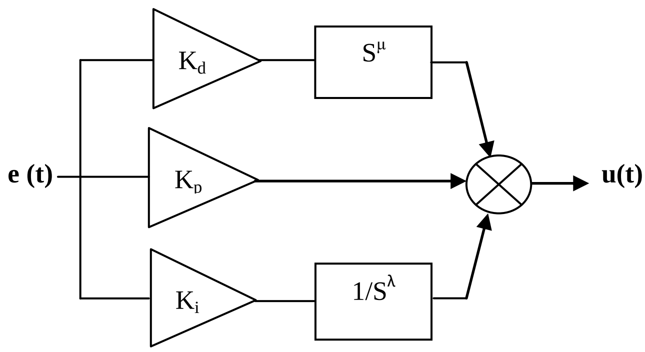

Fig.2 Fractional Order PID Controller

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

$$ u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^{\mu} e(t) \quad (8) $$

and thus the transfer function of the controller becomes

$$ G_{FOID}(s) = K_P + \frac{K_i}{s^{\lambda}} + K_d \cdot s^{\mu} \quad (9) $$

Where λ and μ are an arbitrary real numbers.Taking λ=1 and μ=1, a classical PID controller is obtained.Thus, FOPID controller generalizes the classical PID controller and expands it from point to plane as shown in fig.3.This expansion provides us much more flexibility in designing PID controller and gives an opportunity to better adjust the dynamics of control system.This increses the robustness of the system and makes it more stable.

However, with increase in parameters to be tuned, the optimization problem associated with the system becomes more difficult [14].For achieving a certain performance, it is desired to develop a systematic algorithm for the FOPID optimization.A number of optimization techniques can be implemented for getting the best values of the parameters of the controller.One of the method which the author implements in this paper is particle swarm optimization.

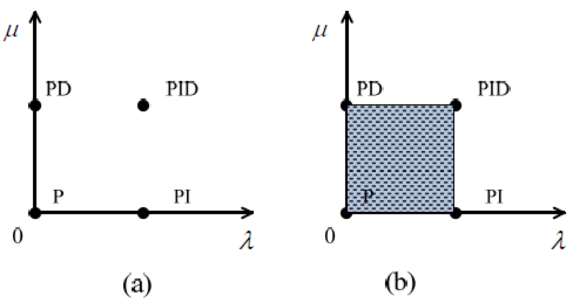

Fig.3 (a) Classical PID Controller (b) FOPID Controller

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

IV. MATHEMATICAL MODELLING OF DC MOTOR FOR SPEED CONTROL

In this section the authors model the transfer function of an armature controlled DC motor for its speed control so as to study the control performance of Fractional order PID controller[18]. The electrical equivalent diagram of an armature controlled DC motor is given in the figure below:

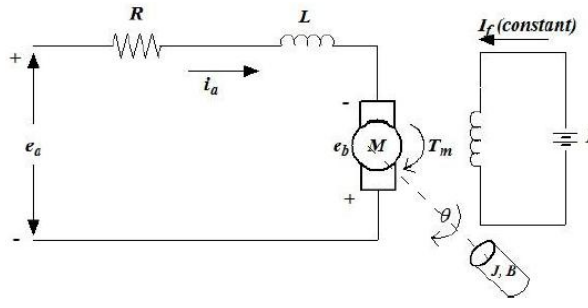

Fig. 4 Electrical equivalent diagram of armature controlled DC motor

where R = armature resistance (Ω), L = self inductance of armature (H), i_a = armature current (A), i_f = field current (A), e_a = applied armature voltage (V), e_b = back emf (V), T_m = 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-m^2), B = equivalent viscous friction coefficient of motor and load referred to motor shaft (Nm*s/rad).

DC motors when applied in servo applications are generally used in the linear range of magnetization curve.Hence, the air gap flux ϕ is proportional to the field current, i.e. $\phi = K_f i_f$ where K_f is constant.

The torque T_m developed by the motor is proportional to the product of armature current and air gap flux, i.e. $T_m = K_1 K_f\ i_f\ i_a$.Here K_1 is constant.Since the field current is constant in armature controlled DC motor,so $T_m = K_T\ i_a$.Here K_T is the motor torque constant.The motor back e.m.f e_b is proportional to speed i.e. $e_b = K_b\omega = K_b \frac{d\theta}{dt}$ .Here K_b is back e.m.f constant.

Now writing the KVL equation for the armature circuit we get,

$$L\frac{di_a}{dt} + R\ i_a + e_b - e_a = 0 \quad (10)$$

And the torque equation is

$$J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} = T_m = K_T\ i_a \quad (11)$$

Applying Laplace Transform

$$e_b(s) = K_b\, s\, \theta(s) \quad (12)$$

$$(Ls+R)I_a(s) = e_a(s) - e_b(s) \quad (13)$$

$$(Js^2 + Bs)\ \theta(s) = T_m(s) = K_T i_a(s) \quad (14)$$

Finally, the transfer function of DC motor is given by:

$$G_p(s) = \frac{s\cdot\theta(s)}{E_a(s)} = \frac{\omega(s)}{E_a(s)} = \frac{k_T}{[(R+Ls)(Js+B)+k_T\ K_b]} \quad (15)$$

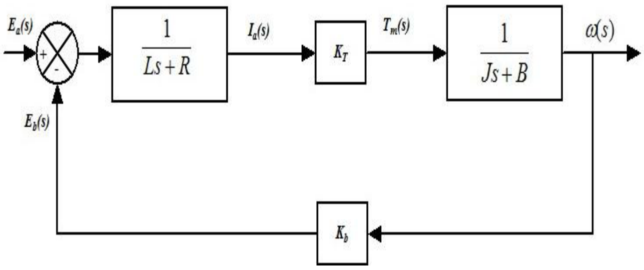

Fig. 5 Block diagram of armature controlled DC Motor

After applying the parameter values of DC motor as shown in the table below:

Table II: Parameter values of 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

the final transfer function of DC motor becomes

$$G_p(s) = \frac{0.01}{0.005s^2 + 0.006s + 0.1001} \quad (16)$$

V. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) was originally developed by Kennedy and Eberhart in 1995 is a population-based evolutionary algorithm [3]. It was inspired by the social behavior of bird and fish schooling, and has been found to be robust in solving continuous nonlinear optimization problems.

In PSO, the 'swarm' is initialized with a population of random solutions. Each particle in the swarm is a different possible set of the unknown parameters to be optimized. Representing a point in the solution space, each particle tries to adjust its flying toward a potential area according to its own flying experience and shares social information among particles [20]. The objective is to efficiently search the solution space by swarming the particles toward the best fitting solution encountered in previous iterations with the intent of encountering better solutions through the course of the process and finally converging on a single minimum error solution.

For a multidimensional problem, the velocity and position of each particle in the swarm are updated using the following equations:

$$v_i(t+1) = w\cdot v_i(t) + c_1\cdot rand\cdot(pbest(t) - x_i(t)) + c_2\cdot rand\cdot(gbest(t) - x_i(t)) \quad (17)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (18)$$

Where,

$v_i(t+1)$ is the velocity of the ith particle at (t+1) iteration, $x_i(t+1)$ is the position of the ith particle at (t+1) iteration, w is the inertial weight factor (weighting function), c_1 and c_2 are acceleration constants called cognitive learning rate and social learning rate respectively,*rand* is the random

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

function in the range [0,1],*pbest* is the individual best position of the particle, *gbest* is the global best position of the swarm of the particles.

The weighting funtion, w is responsible for dynamically adjusting the velocity of the particles, hence it is responsible for balancing between local and global search. Applying a large inertia weight at the start of the algorithm and decaying to a small value through the PSO execution makes the algorithm search globally at the beginning and locally at the end of the execution. The weighting function w is calculated as:

$$w = w_{max} - \frac{(w_{max} - w_{min}) \cdot iter}{iter_{max}} \quad (19)$$

here,w_{max} and w_{min} are the initial and final weights,iter is the current iteration time and $iter_{max}$ is the maximum number of iterations.The proposed Fitness function for the optimization of parameters of FOPID controller is defined as:

$$F(s) = w_{max}\,(M_p+ISE) + w_{min}\,(T_p + T_s) \quad (20)$$

The flow chart depicting the implementation of PSO algorithm for optimizing the parameters of FOPID controller for the given system is as follows:

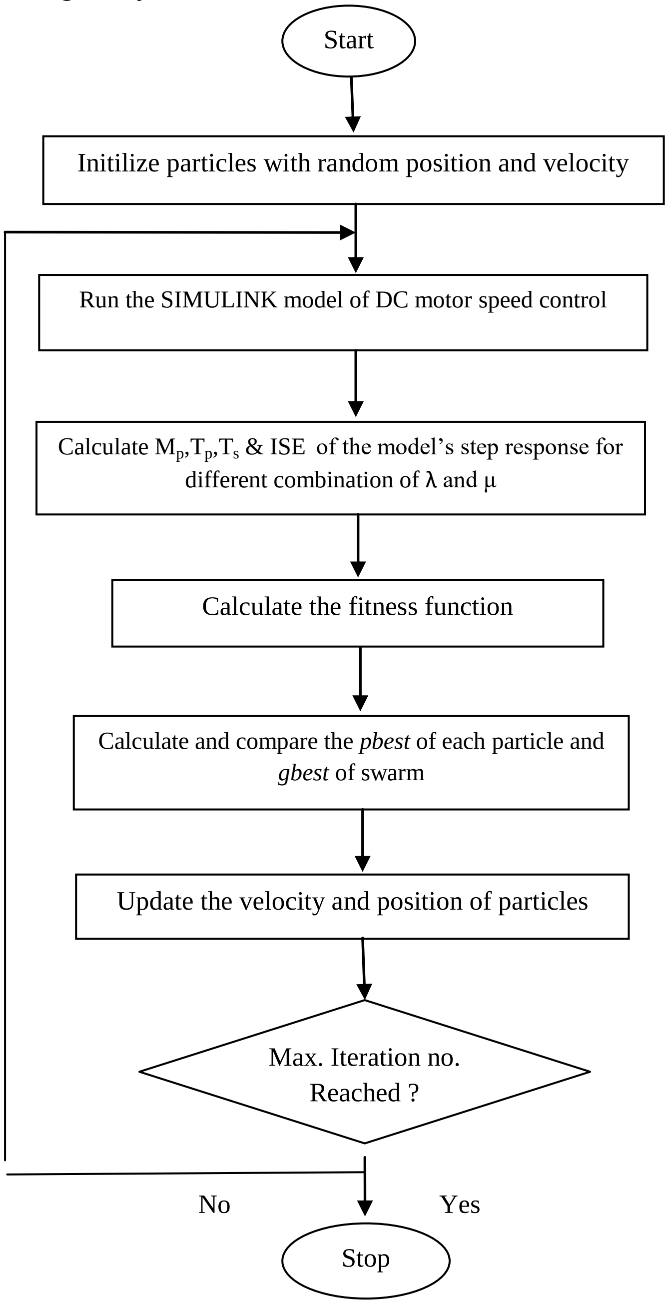

Fig. 6 Implementation of PSO in FOPID tuning for DC motor speed control

VI. SIMULATION & RESULTS

This section shows 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 $K_p = 6$, integral gain $K_i = 28.3$ and derivative gain $K_d = 0.318$.The unit step response and performance parameters such as peak overshoot,peak time and settling time for PID control is shown below:

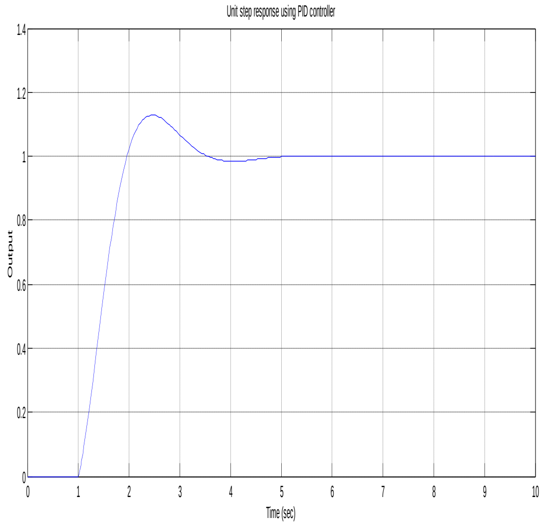

Fig. 7 Unit step response of DC motor using PID controller

Table III: Parameters for PID Control

K_p	K_i	K_d	M_p	T_p	T_s	ISE
6	28.3	0.318	12.85	2.47	3.1	0.3449

The unit step response using PID controller gives an overshoot (M_p) of 12.85%,peak time (T_p) of 2.47 sec and settling time (T_s) 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

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

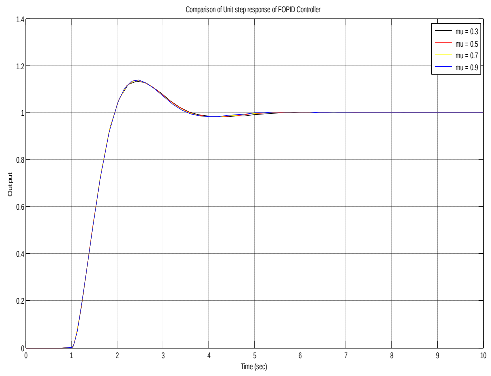

Fig.8 Unit step response of DC motor using FOPID controller for varying values of μ < 1

In figure 8 the integral order λ is kept constant where as derivative order μ is changed. Here derivative order μ <1.

Table IV: Comparison of Parameters for Different Combinations of λ=1 and 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

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

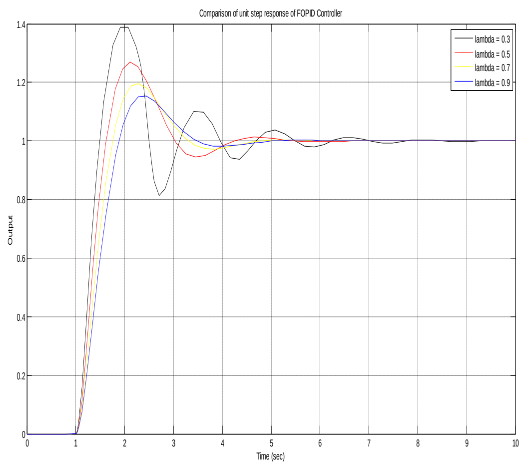

Fig.9 Unit step response of DC motor using FOPID controller for varying values of λ < 1

Figure 9 shows the unit step response of system using FOPID controller where the integral order λ <1 and is variable and derivative order μ is kept fixed.

Table V: Comparison of Parameters for Different Combinations of λ <1 and μ=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

c) With varying values of λ<1 and μ<1

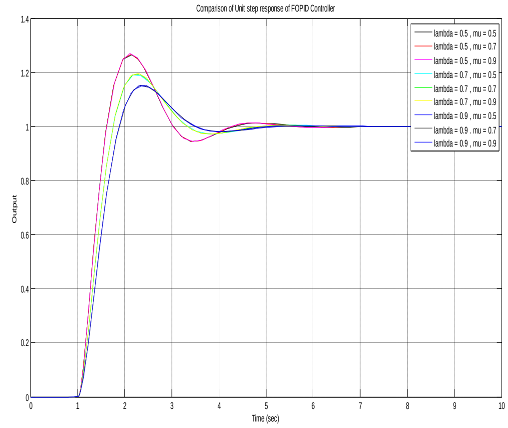

Fig.10 Unit step response of DC motor using FOPID controller for varying values of λ < 1 and μ < 1

Figure 10 shows the unit step response of system using FOPID controller where the integral order λ<1 and derivative order μ < 1.

Table VI: Comparison of Parameters for Different Combinations 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

d) With λ=1 and varying values of μ>1

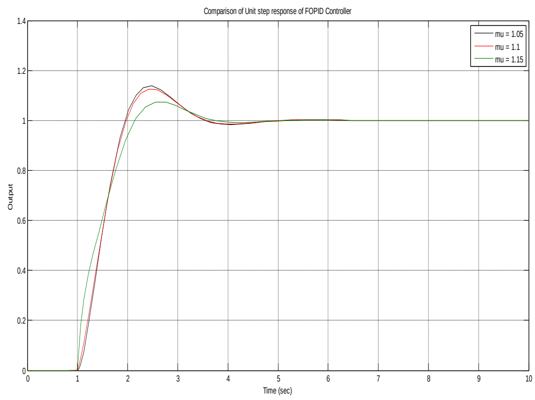

Fig.11 Unit step response of DC motor using FOPID controller for varying values of μ > 1

Figure 11 shows the unit step response of system using FOPID controller where derivative order μ > 1.

Table VII: Comparison of Parameters for Different Combinations of λ=1 and 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

e) With varying values of λ>1 and μ=1

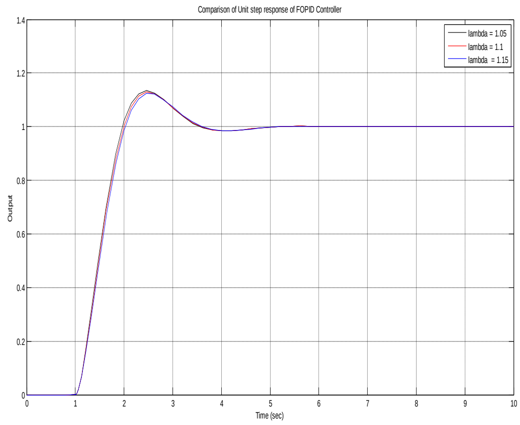

Fig.12 Unit step response of DC motor using FOPID controller for varying values of λ > 1

Figure 12 shows the unit step response of system using FOPID controller where the integral order λ > 1.

Table VIII: Comparison of Parameters for Different Combinations of λ>1 and μ=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

f) With varying values of λ>1 and μ>1

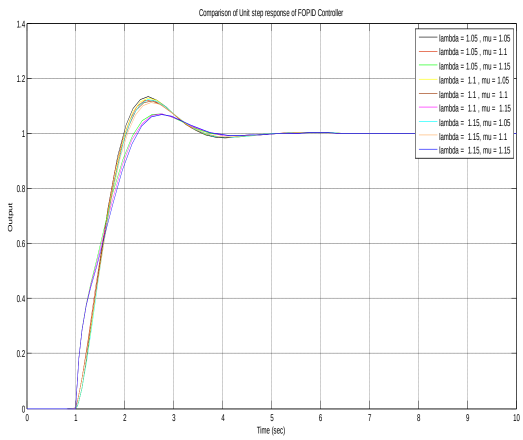

Fig.13 Unit step response of DC motor using FOPID controller for varying values of λ > 1 and μ > 1

Figure 13 shows the unit step response of system using FOPID controller where the integral order λ > 1 and derivative order μ > 1.

Table IX: Comparison of Parameters for Different Combinations 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

g) With varying values of λ<1 and μ>1

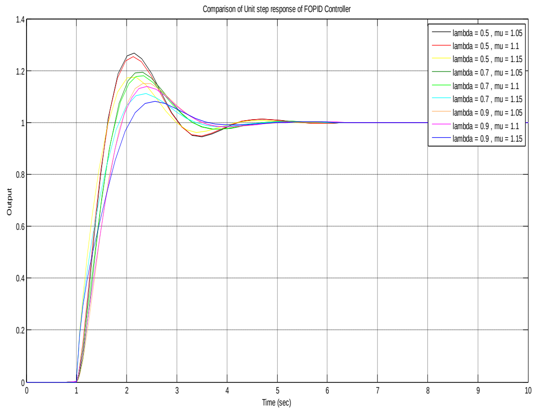

Fig.14 Unit step response of DC motor using FOPID controller for varying values of λ < 1 and μ > 1

Figure 14 shows the unit step response of system using FOPID controller where the integral order λ < 1 and derivative order μ > 1.

Table X: Comparison of Parameters for Different Combinations 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

h) With varying values of λ>1 and μ<1

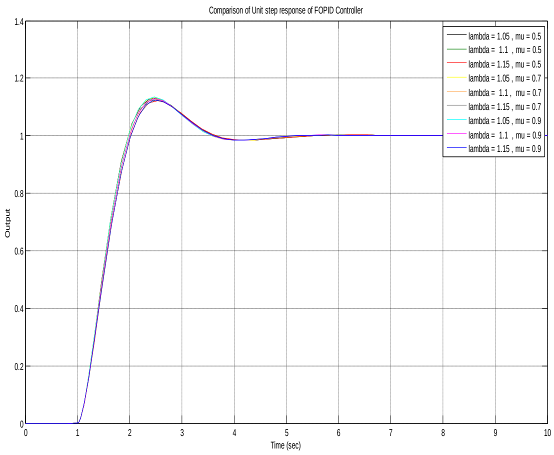

Fig.15 Unit step response of DC motor using FOPID controller for varying values of λ > 1 and μ < 1

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Figure 15 shows the unit step response of system using FOPID controller where integral order λ >1 and derivative order μ < 1.

Table XI: Comparison of Parameters for Different Combinations 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

Implementation of Particle Swarm Optimization:

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

Table XII: PSO parameter values

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

After running the PSO algorithm for different combinations of λ and μ ,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]

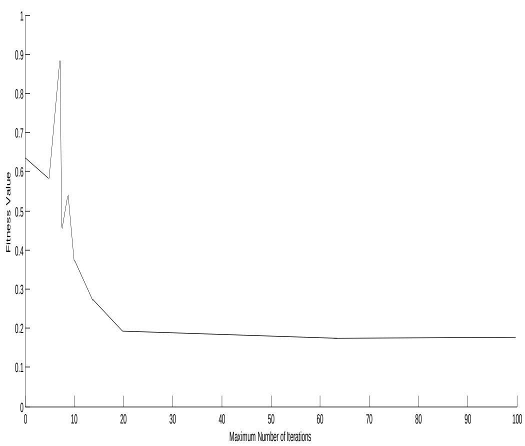

Fig. 16 Graph between fitness value and maximum number of iterations

After getting the optimal values of λ and μ ,we compare the unit step response of optimal FOPID controller and classical PID controller as shown in figure 17.

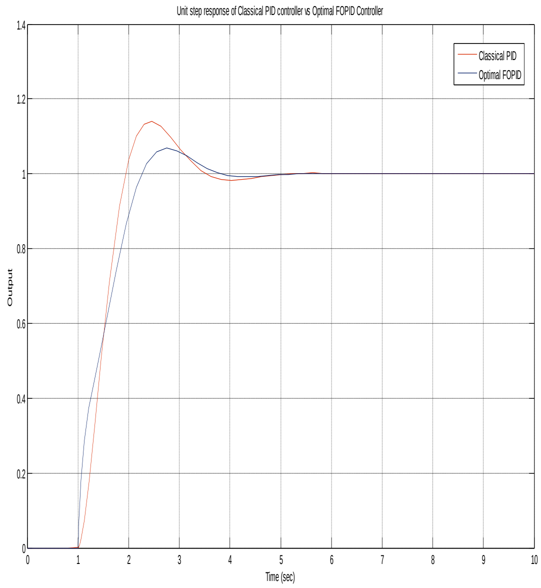

Fig. 17 Comparison of step responses of PID and FOPID controller

Table XIII: Comparison of performance parameters 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

From the above graph and table, it is clear that Fractional order PID controller largely reduces the peak overshoot obtained by classical PID controller.It also improves the settling time and integral square error and thus, enhances the control performance.

VII. CONCLUSION

In this paper, fractional order PID controller has been introduced and its tuning has been done for speed control of DC motor.Simulations has been carried out using MATLAB/SIMULINK software showing variations in unit step response and different performance parameters has been calculated when the derivative and integer order of the FOPID controller is varied.To find the optimal values of derivative and integer order,PSO algorithm has been implemented.From the simulation results, it can be concluded that the proposed FOPID controller improves the performance characteristics and provides flexiblity and robust stability as compared to the conventional PID controller applied to DC motor.

REFERENCES

[1] Jun-Yi Cao and Bing-Gang Cao, “Design of fractional order controller based on particle swarm optimization,” Int. J. Control, Automation and Systems, vol. 4, no. 6, 2006, pp. 775-781.
[2] Dingyu Xue, Chunna Zhao, YangQuan Chen, “Fractional order PID control of a DC motor with elastic shaft: A case study,” Proc. 2006 American control conference, 2006, pp. 3182-3187.
[3] 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.

[4] Deepyaman Maiti, Sagnik Biswas and Amit Konar, “Design of a fractional order PID controller using particle swarm optimization technique,” Proc. ReTIS’08, 2008.

[5] Andrezej Dzielinski and Dominik Sierociuk, “Simulation and experimental tools for fractional order control education,” Proc. IFAC’08, 2008.

[6] Hyo-Sung Ahn, Varsha Bhambhani and YangQuan Chen, “Fractionalorder integral and derivative controller design for temperature profile control,” in Proc. 2008 CCDC-2008, pp. 4767-4771.

[7] 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.

[8] Ivo Petras, “Fractional-order feedback control of a DC motor,” J. Elect. Engg., vol. 60, no. 3, 2009, pp. 117-128.

[9] Arijit Biswas, Swagatam Das, Ajith Abraham, Sambarta Dasgupta, “Design of fractional order $PI^{\lambda}D^{\mu}$ controller with improved differential evolution,” Engineering Applications of Artificial Intelligence, 22, 2009, pp. 343-350.

[10] Venu Kishore Kadiyala, Ravi Kumar Jathoth, Sake Pothalaiah, “Design and implementation of fractional order PID controller for aerofin control system,” Proc. 2009 World Congress NaBIC, 2009, pp. 696-701.

[11] Ammar A Aldair, and Weiji J Wang, “Design of fractional order controller based on evolutionary algorithm for a full vehicle nonlinear active suspension system,” Int. J. Contr Automation, vol. 3, no. 4, 2010, pp. 33-46.

[12] Vishal Mehra, Smriti Srivastava, and Pragya Varshney, “Fractional-Order PID Controller Design for Speed Control of DC Motor,” Third International Conference on Emerging Trends in Engineering and Technology, *IEEE Trans*.,2010.

[13] Radek Matusu, “Application of fractional order calculus to control theory,” International Journal of Mathematical Models and Methods in Applied Sciences, issue 7, vol 5, 2011, pp. 1162 – 1169.

[14] Subhransu Padhee, Abhinav Gautam, Yaduvir Singh, and Gagandeep Kaur , “A Novel Evolutionary Tuning Method for Fractional Order PID Controller “,International Journal of Soft Computing and Engineering (IJSCE), Vol.1, Issue-3, July 2011.

[15] K. Sundaravadivu and K. Saravanan, “Design of Fractional Order PID Controller for Liquid Level Control of Spherical Tank,” European Journal of Scientific Research,Vol.84 No.3,2012, pp.345-353.

[16] Mehdi Ghazavi Dozein, Amin Gholami, Mohsen Kalantar, “Speed Control of DC Motor Using Different Optimization Techniques Based PID Controller,” J. Basic. Appl. Sci. Res., 2(7), 2012.

[17] Karanjkar D. S.,Chatterji S.,Venkateswaran P.R.,“Trends in Fractional Order Controllers,” International Journal of Emerging Technology and Advanced Engineering,Vol.2, Issue 3, March 2012.

[18] Rinku Singhal, Subhransu Padhee, Gagandeep Kaur, “Design of Fractional Order PID Controller for Speed Control of DC Motor”, International Journal of Scientific and Research Publications, Vol 2, Issue 6, June 2012.

[19] Shivaji Karad, Dr. S. Chatterji, Prasheel Suryawanshi, “Performance Analysis of Fractional Order PID Controller with the Conventional PID Controller for Bioreactor Control,” International Journal of Scientific & Engineering Research,Vol.3, Issue 6, June-2012.

[20] Mohammad Reza Dastranj, Mojtdaba Rouhani, and Ahmad Hajipoor, “Design of Optimal Fractional Order PID Controller Using PSO Algorithm,” International Journal of Computer Theory and Engineering, Vol. 4, No. 3, June 2012.

Ankit Rastogi completed his B.Tech in Electrical & Electronics Engineering from Moradabad Institute of Technology, Moradabad, Uttar Pradesh in the year 2011.Currently he is pursuing M.Tech in Control & Instrumentation Engineering from Sam Higginbottom Institute of Agriculture, Technology & Sciences, Allahabad.His areas of interests are soft computing,electrical machines and control systems.

Pratibha Tiwari completed her B.Tech in Electrical & Electronics Engineering from United College of Engineering & Research, Allahabad, Uttar Pradesh in the year 2002 and M.Tech in Control & Instrumentation Engineering from MNNIT,Allahabad in 2006. Currently she is pursuing her Ph.D in Control & Instrumentation Engineering from Sam Higginbottom Institute of Agriculture, Technology & Sciences, Allahabad.Her areas of interests are instrumentation systems,process control and control engineering.