Simulation model of speed control DC motor using fractional order PID controller

Tigo Wati*, Subiyanto*, and Sutarno*

Department of Electrical Engineering, Universitas Negeri Semarang
Office E11 Sekaran Campus 50229, telephone +6224-8508104 Semarang, Indonesia

*Corresponding author email: ggalekputri3@gmail.com, subiyanto@mail.unnes.ac.id, sutarnounnes@mail.unnes.ac.id

Abstract. DC motor speed can be achieved by changing the armature voltage fed through converter that generally employed with conventional PID. However, conventional PID controller has some disadvantages such as the high starting overshoot and sensitivity to controller gains. On the other hand, fractional order PID has potential to accomplish what conventional PID cannot. In this study, fractional order PID controller was applied to control speed of DC motor. By calculated error that occurred by reference speed and actual speed, fractional order PID brought motor run at desired speed. The parameters of fractional order PID controller (proportional constant, integral constant, derivative constant, derivative order and integral order) are optimally tuned by using Genetic Algorithm, and the optimization performance target is based on Integral Time Absolute Error (ITAE) criterion. Oustaloup's approximation method is used to approximate the fractional order differentiator and integrator. This controller performances are tested in simulation mode using MATLAB/Simulink. Speed response of motor DC are compared between fractional order PID and conventional PID controller. The result of fractional order PID controller could reduce overshoot, settling time and steady state error. With this result, show that fractional order PID controller perform better than conventional PID controller.

1. Introduction

The DC motors are widely used in industry and commercial application such as tape motor, disk drive, robotic manipulators and in numerous control applications [1,2]. Therefore, DC motor speed control is very important. DC motors has excellent control of speed for speeding up and slowing down [3]. There are several methods to control speed of DC motor [4], i.e. traditionally armature voltage using rheostatic, conventional PID controllers, neural network controllers, constant power motor field weakening controller based on load-adaptive MIMO linearization technique, single phase uniform PWM ac-dc buck-boost converter with only one switching device or using NARMA-L2 (Nonlinear Auto Regressive Moving Average) controller.

Although many methods have been proposed, the types of PID controller continues to be the most popular controller used in industrial processes [5-6]. However, there are some disadvantages in PID controller like high overshoot and sensitivity to controller gains. In control engineering, a dynamic field of research and practice, better and better performance is constantly demanded. Many techniques on design and tuning of the PID controllers are proposed, i.e. Ziegler-Nichols method, Cohen-Coon rule, modified Ziegler-Nichols scheme, integral performance criteria, Astrom-Hagglund method, so on. Meanwhile, in order to improve the feedback control performance, variant PID controllers have been proposed, for typical examples, PID-dead time Controller, IMC-PID controller, Smith predictor-PID controller, etc [7]. More recently, Podlubny has proposed a generalization of PID controllers, namely the fractional order PID well known as $PI^{\lambda}D^{\mu}$ controller, involving an integrator of order λ and differentiator of order μ (the orders λ and μ may assume real noninteger values) [8-10].

In literature [11-13, 27], an optimal $PI^{\lambda}D^{\mu}$ has been designed by using a Genetic Algorithm, which also shows better performance when used the $PI^{\lambda}D^{\mu}$ controller than the conventional PID controller. In this paper, the study is focused on $PI^{\lambda}D^{\mu}$ controller to optimize speed control of DC motor to get a better performance. For tuning scheme, $PI^{\lambda}D^{\mu}$ controller parameters obtained optimally by Genetic Algorithm.

The basic block diagram of an electrical drive is shown in figure 1. In electrical drives [14], use of various sensors and control algorithms is done to control the speed of the motor using suitable speed control methods. Earlier only DC motors were employed for drives requiring variable speeds due to ease of their speed control methods. On the other hand, modern trends and development of speed control methods of an DC motor have increased in electrical drives extensively.

This paper deal that simulation of optimization speed control using fractional order PID for DC motor designed by MATLAB/Simulink supported SimPowerSystem and FOMCON additional toolbox. After that, DC motor performance is tested by comparing the speed response between $PI^{\lambda}D^{\mu}$ controller and conventional PID controller.

2. Experimental method

Based on figure 1, control unit containing two feedback loops is used in controlling the speed of a DC motor [15].

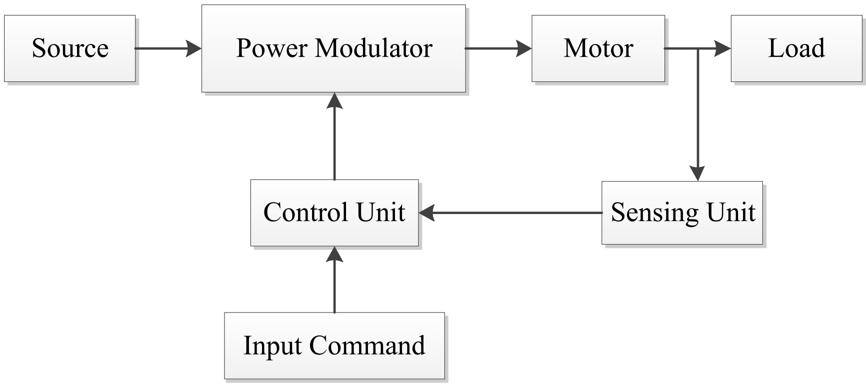

Figure 1. Block diagram of an electrical drive [14]

First, speed controller was used to controlled speed loop by calculate error between reference speed (ω_{ref}) from input command and actual speed (ω_{act}) from sensing unit. Then, the output of speed controller called reference current ($I_{reference}$) that compared with actual current (I_{actual}) as an input to controlled current loop. The speed control of DC motor is achieved by regulating the armature voltage, which is controlled by varying square wave signal from current controller fed to power modulator (converter). The speed controller that used to controlled speed loop are $PI^{\lambda}D^{\mu}$ controller and conventional PID controller.

2.1. PID controller

Based on figure 2, control unit containing conventional PID controller is used in controlling the speed of a DC motor. Fundamentally [5], conventional PID controllers are composed of three basic control actions (see equation (1)). As starting point to study, the conventional PID controller transfer function is [16]:

$$C(s) = Kp + Ki/s + Kd\,s \tag{1}$$

Where K_p is the proportional constant, K_i is the integral constant, K_d is the derivative constant. The function of each constant of a conventional PID controller can be described as follows [28], the proportional part reduces the error response of system to disturbances, the integral part eliminates the steady state error, and the derivative part dampens the dynamic response and improves the system stability. Because of this, choosing the right parameters becomes a crucial decision for putting into practice conventional PID controller [5]. In this work, value of conventional PID controller parameters obtained optimally by Genetic Algorithm.

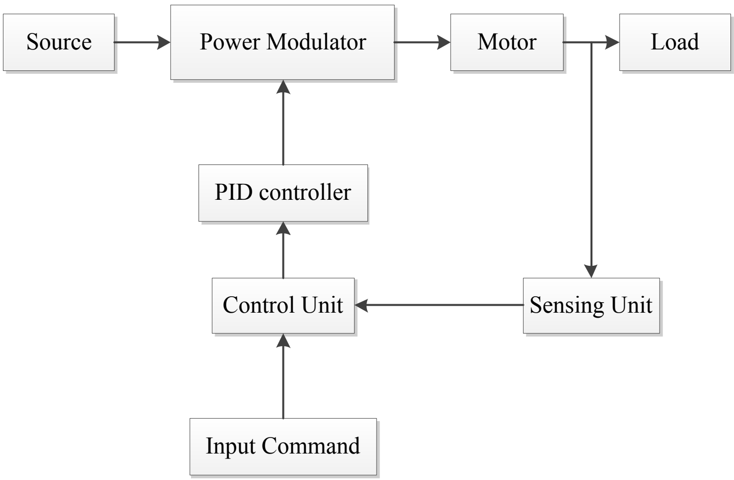

Figure 2. Block diagram of an electrical drive using conventional PID controller

2.2. $PI^{\lambda}D^{\mu}$ controller

Based on figure 3, control unit containing $PI^{\lambda}D^{\mu}$ controller is used in controlling the speed of a DC motor. $PI^{\lambda}D^{\mu}$ controller is the expansion of the conventional PID controller based on fractional calculus [2]. This idea of the fractional calculus application to control theory has been described in many other works [8,10,16]. In order to show the proposed controller, the $PI^{\lambda}D^{\mu}$ controller transfer function is [8]:

$$Gc(s) = K_p + K_i\, s^{-\lambda} + K_d\, s^{\mu} \tag{2}$$

Where K_p is the proportional constant, K_i is the integral constant, K_d is the derivative constant, λ is the integral order and μ is derivative order.

As represented in equation (2), the $PI^{\lambda}D^{\mu}$ controller has five control parameters which add more flexibility and robustness to the system, but becomes more complex obtaining the parameters of the controller [17].

2.3. $PI^{\lambda}D^{\mu}$ controller design

The general procedure of $PI^{\lambda}D^{\mu}$ controller design may be summarized by the following steps [13,18]:

- Depending on the plant characteristics, determine the correct frequency range for approximation. Oustaloup's approximation are always used due to their flexibility.
- Obtain an initial feasible parameter set for the $PI^{\lambda}D^{\mu}$ controller.
- Choose controller gain/exponent constraints using any method based on the plant's model.
- Compute control system constraint (2) using obtained controller gain.
- Decide whether you want to use Simulink for system simulation.
- Next, the choice of an appropriate performance metric is required (ISE, IAE, ITSE, or ITAE).

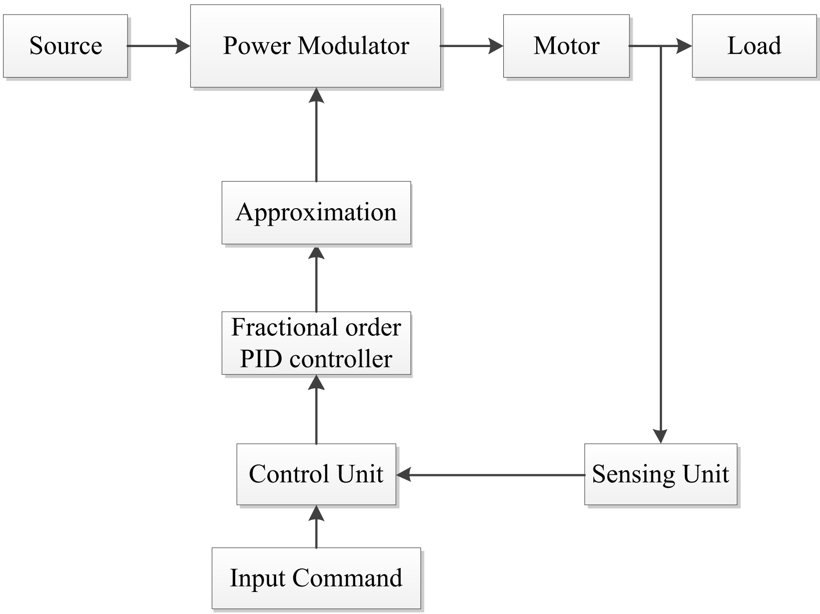

Figure 3. Block diagram of an electrical drive using $PI^{\lambda}D^{\mu}$ controller

2.4. Oustaloup's approximation

Literature [13] explained that fractional that fractional order controller is of infinite order, in the sense of integer. There is a need to approximate from infinite to a finite dimensional system. A detailed review of the various approximation methods and techniques for continuous and discrete fractional order models was done in work [19].

In this paper, Oustaloup's approximation method is used to approximate the $PI^{\lambda}D^{\mu}$ controller. The lower and higher translation frequencies for approximation are ω_b 0.001 rad/sec and ω_h 1000 rad/sec and the approximation order (N) is 5.

2.5. Tuning by Genetic Algorithm

Genetic Algorithm are a powerful search algorithm that performs an exploration of the search space that evolves in analogy to the evolution in nature [20]. Genetic Algorithm consists of three fundamental operators: reproduction, crossover, and mutation. Given an optimization problem, Genetic Algorithm encodes the parameter designed into a finite bit string, and then runs iteratively using the three operators in a random way but based on the fitness function evolution. Finally, Genetic Algorithm finds and decodes the solution to the problem from the last pool of mature strings [2, 11-12]. Table 1 are taken for controller tuning purpose:

Table 1. Parameters for Genetic Algorithm

Parameter	Conventional PID	$PI^{\lambda}D^{\mu}$
Population size	100	50
Creation function	Uniform	Uniform
Selection function	Stochastic uniform	Stochastic uniform
Crossover function	Arithmetic	Arithmetic
Crossover probability	0.65	0.65
Generation	50	25
Initial range	Lower [0 0 0] Upper [20 20 5]	Lower [0.5 1] Upper [1 1.5]

To evaluate control performance, the fitness function (J) based on Integral Time Absolute Error (ITAE) criterion which has an advantage of providing lesser overshoot along with the less settling time [12]:

$$J = \int_0^T t\,|e(t)|\,\mathrm{d}t \tag{3}$$

2.6. Model of DC motor

As reference, a separately excited DC motor equivalent circuit is shown in figure 4. The equations describing the dynamic behavior of the DC motor are as follows equations (4)-(6) [21]:

$$V = e + R_a i_a + L_a \frac{dia}{dt} \tag{4}$$

$$T_m = J\frac{d^2\omega(t)}{dt^2} + B\frac{d\omega(t)}{dt} \tag{5}$$

$$e = e(t) = K_b \frac{d\omega(t)}{dt} \tag{6}$$

Simplification and taking the ratio of *ω(s)/v(s)*, will get the transfer function as equation (7):

$$\frac{\omega(s)}{Va(s)} = \frac{Kb}{[\,JLaS^2+(RaJ+BLa)S+(Kb^2+Ra)]} \tag{7}$$

where R_a is armature resistance in ohm, L_a is armature inductance in henry, i_a is armature current in ampere, V_a is armature voltage in volts, e is back emf in volts, K_b is back emf constant in volt/(rad/sec), T_m is torque developed by the motor in N-m, ω(t) is angular speed of shaft in rad/sec, J is moment of inertia of motor and load in kg-m²/rad, and B is frictional constant of motor and load in N-m/(rad/sec).

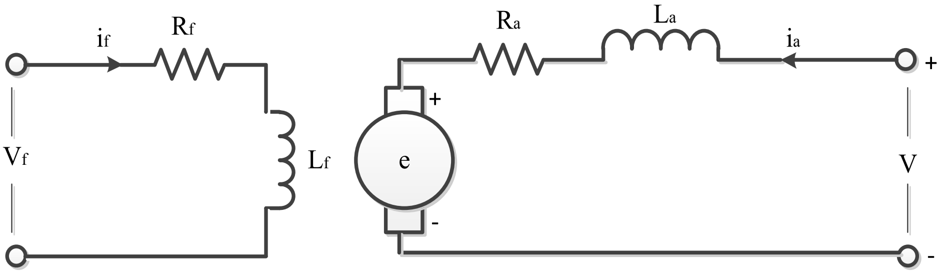

Figure 4. Equivalent circuit of separately excited DC motor [22]

The separately excited DC motor under study has the following specification and parameter [23]:

- Specification: 5 HP, 240 volts.
- Parameters: R_a = 0.5 Ω, L_a = 0.01 H, V_a = 280 V, K_b = 1.23 V/(rad/s), J = 0.05 kg.m², B = 0.02 Nm/s.

From DC motor parameter above, the overall transfer function is given in equation (8):

$$\frac{\omega(s)}{Va(s)} = \frac{1.23}{0.0005\,s^2 + 0.0252\,s + 1.523} \tag{8}$$

2.7. Simulation of DC motor speed control

The simulation build in MATLAB/Simulink environment. The basic model of DC motor speed control is shown figure 5. The model modified from MATLAB demos in power_dcdrive.mdl [23]. As shown in figure 5, a separately excited DC motor fed by a DC source through a chopper circuit. A single GTO as power converter and a free-wheeling diode form the chopper circuit.

The basic principle of speed control of DC motor [24], the output speed of DC motor can be change by changing the armature voltage for speed under and up to the rated speed. The field voltage is kept stable. A PIλDμ controlled speed control loop takes the actual speed of the motor and compares it with the reference speed to determine the reference armature current required by the motor. The current control loop consists of a hysteresis current controller (HCC). HCC is used to generate switching patterns required for the chopper circuit by comparing the actual current motor with the reference current. The chopper output provides the variable voltage essential to bring motor back to the desired speed.

In this paper, for the purpose of speed controller design, FOMCON toolbox for MATLAB/Simulink is used. In the following, a brief description of the FOMCON toolbox and the modules thereof that are applied in this work is provided in [25-26].

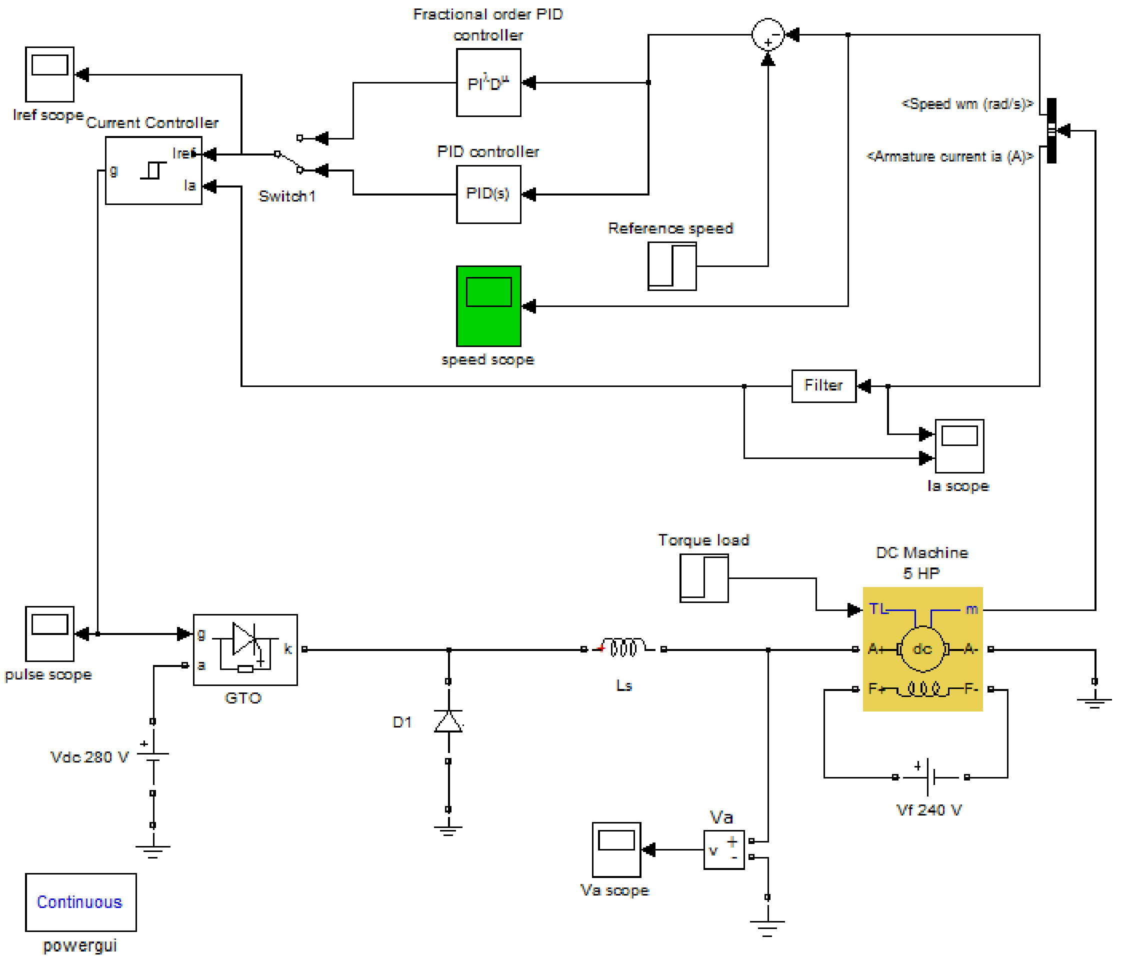

Figure 5. The basic model of DC motor drive system in MATLAB

3. Result and discussions

In this simulation, the $PI^{\lambda}D^{\mu}$ controller that is used to optimize speed control of DC motor is compare with conventional PID. By running Genetic Algorithm, parameters of each controller were obtained.

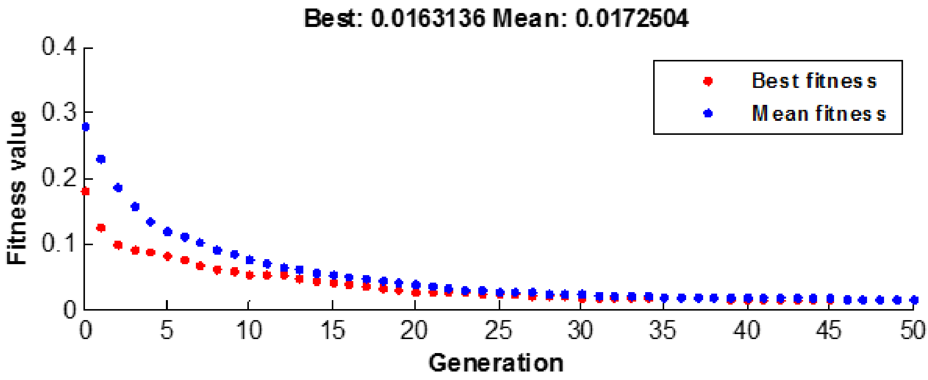

Figure 6. Fitness value vs generation for conventional PID tuning

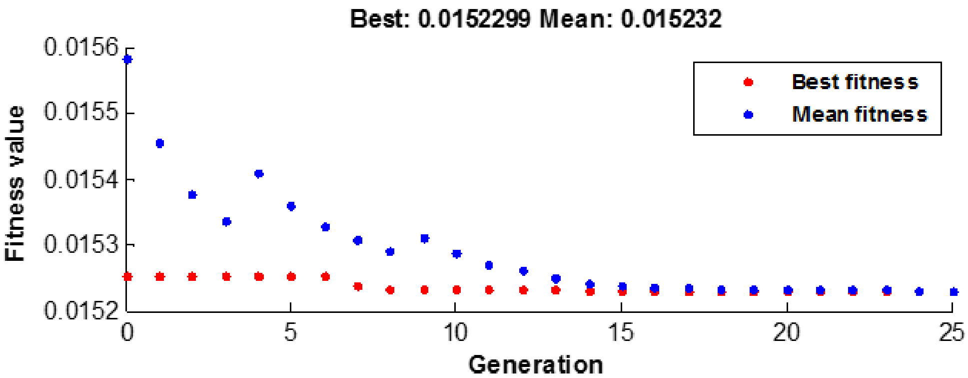

Figure 7. Fitness value vs generation for $PI^{\lambda}D^{\mu}$ tuning

Can be seen on figure 6, after 45 generations, the Genetic Algorithm performs a local search, best fitness value and average fitness value which convergene till the end of the evolution. For conventional PID tuning, the following result were obtained at 50th generation with best fitness 0.0163136 and average fitness 0.0172504. From figure 7, after 20 generations, the Genetic Algorithm performs a local search, best fitness value and average fitness value which convergene till the end of the evolution. Finally, for $PI^{\lambda}D^{\mu}$ tuning, the following result were obtained at 25th generation with best fitness 0.0152299 and average fitness 0.015232.

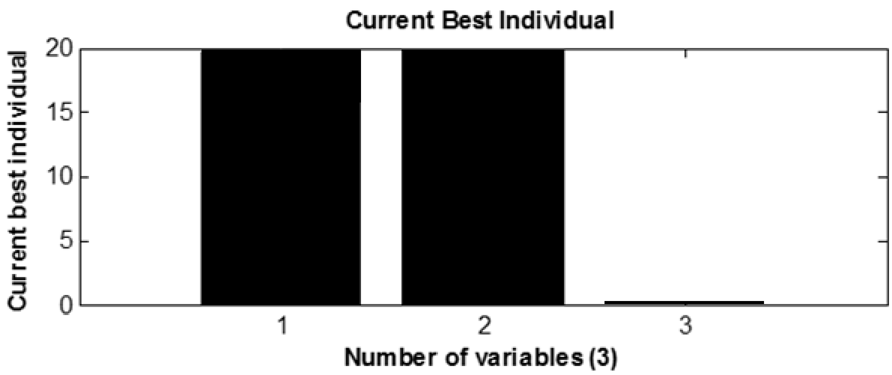

Figure 8. Current best individual vs number of variables for conventional PID tuning

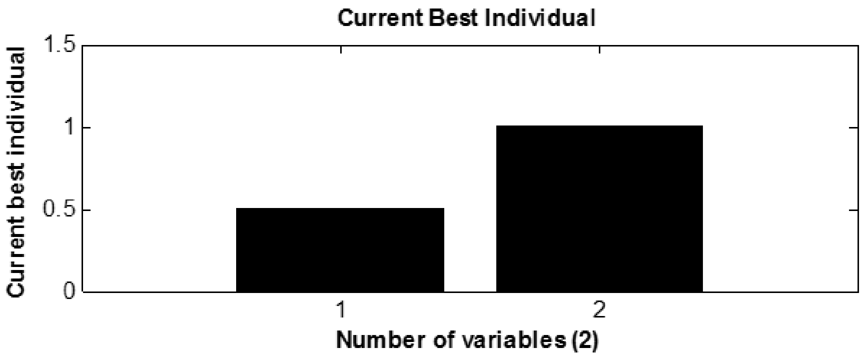

Figure 9. Current best individual vs number of variables for $PI^{\lambda}D^{\mu}$ tuning

Based results of conventional PID tuning on figure 8, number of variables 1, 2, and 3 are depict for K_p, K_i and K_d. Results of $PI^{\lambda}D^{\mu}$ tuning on figure 9, number of variables 1 and 2 are depict for λ and μ. Then, tuning results obtained in both tuning are tabulated in table 2.

Table 2. Optimized controller parameters

Controller	Value				
	K_p	K_i	K_d	λ	μ
Conventional PID	19.856	19.61	0.243	-	-
$PI^{\lambda}D^{\mu}$	19.856	19.61	0.243	0.51	1.004

To analysis the performances of controller, speed response of both controller is shown at various conditions. First, speed response is observed at condition 1 (no load or load torque 0 Nm) with reference speed 120 rad/s and sampling time 5 second.

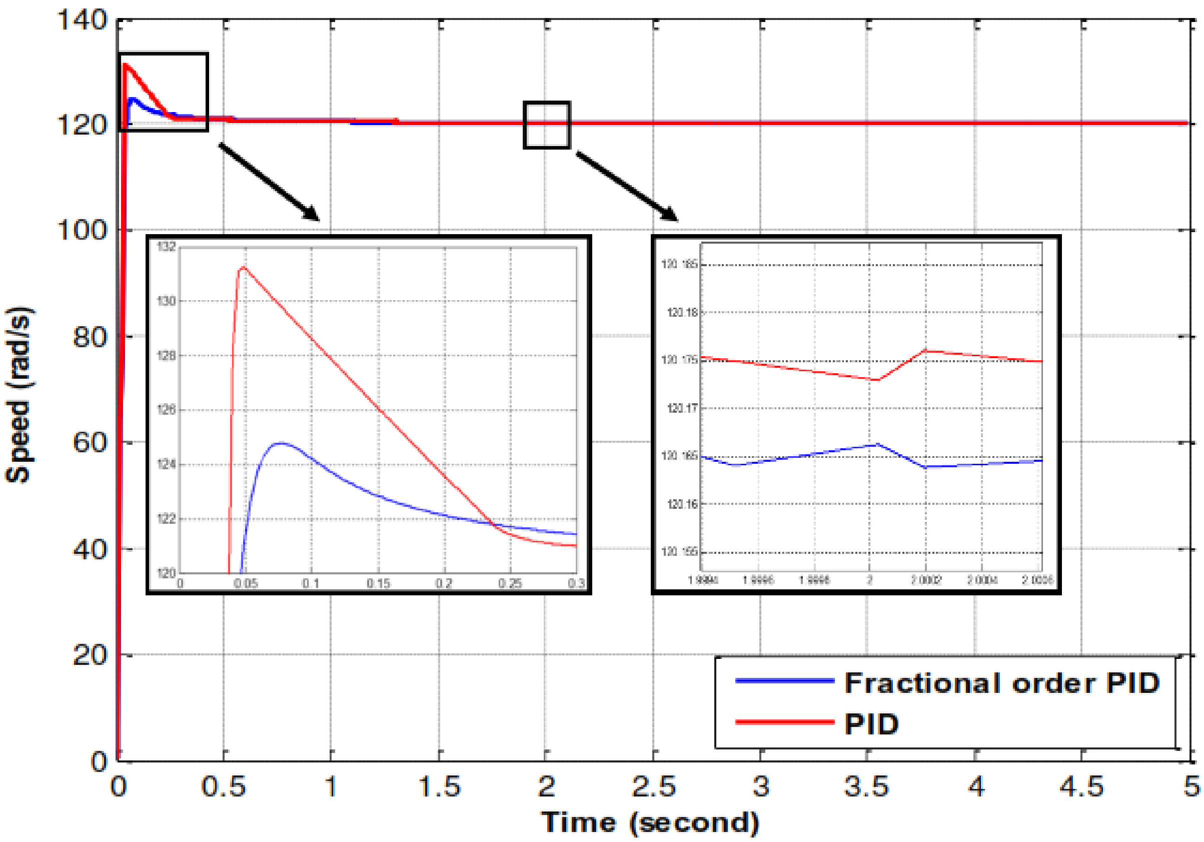

Figure 10. Speed response at condition 1 (no load)

Figure 10 presents the speed responses of the system with the application of $PI^{\lambda}D^{\mu}$ controller and conventional PID controller, respectively. It can be seen that the $PI^{\lambda}D^{\mu}$ controller has remarkably reduced the overshoot (131.25 rad/s to 124.8 rad/s), settling time (1.55 second to 1.1 second), and steady state error (120.175 rad/s to 120.165 rad/s) compared with conventional PID controller. $PI^{\lambda}D^{\mu}$ controller has achieved good performances in both transient and steady state periods. It is clear that $PI^{\lambda}D^{\mu}$ controller smoothly control DC motor with lesser settling time, peak overshoot and steady state error even at no-load condition. The simulation results obtained from figure 10 are tabulated in table 3.

Table 3. Controller performance analysis at condition 1

Controller	Value			
	Overshoot	Settling time	Rise time	Steady state error
Conventional PID	9.38 %	1.55 s	0.034 s	0.14 %
$PI^{\lambda}D^{\mu}$	4 %	1.1 s	0.037 s	0.13 %

Second, speed response is observed at condition 2 (on load or constant torque load 5 Nm) with reference speed 120 rad/s and sampling time 5 second.

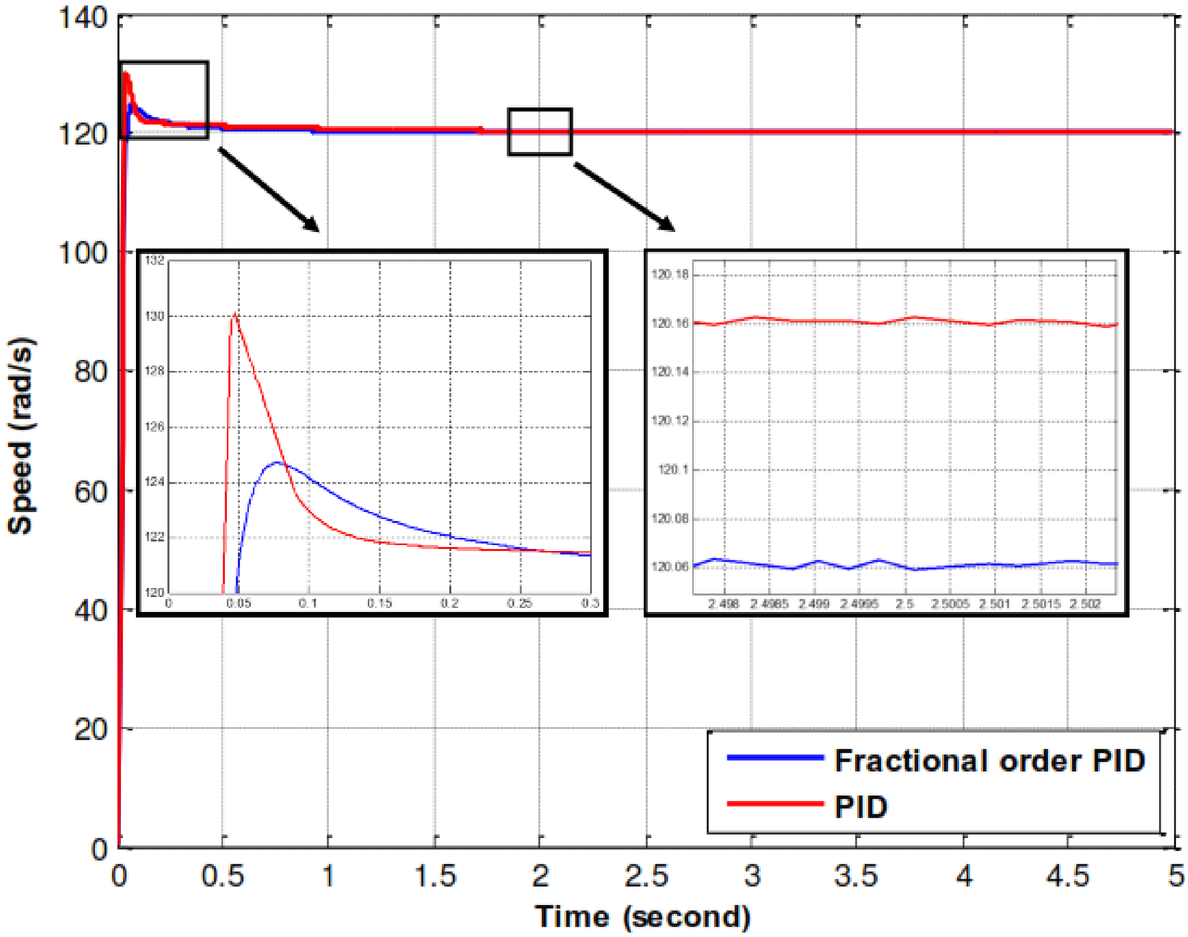

Figure 11. Speed response at condition 2 (on load)

Figure 11 present the speed responses of the system with the application of $PI^{\lambda}D^{\mu}$ controller and conventional PID controller, respectively. It can be seen that the $PI^{\lambda}D^{\mu}$ controller could reduced the overshoot (129.8 rad/s to 124.7 rad/s), settling time (1.13 second to 0.58 second), and steady state error (120.16 rad/s to 120.06 rad/s) compared with conventional PID controller. $PI^{\lambda}D^{\mu}$ controller has achieved good performances in both transient and steady state periods. It is clear that $PI^{\lambda}D^{\mu}$ controller smoothly control DC motor with lesser settling time, peak overshoot and steady state error even at on-load condition. The simulation results obtained from figure 11 are tabulated in table 4.

Table 4. Controller performance analysis at condition 2

Controller	Value			
	Overshoot	Settling time	Rise time	Steady state error
Conventional PID	8.17 %	1.13 s	0.037 s	0.13 %
$PI^{\lambda}D^{\mu}$	3.92 %	0.58 s	0.048 s	0.05 %

Third, speed response is observed at condition 3 (step change of torque load 20 Nm is applied at 2.5 second from the initial value 5 Nm) with reference speed 120 rad/s and sampling time 5 second.

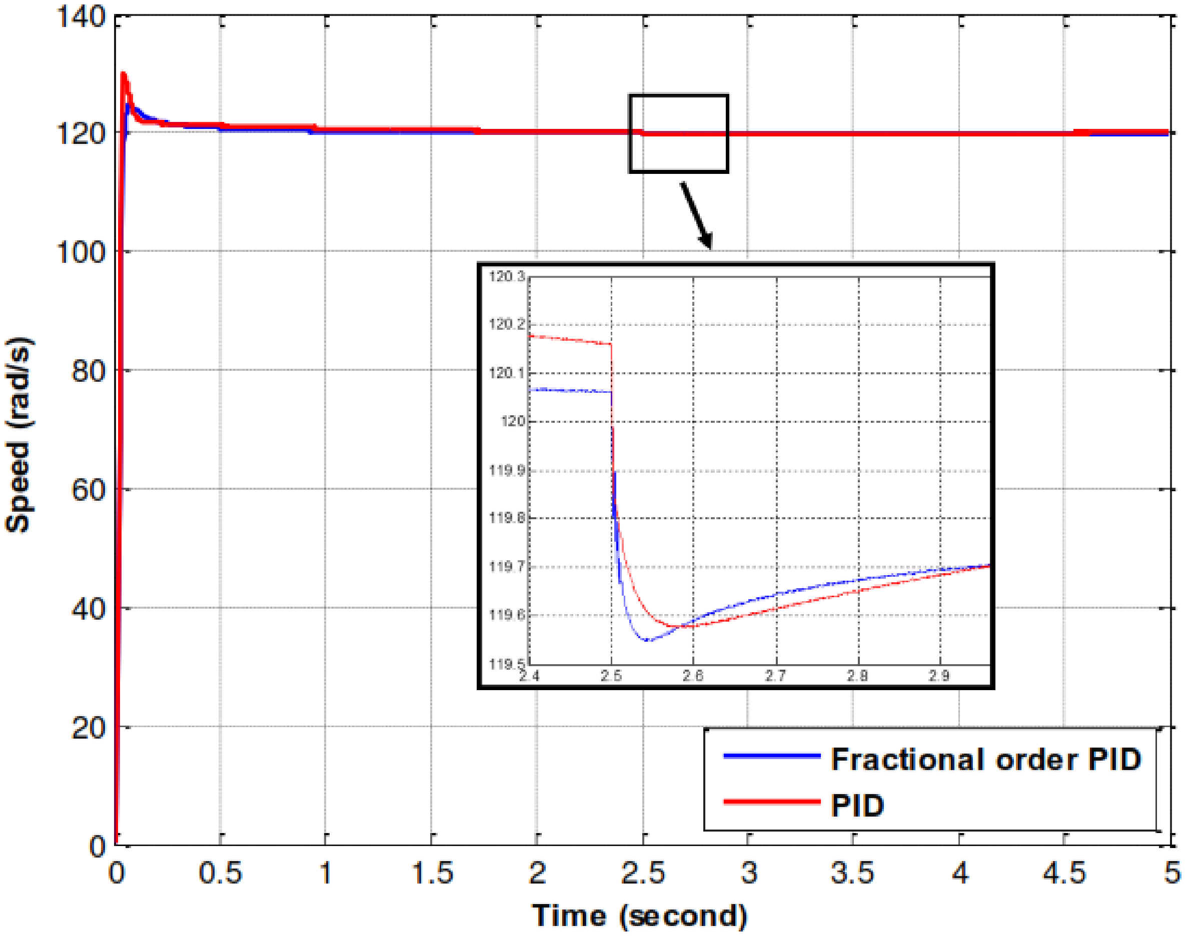

Figure 12. Speed response at condition 3 (step change of load)

With a step-load disturbance of 20 Nm, variation of speed is shown in the figure 12. It can be seen that $PI^{\lambda}D^{\mu}$ controller has undershoot 119.55 rad/s and conventional PID controller has undershoot 119.58 rad/s. When simulation rise up to steady state again, both of controllers had same steady state error 199.7 rad/s. At condition step change of load, conventional PID controller smoothly control DC motor with lesser undershoot. Also, both controllers can recover the desired speed at same time. The simulation results obtained from figure 12 are tabulated in table 5.

Table 5. Controller performance analysis at condition 3

Controller	Value	
	Undershoot	Steady state error
Conventional PID	0.35 %	0.25 %
$PI^{\lambda}D^{\mu}$	0.37 %	0.25 %

Fourth, speed response is observed at condition 4 (step change of speed 140 rad/s is applied at 2.5 second from the initial value 120 rad/s) with torque load 5 Nm and sampling time 5 second.

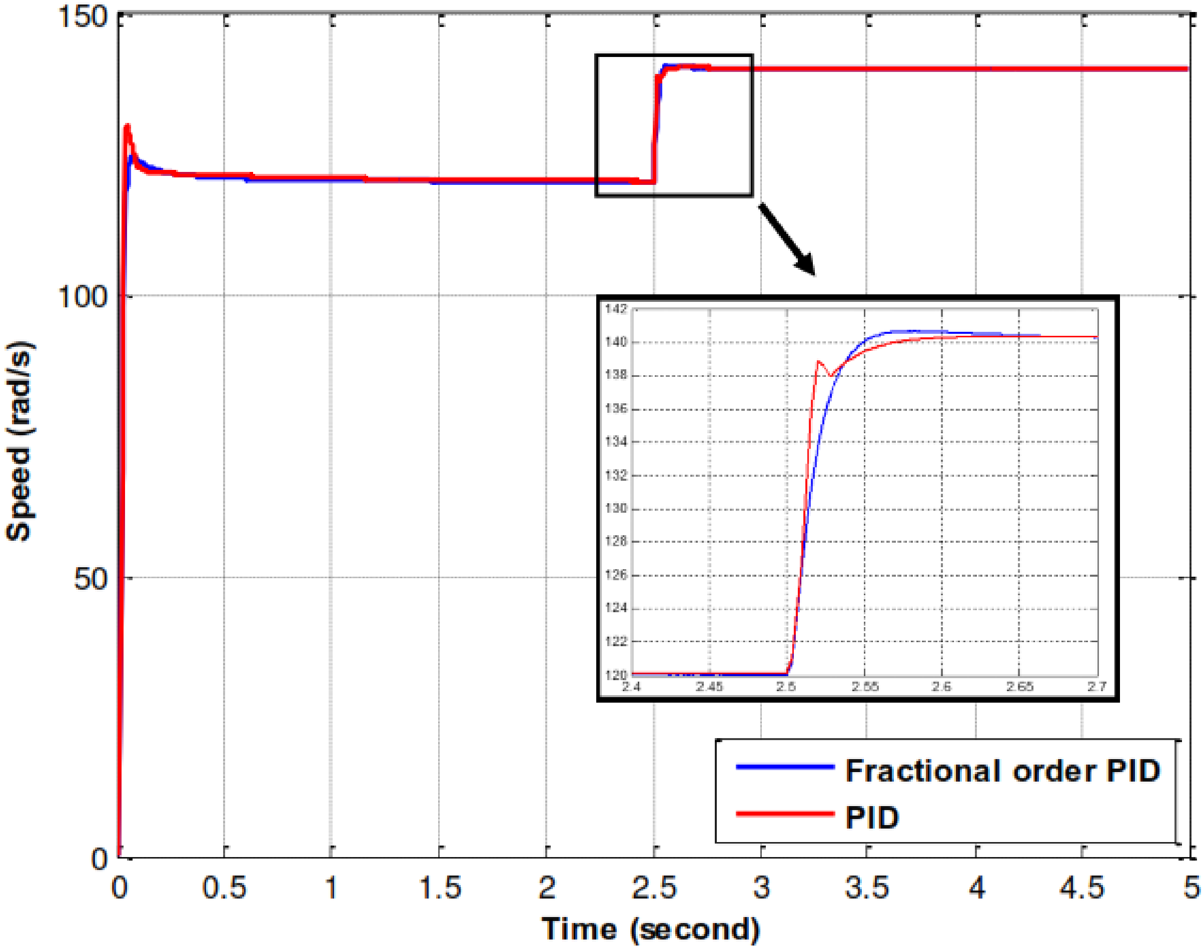

Figure 13. Speed response at condition 4 (step change of speed)

With a step-speed disturbance of 140 rad/s, variation of speed is shown in the figure 13. It can be seen that the $PI^{\lambda}D^{\mu}$ controller more sluggish at transient period, but $PI^{\lambda}D^{\mu}$ controller has overshoot 140.65 rad/s and conventional PID controller has overshoot 140.38 rad/s. When simulation rise up to steady state again, $PI^{\lambda}D^{\mu}$ controller has error 143.5 rad/s and conventional PID controller has error 144.04 rad/s. At condition step change of speed, $PI^{\lambda}D^{\mu}$ controller smoothly control DC motor and lesser steady state error. In other hand, conventional PID controller also proposed lesser overshoot. The simulation results obtained from figure 13 are tabulated in table 6.

Table 6. Controller performance analysis at condition 4

Controller	Value	
	Overshoot	Steady state error
Conventional PID	0.27 %	2.8 %
$PI^{\lambda}D^{\mu}$	0.46 %	2.5 %

Fifth, speed response is observed at condition 5 (stopping motor by changing reference speed to 0 rad/s at 4 second from the initial value 120 rad/s) with torque load 5 Nm and sampling time 5 second.

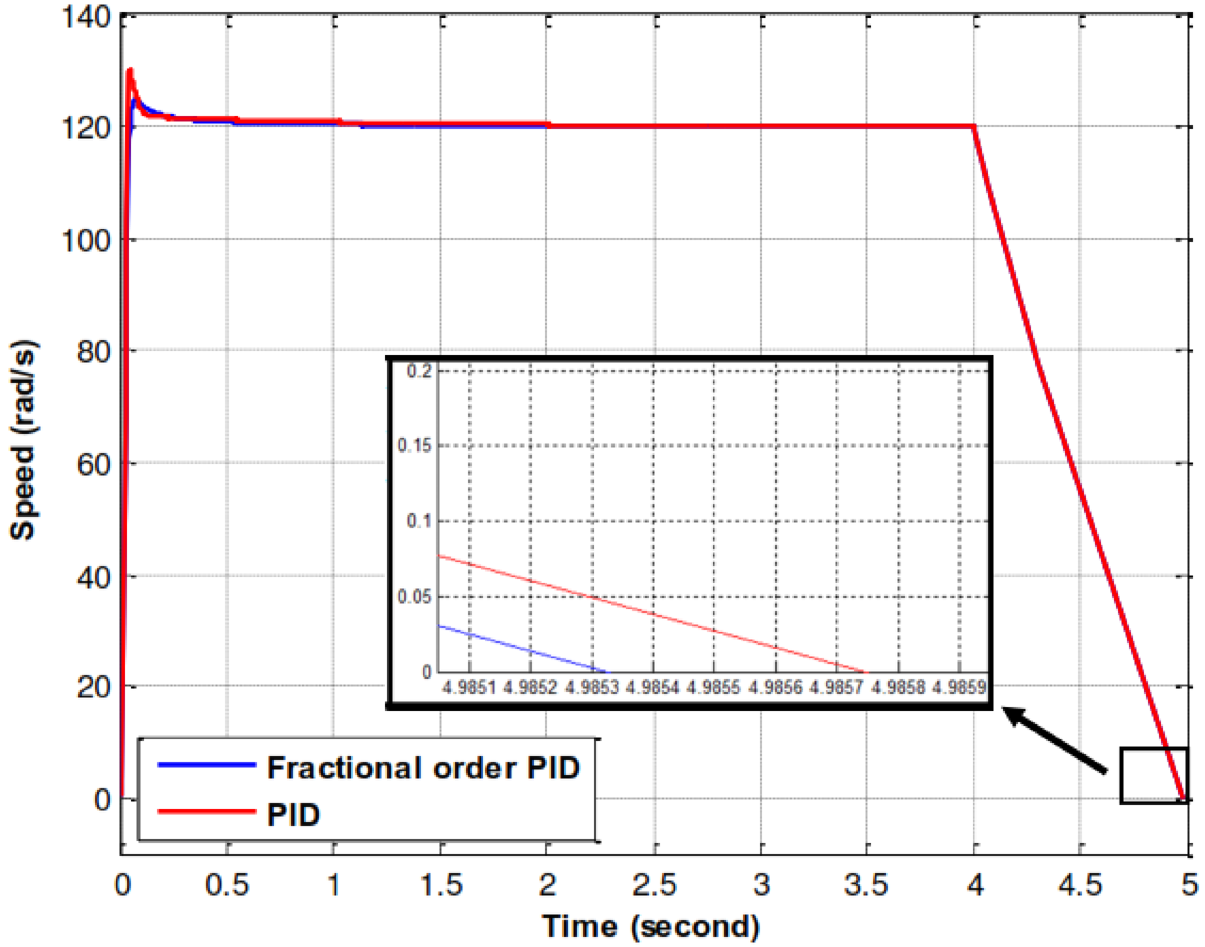

Figure 14. Speed response at condition 5 (stopping motor)

Figure 14 shows that the $PI^{\lambda}D^{\mu}$ controller bring speed motor to 0 rad/s at 4.9853 second and conventional PID controller bring speed motor to 0 rad/s at 4.9857 second. For stopping DC motor until running at 0 rad/s, $PI^{\lambda}D^{\mu}$ controller is faster than conventional PID controller.

4. Conclusions

This paper presented that $PI^{\lambda}D^{\mu}$ controller was used to optimize speed control of DC motor, particularly separately excited DC motor. The parameter of $PI^{\lambda}D^{\mu}$ and conventional PID controller are optimally tuned by Genetic Algorithm. Both controllers are compared in simulation at five various condition, no-load, on-load, step change of load, step change of speed, and stopping motor. The results of $PI^{\lambda}D^{\mu}$ controller could reduce overshoot, settling time and steady state error at no-load and on-load condition. $PI^{\lambda}D^{\mu}$ controller could reduce error, control more sluggish and smoothly at step change of load and step change of speed condition. Also, $PI^{\lambda}D^{\mu}$ controller is faster than conventional PID during stopping motor. With this results, show that $PI^{\lambda}D^{\mu}$ has more flexibility and capability, also verified could optimized speed control of DC motor.

References

[1] Ivo Petras. Fractional Order Feedback Control of a DC Motor. Journal of Electrical Engineering Vol. 60, No. 3, 2009:117-128
[2] Vishal Mehra, Srimiti Srivastava, and Pragya Varshney. Fractional Order PID Controller Design for Speed Control of DC Motor. 3rd International Conference on Emerging Trends in Engineering and Technology, 2010:422-425
[3] Kaustubh S. Deshmukh, Rutuja S. Hiware. Speed Control of Separately Excited DC Motor

using Chopper. IRJET, 2017:799-803

[4] Amir Faizy, and Shailendra Kumar. DC Motor Control using Chopper. Bachelor Thesis. 2011. National Institue of Technology Rourkela: India

[5] Adel A. A. El-Gammal, and Adel A. El-Samahy. Adaptive Tuning of a PID Speed Controller for DC Motor Drives using MOPSO. 11th International Conference on Computer Modelling and Simulation, 2009:398-404

[6] Sajid A. Bhatti, A. Daraz, and Engr S. Abdullah Malik. Comparison of PI and IP Controller by using ZN Tuning Method for Speed Control of DC Motor. International Conference on Intelligent Systems Engineering 2016

[7] Ying Luo, and Yang Q. Chen. Stabilizing and Robust Fractional Order PI Controller Synthesis for First Order Plus Time Delay Systems. 50th IEEE CDC-ECC, 2011:2040–2045

[8] Igor Podlubny, Fractional-Order Systems and Fractional Order Controller. Institute of Experimental Physic, Slovak Acad. Sci., Kosice, 1994:1-18

[9] Hamamci S. E. Stabilization using Fractional-order PI and PID Controllers. Nonlinear Dynamics, 2008:1-24

[10] I. Podlubny, L. Dorcák, I. Kostial. On Fractional Derivatives, Fractional Order Dynamic Systems and $PI^{\lambda}D^{\mu}$ Controllers. 36th IEEE Conference Decision and Control, 1997:4985–4990.

[11] Jun-Yi Cao, Jin Liang, and Bing-Gang Cao. Optimization of Fractional Order PID Controllers Based on Genetic Algorithm. 4th International Conference on Machine Learning and Cybernetics, 2005:5686-5689

[12] Vimal Kumar, A. S. Jhunghare. Fractional Order PID Controller for Speed Control of DC Motor Using Genetic Algorithm. International Journal for Scientific Research & Development, Vol.2, Issue 02, 2014:896-898

[13] Amit S. Chopade, Swapnil W. K., A. S. Junghare, and M.V. Aware. Fractional Order Speed Controller for Buck Converter fed DC Motor. IEEE First International Conference on Control, Measurement and Instrumentation, 2016:331-335

[14] Amitpal S.I.S. Bhatia, Vinit K. Gupita, Sourav A. Sethi. Simulation and Speed Control of Induction Motor Drives. Bachelor Thesis. 2012. National Institue of Technology Rourkela: India

[15] N. C. Pemmaraju. Intelegent Speed Control of DC Motor using ANFIS. Thesis. 2006: Texas A&M University-Kingsville

[16] B.M. Vinagre, I. Podlubny, l. Dorcak, V. Fellu. On Fractional PID Controller: A Frequency Domain Approach. IFAC Digital Control: Past, Present and Future of PID Control, Spain, 2000:51-56

[17] J. Viola, L. Angel, and J. M. Sebastian. Design and Robust Performance Evaluation of a Fractional Order PID Controller Applied to a DC Motor. Journal of Automatica SINICA, Vol. 2, No.2, 2017:304-314

[18] A. Tepljakov, Emmanuel A. Gonzales, E. Petlenkov, J. Belikov, Concepcion A. Monje, and I. Petras. Incorporation of Fractional Order Dynamics into an Existing PI/PID DC Motor Control Loop. ISA Transactions, 2015:1-12

[19] B. M. Vinagre, I Podlubny, A. Hernandez, and V. Feliu. Some Approximations of Fractional Order Operators used in Control Theory and Applications, Fractional Calculus and Applied Analysis 3 No. 3, 2000:231–248

[20] Ecaterina E. Vladu, and Toma L. Dragomir. Controller Tuning Using Genetic Algorithm. 1st Romanian-Hungarian Joint Symposium on Applied Computational Intelegence, 2004.

[21] P. Karpagavalli, and A. E. Jeyakumar. Simulation Analysis on Proportional Integral and Derivatif Control of Closed Loop DC Motor Drive with Bipolar Voltage Switching. American Journal of Applied Science, 2013:714-723

[22] R. Khrisnan. Electric Motor Drives; Modeling, Analysis, and Control. 2001. Prentice Hall, New Jersey:p. 25

[23] Le-Huy H. (2012). SimPowerSystems Demos: Chopper-fed DC Motor Drive. [Online]. Available: http://www.mathwork.com.

[24] Moleykutty George. Speed Control of Separately Excited DC motor. American Journal of Applied Sciences, 2008:227-233
[25] A. Tepljakov, E. Petlenkov, and J. Belikov. A Flexible MATLAB Tool for Optimal Fractional-order PID Controller Design Subject to Specifications. *31st Chinese Control Conference,* 2012:4698–4703
[26] A. Tepljakov, E. Petlenkov, and J. Belikov. FOMCON: a MATLAB Toolbox for Fractional-Order System Identification and Control. *International Journal of Microelectronics and Computer Science,* Vol. 2, No. 2, 2011:51–62
[27] Mihailo P. Lazarevic, Srecko A. Batalov, Tihomir S. Latinovic. Fractional PID Controller by Genetic Algorithms for a Three DOF's Robot System Driven by DC Motors. IFAC Joint Conference SSSC,2013:385-390
[28] Mohamed Shamseldin. Speed Control of High performance Brushless DC Motor. Thesis. 2016. Helwan University:Egypt

Acknowledgments
The authors like to thank members of UEESRG for contributions to the manuscript, and to Engineering Faculty of Semarang State University.