

# Identification and Speed Control of DC Motor Using Fractional Order PID: Microcontroller Implementation

Magdi M. El-Saadawi, Eid Abdelbaqi Gouda, Mustafa A. Elhosseini, and Mohamed Said Essa

**Abstract**—This paper uses Fractional-order PID control (FOPID) to control the speed of the DC motor. FOPID is more flexible and confident in controlling control higher-order systems compared to classical PID. In this work, the FOPID controller tuning is carried out using different methods ranging from classical techniques to most recent heuristic methods are Fractional Grey wolf Optimization and Nelder-Mead. Moreover, parameter estimation of real-world DC motor is carried out experimentally using Matlab/Simulink interfaced to an Arduino Uno board. The feasibility of FOPID is demonstrated through applications to well-known DC motor case study and the estimated DC motor. Based on ISE, ITE, and ISTE performance measures, the proposed approach provide less settling time, rise time and comparable overshoot compared with existing literature approaches. A robustness assessment with differences in the DC motor components is performed. Simulation finding provide validation of the suggested work and the FOPID controller effectiveness as compared to classical PID controller in terms of robustness and control effect.

**Index Terms**— DC engine, Speed control, Experimental work, Matlab simulation, FOPID controller, PID Controller, Genetic Algorithm, Grey wolf optimization.

## I. INTRODUCTION

Most of the DC motors preferred for industrial applications. It is dependable, simple to use and having low cost [1]. System identification is the science to construct a mathematical model able of showing the basic properties of a provided system beginning from watched input-output or output-only data [2]. In the past decades, process control has been developed rapidly in the industrial field, and the controller proportional integral derivative (PID) was the most typical one because of its simple structure and strong robustness. However, it is difficult to achieve the expected performance because of its long adjustment time, large overshoot, especially in multi-input multi-output (MIMO) systems. Three parameters  $K_p$ ,  $K_i$  and  $K_d$  are adjusted to meet the preset performance functions which are usually composed of adjusting time, overshoot, and steady-state error.

The parameter settings of the PID controller are the key to the PID controller design because they directly affect the control performance of the system. In the past few years, the heuristic algorithms are used to tune the initial parameters' values of the PID controller.

A considerable amount of literature has been are not limited to classical techniques for detecting the best value of the three parameters  $K_p$ ,  $K_i$  and  $K_d$ . Ziegler Nichols (Z-N) [3] presented a method to best tune these parameter. In many areas of industrial control, it is hard to get the ideal value using the ZN

method. The Nelder-Mead (NM) simplex algorithm [4] is intended to solve the traditional unrestricted optimization issues to reduce a provided nonlinear function such as PID tuning.

Over the past two decades, much of the current literature on PID tuning payed particular attention to heuristic methods. There has been a growing trend in recent year's amount of literature on applying these methods to get the ideal values of the PID components. These methods include; Neural Network (NN) [5], Genetic Algorithms (GA) [6], Particle Swarm Optimization (PSO) [7], Grey wolf Optimization (GWO) [8].

From many optimization techniques which is inspired by evolutionary biology to use for optimizing the controller parameter such as Genetic Algorithm and GWO. They will be discussed in details as follows:

GA [9] is a worldwide stochastic technique for optimizing adaptive searches depended on the natural selection mechanisms. GA has recently been acknowledged as an effective and efficient method for solving issues of optimization compared to other methods of optimization [10].

The main steps of GA include: (1) randomly create a set of chromosomes, (2) evaluate fitness function, (3) based on fitness function, select the most promising candidates, (4) Repeat the previous steps till reaching the optimal solution, perform crossover, and mutation then determine fitness value. PID Tuner provides a fast and widespread single-loop PID approach for blocks of the Simulink PID Controller [11]. The computer program naturally computes from the Simulink linearized plant demonstrates input and output for submission to the controller. The tuning points to balance performance and robustness whereas accomplishing the control bandwidth and phase margin. Single –loop control is utilized for exact controlling of different linear/nonlinear forms, its simple to settle, alter, and execution misfortune compared to multivariable control. Single–loop control has either independent design which uses when small interactions or consecutive design which begins with quick loop, the most part way better execution, but the external loop is slow, and loops depend on each other.

GWO algorithm is an optimization technique that is utilized to solve optimization issues of diverse changes [12]. Like other heuristic algorithms in the region of developmental computation, GWO need not to the gradient of the function in its optimization prepare. GWO is an optimization method that mimics the authority hierarchy and grey wolves hunting mechanism in nature. This algorithm needs to set a number of parameters, specifically, initialize alpha, beta, and delta,

number of look operators, the greatest number of iterations, number of locales chosen for a look (out of n gone by locales) and the halting basis. The main steps of GWO include (1) following, chasing, and deal with victim, (2) the victim is chased, surround, and harass until the movement stops, (3) attacks on the victim.

Recent research has provided a better showing of the fractional calculus, and thus a control strategy known as fractional control was showed proposing apply of non-integer operators to the model and design control systems. Some fractional control applications have good performance against external disturbances and parametric uncertainties, with good findings in industrial processes control [13].

Podlubny [14] replaced the conventional PID controller by an advanced controller which called FOPID which contented a fractional order in the calculations, symbolized by  $(PI^\lambda D^\mu)$  "where  $(\lambda, \mu)$  correspond fractional order". Not all the conventional and advanced methods used for tuning the conventional PID controller can be applied for the proposed FOPID due to the existence of fractional orders components. So, the applied optimization techniques (such as Nelder-Mead) should take into considerations the complicity of the FOPID controller parameters especially the  $(\lambda)$  and  $(\mu)$  ones. These parameters increase the flexibility and system robustness and better results than the conventional PID controller. This study will explore different aspects of FOPID, different tuning methods, and parameter estimation approaches. The fundamental purpose of the paper is determine the extent to which FOPID is applicable and whether it would be perfect compared to PID controllers. It is expected that the obtained results can make an important contribution to the field of parameter estimation and controlling motors using FOPID. The primary contributions of this research paper are:

- A novel method for estimating the parameter of DC Motor by applying Matlab/Simulink interfaced with Arduino kit.
- DC motor Speed control by applying FOPID.
- Comparing FOPID to most well-known controllers.

The rest of this paper is arranged as follows; part 2 covers the mathematical DC motor model, while the DC motor experiment setup and estimation of the motor transfer function are presented in sections III and IV. Part V demonstrates the suggested FOPID controller. Computer results and simulations are presented in section VI. Finally, work done and results are summarized in the conclusion section.

## II. DC MOTOR MODELING

To test the applicability, and the effectiveness of the suggested controller and the methods of controller tuning, an appropriate electrical application; namely, DC motor was selected, which is a motor for its simplicity and knowledge of its performance. The block diagram of a conventional DC motor with and without PID/FOPID controller is shown in Figs. 1 and 2.

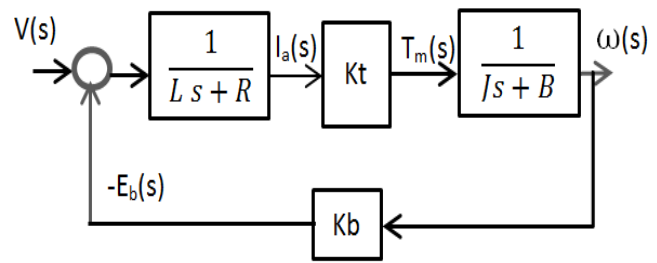


Fig. 1 Block Diagram of DC Motor (Without PID/FOPID controller)

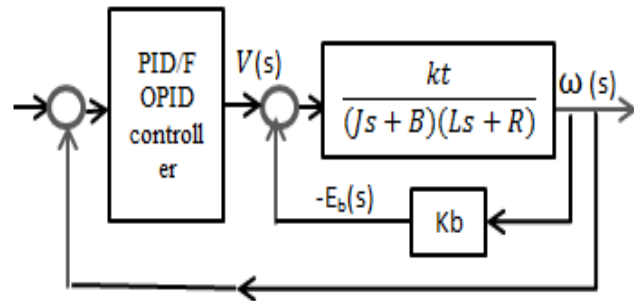


Fig. 2 DC motor speed control (With PID/FOPID controller)

The general transfer function of motor can be described as follows:

$$Gp(S) = \frac{w(s)}{V(s)} = \frac{Kt}{((Js+B)(Ls+R)+KtKb)} \quad (1)$$

If the motor components are provided such as in Table 1 the corresponding transfer function for this motor can be defined as:

$$Gp(S) = \frac{2}{s^2+12s+20.02} \quad (2)$$

Table 1. A DC motor parameter [15]

Parameters	values
Armature resistance R	1 Ω
Angular velocity of rotor ω (rad/s)	N/A
Input terminal voltage V (V)	N/A
Back electromotive force E <sub>b</sub>	N/A
EMF constant K <sub>b</sub>	0.01 V.s/rad
Friction constant B	0.1 N.m.s/rad
Rotor inertia J	0.01 kg.m <sup>2</sup> /rad
Self-inductance of armature L	0.5 H
Torque constant K <sub>t</sub>	0.01 N.m/A

However, in case that there is a difficulty to find the motor parameter due to any practical reasons the authors proposed a new method for estimating the motor transfer function based on Matlab toolbox which will be discussed in the following section.

## III. ESTIMATION OF THE MOTOR TRANSFER FUNCTION

This section presents a new proposed method for estimating the motor transfer function based on Matlab toolbox. The proposed method can be used in case that there is a difficulty

to find the motor parameter due to any practical reasons. The proposed method depends on using Arduino kit interfaced to Matlab/Simulink.

The general motor transfer function was previously given by (1). If the motor parameters ( $k_t$ ,  $k_b$ ,  $J$ ,  $B$ ,  $L$ , and  $R$ ) are not known, then an approximate method to estimate the motor equivalent is proposed. This can be done through toolbox for system identification which is supported by Matlab as shown in Fig. 5.

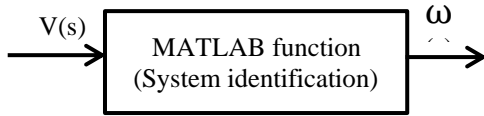


Fig.5 Matlab function (System identification)

A. Verification of the estimation method

To verify the proposed technique, a well-defined dc motor with the components given in Table1 is utilized. The exact transfer function of this motor is defined and given in (2). In case that an input voltage of 5 V is supplied to the motor as shown in Fig. 6, the corresponding output speed can be calculated and drawn as shown in Fig. 7.

The proposed method is then used to estimate the transfer function for the same motor assuming that the motor parameters are unknown. Given the input voltage and the output speed curve (Figs. 6 and 7), the estimated transfer function can be obtained from toolbox for system identification. The estimated transfer function for this motor by is defined as follows:

$$G_p(S) = \frac{2.001}{s^2 + 12.5s + 20.03} \quad (3)$$

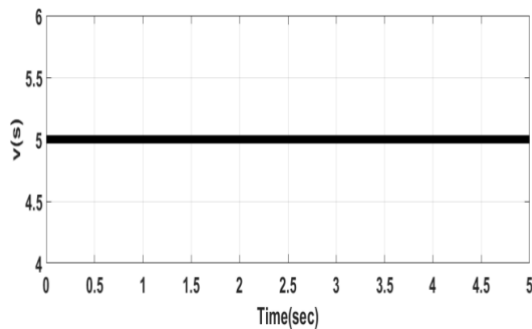


Fig.6 The input supply voltage

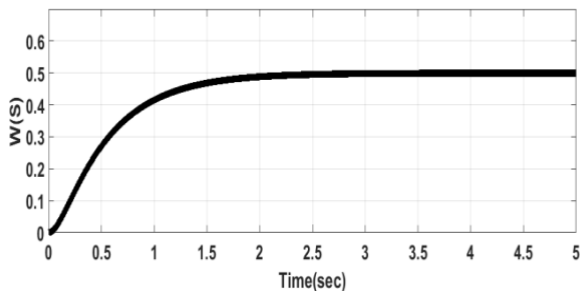


Fig. 7 The output motor angular velocity for the real data  
By comparison between the exact transfer function in (2) and the obtained estimated one (3), the two equations are very close. The obtained error between the real and estimated data

is less than 2% which can be accepted. So, the proposed estimation method can be used with acceptable accuracy in case that any parameters motor is unknown. Figure 8 shows the speed error signal between the exact and estimated method.

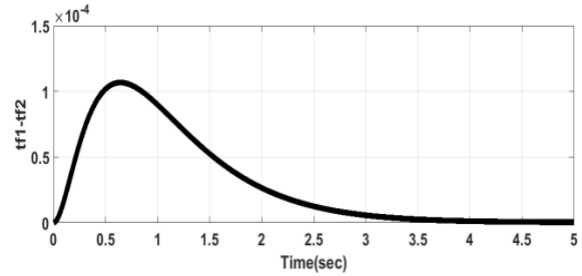


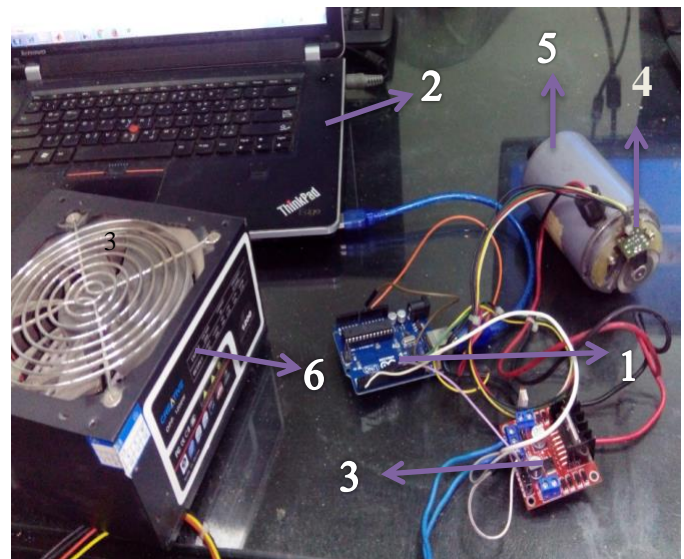
Fig.8 . Speed error signal between exact and estimated method

IV. DC MOTOR EXPERIMENTAL SETUPS

This section demonstrates the experimental setup to estimate the test DC motor to get the transfer function and its conventional and proposed controller.

A. Components of the Experimental Setups

Figure 3 shows the assembly of the used DC motor and its driver. The setups mainly consist of a DC motor, its driver board, Arduino Uno and personal computer (PC). The setup components will be discussed as follows.



1- Arduino UNO board	2- Computer (PC)	3- Driver board
4- Encoder	5- Motor	6- Power supply

Fig.3. Experimental setup

1. Arduino Uno Kit

It is an Atmega328 microcontroller unit that has been used as a digital controller. There are 14 pins for digital inputs and outputs: 6 of them can be used as PWM output pins. It also has 6 analog inputs; its clock crystal is 16MHz, a USB connection that is used to connect it to the PC. Operating Voltage is 5 Volts and Board model is Uno R3 [16].

2. Incremental Optical Encoder

Incremental encoder gives a specific number of pulses per revolution (PPR). Incremental encoder comprises of three fundamental components: a dual light detector, an illumination source, and a slotted disc. The light source sends illumination on the slotted disc, and receives them on a dual light detector. The dual light sensor measures the illumination numbers from the slotted disc and produces two pulse signals of quadrature output. The signals increase and decrease are numbered as measurement of pulses per revolution On the DC motor [17].

### 3. Motor driver

The L298N H-bridge module (Figure 4) can be used as a driver to control the velocity by effective voltage change on the armature voltage. H-bridge module can be operated under an electric voltage not exceed 35V DC and current not exceed 2A. There is also on-board regulator to get volt =5V. Table 3 presents the used truth table for this controller. The motor is connected between output1 (OUT1) and output2 (OUT2).

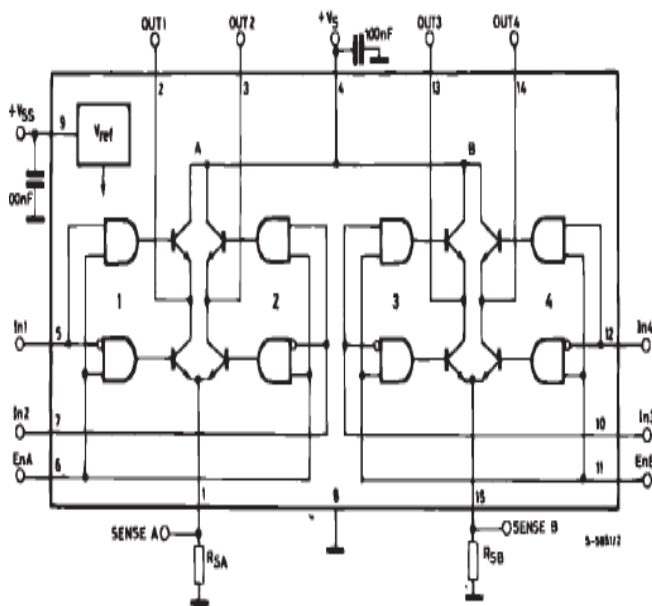


Fig.4 Basic H-bridge diagram [18].

Table3. Truth table H-bridge circuit [18].

Mode		Output
Input1(IN1)	Input2(IN2)	Motor A
0	0	Stop
0	1	Run (Forward)
1	0	Run (Backward)
1	1	Stop

### B. Estimation of the unknown parameters

The motor used in this research (Fig. 3) has some unknown parameters, so the aforementioned estimation method will be used to obtain its transfer function. The input voltage and the output speed for this motor are practically measured and plotted as appeared in Figs. 9 and 10.

By using the suggested method, the corresponding transfer function is obtained as:

$$Gp3(S) = \frac{33.34}{s^2 + 1.51s + 0.3021} \quad (4)$$

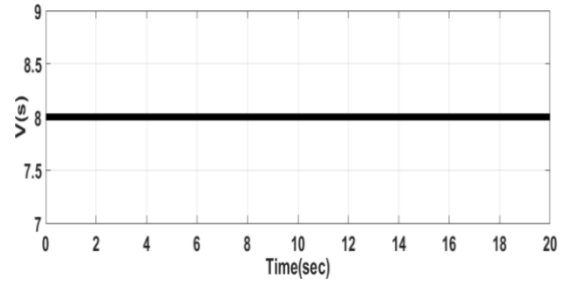


Fig.9 Measured voltage of the used DC Motor

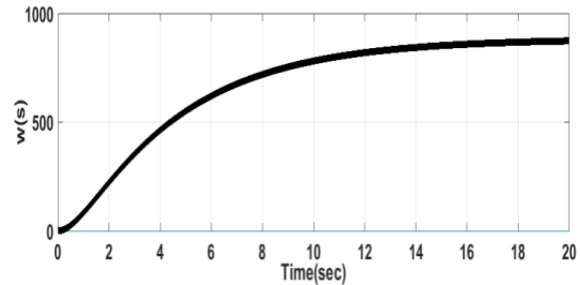


Fig.10. Measured angular velocity of the used DC Motor

## V. FRACTIONAL PID CONTROLLER AND ITS TUNING METHODS

The fractional-order PID (FOPID) controller is the extension of the general control loop feedback mechanism (PID controller) that is commonly applied in industrial systems of control. The FOPID Controller tries to correct the mistake between a measured value and a desired value by calculating and then producing a corrective action that may alter the process accordingly. The FOPID controller transfer function is defined in (5):

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (5)$$

The FOPID equation contains five obscure elements, where  $K_p$  is the proportional factor,  $K_i$  is the integral factor,  $K_d$  is the derivative factor,  $\lambda$  is the integral of fractional-order and  $\mu$  is the derivative of fractional-order and  $\lambda, \mu$  are positive real numbers. Putting  $\lambda=1$  and  $\mu=1$ , a conventional PID controller is gotten. Hence, FOPID controller popularization the conventional PID controller and extends it from point to plane as appeared in Fig.11. This extension gives much more adaptability in planning the PID controller and offers a chance to better alter the control system dynamics. This improves the power of the system and makes it steadier.

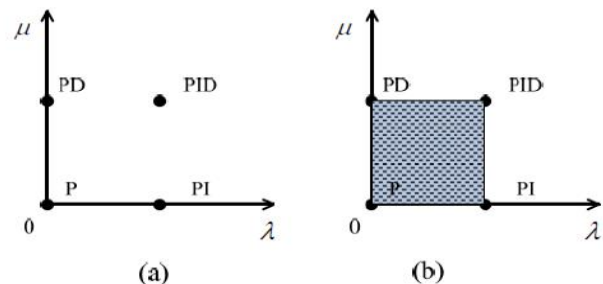


Fig.11 (a) Classical PID Controller (b) FOPID Controller

In spite of the better results of the fractional controller with modifiable elements ( $\lambda$ ) and ( $\mu$ ). The controller's elements are difficult to be modified as the nonlinear transcendental equations obtained by the control theory's design criteria are difficult to fix, leading in limitations on the fractional-order controller's practical applications. The primary problem with the fractional-order controller development is how to solve transcendental nonlinear equations.

In this paper, the tuning of the conventional PID parameters is optimized by using GA and single loop PID. FOPID parameters are optimized by using GWO and NM [19].

## VI. COMPUTER RESULTS AND SIMULATION

It has been provided clear presentations of three cases are prepared for discussion and the simulations performed by means of the MATLAB/Simulink software. for the first case is an analysis of controllers in DC motor velocity control that show the performance of the types of controller with the change of some parameters of DC motor. For the second and third cases are used to Optimize of controllers which appears the performance of the types of controller and which is better. In the following subsections, the significant findings of this research are displayed.

### A. Case No. 1:

The parameters of DC motor are changed for making sure that has it been effect or no on the performance with fixed controller components values for the use of different types of controller. The DC motor transfer function [20] is considered.

$$G_{p1}(S) = \frac{15}{1.6S^2 + 3S + 3.75} \quad (6)$$

It was noted that, the output of the system  $GP_1(S)$  using GWO optimizer (GWO/FOPID), Nelder-Mead (NM/FOPID), Single-loop PID tuning and Genetic Algorithm (GA) are as shown in Figures 12-16 and the response specification is given in tables 4-7. The most striking outcome of the information is that Single-loop PID algorithm-based speed control gives rise time and settling time response equal to zero that means speed of response and it is apparent from this Table 6 that FOPID algorithm is given overshoot equal to zero. In addition, when changing some values such as armature resistor and motor torque constant with the same values of the controller parameters almost no change in performance response. The response of this system using Nelder-Mead, Single-loop PID tuning, and Genetic Algorithm as shown in figure 5 and the response specification is given in Tables 4.

Table 4. Compare transient response elements for working point 1;  $R_a = 0.4; K = 0.015$

Controller	Rise Time (sec)	Settling Time (sec)	Maximum Overshoot (%)
NM/FOPID	0.0025	0.0042	0.012
PID/GA	0.136	0.185	0.019
Single -loop PID	0	0	0.024
GWO/FOPID	0.049	0.0814	0.315

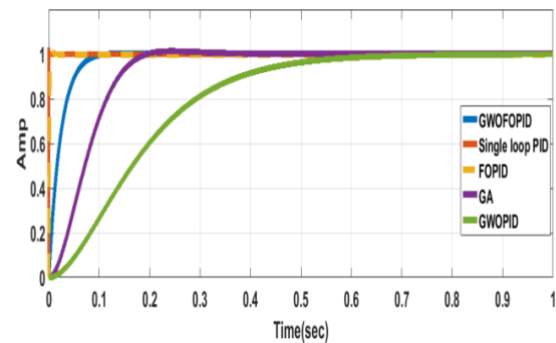


Fig. 12 DC motor velocity comparison for working point 1;  $R_a = 0.4; K = 0.015$

Table 5. Compare transient response elements for working point 2;  $R_a = 0.2; K = 0.012$

Controller	Rise Time (sec)	Settling Time (sec)	Maximum Overshoot (%)
FOPID	0.003	0.0055	0.01
Single -loop PID	0	0	0.024
PID/GA	0.15	0.185	0.03
GWO/FOPID	0.058	0.1172	0.51

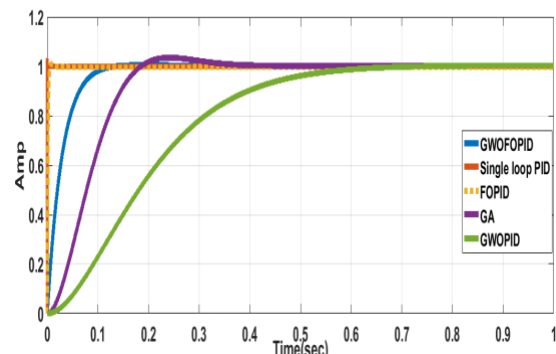


Fig. 13 DC motor velocity comparison for working point 2;  $R_a = 0.2; K = 0.012$

Table 6. Compare transient response elements for working point 3;  $R_a = 0.1; K = 0.014$

Controller	Rise Time (sec)	Settling Time (sec)	Maximum Overshoot (%)
FOPID	0.0025	0.0044	0
PID/Single-loop PID	0	0	0.024
PID/GA	0.137	0.185	0.024
GWO/FOPID	0.052	0.0851	0.57

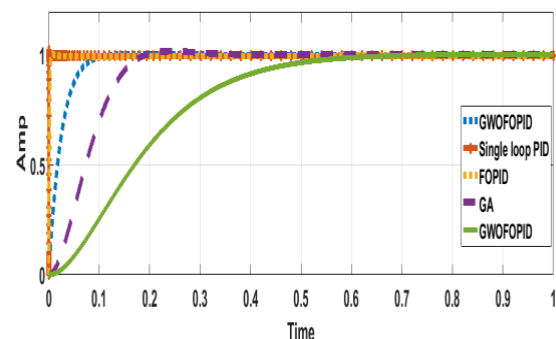


Fig. 14 DC motor velocity Comparison for working point 3;  $R_a = 0.1; K = 0.014$

Table 7. Compare transient response elements for working point 4;  $R_a = 0.3; K = 0.015$

Controller	Rise Time (sec)	Settling Time (sec)	Maximum Overshoot (%)
FOPID	0.0025	0.0044	0.012
PID/Single-loop PID	0	0	0.022
PID/GA	0.136	0.185	0.019
GWO/FOPID	0.055	0.1015	0.33

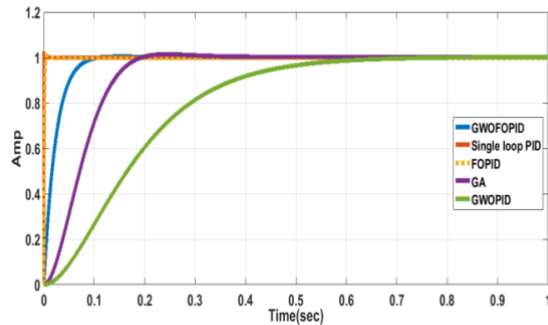


Fig. 15 DC motor velocity Comparison for working point 4; Ra = 0.3; K = 0.015

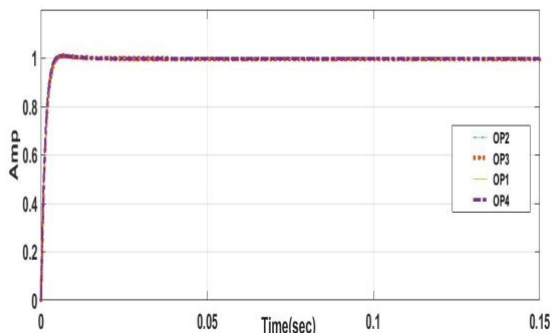


Fig. 16 Compare responses to NM/ FOPID in for all working points

### B. Case No. 2:

The DC motor transfer function [21] is considered.

$$G_{p2}(S) = \frac{0.6S^2 + 3s + 3.75}{0.4S^3 + 1.6S^2 + 3S + 3.75} \quad (7)$$

The response of the  $G_{p2}$  using Grey Wolf Optimizer as shown in Figure 17 and the response specification is given in Table 8. Together these results provide important insights into Single-loop PID algorithm-based speed control gives rise time and settling time response equal to zero that means speed of response and FOPID algorithm is given overshoot equal to zero.

Table 8. Comparison of PID using Single-loop PID tuning and FOPID controller using Nelder - Mead and GWO/FOPID

Controller	Rise time (sec)	Settling Time (sec)	Maximum Overshoot (%)
FOPID	0.0012	0.0024	0
Single-loop PID	0	0	0.105
PID/GA	0.0016	0.003	0
FGWO	0.00924	0.7313	0.003

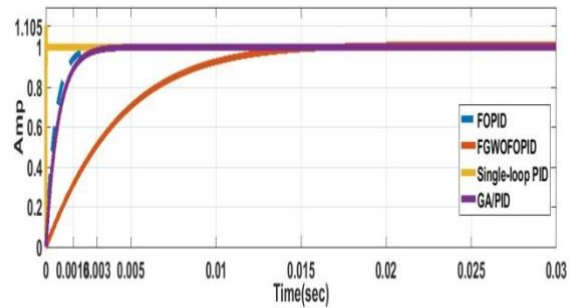


Fig.17 Comparison of responses for all FOPID, Single-loop, and FGWO

### C. Case No. 3:

From the data in Figure 11, it is apparent that the output of DC motor for  $G_{p3}(s)$  “obtained by system identification” without a PID controller is slow response and disturbance. To overcome these problems, controllers which are used PID and FOPID controllers. The controller of PID is tuned by the algorithms GA and Single-loop PID. These algorithms are used in MATLAB® 2016a.

- Fig. 12 display the output of the DC motor using GA and Fig. 13 Single-loop PID by using MATLAB® 2016a. Fig. 14 shows the DC motor response using FOPID. From these plots the variation can be observed between GA, Single-loop PID tuning, NM and the response sharpness in the NM is more than Single-loop PID. This increases the response rise time by using GA. The most surprising aspect of the data is in the overshoot is equal to zero. Additionally, the steady-state of this motor is given better performance by NM algorithm.
- The outcomes acquired from the above algorithms are tabulated in table 9. It is evident from Table 9 that apply of controller will increase the velocity control performance by increasing the response velocity and stability of steady-state. Together these results provide an important insight into FOPID algorithm in the steady-state response is more efficient than the PID.

Table 9. Comparison of PID using Single-loop PID tuning and FOPID controller using Nelder -Mead for lab motor

Controller	FOPID	GA/PID	PID
Rise time (sec)	0.02	0.04	0.03
Settling Time (sec)	0.022	0.052	0.041
Maximum Overshoot (%)	0	0	0
ISE	1.23	0.0059	
ITSE	2.99	1.25e-5	
IAE	3.52	0.01049	

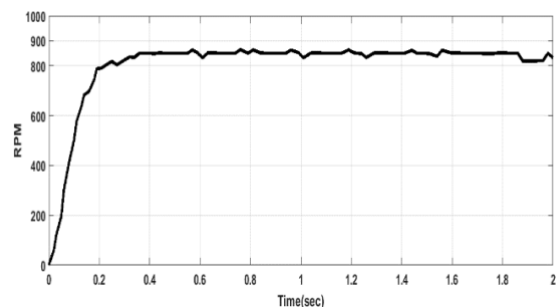


Fig.18. Speed versus Time plot of DC Motor.

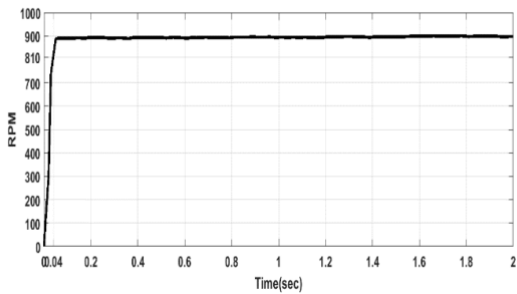


Fig.19. Speed versus Time plot with Reference speed at 900 rpm for GA/PID of lab motor with Arduino

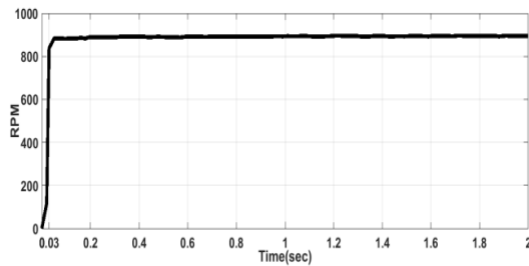


Fig.20. Speed versus Time plot with Reference speed at 900 rpm for Single-loop PID of lab motor with Arduino

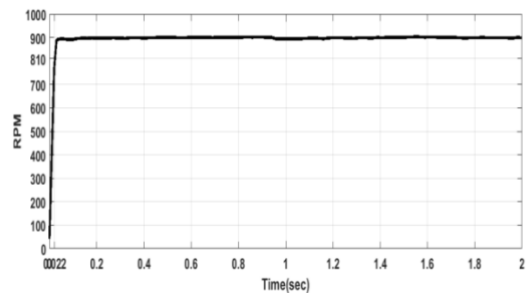


Fig.21. Speed versus Time plot with Reference speed at 900 rpm for FOPID of lab motor with Arduino

## VII. CONCLUSION

This paper presents a parameter estimation method of DC motor. The paper covers the pros and cons of PID and recent FOPID controllers. To judge the performance of FOPID against PID, more than two case studies were presented and discussed along with different tuning methods ranging from Nelder Mead, GA, and Grey wolf. Based on performance metrics used, it is concluded that the best response is provided by the NM algorithm based FOPID DC motor speed control. Where the corresponding rising time, settling time and maximum overshoot are 0.02, 0.022, zero seconds respectively. The obtained results ensured the high quality of the proposed methods.

## REFERENCES

[1] Jain, Ruchi, et al. "Tuning of fractional order PID controller using particle swarm optimization technique for dc motor speed control." 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES). IEEE, 2016.

[2] Kefal, Adnan, et al. "Modeling and parameter identification of electromechanical systems for energy harvesting and sensing." Mechanical Systems and Signal Processing 121 (2019): 890-912.

[3] Lemus, Juan, et al. "PID Controller Design for DC Motor." (2018).

[4] Nayak, Byamaksh, et al. "Parameter Estimation of DC Motor using Adaptive Transfer function based on Nelder-Mead Optimisation." Indonesian Journal of Electrical Engineering and Computer Science 9.3 (2018): 696-702.

[5] Li, Shuhui, et al. "Neural-Network Vector Controller for Permanent-Magnet Synchronous Motor Drives: Simulated and Hardware-Validated Results." IEEE transactions on cybernetics (2019)

[6] Bindu, Mini K, et al. "Tuning of PID controller for DC servo motor using genetic algorithm." International Journal of Emerging Technology and Advanced Engineering 2.3 (2012): 310-314.

[7] Obeng, Benedicta, et al. "PID Speed Control of a DC Motor Using Particle Swarm Optimization." Information Technology-New Generations. Springer, Cham, 2018. 781-789.

[8] Ahmed, Alisha, et al. "GWO/PID Approach for Optimal Control of DC Motor." 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN). IEEE, 2018.

[9] Yazgan, Harun, et al. "Comparison Performances of PSO and GA to Tuning PID Controller for the DC Motor." Sakarya University Journal of Science 23.2 (2019): 162-174.

[10] Kumari, Swati, et al. "GA Based Design of Current Conveyor PLD Controller for the Speed Control of BLDC Motor." 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT). IEEE, 2018.

[11] PID Controller Tuning in Simulink, 1994-2018. [Online]. Available: <https://www.mathworks.com/help/slcontrol/gs/automated-tuning-of-simulink-pid-controller-block.html>.

[12] Das, Kaushik, et al. "Optimal tuning of PID controller using GWO algorithm for speed Control in dc motor." 2015 International Conference on Soft Computing Techniques and Implementations (ICSTI). IEEE, 2015.

[13] Shah, Pritesh, et al. "Review of fractional PID controller." Mechatronics 38 (2016): 29-41.

[14] I. Podlubny, "Fractional-Order Systems and PI $D_{\mu}$  Controllers," IEEE Transactions on Automatic Control, Vol. 44, No. 1, pp.208-214, 1999.

[15] Ahmad, Md Akram, et al. "Speed control of a DC motor using Controllers." Automation, Control and Intelligent Systems 2.6-1 (2014): 1-9.

[16] Chaouch, Saad, et al. "DC-Motor Control Using Arduino-Uno Board for Wire-Feed System." 2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM). IEEE, 2018.

[17] Arslan, Ahmad. "Comparative analysis of speed decoding algorithms for rotary incremental encoders." (2017).

[18] Yin, Liuliu, et al. "Application of drive circuit based on L298N in direct current motor speed control system." Advanced Laser Manufacturing Technology. Vol. 10153. International Society for Optics and Photonics, 2016.

[19] Tepljakov, Aleksei, et al. "Fractional-order controller design and digital implementation using FOMCON toolbox for MATLAB." 2013 IEEE conference on computer-aided control system design (CACSD). IEEE, 2013.

[20] Agarwal, Jeetendra, et al. "Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor." Microsystem Technologies 24.12 (2018): 4997-5006.

[21] Verma, Santosh, et al. "Optimization of fractional order PID controller using grey wolf optimizer." Journal of Control, Automation and Electrical Systems 28.3 (2017): 314-322.



**Mohamed Said Essa** was born in Cairo, Egypt; on 1985. He received a B.Sc. degree from Engineering, Al-Azhar University. He is currently senior of the instrumentation Control Department in Cairo Electricity Production Company. His Email is Eng\_mohamedsaed.2007@yahoo.com.



**Magdi M. El-Saadawi** was born in Mansoura, Egypt in 1959. He received his B.Sc. and M.Sc. from Mansoura University, Egypt in 1982 and 1988, respectively, and his Ph.D. from Warsaw University of Technology in 1997. He was a teaching assistant at El-Mansoura University from 1983–1992. From 1997, he was a staff member of the Electrical Engineering Department, Mansoura University, and has been a professor since May 2011. His fields of interest include, renewable energy, power system analysis, and AI applications in power systems



**Eid Abdelbaki GOUDA** was born in Domiatta, Egypt, on 1975. He received B.Sc. And M.Sc. Degree from electrical engineering Mansoura University in 1997 and 2004 respectively. He is completed his Ph.D at University Henri Poincare of Nancy, the Groupe de Recherche en Electrotechnique et Electronique de Nancy, France in 2011. Now, he is associate professor at Electrical Department in faculty of Engineering, Mansoura University. He is interested in Magnetic Gear, Electric Machine Design/Control and Protection Systems. His E mails are: eid.gouda@yahoo.fr and [caidgoda@mans.edu.eg](mailto:caidgoda@mans.edu.eg).



**MOSTAFA A. ELHOSSEINI** received a B.Sc. degree from the Electronics Engineering Department, Mansoura University, Egypt, the M.Sc., and Ph.D. from Computers and Systems Engineering Department from Mansoura University. He is currently an Associate Professor with the Computer Science Department, College of Computers Science and Engineering, Taibah University, Yanbu, Saudi Arabia. His primary research interests include artificial intelligence (AI), such as genetic algorithms, neural networks, particle swarm optimization, simulated annealing, and fuzzy logic. He is also interested in the application of machine learning in image processing, access control, and optimization and the applications of computational intelligence CI and soft computing tools in bioinformatics. Dr. Elhosseini served as a member of the international program committees of numerous international conferences.