
System Identification and Optimal PI Controller of R/C Servo Motors

*Omar Ahmed¹, A. Okasha Elnady² and Saber Abdrabbo³

¹ Mechatronics Department, Faculty of engineering, October 6 University, Egypt

² Mechatronics Department, Faculty of engineering, October 6 University, Egypt

³ Industrial Department, Shoubra faculty of engineering, Benha University, Egypt

Received: 14-04-2016 Revised: 22-04-2016 / Accepted: 20-5-2016

Abstract

An R/C servo motor is electro-mechanical device that takes the PWM control signal to move to a certain position. This type of system is widely used in undergraduate mechatronics projects such as (small-sized robotics, radio-controlled cars, planes, etc.). However, the embedded close loop controller circuit inside the R/C servomotor represents a drawbacks in many applications need high definition controller. Therefore, in order to improve the performance of R/C servo motor, transfer function model is estimated from input-output measured data using system identification Matlab toolbox. After that, hardware in loop simulation employed to validate the estimated model with physical system. The results show good tracking to both estimated and physical system to the reference signal. PI controller designed firstly using Ziegler-Nichols method in order to limit the search space. After that, optimal controller gains obtained using two intelligent optimization algorithms, Genetic Algorithms (GA), and Particle Swarm Optimization (PSO), then implemented on the physical system in real time environment using low cost components. The controller improves the set point characteristic and achieved satisfactory dynamic performance.

Keywords: System identification, Parameter estimation, Servomechanisms, Servomotors, Robots, Mechatronics, PI controller, DC motors, optimal control, Genetic Algorithms (GA), Particle Swarm Optimization (PSO).

Nomenclature

b = equivalent viscous-friction coefficient of the motor and load referred to the motor shaft, N-m/rad/sec

B = the magnetic field strength

E = applied armature voltage, volt

e = back emf, volt

i_f = field current, ampere

i_a = armature current, ampere

J = equivalent moment of inertia of the motor and load referred to the motor shaft, kg-m²

L_a = armature inductance, henry

l = the length of the conductor, meter

R_a = armature resistance, ohm

T = torque developed by the motor, N-m

θ_m = angular displacement of the motor shaft, radian

1. Introduction

R/C servo motor is DC geared-motor with feedback potentiometer, and motor driver circuit in compact package. R/C servo motors widely used in undergraduate projects, due to compact design, inexpensive, driven by low voltages, and represent an

easy solution for closed loop control. To date, there exist a few research papers, identify and control of R/C servo motor. In that research parameters estimated using SRIVC an identification method for continuous-time-invariant [1].

The main problem to achieve satisfactory dynamic performance using the R/C servo motors, basically due to the existence of embedded position servo controllers inside them. One may expect that the embedded servo controller makes it extremely easy to control, since we just need to send PWM by certain duty cycle. It is true, if the robot is moving freely and is full-actuated. However, if the robot is in contact with the floor or other hard objects, it may not behave as we expected. If worse, it would exhibit a chattering motion caused by the disturbance force from the environment [1]. Moreover, when there is a small change from the target position the feedback mechanism try to correct any drift, but by constant adjustment. That leads motor to create annoying switches while trying to hold a steady position.

PID controllers are the most frequently used in industry due to good understanding, simple structure, easy to implement, and have a sufficient performance

for most industry applications. The performance of PID controller mainly depends on the PID gains. They can be carried out by trial, which takes long time to obtain acceptable performance, or using classical methods such as Ziegler-Nichols, and Chien-Hrones-Reswick, which in general doesn't obtain the optimal gains. In order to obtain the optimal gains two intelligent computation methods are used. Genetic Algorithms (GA), and Particle Swarm Optimization (PSO).

Genetic Algorithms are an optimization technique inspired by the mechanisms of natural selection. They were first proposed and investigated by John Holland at university of Michigan 1970. Genetic algorithms consist of three main operations selection, crossover, and mutation. These operators work with a number of artificial creatures called a generation. By exchanging information from each individual in a population, GA preserves a better individual and yields higher fitness generation such that the performance can be improved [2].

Particle Swarm Optimization is an optimization technique based on the movement and intelligence of swarms. It uses a number of agents, i.e., particles that constitute a swarm moving around in the search space looking for the best solution. Each particle is treated as a point in a N dimensional space which adjusts its "flying" according to its own flying experience as well as the flying experience of other particles [3].

In this work the transfer function model of the whole system, including internal model of R/C servo motors, and the embedded servo controller estimated from measured data. In addition, optimal PI controller designed, the controller gains tuned using two intelligent optimization algorithms, Genetic Algorithms (GA), and Particle Swarm Optimization (PSO). The experimental work utilize low cost components (i.e. Arduino Uno, R/C servo motor, potentiometer), such components are already used in mechatronics projects. The proposed modifications are simple and realistic enough, to use in undergraduate projects.

The rest of this paper is organized as follows; Section 2, gives a brief overview of different objective functions, GA and PSO search algorithms. Section 3, provides the fundamentals of electro-mechanical transfer function model of DC motor. Section 4, describes the main experiment components as well

as, how data transfer in real time between computer, and target hardware. Section 5, presents identification scheme and results. Finally, PI controller designed firstly using classical Ziegler-Nichols method, then PSO, and GA are investigated to find the optimal controller gains are presented in section 6.

2. Review of Optimizations Algorithms

2.1 Genetic Algorithms

Genetic algorithm (GA) is a mature optimization algorithm, and it can solve both constrained and unconstrained optimization problems. GA is an optimization technique inspired by the mechanisms of natural selection. It uses probabilistic transition rules instead of deterministic rules and handles a population of potential solutions known as individuals or chromosomes that evolve iteratively. Each iteration of the algorithm is termed a generation. The evolution of solutions is simulated through a fitness function and genetic operators such as reproduction, crossover, and mutation [4].

2.2 Particle swarm Optimization

Particle swarm optimization (PSO) is a global optimization algorithm inspired by the behavior of birds. It is a simple and efficient algorithm compared with other algorithms. A collection of individuals called particles move in steps throughout a region. At each step, the algorithm evaluates the objective function at each particle. After this evaluation, the algorithm decides on the new velocity of each particle. The particles move, then the algorithm reevaluates.

The velocity of each particle is adjusted according to its own flying experience and the flying experience of other particles.

The i^{th} particle is represented as

$$x_i = x_{i,1}, x_{i,2}, \dots \dots x_{i,d}$$

The velocity for particle i^{th} is represented as

$$v_i = v_{i,1}, v_{i,2}, \dots \dots v_{i,d}$$

The modified velocity and position of each particle can be calculated using the current velocity and the distance from $x_{i,d}^b$ to global best as shown in the following equations [5].

$$v_i^{k+1} = h x v_i^k + c_1 * rand * (x_i^b + x_i^k) + c_2 * rand * (x_i^g + x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

Where

- v_i^k is the particle velocity,
- x_i^k is the current particle position,
- h is the inertia weight factor,
- x_i^b is the best previous position of the i^{th} particle,
- x_i^g is the best particle among all the particles in the population,
- $rand$ is a random function between 0 and 1,
- c_1 and c_2 are acceleration constants,
- k is the pointer of iterations $i = 1, 2, 3, \dots, n$ where n is the number of particles.

2.3 Optimality criteria

To choose a suitable performance index for optimization. Several objective functions can be utilized for the time-domain optimization, namely

Integral square error $ISE = \int_0^t e^2(t) dt$,

Integral absolute error $IAE = \int_0^t |e|(t) dt$

Integral time-square of error $ITSE = \int_0^t t e^2(t) dt$, and

Integral time-absolute of error $ITAE = \int_0^t t |e|(t) dt$.

Where $e(t) = 1 - y(t)$, $y(t)$ is the tuned PI control system closed-loop step response.

3. Model of DC Motor

A motor is an electromechanical component that yields a rotation output when an input voltage is applied. DC schematic shown in figure 1.

A magnetic field is developed by stationary permanent magnets or a stationary electromagnet called the fixed field. A rotating circuit called the armature, through which current $i_a(t)$ flows, passes through this magnetic field at right angles and produce a force, $F = Bli_a$. The resulting torque turns the rotor [6].

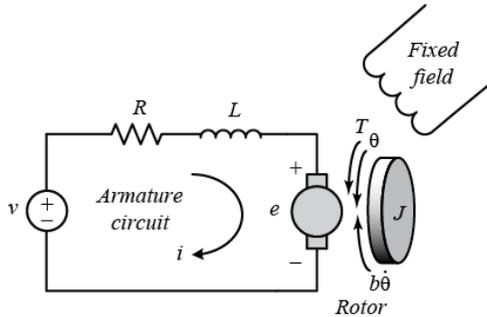


Fig. 1. DC motor schematic

The equilibrium of the system is given by

$$v_b(t) = K_b \frac{d\theta_m(t)}{dt} \quad (1)$$

where; v_b the back electromotive force (back emf); K_b is a constant of proportionality called the back emf constant.

Taking the Laplace transform,

$$v_b(s) = K_b s \theta_m(s) \quad (2)$$

From Kirchhoff voltage law we can get relation between the armature current, i_a , the applied armature voltage, e_a , and the back emf v_b .

$$R_a i_a(t) + L_a \dot{i}_a(t) + v_b(t) = e(t) \quad (3)$$

After applying Laplace transform to Eq(3) will get,

$$R_a I_a(s) + L_a s I_a(s) + V_b(s) = E_a(s) \quad (4)$$

The torque developed by the motor is proportional to the armature current

$$T_m(s) = K_t I_a(s) \quad (5)$$

$$I_a(s) = \frac{1}{K_t} T_m(s) \quad (6)$$

The transfer function of the motor is given by

$$\frac{(R_a + L_a s) T_m(s)}{K_t} + K_b s \theta_m(s) = E_a(s) \quad (7)$$

Now, the mechanical torque in terms of $\theta_m(s)$

$$T_m(s) = (J_m s^2 + D_m s) \theta_m(s) \quad (8)$$

Where J_m , and D_m are motor inertia and viscous damping.

Substituting Eq (7) into (8), the resulting is

$$\frac{(R_a + L_a s)(J_m s^2 + D_m s) \theta_m(s)}{K_t} + K_b s \theta_m(s) = E_a(s) \quad (9)$$

The armature inductance, L_a , is small compared to the armature resistance, R_a , so; Eq (2) becomes

$$\left[\frac{R_a}{K_t} (J_m s + D_m + K_b) \right] s \theta_m(s) = E_a(s) \quad (9)$$

After simplification, the transfer function is

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t / (R_a J_m)}{s \left[s + \frac{1}{J_m} \left(D_m + \frac{K_t K_b}{R_a} \right) \right]} \quad (10)$$

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K}{s(s+\alpha)} \quad (11)$$

4. Experiment Design

R/C servo motors have three wires, first two wires are power and ground, and the third is a digital control line which accept a pulse-width modulation (PWM). The angle and speed of R/C servo motor proportional to pulse width and frequency of pulses respectively. Potentiometer outputs is analog voltage which is proportional to the angle of the DC motor, as shows in fig. 2.

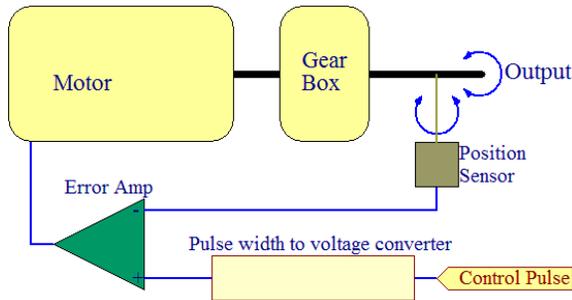
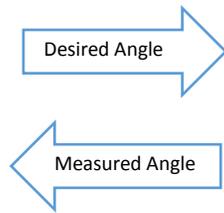
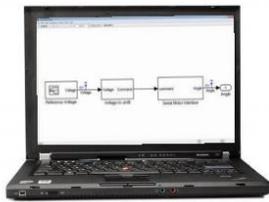


Fig. 2. Internal feedback circuit

As shown in Fig. 3. R/C servo motor is attached to PWM pin in Arduino Uno board, and vcc and motor ground are connected to 5v and ground board, respectively.

The feedback signal comes from the embedded potentiometer inside the R/C servo motor and connected to analog Arduino Uno pin. Arduino Uno board is the interface between computer and R/C servo motor, connected to computer though USB port. Signals transmit and receive between computer and R/C servomotor serially. The data acquisition card, Arduino board, contains an interpreter program receives commands serially form Simulink and send appropriate voltage to R/C servo motor, and also receive the feedback signal from potentiometer and send it to Simulink.

Host Computer



Arduino Uno



R/C servo

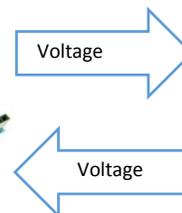


Fig. 3. Experiment hardware

5. Model identification

System identification is a methodology for building mathematical models of dynamic systems using measurements of the system's input and output signals [7].

The process of system identification includes the following steps:

- I. Measure the input and output signals from system.
- II. Model selection and evaluation.
- III. Model evaluation and validation.

Each step of system identification is discussed as the following:

5.1 Measure the input and output signals from system.

Not any signal can be considered as an excitation signal for identification. Excitation signal should be sufficiently variable to give the model enough information about the dynamics of the system to produce an accurate model. Thus, square pulse signal is used as exciting input signal. On computer, Simulink model sends reference angles, various voltage profiles to excite the system and receive measured feedback angle from the potentiometer. Reference and measured angles stored as arrays on MATLAB workspace.

Figure 4 shows the feedback measured signal which seems very noisy, because potentiometer direct connected with motor shaft without any reduction gain, so any small change in motor shaft reflects on feedback signal.

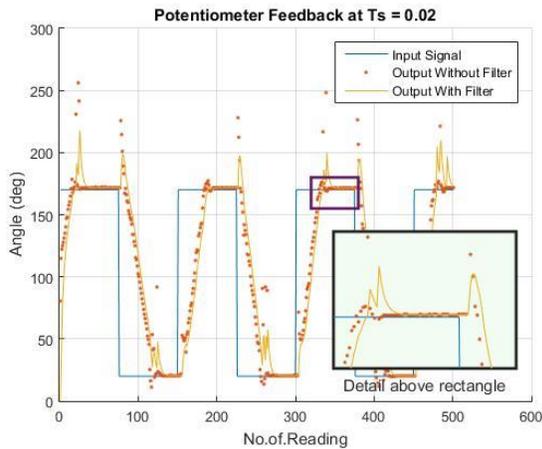


Fig. 4. Input and output signals at ts = 0.02

Figure 5 shows the measured signal which seems very noisy particularly at fluctuating points. When motor shaft rotated from angle 20° degree to reach the reference angle 170° degree the measured value recorded more than 250°. Moreover, the measured angle recorded high value then decreased to reach the low reference angle. In both cases the actual shaft angles didn't jump suddenly to these angles, simply because the motor shaft limited to rotate from 0 to 180 degree as shown in Fig. 5.

An examiner look to the measured angles in Fig. 5, only a few instantaneous-angle values far from the actual motor shaft-angle values. The major recorded angle values are consistent and represent a second order response. Therefore, these far recorded angle values should be neglected to obtain a signal represent the actual motor shaft angle values.

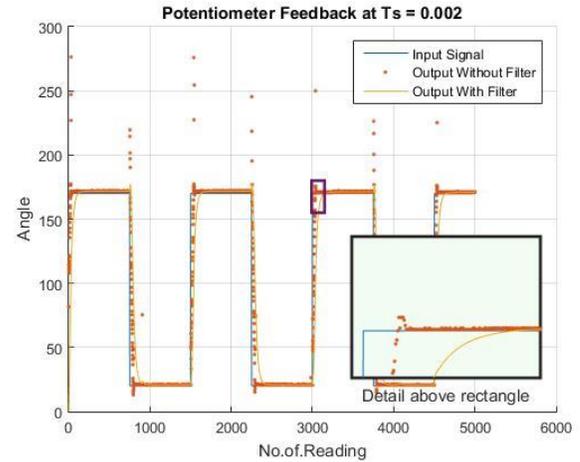


Fig. 5. Input and output signals at Ts = 0.002

In order to reduce the effect of the rest of noise, sampling rate was increased, and a first order filter was added to smooth the signal. The sampling rate was $TS = 0.002$ and filter constant = 0.065. The drawback of filter is added a delay to the signal, which reflected on the transient response of the system. Therefore, the filter is removed but sampling rate is still the same.

5.2 Model selection and evaluation

There are many possibilities to choose the suitable model. The quality of the estimate model depends on which model you choose. The more accurate model, the higher cost in terms of effort and computational time. The goal is to find the simplest model that will adequately capture the dynamic system performance [8]. The mathematical model of DC motor is discussed in details above in section 2. It contains two poles but no zeroes.

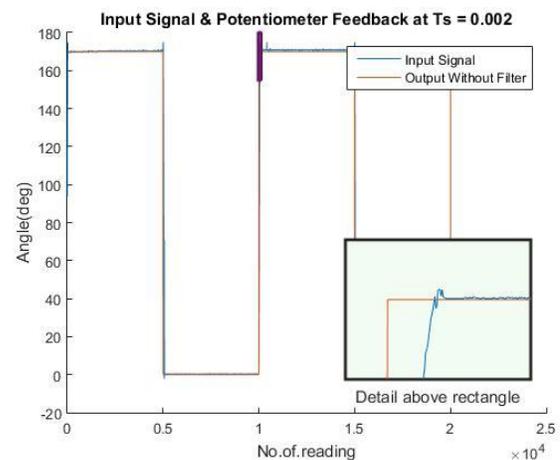


Fig.6. Input and measured signals with remove the far angle values

5.3 Model evaluation and validation.

The Transfer function of the system obtained using MATLAB system identification toolbox. The model estimated is a linear dynamic model for the plant as a continuous-time transfer function.

$$Y(s) = \frac{num(s)}{den(s)} U(s) + E(s)$$

$Y(s)$, $U(s)$ and $E(s)$ represent the Laplace transforms of the output, input and error, respectively.

In order to obtain the suitable model a lot of experiments were performed. The best fit model is shown in figure 7, the Transfer function is

$$\frac{\theta_m(s)}{E(s)} = \frac{20880}{s^2 + 219.6s + 20950}$$

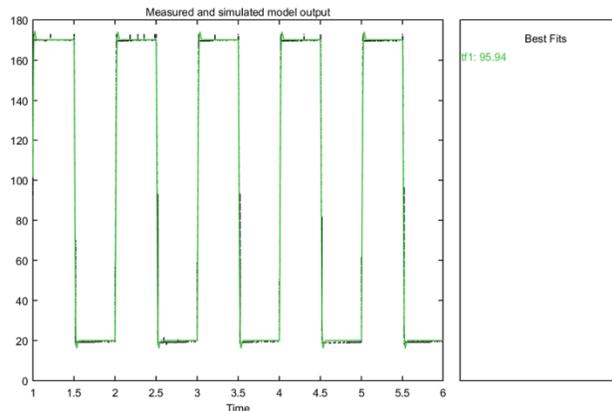


Fig. 7. Measured simulated model output

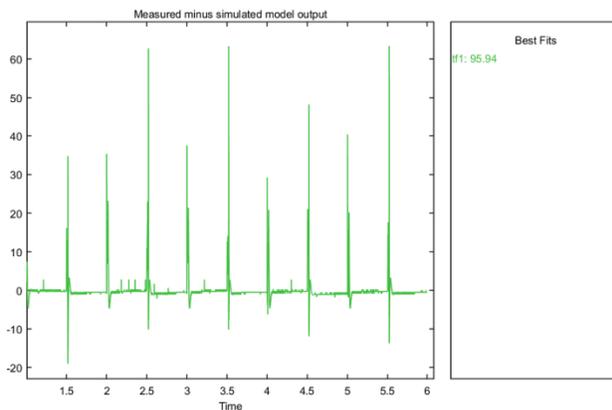


Fig.8. Measured minus simulated model output

6. PI controller

Proportional-Integral-Derivative (PID) controller is the most common controller use in industry and has

been universally accepted in industrial control. The popularity of PID controller Comes from it has simple structure, easy to design and implement, and also it can adjusted on site in case there no mathematical model [9]. In fact, most loops are PI because derivative action is not used very often [10].

The challenge in PID controller design is to tune the values of the proportional gain K_p , integral gain K_i , and derivative gain K_d . Tuning work is usually performed manually by trying out different tuning parameter combinations on-line until satisfactory or, at least, acceptable results are achieved. This method is laborious, time-consuming, unsafe, and does not always give the best possible solution [11].

In order to obtain the optimal gains two intelligent optimization algorithms are utilized, POS, and GA.

Firstly PI controller is designed to get the constrained values for optimization methods using Zigler-Nichols. The upper and lower search limits values are listed in Table 1.

Table 1. PI controller upper and lower search limits

	K_p	K_i
Upper Limits	0.01	1
Lower Limits	2	60

Genetic algorithm and particle swarm optimization are an iterative random search algorithm, so that many runs are performed to obtain the optimal PI controller gains. Genetic algorithm parameters are listed in Table 2.

After many runs to get the suitable gains using upper and lower constrain gains which obtained using classical Zigler-Nichols method. The GA parameters are the

Table 2. Parameters used in genetic algorithm.

Parameter	Type/value
Generations	50
Population size	30
Selection	Uniform
Crossover	Constrained dependent
Mutation	Constrained dependent
Encoding	Binary
Fit. Fun.	Integral time-absolute of error

The optimal gains are $K_p = 0.1984$, and $K_i = 52.5318$ at upper and lower value concertinaed up_vlaue = [0.01 30], and Low_value = [2 60] K_p and K_i respectively. The fitness function is Integral time-absolute of error (ITAE). The system response tuned using GA compared with POS shown in figure

[9 & 10]. Show a better performance in terms of the percent overshoot and the settling time.

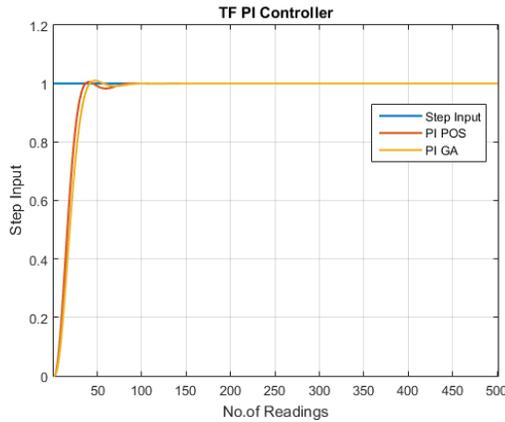


Fig. 9 PI Controller tuned using GA & POS

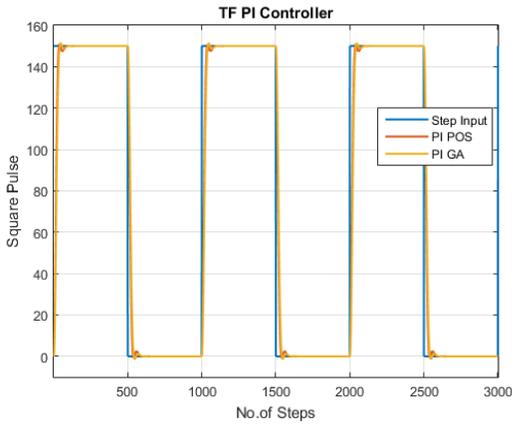


Fig.10. PI Controller tuned using GA & POS

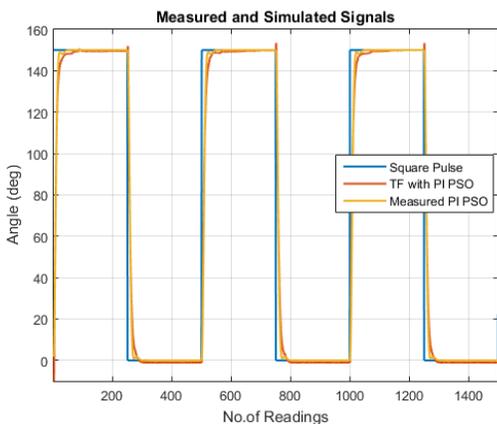


Fig.11. Measured and simulated output with PI controller

Table 3. Parameters used in PSO.

Search Criteria

No.of.Particals	10
No.of. Iterations	30
Fitness Fun.	Integral time-absolute of error

PSO search parameters listed in Table 3. The optimal gains are $K_p = 0.3184$, and $K_i = 60$, At the same constrained vales, and fitness function used in GA. The results show a slightly different performance between two algorithms, in term of percent overshoot GA is better than PSO, in contrast PSO is faster rise time than GA.

7. Conclusion

System identification is an iterative technique to obtain the mathematical relation form measured data. Hence, many attempts were conducted to obtain the suitable model. In this system the main issues were how to capture the dynamic response that represent the system, as well as how to choose a suitable excitation signal, signal conditioning and sampling rate. Other steps such as model structure is known from the physical relations, and linear identification method is suitable for this system, and also easy to control.

PI controller is designed firstly using Zigler-Nichols to determine the limits search space, so that the controller can acquire the optimal gain quickly. Then PSO and GA are employed to obtain the optimal PI controller gains. The results show that the system repose using PSO is almost the same as the system response using GA. However, the PSO is simpler algorithm and doesn't need complex computations compared with GA.

In this study, the main components and theory of operation of R/C servo motor has been discussed, in order to understand how it work. After that, the mathematical relation between input voltage and shaft motor angle as output was estimated with minimum error under real time environment, the DC motor has been modeled before to know the structure of the transfer function. PI controller designed and optimized using PSO, and GA. Then implemented on the target system, the results show good dynamic response and good tracking to the target hardware.

8. References

- [1] T. Wada, M. Ishikawa, R. Kitayoshi, I. Maruta and T. Sugie. "Practical modeling and system identification of R/C servo motors," in Proc. Control Applications, (CCA) & Intelligent Control, (ISIC), 2009 IEEE, 8-10 July 2009, pp.1378-1383

- [2] Gupta, S.K.; Varshney, P. "Fractional-order PID Controller Design for Speed Control of DC Motor ", Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on.
- [3] Du, Ke-Lin, and M. N. S. Swamy. "Particle swarm optimization." Search and Optimization by Metaheuristics. Springer International Publishing, 2016. 153-173.
- [4] Jayachitra, A., and R. Vinodha. "Genetic algorithm based PID controller tuning approach for continuous stirred tank reactor." Advances in Artificial Intelligence 2014 (2014): 9.
- [5] Hazem I Ali, Samsul Bahari B Mohd Noor, S M Bashi, Mohd Hamiruce Marhaban, "Quantitative feedback theory control design using particle swarm optimization method" 2012 Month:06 Day:01 Volume:34 Issue:4 First page:463 Last page:4
- [6] Nise, Norman S. CONTROL SYSTEMS ENGINEERING, John Wiley & Sons, 2011.
- [7] Ljung, L. "System Identification Toolbox: Getting Started Guide, The Math Works." Inc., Natick (2012).
- [8] Motor Control with Arduino: A Case Study in Data-Driven Modeling and Control Design. Mathworks.
- [9] A. Saleem & S. Abdrabbo & T. Tutunji "On-line identification and control of pneumatic servo drives via a mixed-reality environment" Int J Adv Manuf Technol (2009) 40:518–530
- [10] Åström, Karl Johan, and Tore Hägglund. "The future of PID control." Control engineering practice 9.11 (2001): 1163-1175.
- [11] <http://www.ni.com/white-paper/3782/en/>