

---

## Online Identification and Artificial Intelligence Control of a ServoPneumatic System

Omar A. Mohamed<sup>1</sup>, Abdelrady Okasha<sup>1</sup>, Saber Abdrabbo<sup>2</sup>

<sup>1</sup>October 6 University, Cairo, Egypt

<sup>2</sup>Faculty of Engineering in Shobra, Cairo, Egypt

**Received: 10-08-2018**

**Revised: 18-09-2018**

**/ Accepted: 20-11-2018**

---

**Abstract**—The aim of this paper is to identify and control a pneumatic servo drive in real-time environment. Obtaining the system's dynamic model accurately can be difficult once the pneumatic servo-system has been assembled since its highly nonlinear in nature, as a result, some difficulties in servo-pneumatic system modeling and control. In order to, overcome the complexity associated with the system nonlinearity, auto-regressive moving-average (ARMA) model is employed to identify the system's dynamic model in real-time environment. The advantages of this approach include high accuracy in the estimated model, low cost, and time reduction in controller design. The results acquired from the online experimental measured data are used to predict a discrete transfer function of the pneumatic servo system. The fourth-order model with one-step prediction shows the best performance compared with different order estimated model with varying sizes of step. Due to the highly nonlinearity of the system under study, two sophisticated controllers, PID-type fuzzy logic controller and Fractional order PID controller were chosen and designed using three optimization algorithms, namely particle swarm optimization (PSO), genetic algorithm (GA), grey wolf optimization (GWO).

**Keywords**—Servo-pneumatic, System Identification, Fuzzy Control, Fractional Order PID, Optimization algorithms.

---

### Introduction

Pneumatic servo drives are used in many applications in the industry due to several advantages, free and available source, environmentally friendly, has a simple structure so that easy to maintain, easy to transmit the power through pipes. pneumatic systems are highly nonlinear, and time-variant system due to the air compressibility, internal and external disturbances.

The First theoretical and experimental works on closed loop servo-pneumatic system were by Shearer in the 1950's (1954 & 1957) [1]. He did a complete theoretical analysis of a double-rod pneumatic cylinder involving the air compressibility in the two actuator chambers and the characteristics of the airflow through a proportional control valve. He also made several simulations runs on an analog computer,

and the dynamic model he obtained was a linear third-order transfer function.

Liu and Bobrow (1988) [2] derive a linear state-space model of the pneumatic servo-actuators in robotic based on an arbitrary operating point and experimentally determined the unknown flow constant of the linear model for doing LQ and PD based control. They pointed out that the linear model captures the dominant dynamics of the pneumatic servo system with an adequate precision so that pneumatic systems are practical for use in servo control applications.

In (Pu et al., 1992) [3], they studied digital motion control and profile planning for pneumatic servo systems. They claimed that it is quite difficult to predict actual pneumatic motion characteristics when several types of motion are required since the transient response and stability of pneumatic servo systems are

inherently complex, highly nonlinear, and difficult to model; being velocity/acceleration dependent, position-dependent as well as direction dependent. This complexity is mainly caused by the air compressibility, the pressure drops along transmission tubes, possibly time-varying and leakage. In their research, a simplified linear dynamic model was used for analysis, and controller design.

A paper by (E Richer, Y Hurmuzlu, 2000) [4] presented a detailed mathematical model of dual acting pneumatic cylinder controlled with the proportional control valve. They studied the influence of nonlinear air flow through the proportional valve, leakage between chambers, the compressibility of air in the cylinder chambers, end of stroke inactive volume, time delay, and attenuation in pneumatic tube lines considered carefully. Then numerical simulation, parameter estimation, and model validation were conducted for two types of pneumatic cylinders, and different connecting pipes length shows that perfect agreement.

linear control strategies are modified in order to show a good performance. In fact, many modified linear controllers have been introduced for servo-pneumatic systems. Linearized Feedback and disturbance rejection with PID controller studied in the works [5, 6]. They demonstrate that a pneumatic actuator with a payload can be linearized. In [7-11], they proposed a control law composed of a linearized input/output via static nonlinear state feedback. A recently, (Sobczyk et al., 2012) [12] proposed linearized feedback with friction compensation conducted on a pneumatic positioning system.

The most common controller in servo pneumatic positioning system is the state feedback controller, called PVA controller and used three feedback signals; position, velocity, and acceleration [13-17]. However, the acceleration feedback signal is difficult to measure and involved much noise when obtains from the derivative of velocity or the second derivative of position. For this reason, [18, 19] modified the state feedback controller by measure the pressure difference in the cylinder chambers and feed it to the controller instead of the acceleration signal. The PVA and modified state controllers can enhance both dynamic and static performance so that pneumatic systems are the excellent practical choice for use in servo position control applications. On the other hand, they are not robust enough to payload changes as well as to friction force effects.

Adaptive control approaches usually apply to improve the control performance of the dynamic systems in the case where the dynamic model of the system changes during the operation. As well as, they reduce the effects of system nonlinearities. Adaptive controllers generally apply three principles; parameter estimation, decision, and modification. Several adaptive control

techniques have been proposed [20, 21], self-tuning PID control [22], adaptive PI control [23], self-tuning Proportional controller [24], fuzzy PID gain scheduling [25]. One problem with this control approach is that there is no general algorithm to design an adapter. Hence, most adapter functions are especially for some sort of application.

In this paper, ARMA models and RLS algorithms were utilized to identify the servo-pneumatic system online. Due to the highly nonlinearity of the system under study, two sophisticated controllers, PID-type fuzzy logic controller and Fractional order PID controller were chosen and designed using three different global optimization algorithms to fulfill the system needs. The PID-type logic fuzzy controller should provide means for accurate control of nonlinear systems and guarantee the robustness to modeling errors and uncertain parameters. fractional order PID controller is more flexible than the conventional PID controller.

The rest of the paper is organized as follows. Section 2 provides the ARMA models and the recursive estimation algorithm. The experimental setup is presented in Sect. 3. the employment of online identification is provided in Sect. 4. Sect 5 first gives a brief introduction to the controllers, fitness functions, and optimization algorithms, then obtained the optimal paraments of the controllers and validate them on real time simulation. Finally, the conclusion is drawn in section 6.

## I. ARMA MODELS AND RLS ALGORITHMS

The least square algorithm is used to identify autoregressive moving average ARMA parameters using input/output measured data as shown in figure 1. [26, 27]. For simplicity of derivation, we consider single input single output model structure.

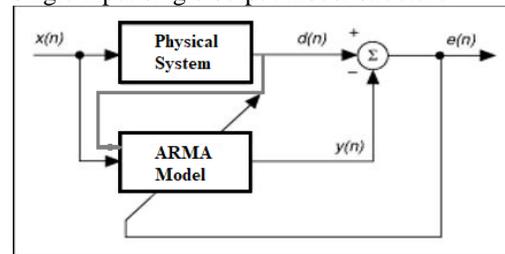


Figure 1. Block diagram of the online system identification.

$$\hat{y}(\tau) = \sum_{i=1}^n \hat{a}_i(k)y(\tau - i) + \sum_{j=1}^m \hat{b}_j(k)y(\tau - j) \quad (1)$$

Where  $\hat{y}(\tau)$  = the estimated output samples,  $y(\tau)$ , and  $u(\tau)$  are input and output samples,  $\hat{a}_i(k)$  and  $\hat{b}_i(k)$  are the unknown system parameters.

For all integers values of  $\tau \in [1, k]$  we would like to find the set of parameters which satisfy the system of equations

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(k) \end{bmatrix} = \begin{bmatrix} y(0) & \cdots & y(1-n) & u(0) & \cdots & u(1-m) \\ y(1) & \cdots & y(2-n) & u(1) & \cdots & u(2-m) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y(k-1) & \cdots & y(k-n) & u(k-1) & \cdots & u(k-m) \end{bmatrix} \begin{bmatrix} \hat{a}_1(k) \\ \vdots \\ \hat{a}_n(k) \\ \hat{b}_1(k) \\ \vdots \\ \hat{b}_n(k) \end{bmatrix} \quad (2)$$

The compact form of Eq (2) is

$$Y_k = X_k \hat{\theta}(k) \quad (3)$$

Eq. (3) can't be realized in the real world due to measurement noise and uncertainty, so the more realized form of Eq (4) Is,

$$Y_k + E_k = X_k \hat{\theta}(k) \quad (4)$$

Now the goal is to determine the  $\hat{\theta}(k)$ , which represent the unknown parameters, by minimizing the error in least square

$$J(k) = \sum_{i=1}^k e^2(i) \quad (5)$$

$$J(k) = E_k^T E_k = \sum_{i=1}^n [y(i) - X^T(i-1)\hat{\theta}(k)]^2 \quad (6)$$

The minimum error can find by solving the following equation

$$\frac{\partial (E_k^T E_k)}{\partial \hat{\theta}(k)} = 0 \quad (7)$$

The solution of Eq (7) is

$$\hat{\theta}(k) = (X_k^T X_k)^{-1} X_k^T Y_k \quad (8)$$

Since measurements samples have varying influence, the weighted least square equation can be written as

$$\hat{\theta}(k) = (X_k^T W X_k)^{-1} X_k^T W Y_k \quad (9)$$

Where W is a diagonal weight matrix

$$W = \text{diag} [\lambda^{k-1} \cdots \lambda \ 1] \quad (10)$$

$\lambda$  is forgetting factor  $0.95 < \lambda < 0.99$

The quality of measurements can be determined by the covariance matrix of the observation

$$\hat{\theta}(k) = (X_k^T Q_k^{-1} X_k)^{-1} X_k^T Q_k Y_k \quad (11)$$

The main problem with the above algorithm is that requires matrix inversion  $(X_k^T Q_k^{-1} X_k)^{-1}$ , as the size of the matrix depends on the number of parameters, as the number of (n, m) increases, therefore, taking the inverse matrix for all previous measured data will be computationally cost. Moreover, it requires large memory for all pervious measured data. Estimating the parameter vector  $\theta$  recursively can solve these two problems, reducing the memory usage and the computations, where the parameter vector  $\theta$  is updated every iteration until convergence. instead of one time after collecting a dataset.

$$\hat{\theta}(k+1) = \left( \begin{bmatrix} X_k \\ X^T(k) \end{bmatrix}^T \begin{bmatrix} Q_k X_k \lambda \\ X^T(k) \end{bmatrix} \right)^{-1} \begin{bmatrix} X_k \\ X^T(k) \end{bmatrix}^T \begin{bmatrix} Q_k Y_k \lambda \\ y^T(k+1) \end{bmatrix} \quad (12)$$

$$\hat{\theta}(k+1) = [X_k Q_k X_k \lambda + X^T(k) X^T(k)]^{-1} [X_k Q_k Y_k \lambda + X^T(k) y^T(k+1)] \quad (13)$$

Assume the following

$$P(k) = [X^T Q_k X_k]^{-1}, \text{ and} \quad (14)$$

$$P(k+1) = [X_k^T Q_k X_k \lambda + X^T(k) X^T(k)]^{-1} \quad (15)$$

The matrix inversion lemma is very useful here

$$[A + BCD]^{-1} = A^{-1} - A^{-1} B (DA^{-1} B + C^{-1})^{-1} D A^{-1} \quad (16)$$

By applying this Lemma for the calculation of  $P(k+1)$ ,  $A = X_k^T Q_k X_k \lambda$ ,  $B = X(k)$ ,  $C = I$ ,  $D = X^T(k)$ , so we will obtain a recursion formula for solving and updating the matrix  $P(k)$

$$P(k+1) = \lambda^{-1} [P(k) - P(k) X(k) [X^T(k) P(k) X(k) + \lambda]^{-1} X^T(k) P(k)] \quad (17)$$

Using Eq (15), we can write

$$\hat{\theta}(k) = P(k) X_k^T Q_k Y_k \quad (18)$$

After some calculations, the update equation will be

$$\hat{\theta}(k+1) = \hat{\theta}(k) + P(k) X(k) \frac{y(k+1) - X^T(k) \hat{\theta}(k)}{\lambda + X^T(k) P(k) X(k)} \quad (19)$$

## II. EXPERIMENTAL SETUP

The test rig integrates all mechatronics components which include mechanical elements, e.g., pneumatic circuit, electronics elements e.g. embedded closed-loop control for the proportional valve as well as distance sensor and real-time computer interface, with excellent computational capacity and good software capability which receive signals from the sensor process them then send appropriate controlled signal to the actuator drive.

The test rig consists of a pneumatic circuit which composed from a pneumatic power supply 2 HP connected to air service unit, 5/3 Fasto pneumatic servo valve, the pneumatic cylinder has a piston diameter of 42 mm, rod diameter of 16 mm, and stroke length of 140 mm. A pneumatic cylinder attached with timing built and two pullies mechanism in order to measure the linear distance by rotary potentiometer. The feedback signal comes from sent to the computer via a National Instruments (NI) (USB-6009) DAQ card through analog pins. The DAQ card has two analog outputs, three analog inputs, a sampling rate of 48 kS/s, and input voltage range of  $\pm 10$  V. the computer runs MATLAB/Simulink software program to read the analog signal from the sensor and feed the pneumatic servo valve by appropriate controlled signals, the whole process happens in real-time environment. A photograph of the test rig of the Pneumatic System is shown in Figure 3.

## I. ONLINE SYSTEM IDENTIFICATION

Online system Identification estimates the model parameters when new measured data is available while the physical system is working, typically performed using a recursive algorithm which they use the current input and output measured data and previous parameter estimates. Therefore, it is important to have initial model parameters. In order to, obtained a suitable dynamic model reflect the plant characteristics, many experiments were performed on the test rig. First, online identification using the ARMAX model was accomplished using the impulse response of the physical system. Figure 2 shows the Simulink block diagram of parameter estimation that was interfaced to the physical system through the data acquisition card. The Simulink block diagram sends the appropriate input signal to the servo drive through an analog output. The response of the actuator drive (piston displacement) was measured and transmitted to the computer through the analog input (DAQ input). In order to, estimate the model parameters, several prediction strategies were performed with 8-bar supplied pressure and 10-ms sampling time.

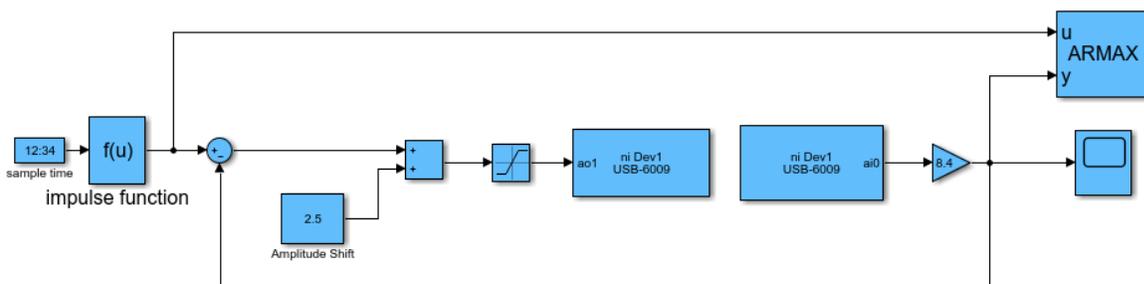


Figure 2. Simulink Block Diagram of the Real System with ARMAX

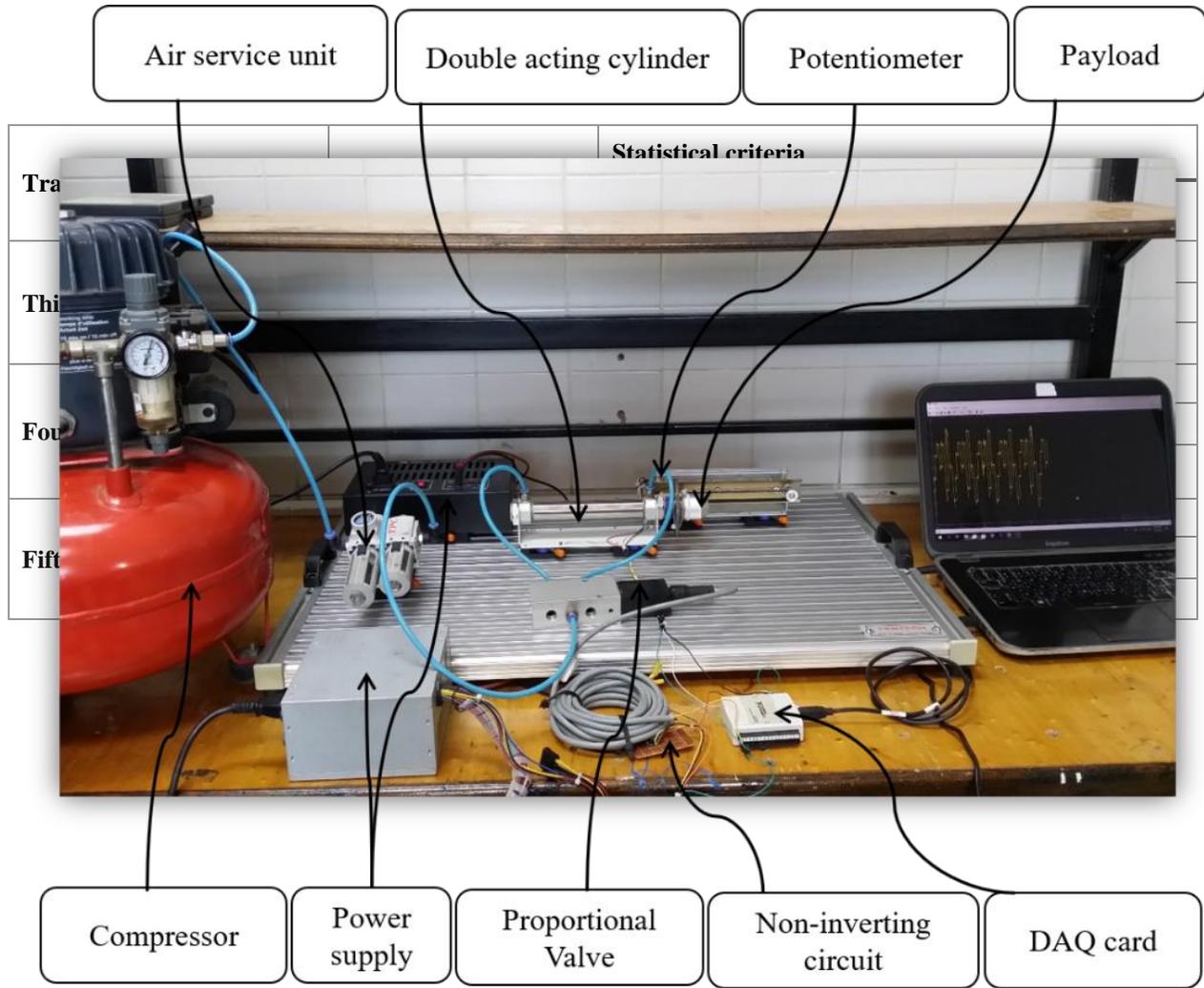


Figure 3. Photograph of the Test Rig of the Pneumatic System

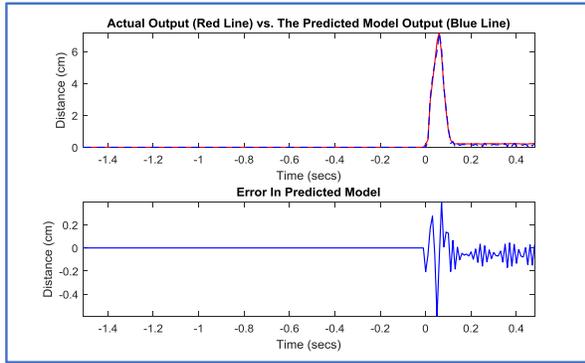


Figure 4. Variation of the measured output, third-order predicted model with one-step prediction and error versus time.

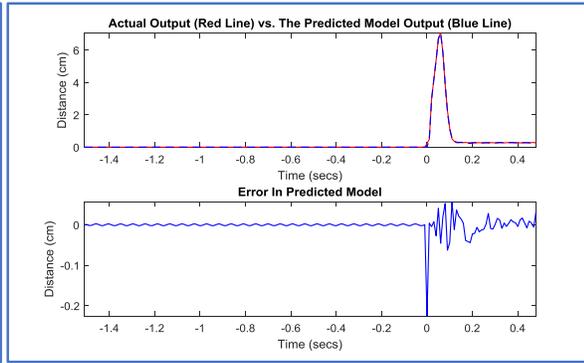


Figure 5. Variation of the measured output, fourth-order predicted model with one-step prediction and error versus time.

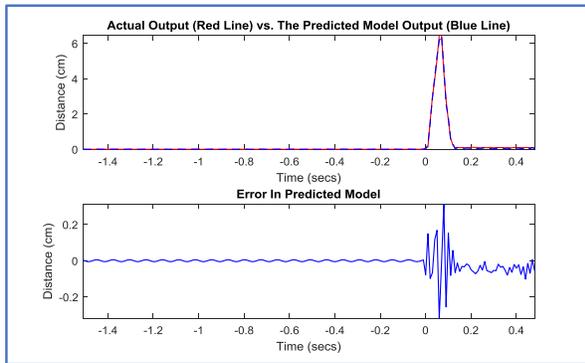


Figure 6. Variation of the measured output, fifth-order predicted model with one-step prediction and error versus time

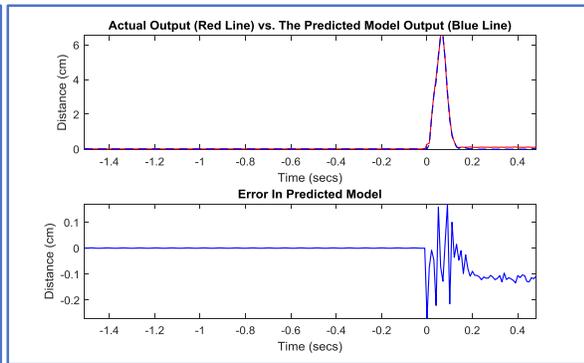


Figure 7. Variation of the measured output, fourth-order predicted model with five-step prediction and error versus time

Table 1. Comparison of statistical error analysis of different transfer function orders with the one-step prediction

**a. Effect of prediction orders**

The model order can be driven mathematical or guessed from prior physical knowledge. The quality of the estimated mathematical model highly depends on the model structure. In all cases, many models have to estimate in order to obtain the best model. Therefore, several experiments were performed using different mathematical model orders (e.g., third, fourth, and fifth orders) and different prediction step sizes (e.g., one step, five steps, and ten steps) to show the effect of the estimation orders and the effect of step size prediction.

Figure 4. shows the online measured output and third-order estimated model with one-step prediction. The square error is  $2.34e-4$  cm, and the Standard deviation is 0.04141 cm. Thus, the obtained transfer Function is:

$$-0.0086731 z^2 + 0.24321 z + 0.06598$$

$$G(z) = \frac{\dots}{z^3 - 0.66561 z^2 - 0.30705 z + 0.34817}$$

$$z^3 - 0.66561 z^2 - 0.30705 z + 0.34817$$

Figure 5. shows the online measured output and fourth-order estimated model with one-step prediction. The square error is  $8.46e-5$  cm, and the Standard deviation is 0.003269 cm. Thus, the obtained transfer Function is:

$$0.0103z^2 + 0.0761z + 0.0761$$

$$G(z) = \frac{\dots}{z^4 - 2.0470 z^3 + 1.0700z^2 + 0.0787z - 0.1017}$$

figure 6. shows the online measured output and fifth-order estimated model with one-step prediction. The square error is  $9.28e-4$  cm, and the Standard deviation is 0.00537 cm. Thus, the obtained transfer Function is:

$$G(z) = \frac{0.044675 z^4 + 0.14976 z^3 - 0.3674 z^2 + 0.44021 z - 0.26773}{z^5 - 2.4258 z^4 + 2.7044 z^3 - 1.8968 z^2 + 0.83927 z - 0.22339}$$

different estimated models and statistical error analysis proved that the best results were the fourth order with one-step prediction, this is because the pneumatic systems are highly nonlinear systems, so that, the one-step prediction succeeded in tracking the system changes. In contrast, five-step predictions in Figures 7.introduced delay, which prevented the predicted models from following the transient response accurately

Table 1.illustrate a comparison of the statistical analysis of the ARMAX estimated models with different transfer function orders and one step prediction. The fourth transfer function shows minimum square error, as a result, was adopted as the system a dynamic model for the controller design.

## II. CONTROLLER DESIGN

### a. Fractional Order PID Controller

Fractional order calculus is a generalization of the integral integration and differentiation to the non-integer order.  $aD_t^r$  is the fundamental operator where the lower and upper bundies of the operation are denoted by a and t respectively and r is the fractional order. Different types of fractional order controllers have been proposed in the literature [28-31] Since A.Oustaloup presents the idea of fractional calculus to control a dynamic system. The most common form of fractional order controllers was introduced by I.Podlubny in 1994 [32], a generalization of popular PID controller, which is known as the  $PI\lambda D\mu$  controller, where  $\lambda$  and  $\mu$  are fractional orders of the integral and derivative parts of the controller, respectively. Therefore, if  $\lambda$  and  $\mu$  are equal one will obtain the famous conventional integral PID controller.

The transfer function of  $PI^\lambda D^\mu$  controller has the form:

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_I}{S^\lambda} + K_D S^\mu, \quad (\lambda, \mu > 0) \quad (20)$$

Where C(s) is the controller output, U(s) and E(s) are control and error signals,  $K_p$  is the proportional gain,  $K_i$  the integration gain and  $K_d$  the differentiation gain.

In general, all four different forms of classic PID controllers can obtain by set  $\lambda$  and  $\mu$  positive integral number. However, we are not interested to obtain integral order PID controller from fractional order PID controller, since FOPID controller is infinite dimensional linear filter, usually controller  $\lambda$  and  $\mu$  are fractional. Therefore, in fractional order PID controller not only we have three controller gains  $K_p$ ,  $K_i$ , and  $K_d$ , but also, we tune the controller integral and derivative order, which will give more flexibility to adjust the dynamic properties of the controller, and as a result we will enhance the performance of the system, as well as the gain tuning process will become more challenge since we have five parameters instead of three.

### b. PID-type fuzzy logic controllers

Many fuzzy logic controller structures have been proposed in the literature (WJM Kickert, EH Mamdani, 1978), (CC Lee, 1990), (J Lee, 1993), (Mann G, el al, 1990) since Mamdani-type fuzzy controller mimic the human operator behavior in the 1970s. The significant work of using fuzzy logic in control systems based on obtaining the mathematical model first then design and analysis a nonlinear fuzzy logic controller was by Sugeno in the 1990s. This researcher focusses on direct action fuzzy controller which called PID-type fuzzy controller (PID-FLC). (Mann G, el al, 1990) Analyzed the direct action fuzzy PID controller structures. The PID-FL controllers are classified according to the number of inputs to the PID-FL controller.

The PID-type fuzzy controller used in the research illustrated on the Figure . which consists of two, the error and derivative of error which represent the PD part of the PID-FL controller and output part represent an integral part.

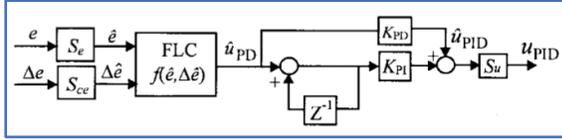


Figure 8. Two-inputs fuzzy PID-FL controller

The advantages of the two inputs fuzzy PID-FL controller do not need much computations, and stable for real-time implementation. As two inputs are supplied to the fuzzy controller, a smaller number of rule-base is required to be designed compared to the three-input structures. In term of the processing time, the Sugeno-type fuzzy inference is much faster than the Mamdani-type fuzzy inference, since the output membership functions of Sugeno-type fuzzy inference are functions. Therefore, the Sugeno-type fuzzy inference is more reliable for real-time applications.

The PID-type fuzzy controller we designed consists of twenty-five rules, as shown in Figure 9. and Table 2. two inputs and one output, each one of them composed of five membership functions. Then, serval optimization algorithms are applied to tune the controller parameters, which transform the system input/output into the normalized range of [-1, 1].

Table 2. 5 X 5 fuzzy logic controller rule-base

e/de	NB	NS	Z	PS	PB
NB	NB	NB	NB	NS	Z
NS	NB	NB	NS	Z	PS
Z	NB	NS	Z	PS	PB
PS	NS	Z	PS	PB	PB
PB	Z	PS	PB	PB	PB

**c. Objective functions**

For design an optimal FOPID controller. Several objective functions can be used for the time-domain optimization [33]. For example,

Integral absolute error:

$$IAE = \int_0^t |r(t) - y(t)| dt = \int_0^t |e(t)| dt, \quad (21)$$

$$Integral\ square\ error: ISE = \int_0^t e^2(t) dt, \quad (22)$$

Integral time-square of error:

$$ITSE = \int_0^t te^2(t) dt, \text{ and} \quad (23)$$

Integral time-absolute of error:

$$ITAE = \int_0^t t|e(t)|dt. \quad (24)$$

It is worth to notice that the different performance indices will obtain different controller gains consequently will produce different controller performance. Specifically, IAE and ISE criteria can eliminate the overshoot but, in a bit, long settling time, because of both IAE and ISE performance criteria weights all errors equally independent of time. In contrast, ITSE and ITAE can overcome that disadvantage [34]. The optimization process works as shown in figure 10. Table 2, table 4, and table 5 contain the search domains of the controllers for all optimization algorithms.

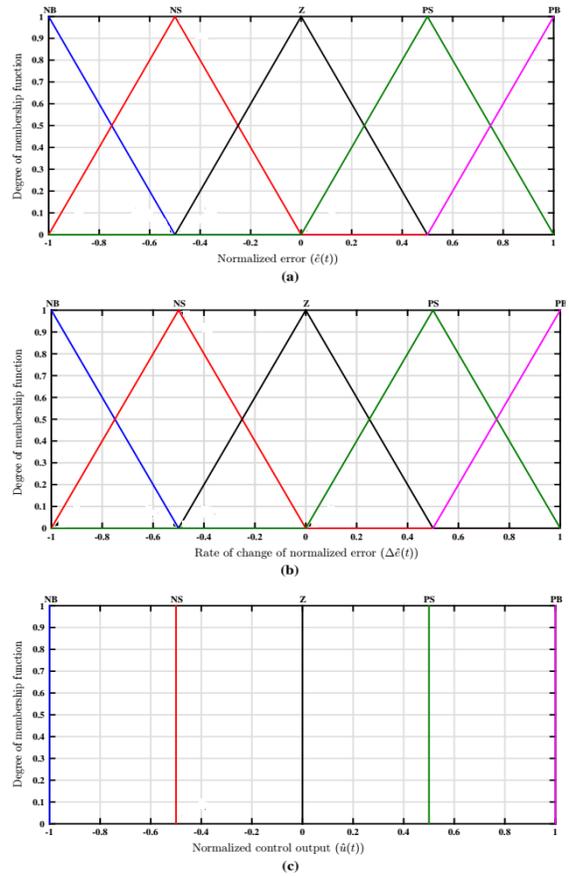


Figure 9 The distribution of the inputs/output membership functions

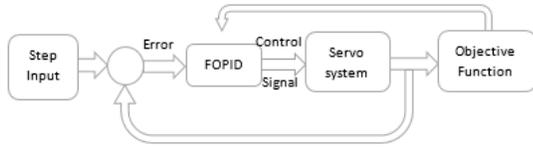


Figure 10. Optimization process flow diagram

Table 3. PID Controller search domain for all optimization algorithms

	Lower Bounds	Upper Bounds
<b>Kp</b>	0	0.5
<b>Ki</b>	0	1
<b>Kd</b>	0	1

In integral order PID controller case, all search algorithms failed to find the optimal controller parameters from the first time, so it was necessary to repeat the search process several times with different search domain.

Table 4. FPID Controller search domain for all optimization algorithms

	Lower Bounds	Upper Bounds
<b>Kp</b>	0	1
<b>Ki</b>	0	2
<b>Kd</b>	0	2
<b>Mu</b>	0	1
<b>Lambda</b>	0	1

Table II. PID-type Fuzzy Controller search domain for all optimization algorithms

	Lower Bounds	Upper Bounds
<b>K1</b>	0	1
<b>K2</b>	0	2
<b>K3</b>	0	2
<b>K4</b>	0	1

**d. Genetic algorithm (GA)**

Genetic algorithm (GA) is the most popular optimization algorithm based on the natural selection process and genetic evolution of the population. GA starts with a set of solutions (called population) of randomly generated chromosomes. The population in each iteration called generation. At each generation, the fitness of every chromosome is evaluated by the objective function. The fittest chromosomes will select to be parents and the next generation will (children) will produce by crossover and mutation of the parents [35].

The procedure of GA is given by the following algorithm

```

Set t = 0.
Initializes the population P (0).
While (condition true) do
Evaluate fitness of everyone of P(t).
Select the best-fit individuals from P(t)
and set them as parents
Apply crossover and mutation operations
on parents to generate a new generation
P(t+1).
t = t + 1.
    
```

Table 6. Parameters used in the genetic algorithm.

Parameter	Type/value
Generations	100
Selection	Uniform
Crossover	Constrained dependent
Mutation	Constrained dependent
Encoding	Binary
Fit. Fun.	Integral absolute error

The closed loop unit step response simulation results with fractional and integral order PID controller, and PID-type fuzzy controller designed using GA are plotted in figure 11. Obviously, both fractional and PID-type fuzzy controller improved the time response characteristics and eliminated the overshoot. In contrast, integral order PID controller struggles to find the optimal controller gains. As a result, it doesn't improve the system response. Table 7. reported all optimal controllers' parameters.

Table 7. The optimal controllers' gains tuned by GA

	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>	<b>Lambda</b>	<b>Mu</b>
<b>PID</b>	1	0.0014 92	0.04991 3	1	1
<b>FPID</b>	0.821 40	0.1247 12	0.69211	0.00508	0.124 712
	<b>K1</b>	<b>K2</b>	<b>K3</b>	<b>K4</b>	
<b>Fu-PID</b>	9.999 4	0	1.2413	3.9218	0

**e. Particle swarm optimization (PSO)**

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 moves in steps throughout a region. At each iteration, the PSO algorithm evaluates the fitness

function for each particle. After the evaluation process, the PSO algorithm gives the new velocity of each particle. The particles move, then the PSO algorithm re-evaluates [35].

The velocity of each particle is adjusted according to its own flying experience and the flying experience of other particles. The modified velocity and position of each particle can be calculated using the current the distance and velocity from  $x_{i,d}^b$  to global best as shown in the following equations [36]

$$v_i(t+1) = \alpha v_i(t) + c_1 r_1 (x_i^b(t) - x_i^k(t)) + c_2 r_2 (x_i^g(t) - x_i^k(t)) \quad (25)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (26)$$

$$\alpha(t) = \alpha_{max} - (\alpha_{max} - \alpha_{min}) \frac{t}{T} \quad (27)$$

Table 8. Parameters used in PSO.

	Search Criteria
No.of.Particles	15
No.of. Iterations	40
Fitness Fun.	Integral time-absolute of error

```

Initialize
repeat
  for each particle i in S do
    if f(xi) < f(pbi) then pbi = xi
    end if
    if f(pbi) < f(gb) then gb = pbi
    end if
  end for
  for each particle i in S do
    for each dimension d in D do
      calculate the velocity
      calculate the position
    end for
  end for
  it = it + 1
until it > MAX_ITERATIONS
    
```

Figure 12. shows compassion among three different controllers PSO-PID, PSO-FPID, and PSO-type Fuzzy Controller. It is observed that both fractional order PID controller and a PID-type fuzzy controller that tuned by PSO improved the time response characteristics and eliminated the overshoot. In contrast, integral order PID controller has the same rise time as fractional order PID, but it struggles to

show an acceptable response.table 9. illustrates the different parameters of controllers.

Table 9. The optimal controllers' gains tuned by PSO

	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>	<b>Lambda</b>	<b>Mu</b>
<b>PID</b>	1	-0.0018	0.0499	1	1
<b>FPID</b>	0.94 18	0.0760 9	0.0977 9	0.25466	0.844 2
	<b>K1</b>	<b>K2</b>	<b>K3</b>	<b>K4</b>	-
<b>FPID</b>	0.39 18	10.021 3	1.4529	2.500.1	-

### f. Gray Wolf Optimization (GWO)

Gray Wolf Optimization is a mathematical model for the social hierarchy and hunting techniques of gray wolves social behavior. The algorithm made up of four different kinds of wolves ( $\alpha$ ,  $\beta$ ,  $\delta$ , and  $\omega$ ). Alpha ( $\alpha$ ) is the best solution, beta ( $\beta$ ) and delta ( $\delta$ ), considered as second and third best solutions. The rest of the wolves are omega ( $\omega$ ). At the beginning of optimization search, all solutions are considered as  $\omega$ , until the first iteration finished, the three best solutions are assigned to ( $\alpha$ ,  $\beta$ , and  $\delta$ ) wolves [37]. After assuming the random solutions, wolves start encircling the prey according to equation (26):

$$\vec{D} = |\vec{C} \times \vec{X}_p(t) - \vec{X}(t)| \quad (28)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \times \vec{D} \quad (29)$$

Where  $\vec{X}$  indicates the position vector of a grey wolf, and  $\vec{X}_p$  represents the position vector of the prey, t is the current iteration,  $\vec{A}$  and  $\vec{C}$  are coefficient vectors and given in equations.

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \quad (30)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (31)$$

Where components of  $\vec{a}$  are linearly decreased from 2 to 0 over the course of iterations and  $r_1, r_2$  are random vectors in [0, 1]

The optimizations process continuous by sorting and save the best three solutions ( $\alpha$ ,  $\beta$ , and  $\delta$ ) obtained so far. The rest of the candidate solutions update their positions according to equations (31, 32, and 33).

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|,$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \quad (32)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \quad (33)$$

$$\begin{aligned} \vec{X}_2 &= \vec{X}_\beta - \vec{A}_1 \cdot \vec{D}_\beta, \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_1 \cdot \vec{D}_\delta \\ \vec{X}(t+1) &= \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \end{aligned} \quad (34)$$

The last closed-loop simulation results, shown in figure 13. are the three controllers designed using GWO. Similar to the two previous search algorithms, GWO successfully obtained the optimal controller parameters, as a consequence, both fractional and PID-type fuzzy controller improved the time response characteristics. In contrast, integral order PID doesn't struggle dealing with the system nonlinearly. Table 10. reported all optimal controllers' parameter.

```

Initialize the grey wolf population  $X_i$  ( $i = 1, 2, \dots, n$ )
Initialize  $a, A$ , and  $C$ 
Calculate the fitness of each search agent
 $X_\alpha$ =the best search agent
 $X_\beta$ =the second-best search agent
 $X_\delta$ =the third best search agent
while ( $t < \text{Max number of iterations}$ )
  for each search agent
    Update the position of the current search agent by equation (3.7)
  end for
  Update  $a, A$ , and  $C$ 
  Calculate the fitness of all search agents
  Update  $X_\omega, X_p$ , and  $X_o$ 
   $t=t+1$ 
end while
return  $X_\alpha$ 
    
```

GWO

	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>	<b>Mu</b>	<b>lambda a</b>
<b>PID</b>	0.994 2	0.0036	0.0500	1	1
<b>FPI</b>	0.831	0.1076	0.1007	0.729	0.342
<b>D</b>	11	00	03	01	01
	<b>K1</b>	<b>K2</b>	<b>K3</b>	<b>K4</b>	-
<b>Fu-PID</b>	0.486 5	9.4414	1.0529	1.824 3	-

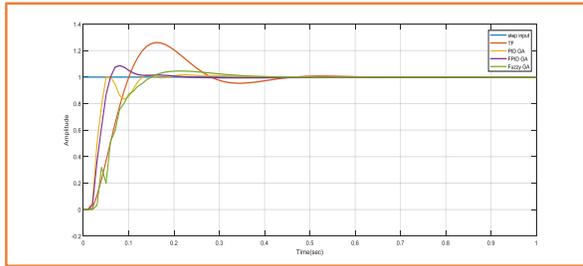


Figure 11. Step response of the system with controllers tuned by GA

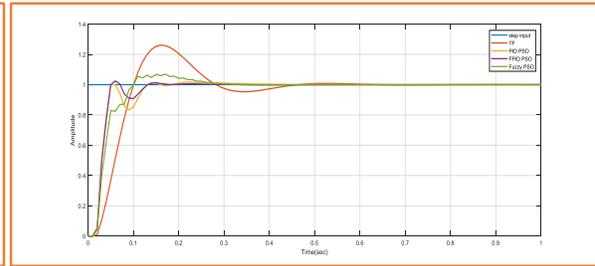


Figure 12. Step response of the system with controllers tuned by PSO

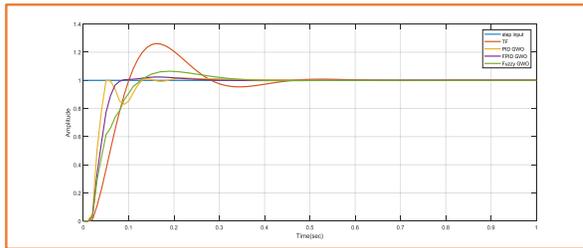


Figure 13. Step response of the system with controllers tuned by GWO

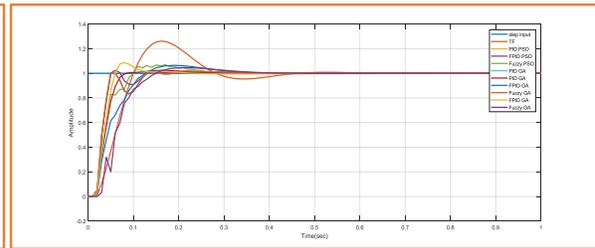


Figure 14. Step response of the system with controllers tuned by the three controllers

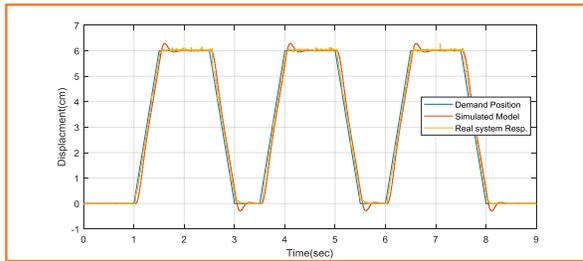


Figure 15. Variation of the real-time system response and demand multiple speed profiles versus time

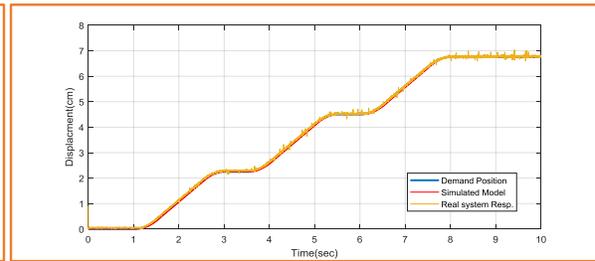


Figure 16. Comparison between real system position and simulated model position with demand multiple position profiles versus time

**g. Comparison of Three Controllers**

Three different search algorithms are employed to seek the optimal parameters values for three controllers, fractional order PID controller, integral order controller and PID-type fuzzy logic controllers. The results show that all the three algorithms are guaranteed to find the optimal controller parameters of the pneumatic servo system. Figure 14. shows the controllers designed using the three different search algorithms have quite similar performance. The different between GA and POS and GWO are the number of iteration and the computational time, and it is not surprising since both PSO and GWO has simple structure compared with GA which has serval computation processes such as selection and crossover.

The second observation is both fractional PID controller and PID-type fuzzy controller, show a good performance, but the design steps are completely different also several runs were required to find the optimal controller parameters as all optimization search algorithms failed to find the optimal values on the first attempt. On the other hand, integral order PID doesn't struggle dealing with the system nonlinearly.

To summarise, Integral order PID controller is an especial case of fractional order PID controller where the derivative exponent mu and integral exponent lambda are equal to one. Therefore, if the integral order controller fulfills the intended system needs, in this case, integral order PID controller is the best choice, since it has a simple structure, easy to

implement, and doesn't need higher order filter. But in the case of nonlinear systems integral order PID doesn't show a good performance in comparison with the linear controllers, the presented Fractional order PID and PID-type fuzzy logic controller showed excellent performances.

#### **h. Real-time implementation of fuzzy controller**

After achieving satisfactory results, the optimized controller(s) was applied to the real system through the real time environment in order to validate the controller performance on the real system as well as ensure the estimated model accuracy figure 15. shows the variations of the desired speed profile, the real time system response with controller and the simulated system speed response. The results assure much closer tracking of the speed profile, with good time response characteristics.

Figure 16. shows the results of the accumulated displacements of real time system response with controller and the simulated system, due to multiple profiles. The pneumatic system succeeded in tracking the desired position and showed minimum sustained oscillations.

### **III. COLCLSION**

In this paper, a method to identify and control pneumatic servo drive is proposed and implemented. In order to avoid the great difficulty associated with servo-pneumatic system modeling and control, an online-system identification method is employed to approximate the dynamic system model from the input/ output measured data using the recursive least squares algorithm (RLS) based on the auto-regressive moving-average model (ARMA). The advantages of this method apart from high accuracy in the identified model, reduction in the tuning time required of the controller parameters, and low cost. The results showed a good match between the simulated model and real system response. This implies that the accuracy of the predicted model obtained through on-line identification is high. Furthermore, the estimated model used to control the pneumatic servo system showed an excellent performance in tracking the reference positions of multiple profiles.

Due to the highly nonlinear nature of the system under study, two sophisticated controllers, PID-type fuzzy logic controller and Fractional order PID controller were chosen as well as the integral order PID controller. The results indicated that the both presented controllers, fractional order PID, and PID-

type fuzzy logic controller, showed excellent performances. On the other hand, several runs were required in order to capture the excellent PID-type fuzzy logic controller performance as all algorithms failed to find the optimal values on the first attempt. That is because the performance of the fuzzy controllers depends highly on the proper selection of many design parameters. In contrast, integral order PID struggles to deal with the system nonlinearly.

Finally, the three search algorithms, namely genetic algorithms (GA), particle swarm optimization (PSO), and gray wolf optimization (GWO), were utilized to find the optimal parameters of the three controllers. Similarly, both PSO and GWO were able to find the optimal controller parameters as GA, but differently, both of them did not need many computations as GA.

### **References**

- [1] Shearer J L (1954) Continuous control of motion with compressed air. Ph.D. thesis, Massachusetts Institute of Technology
- [2] Liu, S., and J. E. Bobrow. "An analysis of a pneumatic servo system and its application to a computer-controlled robot." *Journal of Dynamic Systems, Measurement, and Control* 110.3 (1988): 228-235.
- [3] Pu, J., R. H. Weston, and P. R. Moore. "Digital motion control and profile planning for pneumatic servos." *Journal of dynamic systems, measurement, and control* 114.4 (1992): 634-640.
- [4] Richer, Edmond, and Yildirim Hurmuzlu. "A high performance pneumatic force actuator system: Part I—Nonlinear mathematical model." *Journal of dynamic systems, measurement, and control* 122.3 (2000): 416-425.
- [5] Kimura, Tetsuya, et al. "Control for pneumatic actuator systems using feedback linearization with disturbance rejection." *American Control Conference, Proceedings of the 1995. Vol. 1. IEEE, 1995.*
- [6] Kimura, T., et al. "Feedback linearization for pneumatic actuator systems with static friction." *Control engineering practice* 5.10 (1997): 1385-1394.
- [7] Richard, E., Scavarda, S. (1996). *Comparison between Linear and Nonlinear Control of a Pneumatic Servodrive. ASME*

Journal of Dynamic Systems, Measurement and Control , Vol. 118, No.2, pp. 245-252.

[8] Brun, X., Belgharbi, M., Sesmat, S., Scavarda, S. (1999). Control of an electropneumatic actuator: Comparison Between Some Linear and Non-linear Control Laws. Proceedings of the Institution of Mechanical Engineers, Part 1: Journal of Systems and Control Engineering, pp. 387-405.

[9] Brun, X., Thomasset, D., Bideaux, E. (2002). An Accurate Tracking Control of an Electropneumatic Actuator. 1st Fluid Power Net International Ph.D Symposium, pp. 215-226.

[10] Wang, J., Kotta, U., Ke, J. (2007). Tracking Control of Nonlinear Pneumatic Actuator Using Static State Feedback Linearization of Input/Output Map. Proceedings of Estonian Academy on Science, Physics and Mathematics , Vol. 56, p. 47-66.

[11] Perondi, E., Suzuki, R., Sobczyk, M. (2010). Feedback Linearization Control Applied to a Pneumatic Actuator System. In XVIII Congresso Brasileiro de Automatica (CBA2010), pp. 24732478.

[12] Sobczyk, M., Perondi, E., Suzuki, R. (2012). Feedback Linearization Control with Friction Compensation Applied to a Pneumatic Positioning System. ABCM Symposium Series in Mechatronics, Vol. 5 , pp. 252-261.

[13] Weston, R. H., Moore, P., Thatcher, T. W. (1984). Computer Controlled Pneumatic Servo Drives. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture , Vol.198, pp. 275-281.

[14] Moore, P. R., Weston, R., Thatcher, T. W. (1985). Compensation in Pneumatically Actuated Servomechanisms. Trans Inst M C , Vol.7, No.5, pp. 238-244.

[15] Moore, P., Ssenkungo, F. W., Weston, R. H., Thatcher, T. W., Harrison, R. (1986). Control Strategies for Pneumatic Servo Drives. International Journal of Production Research , Vol. 24, No.6, pp. 1363-1382

[16] Wang, Jihong, Junsheng Pu, and Philip Moore. "A practical control strategy for servopneumatic actuator systems." Control Engineering Practice 7.12 (1999): 1483-1488.

[17] Ning, Shu, and Gary M. Bone. "High steady-state accuracy pneumatic servo positioning system with PVA/PV control and

friction compensation." Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. Vol. 3. IEEE, 2002.

[18] Surgenor, B.W. and Iordanou, H.N. (1994), "Experience with Advanced Position Control of Pneumatic Systems," IFAC Workshop: Trends in Hydraulics and Pneumatics, Chicago, IL.

[19] Perondi, E.A. and Guenther, R. (2000), "Control of Servopneumatic Drive with Friction Compensation," First FPNI — PhD Symposium, Hamburg, pp. 117–127.

[20] Bobrow, J. E., and F. Jabbari. "Adaptive pneumatic force actuation and position control." Journal of dynamic systems, measurement, and control 113.2 (1991): 267-272.

[21] McDonell, B. W., and J. E. Bobrow. "Adaptive tracking control of an air powered robot actuator." Journal of dynamic systems, measurement, and control 115.3 (1993): 427-433.

[22] Shih, M. C., Tseng, S. I. (1994). Pneumatic Servo-Cylinder Position Control by PID Self-tuning Controller. JSME International Journal, Series C , Vol. 37, No.3, pp. 565-572.

[23] Hamiti, K., Voda-Besanqon, A., Roux-Boisson, H. (1996). Position Control of a Pneumatic Cylinder Under the Influence of Stiction. Control Engineering Practice , Vol. 4, No. 8, pp. 10791088.

[24] Richardson, R., Plummer, A. R., Brown, M. D. (2001). Self-tuning Control of a Low-Friction Pneumatic Actuator Under the Influence of Gravity. IEEE Transactions on Control System Technology , Vol. 9, No.2, pp. 330-334.

[25] Situm, Z., Pavkovic, D., Novakovic, B. (2004). Servo-Pneumatic Position Control Using Fuzzy PID Gain Scheduling . ASME Journal of Dynamic Systems, Measurement and Control, Vol.12, No.2, pp. 376-387.

[26] Saleem, Ashraf, Saber Abdrabbo, and Tarek Tutunji. "On-line identification and control of pneumatic servo drives via a mixed-reality environment." The International Journal of Advanced Manufacturing Technology 40.5-6 (2009): 518-530.

[27] T. Kara, I. Eker, Nonlinear modeling and identification of a DC motor for bidirectional operation with real time experiments, Energy Conversion and Management 45 (2004) 1087–1106.

- [28] Y. Q. Chen, I. Petras, and D. Xue, "Fractional order control a tutorial," in Proc. ACC '09. American Control Conference, 2009, pp. 1397–1411
- [29] A. Oustaloup, P. Melchior, P. Lanusse, O. Cois, and F. Dancla, "The CRONE toolbox for Matlab," in Proc. IEEE Int. Symp. Computer-Aided Control System Design CACSD 2000, 2000, pp. 190–195.
- [30] I. Petráš, S. Grega, and L. Dorcak, "Digital fractional order controllers realized by PIC microprocessor: Experimental results," in Proc. of the ICCC'2003 conference, High Tatras, Slovak Republic, 2003, pp. 873– 876.
- [31] Concepción A. Monje, YangQuan Chen, Blas Vinagre, Dingy u Xue and Vincente Fileu. "Fractional Order Controls - Fundamentals and Applications". Springer-Verlag London, Advances in Industrial Control series, (Invited book project, to be published in 2009)
- [32] Podlubny, Igor. "Fractional-order systems and  $PI^{sup/spl \lambda}/D^{sup/spl \mu}/-$  controllers." IEEE Transactions on automatic control 44.1 (1999): 208-214.
- [33] Aleksei, Tepljakov, Petlenkov Eduard, and BelikovJuri. "A flexible MATLAB tool for optimal fractional-order PID controller design subject to specifications." Control Conference (CCC), 2012 31st Chinese. IEEE, 2012.
- [34] Gaing, Zue-Lee. "A particle swarm optimization approach for optimum design of PID controller in AVR system." IEEE transactions on energy conversion 19.2 (2004): 384-391
- [35] Du, Ke-Lin, and M. N. S. Swamy. Search and Optimization by Metaheuristics. Springer, 2016.
- [36] Hazem I Ali, SamsulBahari B Mohd Noor, S M Bashi, MohdHamiruceMarhaban, "Quantitative feedback
- [37] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." Advances in Engineering Software 69 (2014): 46-61