HARDWARE-IN-THE-LOOP FOR THE DESIGN AND IMPLEMENTATION OF SERVO PNEUMATIC SYSTEM

By

Bashar Saad Taha

Supervisor

Dr. Ahmad Al-Qaisia, Prof.

Co-Supervisor

Dr. Ashraf Saleem

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree of Science in Mechanical Engineering

Faculty of Graduate Studies

The University of Jordan

December, 2010

The University of Jordan

Authorization Form

I, Bashar Saad Taha , authorize the University of Jordan to supply copies of my Thesis/ Dissertation to libraries or establishments or individuals on request, according to the University of Jordan regulations.

Signature:

Date: 12/12/2010

الجامعة الأردنيـــة
كلية الدراسات العليا

التاريـــخ: / /

نموذج رقم (١٨)
اقرار والتزام بالمعايير الأخلاقية والأمانة العلمية
وقوانين الجامعة الأردنية وأنظمتها وتعليماتها
لطلبة الماجستير

أنا الطالب: بشار سعد مصطفى لطوف الرقم الجامعي: (٨٠٦٢٠٨٧)

تخصص: هندسة الميكانيك الكليـــــة: الهندسة

عنوان الرسالة: Hardware-In-The loop for the design and implementation of servo Pneumatic systems.

اعلن بأنني قد التزمت بقوانين الجامعة الأردنية وأنظمتها وتعليماتها وقراراتها السارية المفعول المتعلقة باعداد رسائل الماجستير عندما قمت شخصيا" باعداد رسالتي وذلك بما ينسجم مع الأمانة العلمية وكافة المعايير الأخلاقية المتعارف عليها في كتابة الرسائل العلمية. كما أنني أعلن بأن رسالتي هذه غير منقولة أو مستلة من رسائل أو كتب أو أبحاث أو أي منشورات علمية تم نشرها أو تخزينها في أي وسيلة اعلامية، وتأسيسا" على ما تقدم فانني أتحمل المسؤولية بأنواعها كافة فيما لو تبين غير ذلك بما فيه حق مجلس العمداء في الجامعة الأردنية بالغاء قرار منحي الدرجة العلمية التي حصلت عليها وسحب شهادة التخرج مني بعد صدورها دون أن يكون لي أي حق في التظلم أو الاعتراض أو الطعن بأي صورة كانت في القرار الصادر عن مجلس العمداء بهذا الصدد.

توقيع الطالب: بشار سعد التاريخ: ١٢ / ١٢ / ٢٠١٠

COMMITTEE DECISION

This Thesis/Dissertation (The Relationship Between Socioeconomic Status and Disruptive Behaviors in Early Elementary School Children in Greater Amman Municipality) was Successfully Defended and Approved on 2/12/2010

Examination Committee	**Signature**
Dr. Ahmad Al-Qaisia, (Supervisor) Prof. in Mechanical Engineering	
Dr. Ashraf Ismail Saleem. (Co-Supervisor) Assist. Prof. in Mechatronics Enginering	
Dr. Lutfi Rawhi Al-Sharif (Member) Assist. Prof. in Mechatronics Engineering	
Dr. Nidal Mohammad Aleef (Member) Assist. Prof in Mechanical Engineering	
Dr. Tarek Aqail Tutunji (Member) Assoc. Prof in Mechatronics Engineering (Philadelphia University)	

ACKNOWLEDGMENTS

I thank my wife Nour for her love, support, and understanding. I would also thank my family. I also extend my thanks to my extended family, whom have helped both Nour and myself in many ways.

I thank my supervisor Prof.Ahmad Al-Qaisia for his help along the work.

I thank my Co-Supervisor, Dr. Ashraf Saleem, for his ongoing encouragement in my academic endeavours. His friendship and guidance has made this work possible.

My appreciation goes to my boss Mr. Georges Sleiman, Mr.Nidal Salman and all people in United Traders for their unwavering support and guidance throughout the study.

Finally, thanks to Nour again for your patience while I spent many hours glued in front of the PC. You have shown great understanding and given selfless support.

Table of contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ARMA	Auto-Regressive Moving-Average
CC	Cascade Control
HIL	Hardware In the Loop
MRE	Mixed Reality Environment
PSO	Particle Swarm Optimization
PID	Proportional-Integral-Derivative controller
ZN	Ziegler and Nichols
DCS	Distributed Control System
SCADA	Supervisory Control And Data Acquisition

HARDWARE-IN-THE-LOOP FOR THE DESIGN AND IMPLEMENTATION OF SERVO PNEUMATIC SYSTEM

By
Bashar Saad Taha

Supervisor
Dr.Ahmad Al-Qaisia,Prof

Co- Supervisor
Dr. Ashraf Saleem

ABSTRACT

Servo pneumatic systems have properties that can make them favourable for servo applications. The actuators themselves are of simple construction, widely sourced, and easily maintained, making them low in cost. They have a high power-to-weight ratio, are fast acting. Compressed air is readily available in most industrial environments. Air is highly compressible, which makes the systems' behaviour highly non-linear, and designing a controller for such systems very complicated. The complexity arises in acquiring the system's transfer function accurately, which causes a great difficulty in servo-pneumatic system modeling and control, Moreover, it is highly complicated to adopt a strategy that does not require a lot of time in building a controller structure.

The recent availability of low-cost, high-performance computer processors is allowing servopneumatic actuators to take advantage of advanced control algorithms in industrial applications. Several manufacturers already offer industrial servopneumatic controllers, and the technology is finding new applications. Still, there is room for further research into design and control methodologies for servopneumatic systems.

Therefore, this research strives to apply a methodology for identification and control of servo pneumatic using a particle swarm optimization technique as the tuning method in the controller design. Moreover it aims at applying the concept of Hardware in the loop (HIL) as a real time simulation environment which allows the system to be controlled easily without damaging any component of the system or extra cost in building the controller itself by replacing components according to application requirements.

Chapter 1
Introduction

1.1 Background

A considerable amount of interest has been shown by researchers in the modelling and control of pneumatic drives over the past decade. This increasing acceptance is due to the fact that pneumatic systems offer a number of advantages: typically they are fast, robust, simple to maintain and low in cost. However, the dynamic model of servo-pneumatic system is highly non-linear, which makes designing a controller for such systems very complicated. The complexity arises in acquiring the system's transfer function accurately, which causes a great difficulty in servo-pneumatic system modelling and control, Moreover, it is highly complicated to adopt a strategy that does not require a lot of time in building a controller structure to end up with a reliable response attaining a desired set point accurately.

1.2 Motivation

Pneumatic systems have properties that can make them favourable for servo applications; they are widely sourced, easily maintained and low in cost. They have a high power-to-weight ratio, are fast acting, and they are generally clean and reliable in operation. (Moore, Pu, & Harrison, 1993; Van Varseveld & Bone, 1997). However, they exhibit highly nonlinear dynamic characteristics due to the compressibility of air, complexity of friction presence and time delay in response.

Lack of simulation tools dedicated for such systems made the design, implementation of different control methodologies, and the controller optimization for pneumatic systems more complex jobs for control engineers. To cope with these complexities, it would be more advantageous if the simulation environment Hardware in the loop (HIL) could be applied and run in the time domain so that it can be mixed virtually with the real systems. This would make the simulation more accurate and reliable especially when dealing with nonlinear systems. In addition this methodology facilities the identification and tuning techniques where in this research a particle swarm optimization has been adopted as identification and tuning for the pneumatic systems.

The proposed environment also provide the capability of predicting the dynamic behaviour of servo-pneumatic systems and increase their usability which will enhance their market potential considerably and help both of designers and end users to cope with these highly non-linear systems.

1.3 Aim and Objectives

1.3.1 Aim

This research strives to apply a methodology for identification and control of servo pneumatic system using a particle swarm optimization technique as the tuning method in the controller design. Moreover it aims at applying the concept of Hardware in the loop (HIL) as a real time simulation environment which allows the system to be controlled easily without damaging any component of the system or extra cost in building the controller itself by replacing components according to application requirements.

1.3.2 Objectives

The three streams of research efforts objectives are ; 1- Applying the concept of Hardware In the Loop (HIL) - that bridges the gap between the virtual and real systems. This is very important in servo pneumatic control field and have the advantages of reducing the risks of discovering an error in the last stage of design, the cost of extra parts, time saving in designing and troubleshooting any servo pneumatic system. 2- Applying a single and cascaded PID control structure through HIL. 3- Develop a method for single and cascade controller tuning using an optimization technique - Particle Swarm Optimization (PSO).

1.4 Research methodology

The research study was initiated with literature review for the recent technologies in servo pneumatic systems, identification and acquiring the system's transfer function which is a very challenging task to get an accurate mathematical model for describing pneumatic servo drives behaviour. The second stage- implementation, testing and evaluation of different control structures (cascade and single PID structure). Third and fourth stages- developing and applying the particle swarm optimization as the tuning technique for the proposed cascaded and single PID controllers. In the last stage the concept of 'Hardware in the loop' was demonstrated and implemented effectively.

Chapter 2

Literature review

2.1 Introduction

This chapter presents an overview of identification techniques that have been used in identifying pneumatic models focuses on the recent developments in this track. Brief descriptions of using Particle swarm optimization as identification method is also presented, and finally review the recent technology of servo pneumatic control using single and cascaded PID and the tuning algorithms applied on such controllers. The objective of this chapter is to provide an overview of identification techniques, control methodologies and tuning algorithms to extract the knowledge base in servo pneumatic control in order to widen the applications range of pneumatic systems in the future.

2.2 Identification of pneumatic systems

System identification is concerned with the estimation of a system on the basis of collecting a large amount of reliable data carefully; this involves specification of the model structure that describes the relations between the observed variables, estimation of the unknown model parameters and validation of the resulting model.

Classical approaches to position control of pneumatic drives is based on the linearization of the nonlinear system dynamics around the desired set point, and such models are valid only for small deviations around an operating point. So it is very challenging task to get an accurate mathematical model for describing pneumatic servo drives behavior. In practical applications, one often does not have available values for the model parameters and/or part of the model structure. Therefore, one tries to obtain these parameters and/or structural elements using experimental data from the real process. Researchers developed various parameter-identification methods and applied them to many engineering systems. One of these techniques presented by G. Carducci, N.I. Giannoccaro, where an optimization method used for identification of viscous friction coefficients for a pneumatic system model. They focused on developing a mathematical model of a pneumatic actuator driven by two on/off two-way valves based on the identified friction parameters. They reported that pneumatic systems are not only nonlinear, but also involve several tuning parameters (G. Carducci, N.I. Giannoccaro, 2006).

2.3 PSO for identification

System identification is an important step for the design of a controller, because the whole controller will be built later on according to the identified model.

The least-squares approach is a basic technique often used for parameters estimation. It has been successfully used to identify the parameters in the static and dynamic systems, respectively. But, it is only valid for the model structure of system having the property of being linear in the parameters. Once the form of model structure is not linear in the parameters, this approach may be invalid. (Alfi. et al 2010).

To overcome this problem Particle Swarm Optimization (PSO) algorithm seems to be a more helpful approach for identifying the transfer function of the highly non-linear pneumatic systems. Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. Chapter three will describe in details the particle swarm optimization technique.

2.4 Servo-pneumatic control and tuning methodologies

2.4.1 Background

One of the first attempts to analyse pneumatic control systems was reported by Shearer, where theoretical and experimental results of open and closed-loop position controlled pneumatic systems were presented in 1954. Burrows and Webb (1966) extended Shearer's analysis using the root locus technique from linear control theory. (Shearer, J. L, 1956).

In 1969 Burrows described a system structure that was very similar to those used for later work by others: "The valve is operated by a torque motor fed by a controller which sums the effect of position, velocity and transient pressure feedback". This means there was an electrically operated directional control valve and a feedback of the control signal position and the additional signals velocity and pressure change. The next important contribution was published by Barker (1976). He used the Luenberger observer to reconstruct the velocity and acceleration signals from the measured position signal and could thus omit two sensors. The additional feedback of these two signals significantly improved the performance of his system, a linear pneumatic actuator in a missile flight control system.

There was no substantial advancement made available until 1980s when Mannetje (1981) introduced a high gain differential pressure feedback loop which alters the valve characteristics to a constant pressure source independent of the amount of flow. The method showed significant improvement of the system bandwidth. (Mannetje.J.J.1981). Weston, et al., (1984) considered microprocessor based low cost servo-pneumatic drives. Based on a linearized model, a three loop controller with position, velocity, and acceleration feedback was implemented. It achieved a substantial improvement in both the static and dynamic performance of the drive when positioning loads in a "point-to-point" mode with a brake subsystem.

Moore, et al. (1985) conduct a modelling study to investigate the nonlinear characteristics of pneumatic actuator systems. The pneumatic system is modelled as a cascade connection of two nonlinear subsystems affine in the control input. A robust servo control strategy is proposed using the deterministic control design method developed by Wang and Goodall; the major disadvantage of the servo control strategy is that the method requires the full state feedback which will definitely increase the cost of the whole system.

Yang and Tomizuka (1988) presented an adaptive pulse width control (PWC) scheme for a precise point to point positioning system; the disadvantages of such systems were difficulties in generating impulses, saturation effects, and problems in managing inputs in presence of nonlinearities. Yang and Chu (1993) proposed an adaptive controller which consists of a friction compensator and a PID controller. The results are shown to be superior but such approaches are often complex and the tuning process tends to be difficult and sometimes unreliable. In 1996 Hamiti proposed a new control scheme consists of two parts, an analog P controller part cascaded with a digital PI controller part, the inner loop is used to stabilize the system and reduce the effect of nonlinearities, outer loop to specify the characteristics of the whole system. This modified plant model was approximated as a first order with time delay model. However, time delay in the first order approximated model is generally large, this affects controller synthesis such that large time delay imposes the limitations of bandwidth in controller design. In 1999 Jihong Wang and Moore apply a control strategy with the following main features (1) using acceleration feedback to improve the stability of the system; and (2) introducing time-delay minimisation and optimised null offset compensation to address the problem of time delay and dead zone. The results showed that the system performance has been much improved when compared with a conventional PID controller but the problem was in the complexity of tuning the PID parameters.

In 2001 Han Koo Lee proposed a controller with an inner pressure control loop and an outer position control loop. A PID controller with feedback linearization is used in the pressure control loop to nullify the nonlinearity arising from the compressibility of air; he used friction compensation using the multi-layered perceptron type neural network model and the reduced-order nonlinear observer, the proposed strategy was valid on a 5-port proportional valve only, and if the model of dynamic nonlinearities such as friction and compliance are identified and incorporated into the feedback linearization, further improvement of tracking accuracy may be achieved. In parallel with that, some efforts have been spent on using robust controller; this approach has been implemented on servo pneumatic positioning control by many researchers, Somyot Kaitwanidvilai. (2004). Applied evolutionary controller at the outer loop of Hamiti model of the pneumatic plant and analog controller at the inner loop, the model was approximated to second order with time delay with accuracy compared with Hamiti model. He used output error (OE) to identify the plant with batch data, and a genetic algorithm as an optimization method to optimize the parameters of the robust controller. But the approach cannot guarantee existence of the satisfied solution. In (2005, Xiang Gao) proposed a new adaptive Fuzzy-PD controller for controlling a pneumatic servo system with strong nonlinear characteristics. In order to reach high control precision of position controlling, a friction compensator is introduced into the fuzzy controller, but the disadvantages of using this method were in consuming a lot of time in carrying experiments find suitable controller parameters and scaling factors of the fuzzy controller.

There was no substantial advancement between 2006-2008, in 2008 Yi-Chang Tsai proposed a multiple-surface sliding controller (MSSC) for pneumatic servo systems with variable payload and mismatched uncertainties. The closed loop system stability is proved to be asymptotically stable by using the Lyapunov method, and experimental results showed that the proposed controller can give good tracking performance regardless of the uncertainties and time-varying payload. The proposed controller unfortunately was complex with a lot of assumptions, not preferred commercially and requiring expert control engineers to deal with.

In 2009 Saleem et.al proposed a new methodology to identify and control servo pneumatic systems in a real time environment. He used the recursive least squares (RLS) algorithm based on the auto-regressive moving-average (ARMA) model to identify the transfer function of the system with a mixed reality environment (MRE). Advantages of the proposed method include high accuracy in the identified system, low cost, and the method showed good performance in tracking the demand positions of multiple profiles with different widths. After that Zhao ting used the extended kalman filter (EKF) to estimate friction, and then put it to the control input for friction compensation, the experimental results demonstrated that the system with friction compensation has better dynamic and statistic characteristics, and the system output can effectively track input. In 2010 Yu Xiaolin carried out a comparative study between traditional PID controller and the structure of the Neural Network model adaptive controller. The results showed neural network model reference self adaptive control of the dynamic performance is better than the incremental PID algorithm, and also gives a good position to control the outcome, which shows that the neural network model reference self adaptive control of pneumatic servo system has a strong tracking ability and good robustness of anti-parameter disturbance.

2.4.2 PID-based control

The PID controller is the most commonly used controller strategy in the process control industry (Åström and Hägglund, 1995). Its widespread use is attributed to its simple structure and robust performance over a wide range of operating conditions (Gaing, 2004). PID control is implemented as either stand-alone control, or on DCS, SCADA and PLC control systems. The popularity and widespread use of PID control in the process control industry necessitates a detailed discussion on the fundamental theory that underpins this type of three-term process control. The dynamics associated with each control mode will also be discussed and the advantages and shortcomings associated with each type of control will also be given.

2.4.2.1 Proportional control

Proportional control is defined as the control action that occurs in direct proportion with the system error. The output of a proportional controller varies proportionally to the system error according to (1)

$$u_p(t) = K_c e(t) + b \qquad (2.1)$$

Where: u_p (t) is the controller output, e (t) is the error, b is the controller bias and K_c is the controller gain (referred to as the proportional gain). Proportional control action responds to only the present error. For a small value of proportional gain, a large error yields a small corrective control action. Conversely, a large proportional gain will result in a small error and hence a large control signal. The controller bias is necessary in order to ensure that a minimum control action is always present in the control loop.

2.4.2.2 Integral control

Integral control is used in systems where proportional control alone is not capable of reducing the steady-state error within acceptable bounds. Its primary effect on a process control system is to permanently attempt to gradually eliminate the error. The action of the integral controller is based on the principle that the control action should exist as long as the error is different from zero, and it has the tendency to gradually reduce the error to zero. The integrator control signal (u_i (t)) is proportional to the duration of the error and is given by:

$$u_t(t) = \frac{K_c}{T_i}\int_{t_i}^{t_f} e(t)dt = K_i \int_{t_i}^{t_f} e(t)dt \qquad (2.2)$$

T_i is the integral time constant, Kc is the proportional gain, Kc /Ti = Ki is the gain of the integral controller, e(t) is the instantaneous error signal and the limits i t and f t represent the start and end of the error, respectively. The smaller the integral time constant, the more often the proportional control action is repeated, therefore resulting in greater integral contribution toward the control signal. For a large integral time constant, the integral action is reduced. Integral control can be seen as continuously looking at the total past history of the error by continuously integrating the area under the error curve and reducing any offset. The greater the error signals the larger the correcting action from the integral controller will be.

2.4.2.3 Derivative control

In linear proportional control the strength of the control action is directly proportional to the magnitude of the error signal and P-action becomes assertive only when a significant error has occurred. The integral controller performs corrective action for as long as an error is present but its gradual ramp shaped response means that significant time expires before it produces a sizeable control response. Both these control modes are incapable of responding to the rate of change of the error signal. D-control action positively enhances system closed-loop stability (Åström and Hägglund, 1995). When operating in the forward path, the derivative controller responds to the rate at which system error changes according to:

$$u_d(t) = K_c T_d \frac{de(t)}{dt} = K_d \frac{de(t)}{dt} \qquad (2.3)$$

Where: $\frac{K_c}{T_d}$ =K_d is the derivative gain, T_d denotes the derivative time constant and $\frac{de(t)}{dt}$ = De(t) is the rate of change of the error feedback signal. It is obvious that D-action is only present when the error is changing; for any static error the contribution of the D-controller will be zero. Derivative action on its own will therefore allow uncontrolled steady-state errors. It is for this reason that derivative control is usually combined with either P-control or PI control. Another shortcoming of the D-controller is its sensitivity. The D-controller can be regarded as a high-pass filter that is sensitive to set-point changes and process noise when operating in the forward path (Lipták, 1995). To reduce this sensitivity, it is quite common to find the D-controller operating in the feedback loop enabling it to act on the feedback signal.

2.4.2.4 PID Algorithms

The transfer functions for PID algorithms are classified as follows: standard non-interacting:

$$\frac{U(s)}{E(s)} = K_c\left[1 + \frac{1}{T_I s} + T_d s\right] + b \tag{2.4}$$

Series interacting:

$$\frac{U(s)}{E(s)} = K_c\left[(1 + \frac{1}{T_I s}) + (1 + T_d s)\right] + b \tag{2.5}$$

And parallel non-interacting PID:

$$\frac{U(s)}{E(s)} = \left(K_c + \frac{K_i}{s} + K_d s\right) + b \tag{2.6}$$

2.4.3 PID Tuning

The dynamical nature of process control loops leads to changes of operating conditions within the loop, and hence loop performance. Changes in system performance may be attributed to the presence of process nonlinearities within the control channel, production strategy changes, modifications to the properties of raw materials, and changes over equipment maintenance cycles (Pomerleau and Poulin, 1996). Given these dynamical conditions, loop tuning is necessary to ensure the continued satisfactory performance of the control loop.

The goal of PID controller tuning is to determine parameters that meet closed loop system performance specifications, and the robust performance of the control loop over wide range of operating conditions should also be ensured. Practically, it is often difficult to simultaneously achieve all of these desirable qualities. For example, if the PID controller is adjusted to provide better transient response to set point change, it usually results in a sluggish response when under disturbance conditions. On the other hand, if the control system is made robust to disturbance by choosing conservative values for the PID controller, it may result in a slow closed loop response to a set point change. A number of tuning techniques that take into consideration the nature of the dynamics present within a process control loop have been proposed (Ziegler and Nichols, 1942; Cohen and Coon, 1953; Åström and Hägglund, 1984; De Paor and O'Malley, 1989; Zhuang and Atherton, 1993; Venkatashankar and Chidambaram, 1994; Poulin and Pomerleau, 1996; Huang and Chen, 1996). All these methods are based upon the dynamical behaviour of the system under either open-loop or closed-loop conditions. These tuning methods are discussed in the following sections.

2.4.3.1 Ziegler-Nichols Tuning

The earliest known and most popular tuning methodology was proposed by Ziegler and Nichols (ZN) in 1942 (Åström and Hägglund, 2004). They proposed the closed-loop (or ultimate sensitivity) method and the open-loop (or process reaction curve) method. The ZN tuning rules has a serious shortcoming in that it uses insufficient process information to determine the tuning parameters (Åström and Hägglund, 2004). This disadvantage leads to system performances that have poor robustness (Åström and Hägglund, 2004). The Ziegler-Nichols tuning method is based on the determination of processes inherent characteristics such as the process gain (K_p), process time constant (T_p) and process dead time (L_p). These characteristics are used to determine the controller tuning parameters. Although the Ziegler-Nichols methods attempt to yield optimum settings, the only criterion stated is that the response has a decay ratio of quarter (Ziegler and Nichols, 1942). This is viewed as a shortcoming because a controller tuned with this criterion may not be at its optimal setting (Lipták, 1995).

2.4.3.2 Cohen-Coon tuning (Open-loop tuning)

The ZN method was designed for a process that cannot regulate itself. To account for self regulation, Cohen-Coon (CC) introduced the self-regulation index or controllability ratio given:

$$\varepsilon = \frac{L_p}{T_p} \tag{2.7}$$

L_p refers to the process dead time and T_p denotes the process time constant. This method is based on a first-order-plus-dead-time (FOPDT) process models.

Comparison between ZN and CC Tuning

A fundamental difference between the ZN and CC methods is as follows: The ZN method associates the integral and derivative constants completely with the process dead-time, whereas the CC method adjusts the integral and derivative time constants according to the particular relationship between the process dead time and the process time constant. For both methods, the controller gain is a function of this relationship. Since processes having different controllability ratios experience different dynamic behaviors, the Cohen- Coon method may perform better than the Ziegler-Nichols method (Lipták, 1995). For example, for dead-time dominant processes i.e. processes having a large controllability ratio, the derivative time constant tends towards zero according to the Cohen-Coon tuning formulae. This is reasonable since the derivative action should not be used when the process contains large process time lag (Åström and Hägglund, 2004, Hägglund, 1992). The method does suffer from the decay ratio being too small. This results in closed-loop systems that are characterized by low damping and high sensitivity (Åström, 1995). Furthermore, the tuning formula tends to produce a very oscillatory set-point change closed-loop response because it was derived to give a quarter wave decay ratio following a load disturbance response (Hang *et al.,* 1991).

2.4.3.3 Åström - Hägglund Gain and Phase Method (Closed-Loop Method)

The tuning method proposed by Åström- Hägglund (1984) is based on the idea of moving the critical point on the process Nyquist curve to a given position. Åström and Hägglund suggested that this point be located at unity gain and at a phase of (ϕ_m $-180°$) on the Nyquist plot, where ϕ_m denotes the desired phase margin and A_m represents the desired gain margin. The phase and gain margins of a control system are a measure of closeness of the polar plot of the system to the ($-1 + j0$) point. For a system to be stable both the phase and gain margins must be positive. Negative margins indicate instability (Ogata, 1970). For satisfactory performance, the phase margin should be between 30° and 60°, and the gain margin should be greater than 6 dB (Ogata, 1970). The phase margin is that amount of additional phase lag at the gain crossover frequency (ω_c) required to bring the system to the verge of instability, where ω_c is defined as the frequency at which $G(jw)$ (the magnitude of the open-loop transfer function) is unity. The phase margin (ϕ_m) is 180° plus the phase angle ϕ of the open-loop transfer function at the gain crossover frequency.

A fundamental weakness in the ZN closed-loop method is that the method relies on trial and error adjustments to set the ultimate gain and ultimate period. To overcome this weakness, Åström Hägglund (1984) proposed their gain and phase method for determining specific points on the Nyquist curve to assist in determining controller pre-tuning parameters. Their approach is based on the use of a simple relay in series with the process. When the switch is in position two, the PID controller is disconnected from the closed-loop and is replaced by the relay. This mode is generally considered a "*pre-tuning*" phase where specific dynamics of the process are determined in the closed-loop.

2.5 Cascade Control structure

2.5.1 Background

One of the most important control structures that is widely applied in industry, namely Cascade Control, which still the subject of concern in order to find methodologies that improve the performance of the overall control system design. Cascade control structure can widely be found in systems with load disturbances in order to provide an effective disturbance rejection, and improve the system dynamics. (Antonio Visioli, Practical PID).

Cascade Control (CC), which was first introduced many years ago by Franks and Workey, which is one of the most important methodologies that can be used to improve the system performance, particularly in the presence of disturbances. Ibrahim Kaya introduce an improved cascade control strategy for a low order Plant transfer function and used an optimization method for the tuning the inner PI parameters and outer PID parameters but the disadvantages of this method were: it applicable on low order transfer functions and some constrained must be applied in the optimization criteria in order to optimize the tuning parameters. Sihai Song applied in his research auto-tuning method for the cascade control system he used a model-matching algorithm is to obtain the PID control parameters for overall system performance, Tan proposed a method to carry out the entire tuning process in one experiment, but the experiment requires prior information of the process. Furthermore, the ultimate frequency used for outer loop design is based on initial ultimate frequency without considering changes in inner loop control parameters.

From the previous mentioned in servo pneumatic background chapter, the first who attempted to use cascaded controller in pneumatic drives was Hamiti in 1996, where he proposed a cascaded analog P controller with a digital PI controller, the tuning approach that he adopted in his work was Chien-Hrones-Reswick method which based on minimizing criteria, and it was apply for only three parameters starting with trial and error guess.

2.5.2 Cascade control configuration

Cascade control structure consists of two nested loops configuration so that the fast dynamics of the process is separated as much as possible from the slow dynamics, each loop has its associated PID controller, the inner loop which is the fast one called the secondary controller whereas the controller of the outer loop as the primary controller. The typical cascade control structure is shown below.

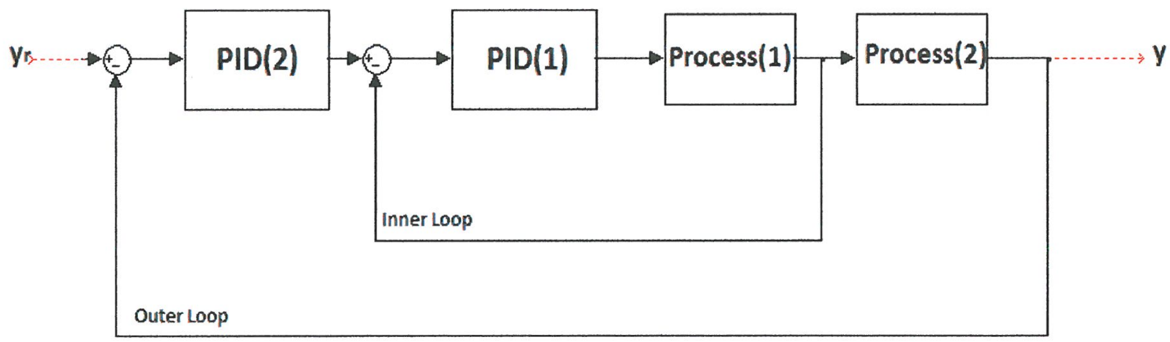

Figure 2.1: Cascade Control structure

The design of the overall cascade control system is usually performed by tuning the inner loop, and then the main PID controller is designed on the basis of determining the parameter of the secondary PID.

2.5.3 Cascade control tuning

In general it appears that the tuning procedure is performed sequentially and for several times which consumes a lot of time when it compares with single loop control system; first, the outer loop controller is put on manual and the inner loop controller is tuned. Subsequently, the inner loop controller is commissioned and the outer loop controller is tuned to complete the tuning process. If the control performance achieved is unsatisfactory, the entire sequence must be repeated. Therefore it is a challenge to be able to provide a potential tuning that guarantee a simultaneous tuning for both controllers.

2.6 Summary and conclusion

This chapter has presented an overview of current developments in the area of servo-pneumatic modelling, control and tuning methodologies. The review also has pointed out the cascade control strategy that has been employed in servo-electric systems and has received low attention for servo-pneumatic systems. In general, the previous issues can be summarized into the following:

(1) Lack of an optimal strategy for controller tuning and optimization.

(2) Lack of working environment which allows the implementation of different control methodologies for servo-pneumatic systems and facilitate the procedure of tuning and optimisation.

This lack of such tools, capabilities and methodologies which required-depth knowledge from the designers and user to be able to deal with such systems make it a challenge for this research to focus on developing and applying HIL concept for design and control of servo-pneumatic and deal with such highly non-linear systems easily.

Chapter 3

Identification and Control of Pneumatic systems

3.1 Introduction

Classical approach to position control of pneumatic drives is based on the linearization of the nonlinear system dynamics around the desired set point, and such models are valid only for small deviations around an operating point. So it is a very challenging task to get an accurate mathematical model for describing pneumatic servo drives behavior.

In practical applications, one often does not have available values for the model parameters and/or part of the model structure. Therefore, one tries to obtain these parameters and/or structural elements using experimental data from the real process. Researchers developed various parameter-identification methods and applied them to many engineering systems.

3.2 Background

System identification techniques have been used in many fields for building accurate mathematical models of dynamic systems, based on observed input-output data. However, most of the identification methods, such as those based on least mean-squares or maximum likelihood estimates, are in essence gradient-guided local search techniques. They require a smooth search space or a differentiable error energy function. These conventional approaches can thus easily fail in obtaining the global optimum if the multi model search space is not differentiable or the performance index is not "well behaved" in practice.

Carducci et al. (2006). presented the identification of viscous friction coefficients for a pneumatic system model using optimization methods. Their work focused on developing a mathematical model of a pneumatic actuator driven by two on/off two-way valves based on the identified friction parameters. They reported that pneumatic systems are not only nonlinear, but also involve several tuning parameters. Daw et al. (2003). employed a genetic algorithm in order to identify the dynamic friction parameters along the pneumatic cylinder. The evaluation function has been formed using the statistical expectation of the mean squared error (MSE). Further study has been conducted by Wang et al. (2004) to improve the convergence rate and the accuracy of the algorithm. Their work concentrated on measuring the friction parameters of the cylinder rather than a complete system. Ziaei and Sepehri, (2000). discussed some practical issues concerning the identification of electro-hydraulic actuators using discrete-time linear models. They considered a discrete-time linear model and estimated its unknown parameters. Other researchers used neural networks to identify system parameters. Angerer et al. (2000). used a structured recurrent neural network to identify the physically relevant parameters and nonlinear characteristics of a nonlinear two-mass system with friction and backlash. None of the above researchers identified the transfer function of a servo-pneumatic system. Rather, they concentrated in their research on identifying certain nonlinear parameters within their plant model.

3.3 Identification using Particle Swarm Optimization (PSO)

In this work, the PSO is utilized in order to identify the servo-pneumatic parameters system transfer function the system is initialized with random particles and the algorithm searches for optimal particle by updating generations. The swarm adjective is the best to describe the behaviour of the particles that is because the particles do interact with the surrounding environment, and in the same time transmit data between each other. The updating procedure depends on the velocity of this particle v_i and the position of this particlep_i.

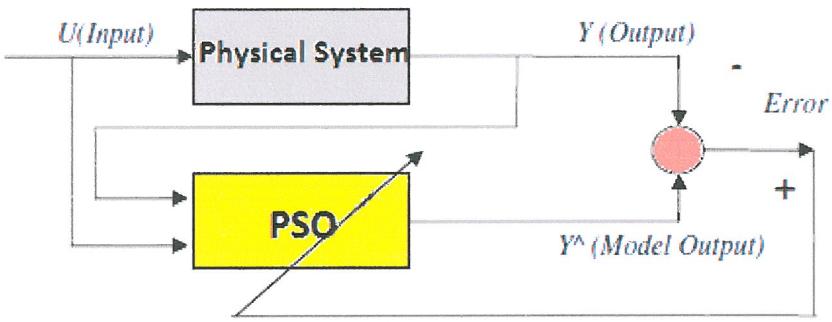

Figure 3.1: Block diagram of PSO identification model.

The parameters of the transfer function are updated to minimize the least square error as shown in figure 3.1, to successfully use PSO algorithm to solve the problem of system identification, the key lies in how to conduct particle coding and establish a proper fitness function.

3.3.1 Design of fitness function

The purpose of system identification is to offer a model which can fit the sample data very well, which means making the calculated system output Y_m approach the actual system output Y_a as much as possible. The closer those two values are, the better the fitting effect will be. Therefore, the following criterion function can be taken as the fitness function.

$$f = \sqrt{\sum_{i=1}^{n}(Y_m - Y_a)} \tag{3.1}$$

Where Y_m represents the model output.

3.3.2 Procedures of system identification

The main procedures of the algorithm for system identification based on PSO are as follows:

Step 1: Initialization, randomly generate initial solutions (particles) x_i of N and initial velocity v_i of N.

Assess the fitness of each particle, initialize p_i with its own position of each particle and initialize g with the particle of the minimum fitness in the swarm.

Step 2: Update the velocity and the position of each particle according to equations (3.1) and (3.2).

$$v_i^{k+1} = v_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k) \qquad (3.2)$$

$$s_i^{k+1} = s_i^k + v_i^{k+1} \qquad (3.3)$$

Where:

v_i^k = current velocity of agent I at iteration k

v_i^{k+1} = new velocity of agent I at iteration k,

c_1 = adjustable cognitive acceleration constant,

c_2 = adjustable social acceleration constant,

$Rand_{1,2}$ = random number between 0 and 1,

s_i^k = current position of agent i at iteration k,

$Pbest_i$ = personal best of agent i,

gbest = global best of population.

Step 3: Calculate fitness of the new position of each particle.

Step 4: For each particle, if the fitness of current position x_i is better than the fitness of the historically optimal position p_i, replace p_i with x_i.

Step 5: For all particles, if the fitness of currently optimal position is better than the fitness of the historically optimal position g, replace g with x_i .

Step 6: if the stopping rule of algorithm is met (e.g., specified iteration steps are achieved or the fitness value exceeds certain threshold), the algorithm stops, and output optimization result, or else return to step 2.

3.3.3 Selection of PSO parameters

To start up with PSO, certain parameters need to be defined. Selection of these parameters decides to a great extent the ability of global minimization. The maximum velocity affects the ability of escaping from local optimization and refining global optimization. The size of swarm balances the requirement of global optimization and computational cost. Initializing the values of the parameters is as per table.3.1.

Table 3.1: PSO selection parameters.

Population size	500
Number of iterations	100
Velocity constant,c1	2
Velocity constant,c2	2

3.4 ARMA models and recursive estimation algorithms

ARMA models are one of the most important and general purpose modelling processes. ARMA models contain two parts; autoregressive (AR) and moving average (MA) models. The goal is to determine ARMA (p, q), where q is the degree of numerator; i.e., the number of numerator coefficients, and p is the degree of denominator; i.e., the number of denominator coefficients. Then the values of these coefficients can be estimated.

The modelling of a system means finding its transfer function; i.e., finding the coefficients of numerator and denominator polynomials. Parameters can't be found without knowing the order of the numerator and denominator. Previous methods of modelling ARMA model assumed the order and then estimated the parameters according to this assumption. Practically, the order is unknown, so, it has to be found to estimate the correct parameters of the model.

ARMA models have a wide range of applications such as in 1-communication, 2-signal processing, 3-control systems, 4-biomedical engineering,5- image processing and compression, 6- prediction of spectrum estimation, 7- controlling of dynamic systems, and can even be used in such broad fields as stock price estimation and other business related field. In the scope of digital signal processing, an ARMA model is treated as an Infinite Impulse Response (IIR) filter.

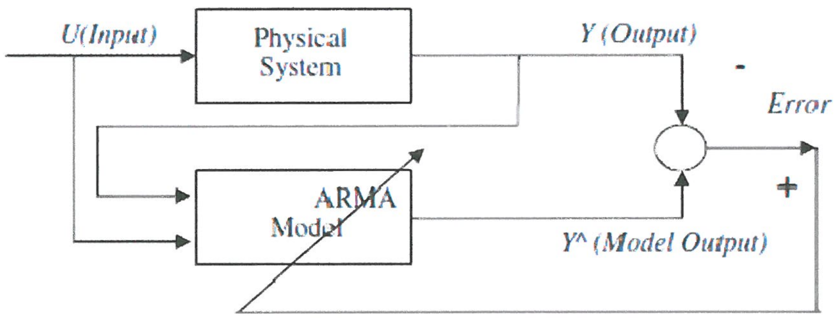

Figure 3.2: Block diagram of the auto-regressive moving-average (ARMA) identification model.

Figure 3.2 shows how the parameters of ARMA model updated to minimize the least square error.

3.4.1 Mathematical representation of ARMA model

A signal y(n) is said to be an ARMA process ARMA(p, q) if this signal is stationary and satisfies the following equation:

$$y(n) = -\sum_{i=0}^{p} a_i y(n-i) + \sum_{j=0}^{q} b_j x(n-j) \tag{3.4}$$

where y(n) is the observed data and x(n) is the input data sequence. The input x(n) is a zero-mean, stationary, and Independent and Identically Distributed (I.I.D) random variable. The coefficients a_i and b_j are the AR and MA parameters, respectively. The order of the AR part in this equation is p, while q is the order of the MA part.

The ARMA model can be written in the complex Z-domain as a transfer function of the output sequence to the input sequence in Equation (2.1):

$$H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} = \frac{(\sum_{j=0}^{q} b_j z^{-j})}{1 + (\sum_{i=1}^{p} a_i z^{-i})} \tag{3.5}$$

Where the numerator and denominator polynomials of this transfer function are as follow:

$$B(z) = \sum_{j=0}^{q} b_j z^{-j} \tag{3.6}$$

$$A(z) = \sum_{i=0}^{p} a_i z^{-i} \tag{3.7}$$

And the value of a_0 is set to 1 to normalize the transfer function of the process. In some cases even b_0 is set to 1, which is the case for all examples in this thesis. This transfer function is referred to as a casual linear shift-invariant filter. These polynomials have no common roots. The roots of B(z) are known as the zeros and their number is determined by the value of q, while the roots of A(z) are called the poles of the system and their number is related to the value of p.

3.5 HIL Concept

Generally the pneumatic systems have a considerable nonlinearities, a lot of properties such as poor damping and low stiffness which causes a lot of difficulties in design and control of servo pneumatic systems, Besides, uncertainties in system parameters make the controller design problem more challenging and using the traditional simulation software has the disadvantage of being unable to replicate the real conditions due to previous mentioned conditions. So the need of applying a new methodology that bridges the gap between the virtual and real systems is very important in servo pneumatic control field.

Applying the concept of Hardware in the Loop (HIL) has the advantages of reducing the risks of discovering an error in the last stage of design, the cost of extra parts, time saving in designing and troubleshooting any servo pneumatic system.

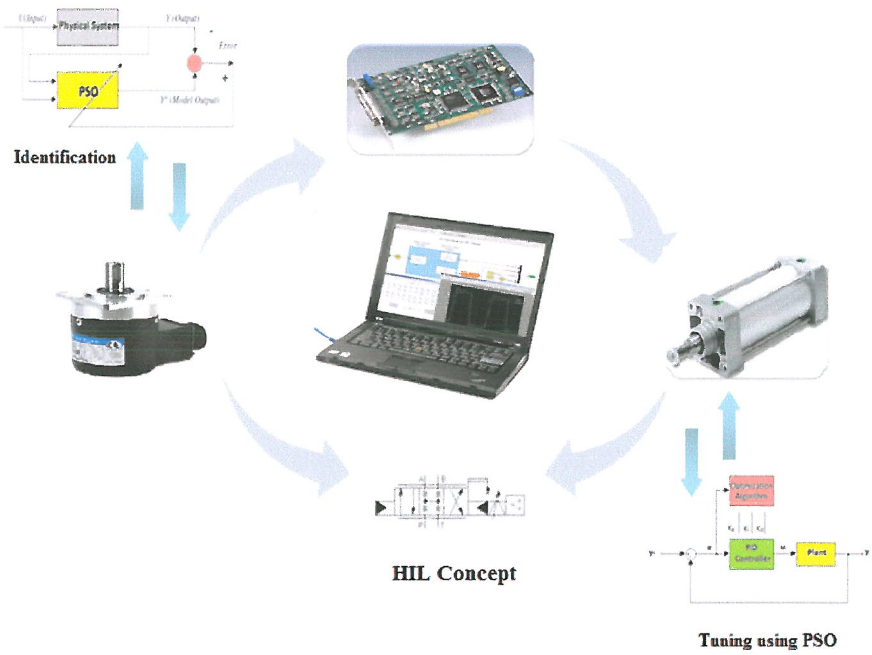

Figure 3.3: HIL with PSO for Identification and tuning servo-pneumatic system.

Generally, servo-pneumatic systems consist of a controller, actuators, and sensors. The controller generates an output to the feedback signal from the sensors and sends it to the actuator which performs an action. According to the above situation, some of the hardware components, such as the controller, can be substituted by its model and simulated in real time. The simulated component(s) can be run in conjunction with real components under the same environment. This environment can be regarded as an HIL. Figure 3.3 shows the concept of the proposed environment.

3.5.1 Control via HIL

Control systems design is a common subject for engineering students world-wide. Many tools exist to help engineers in design and simulate digital controllers, such as MATLAB and SIMULINK, but actually implementing and testing a designed controller is important as well.

Most simulation frameworks readily available for students focus on the controller itself. They aid in design of the controller's mathematical model, but do not aid in physically testing the actual implementation of the controller. The Control System Plant Simulator follows the Hardware-in-the-loop concept in that it takes the place of a physical plant.

This can reduce production costs of control systems, and make possible more realistic control systems development in academic environments where resources are more limited. This allows for the education of the next generation of control system designers and implementers in a more realistic setting.

This technology provides a way for testing control systems over the full range of operating conditions, including failure modes. Testing a control system prior to its use in a real plant can reduce the cost and the development cycle of the overall system.

Hardware-in-the-loop simulation has been used, with success, in the aerospace industry and is now emerging as a technique for testing electronic control units.

3.6 HIL in servo-pneumatic systems

The HIL is an environment whereby virtual components can be applied on real systems' components. From the control perspective, working with an HIL should include control system synthesis off-line (or under a simulation environment) and then apply the simulated model on the real system under the HIL. Off-line simulation will normally take place before moving onto the real system, where the system should be tested and the controller should be tuned or optimized. Then, the optimized virtual controller will be applied on the real system. This environment should allow the system to be controlled with different control schemes by simply replacing the "controller" component according to the application requirements. Furthermore, the HIL gives the capability to monitor the system's behaviour by observing the output signals, such as speed and position signals. These signals can be utilized to identify the real system using one of the system-identification methods.

3.7 Summary

This chapter has presented an overview of identification techniques used in servo-pneumatic field. Also in this chapter a particle swarm optimization technique is employed for the model identification of servo-pneumatic systems.

Finally HIL methodology is introduced to bridges the gap between the virtual and real systems. This environment allows the system to be controlled with different control schemes by simply replacing the "controller" component according to the application requirements.

Chapter 4
PSO for controller tuning

4.1 Introduction

This chapter presents tuning methodologies applied in the area of servo pneumatic systems and the implementation of PSO tuning methodology as an optimization technique to determine the optimal tuning parameters of PID controllers.

4.2 Tuning methodologies

4.2.1 Background

Tuning a PID controller means setting the proportional, integral and derivative values to get the best possible control for a particular process. The dynamical nature of process control loops leads to changes of operating conditions within the loop, and hence loop performance. Changes in system performance may be attributed to the presence of process nonlinearities within the control channel, production strategy changes, modifications to the properties of raw materials, and changes over equipment maintenance cycles (Poulin,et al 1996). Given these dynamical conditions, loop tuning is necessary to ensure the continued satisfactory performance of the control loop.

Many methods for tuning PID controllers have been proposed but every method has some limitations. In the context of pneumatic drives, the most popular methods for PID tuning are gain-scheduling, fuzzy logic, neural network, self-tuning and optimization techniques.

4.2.2 Gain-scheduling

One method for tuning the controller is “gain-scheduling”. This method aims to overcome the nonlinear behaviour by “scheduling” the associated control gains in response to variations in the system conditions or status. A study of a gain-scheduling method for controlling the motion of pneumatic actuators was introduced by Pu, et al., (1993). Basically, to compensate for the nonlinearity arise, for instance, from the valve, a look-up table can be built to characterize the actual performance characteristics of the control valve. As a result, the table can be used to associate the control gain constants, typically, the proportional gain term can be “scheduled” accordingly. Another approach for “gain-scheduling” is to use a “gain table”, where table entries are made automatically during system operation. Such a system could establish (or learn) appropriate controller parameters for different operating and initial conditions.

Therefore gain-scheduling provides a means of compensation for known and repeatable nonlinearities. However, one major drawback of the method is that the system design is time-consuming if the learning capabilities are not included with the control system (Wang, et al., 1999). Another drawback is that the method is extremely difficult to be employed if the dynamics of the system are not known sufficiently either through the use of learning or priori modelling. More intensive research and critique for this method can be found in (Rugh and Shamma, 2000).

4.2.3 Fuzzy Logic

The tuning of the controller gains tends to be of a “fuzzy” nature. Therefore, some researchers utilised fuzzy logic for PID tuning (de Bruijn and van Wal, 1993; Visioli, 2001; Situm, et al., 2004). With fuzzy logic, new forms of tuning techniques can be realized without the need for mathematical identification. To describe the control logic, the knowledge of human experts in the form of a rule base is used instead of mathematical equations based on linear analysis.

This approach does not require significant potential power; therefore it can be easily embedded within a digital controller. However, the process of fuzzy logic tuning is related largely on a trial and- error basis. It can be difficult to predict the learning steps and the length of time required. It may also be difficult to determine the maximum allowable gain values and they may have to be chosen through trial-and-error experimental means, which involve a tedious process.

4.2.4 Neural Network

Tuning approaches based on neural networks negate the need for detailed knowledge of the system being controlled. The network can be used to emulate the servo-pneumatic system after being trained and validated. As soon as the designed motion profile is fully defined, a feature extraction module can be used to analyse and extract the response features (e.g. overshoot, damping ratio, and steady state error). If the chosen criteria are not met, these features will then be fed to the rule-based parameter adjustment mechanism (PAM). New values can then be assigned to the associated control parameters by the PAM, based on the characteristic values of the response features, the pre-defined rules and the specified performance requirements. Iteration continues until the specified performance achieved. Fujiwara and Matsukuma (Fujiwara, et al., 1995; Matsukuma, et al., 1997) present neural network based tuning PID controller.

Tuning using neural networks has a number of advantages over conventional, manually based, tuning practice. The method requires only a representative input-output data set, and therefore no detailed models need to be derived or used. Furthermore, tuning is achieved off-line, thereby reducing potential for accidental damage to the drive system during the commissioning phase of the system. However, this method suffer from two main disadvantages; firstly, the quality of the emulated results, i.e. how closely and with what level of robustness it can mimic the actual response characteristics, and secondly, the length of time involved before a converged result can be obtained.

4.2.5 Self-tuning

Self-tuning is also referred to as automatic tuning. The term automatic tuning covers a variety of concepts such as adaptive tuning, self tuning, tune on demand, and pre tuning. Hardie (1988) in his article clarified some of the terminology related to this issue. However, self-tuning is not just a matter of automating the tuning process which typically carried out manually. With self tuning more mathematical mechanisms are employed which may incorporate some methods which are difficult to realise by manual operation. Two approaches currently adopted for process control which may be widely applicable for tuning pneumatic drives are (1) the pattern-recognition method (Kraus and Myron, 1984; Swiniarski, 1990; Xing, et al., 2001), and (2) the relay auto tuner (Hang and Sin, 1991; 1992; Liu, et al., 2004).

With the pattern recognition approach, the response to step changes or disturbances is observed, and the controller parameters are adjusted based on the response pattern. The procedure imitates the procedure used by an expert. It is necessary for reasonable controller setting to be known and selected prior to the self-tuning process. On the other hand, the relay auto-tuner is based on the belief that knowledge of the ultimate frequency (i.e. frequency where the phase lag of the open loop system is 180 degrees) is the critical information for tuning the controller. Many works are dedicated to the PID self-tuning for pneumatic systems (Shih and Huang, 1992; Shih, et al., 1994; Fok and Ong, 1999).

4.2.6 Optimization Techniques

Genetic algorithms (GA) are general-purpose search procedures, optimization methods, or learning mechanisms based on Darwinian principles of biological evolution. When provided with a suitable objective function that evaluates the performance of the control systems, using genetic algorithms can readily optimize control parameters for servo pneumatic systems. Applying genetic algorithms to optimize the control gains of a three-loop controller for a pneumatic servo cylinder drive was presented by (Yong et al., 1998). The three loops controller has position, velocity, and acceleration feedback gain, all of which have formerly been manually tuned. A general procedure for optimizing the control parameters of servo pneumatic systems by using genetic algorithms was presented. The optimized gains were confirmed by a plot of the fitness distribution defined in this study, which represented the control performance of the system in a given gain space.

4.3 Particle swarm optimization

Particle swarm optimization (PSO) is a computational algorithm technique based on swarm intelligence proposed by Eberhart and Kennedy (1995). It was inspired from the computer simulation of the social behaviour of bird flocking by Reynolds (1987). Reynolds used computer graphics to model complicated flocking behaviour of birds.

He was mainly interested in simulating the flight patterns of birds for visual computer simulation purposes, observing that the flock appears to be under central control. Reynolds proceeded to model his flocks using three simple rules, namely collision avoidance, velocity matching and flock centering . Using these rules Reynolds showed how the behaviour of each agent inside the flock can be modeled with simple vectors. This characteristic is one of the basic concepts of PSO.

Boyd and Recharson (1985) examined the decision making process of human beings and developed the concept of individual learning and culture transmission. According to their examination, people utilize two important kinds of information in decision-making processes, namely:

Their own experience: They have tried the choices and know which state has been better so far, and they know how good it was and;

Other people's experiences: They have knowledge of how the other agents around them have performed. In other words, they know which choices their neighbors have found positive so far and how positive the best pattern of choice was. Each agent's decisions are based upon his own experience and other people's experience. This characteristic is another basic concept of PSO.

Eberhart and Kennedy (1995) incorporated these ideas into the development of their PSO method and invented simple velocity and position algorithms that mimic natural swarm behavior. In PSO, a set of randomly generated agents propagate in the design space towards the optimal solution over a number of iterations. Each agent has a memory of its best position and the swarm's best solution.

A swarm consists of individuals, called particles; each particle represents a candidate solution to the optimization problem. In a PSO system, particles fly around in a multidimensional search space adjusting its position according to its own experience and the experience of its neighboring particle. The performance of each particle is measured using a fitness function that varies depending on the optimization problem.

The use of PSO has been reported in many of the recent works in different fields; it has been regarded as a promising optimization algorithm due to its simplicity, low computational cost and good performance. But nobody so far apply it on tuning of servo-pneumatic controllers. In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods. Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications.

4.3.1 Applications of PSO

PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas.

The various application areas of Particle Swarm Optimization include:

- Power Systems operations and control.
- Job Scheduling problems.
- Modeling optimized parameters.
- Multi-objective optimization problems.
- Image processing and Pattern recognition problems.

4.3.2 Description of the PSO tuning methodology

Consider Figure 4.1, which represents a 3-dimensional search space being traversed by intelligent agent *"w1"*. Each dimension's space represents a potential optimal value for K_p, K_i and K_d. The position of a*gent "w1"* determines the controller's tuning parameters. Modification of an agent's position is realized by responding to velocity and position information according to (4.1) and (4.2). For PID control each agent is given an initial position within a within a 3- D search space.

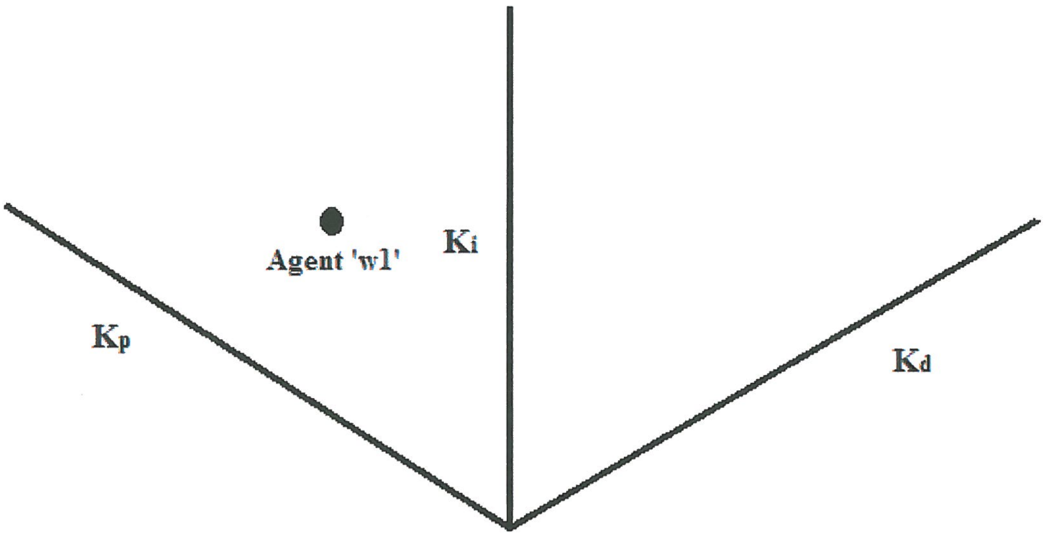

Figure 4.1: Position of swarm agent within a 3-D search space

The algorithm proposed by Eberhart and Kennedy (1995) uses a 1-D approach for searching within the solution space. In this research the PSO algorithm will be applied to a 3-D solution space in search of optimal tuning parameters PID control. Consider position $s_{i,n}$. of the i-th particle as it traverses a n-dimensional search space.

The previous best position for this i-th particle is recorded and represented as $pbest_{i,n}$. The best performing particle among the swarm population is denoted as $gbest_{i,n}$ and the velocity of each particle within the n-th dimension is represented as $v_{i,n}$, . The new velocity and position for each particle can be calculated from its current velocity and distance with (4.1) and (4.2), respectively.

4.3.3 Steps in implementing the PSO technique.

Figure 4.2 illustrates the general flowchart for the PSO technique. The sequence can be described as follows:

Step 1: Generation of initial conditions of each agent.

Initial searching points (s_i^0) and the velocities (v_i^0) of each agent are usually generated randomly within the allowable range. The current searching point is set to p best for each agent. The best-evaluated value of p best is set to gbest and the agent number with the best value is stored.

Step 2: Evaluation of searching point of each agent.

The objective function is calculated for each agent. If the value is better than the current p best value of the agent, then p best is replaced by the current value. If the best value of p best is better than the current gbest, the gbest value is replaced by the best value and the agent number with the best value is stored.

Step 3: Modification of each searching point.

Step 4: Checking to exit condition.

The terminating criterion is checked to determine whether it has been achieved. If the terminating criterion is not met then the process is repeated from Step 1, otherwise the algorithm is stopped.

PSO tuning is implemented *offline* and then *online*. In *offline tuning*, the plant model is identified and tuning was then performed under simulated conditions within the MATLAB- Simulink environment. Then the tuned controller was applied to real system through HIL. Results will be discussed in later chapter.

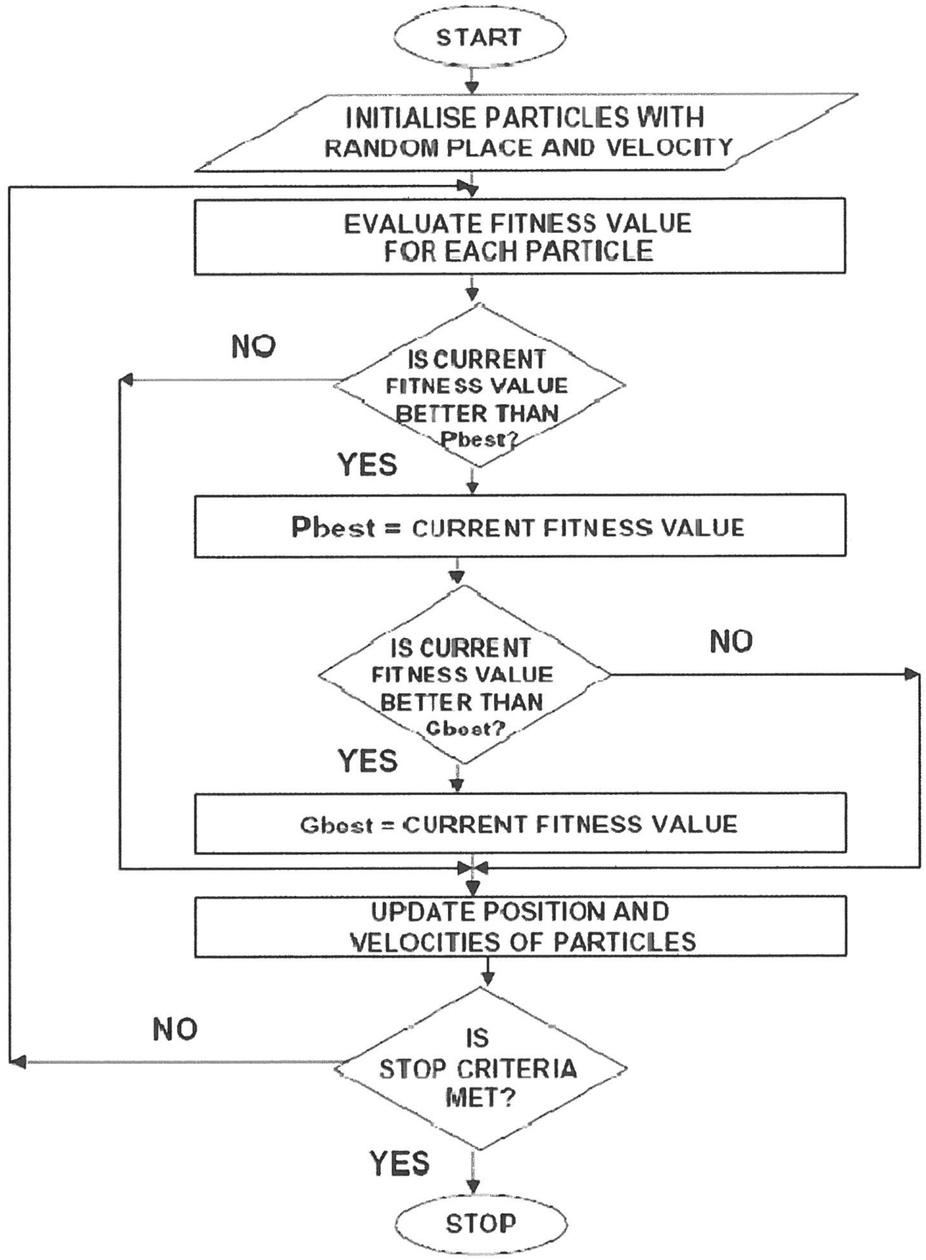

Figure 4.2: PID tuning using PSO

Figure 4.3 illustrates the block diagram of the PSO tuning algorithm for the servo pneumatic system used in this study.

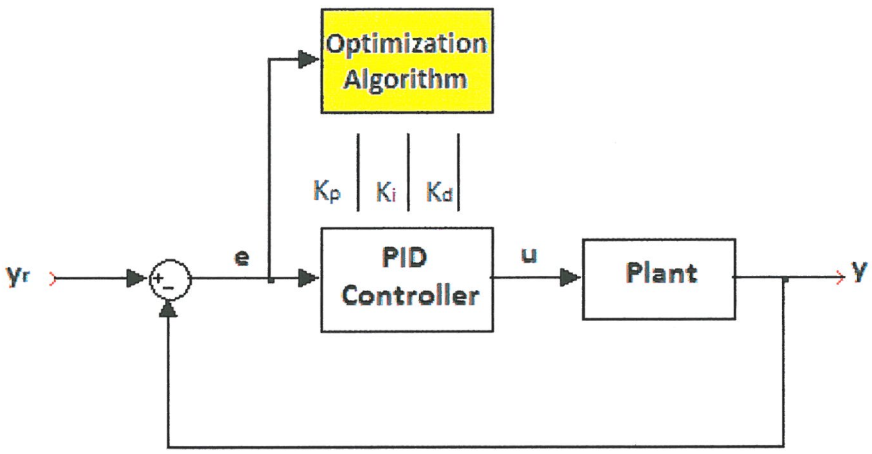

Figure 4.3: PSO block diagram

4.3.4 Selection of PSO parameters

To start up with PSO, certain parameters are needed to be defined. Selection of these parameters decides to a great extent the ability of global minimization. The maximum velocity affects the ability of escaping from local optimization and refining global optimization. The size of swarm balances the requirement of global optimization and computational cost. Initializing the values of the parameters is as per table 4.1.

Table 4.1: PSO selection parameters

Population size	200
Number of iterations	5000
Velocity constant,c1	2
Velocity constant,c2	2

4.3.5 Performance indices for the PSO algorithm

The objective function considered is based on the error criterion. The performance of a controller is best evaluated in terms of error criterion. In this proposed work, controller's performance is evaluated in terms of least square error as previously mentioned in chapter three.

4.3.6 Termination Criteria

Termination of optimization algorithm can take place either when the maximum number of iterations gets over or with the attainment of satisfactory fitness value. Fitness value, in this case is nothing but reciprocal of the magnitude of the objective function, since we consider for a minimization of objective function. In this paper the termination criteria is considered to be the attainment of satisfactory fitness value which occurs with the maximum number of iterations as 5000.

4.4 Attractive features of PSO in PID tuning

- Fast convergence: The PSO is influenced by the simulation of social behaviour where each individual benefits from its history and its interactions with other agents within the population. This sharing of knowledge helps facilitates faster convergence to an optimal solution.
- Simple operating algorithm: The use of simple mathematical operators facilitates a faster computational time and makes the algorithm suitable for determining tuning parameters under high-speed dynamical conditions.
- PSO relies on a memory based progression, in which the previous solutions are remembered and is continually improved upon until convergence is reached.

- Traditional tuning methods require further fine tuning to improve control performance.

4.5 Comparison between PSO and other tuning methods

Tuning using ZN: significant drawback of this tuning method is that the ultimate gain has to be determined through trial and error and the system has to be driven to its stability limits. Another disadvantage is that when the process is unknown, the amplitudes of the oscillations can become excessive when using trial and error to determine the ultimate gain of the system. This could lead to unsafe plant conditions.

The tuning method proposed by Åström- Hägglund which based on the idea of moving the critical point on the process Nyquist curve to a given position, suffer from a large time delay since this method may result in a very oscillatory closed loop response.

GA method which depends on genetic operators one of the most significant drawback that operate according to a sharing mechanism during their evolutionary process whereby the previous solutions are potentially lost, The high degree of stochasticity that the GA suffers from means that there is a strong possibility of the algorithm yielding poor results over a small number of iterations which consumes lot of time for optimization.

For these reasons the research proposes a simple methodology based on the PSO computational algorithm, for determining PID tuning parameters.

The control performance enhanced using PSO, this attributes to the following reasons:

- PSO relies on a memory based progression, in which the previous solutions are remembered and is continually improved upon until convergence is reached.

- Simple operating algorithm with minimum number of parameters facilitates a faster computational time and makes the algorithm suitable for determining tuning parameters under high-speed dynamical conditions

4.6 Summary

This chapter presented a brief overview of PID tuning methods including: gain-scheduling, fuzzy logic, neural network, self-tuning techniques and PSO. With special emphasis was given to the PSO approach as a new tuning approach applied for optimizing PID controllers.

The study explained the PSO computational algorithm that will be used for PID control. The objective was to improve the performance of servo pneumatic systems that experience poor control behaviour when tuned using conventional tuning methodologies.

Chapter 5

Results and analysis

5.1 Introduction

This Chapter presents the results obtained from the identification and control of servo-pneumatic system. The system model has been identified using PSO technique. Implementing PSO was employed as a tuning methodology off-line, and then applies the optimized control parameters were applied on-line. Moreover, a comparative simulation and experimental analyses have been carried out between single PID and cascaded PID structures.

5.2 Experimental Setup

The pneumatic unit consists of a pneumatic power supply, which includes a compressor with an air conditioning unit, lubricating unit, and manifold. The servo-pneumatic valve is an 1/8-inch port and has an operating voltage from zero to 10 V, while the pneumatic actuator has a piston diameter of 27 mm, rod diameter of 8 mm, and stroke length of 100 mm. A rotary potentiometer is used for displacement measurement. It has a resistance range from 2 to 12 kΩ, with a voltage source of 10 V, and is fixed on the cylinder to provide position feedback.

All signals were sent to a computer via a National Instruments (NI) DAQ card 1036E through an A/D converter terminal. The DAQ card has 16 analog inputs, two analog outputs, a sampling rate of 200 kS/s, and an input voltage range of ±10 V. The final signals are used to activate an analog input block in MATLAB's real-time windows target (rtwt). The input signal of the valve is the controlled voltage from the analog output block of MATLAB (rtwt) to the D/A converter of the DAQ card and, finally, to the servo valve.

The change of input voltage from zero to 5 V produces the change of air flow through the valve to control the motion of the piston of pneumatic actuator. Figure 5.1 shows a photo of the described system.

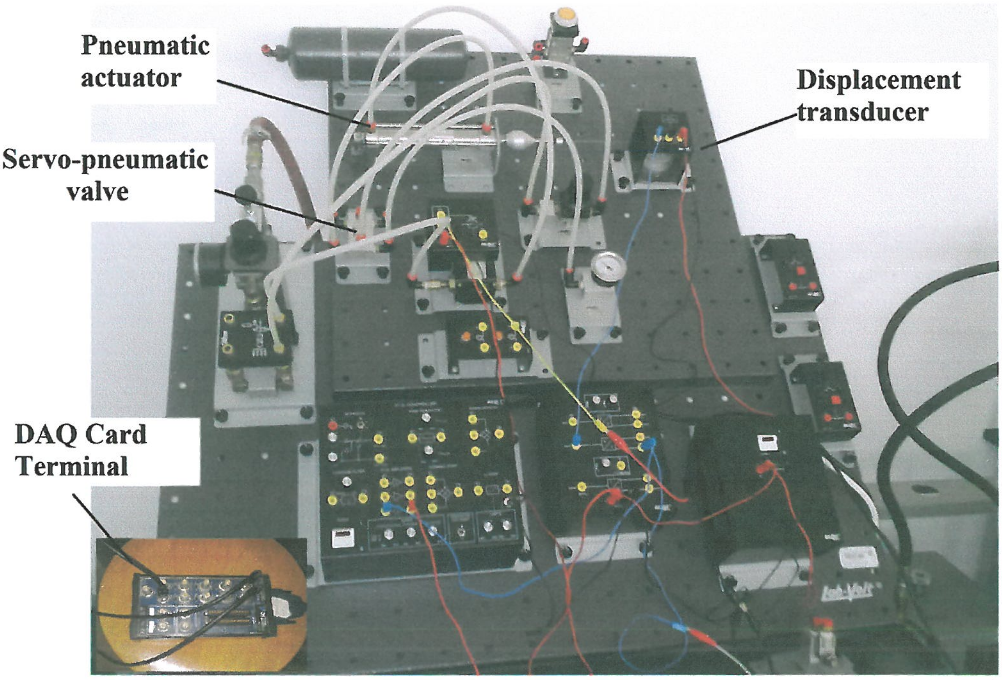

Figure 5.1: Pneumatic test setup.

5.3 Identification using ARMA model

The most basic relationship between the input and output of a system is the linear difference equation (Ljung, 1999) given by:

$$y(t) + a_1 y(t-1) + \cdots + a_n y(t-n) = b_1 u(t-1) + \cdots + b_m u(t-m) \quad \text{......} \; .5.1$$

where y(t) is the model output at time t and [y(t-1),...,y(t-n), u(t-1),...,u(t-m)] are past observed data,(Tavakolpour,et al, 2010). Since the observed data would be collected from input output data by running the system as open loop. A useful way to visualize Equation 5.1 is to view it using a backshift operator z^{-1} as defined by:

$$y(t) = \frac{B(z^{-1})}{A(z^{-1})}u(t) \qquad 5.2$$

Where A (z^{-1}) and B (z^{-1}) are polynomials with associated parameters:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}$$

$$B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}$$

The corresponding transfer function of the system and can be represented as follows:

$$\mathrm{G}(\mathrm{z}^{-1}) = \frac{\mathrm{B}(\mathrm{z}^{-1})}{\mathrm{A}(\mathrm{z}^{-1})} = \frac{\mathrm{b}_1\mathrm{z}^{-1}+\mathrm{b}_2\mathrm{z}^{-2}+\cdots+\mathrm{b}_m\mathrm{z}^{-m}}{1+\mathrm{a}_1\mathrm{z}^{-1}+\mathrm{a}_2\mathrm{z}^{-2}+\cdots+\mathrm{a}_n\mathrm{z}^{-n}} \qquad 5.3$$

In Saleem, et al (2009) they used ARMA model to identify the servo-pneumatic model parameters, as a first step they run the system as open loop and the required input-output data were collected, a different experiments were conducted on the test rig outlined above, then using a sampling time of 1ms all obtained data were employed in ARMA model to finally end with plant model transfer function.

A set of experiments using third, fourth, and fifth orders were conducted to show the effect of the identification orders. Mean square error was employed as comparison criteria between the predicted outputs of the different models order.

Figure 5.2 show the results of the on-line actual output and the fourth order predicted model with one step prediction with minimum square error among all models that had been depicted in Saleem, et al (2009) work.

The identified transfer function using ARMA model and RLS algorithm which was adopted for the servo-pneumatic system controller design:

$$G_P = \frac{0.0070699z^4 - 0.017738z^3 + 0.019649z^2 + 0.00027783z}{z^4 - 1.9827z^3 + 1.2989z^2 - 0.43504z + 0.12436}$$

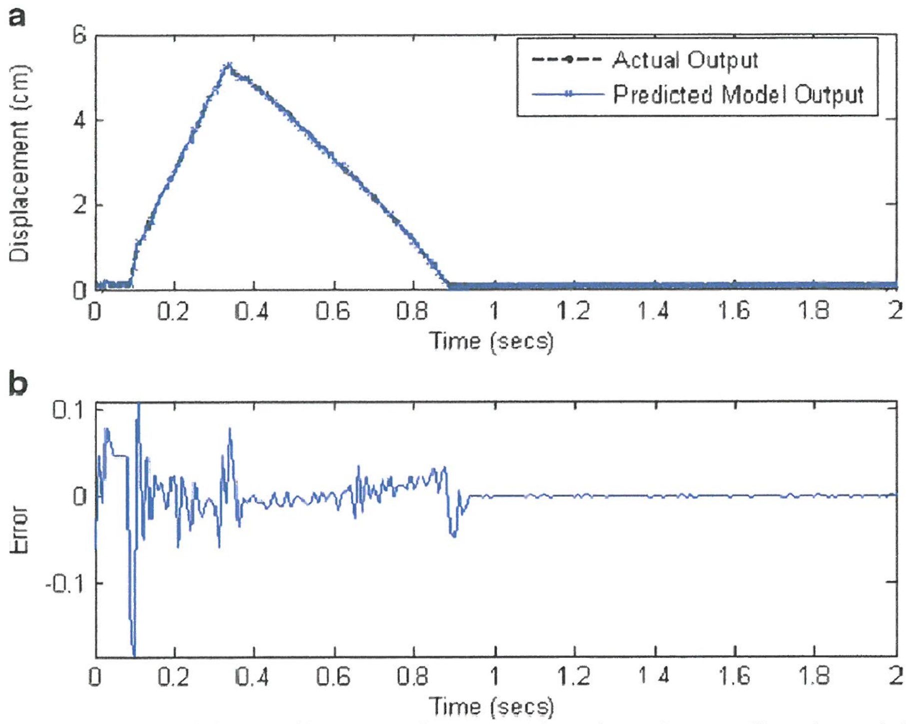

Figure 5.2: Variation of the on-line actual output, fourth-order predicted model, and the error verses time

5.4 Identification using PSO

Table 5.1 shows the PSO parameters used in the experiment. So, here PSO is used for identifying parameters ai and bj where each solution has 9 values for ai and bj. Several trials were conducted under the following conditions: Upper bound of initialization (u_b) = 10, Lower bound of initialization (l_b) = -5. All tests were conducted using the control loop shown in Figure 5.3.

Table 5.1: PSO selection parameters.

Population size	500
Number of iterations	100
Velocity constant,c1	2
Velocity constant,c2	2

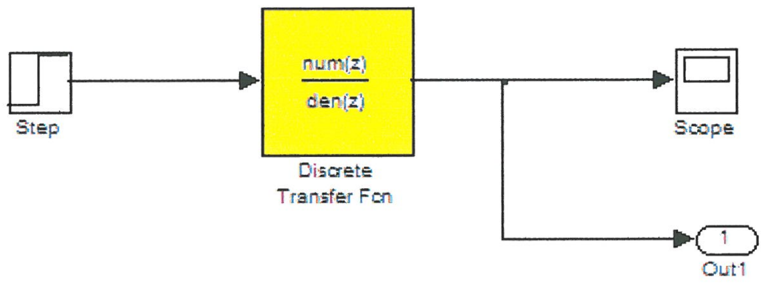

Figure 5.3: Off-line identification using PSO.

Once the significant terms have been identified and the estimates of associated parameters have been obtained, the model can be constructed. It is important to know whether the model has successfully captured all the system dynamics.

Therefore, a strategy for evaluating the correctness and validity of the model is necessary. If validation shows that the model is not good, then some of the design variables should be changed, and the identification procedure should be redone.

The validation strategy adopted in this research is to test if the predicted model output in a good agreement with the well known input or not.

Figures 5.4 to 5.7 show different input data (step and ramp) used to test if the predicted model follows the actual output. As shown below in both step and ramp input the predicted model is in a good agreement with the actual output with minimum error.

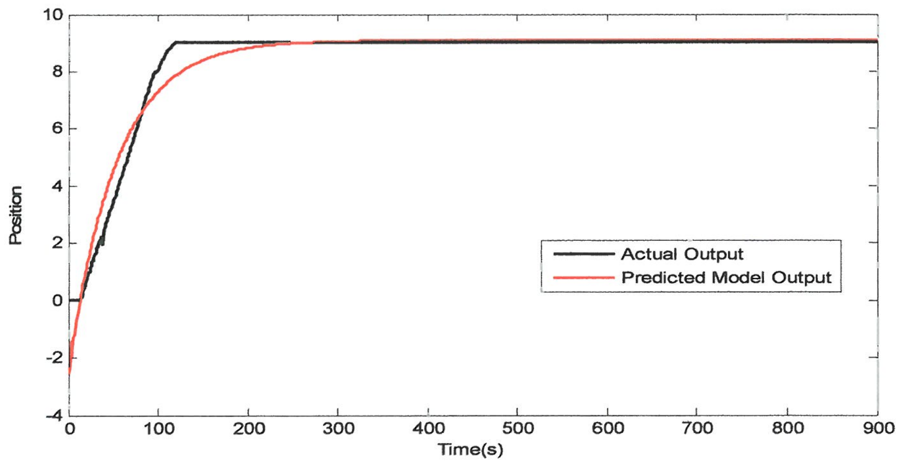

Figure 5.4: Step input response of the fourth order predicted model output using PSO.

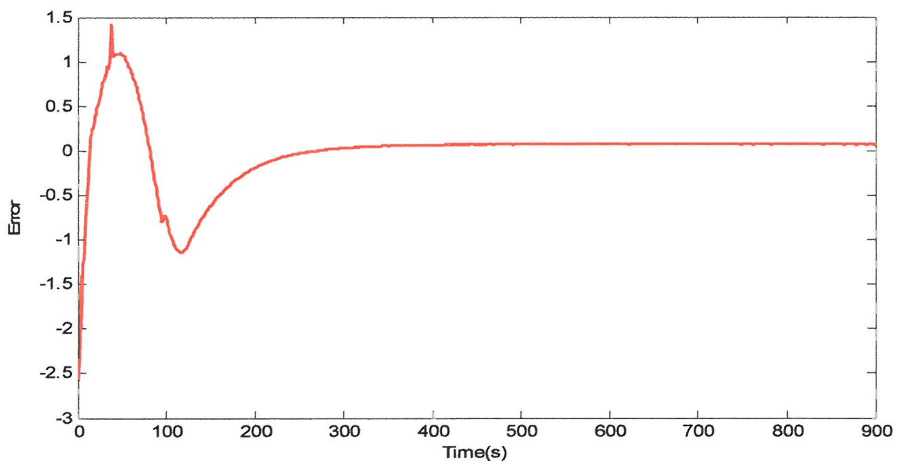

Figure 5.5: Error verses time between predicted model output and actual output.

Figure (5.6) and (5.7) reveal the model capability to predict the system output with different types of input. As can be seen below with a ramp input used to test the model the predicted output follows the input, but it can be observed clearly that there was a high error at the beginning of simulation but after that the error was almost zero. The explanation of this high error may be attributed to the collected input-output data obtained from the real system (time delay in response) of the pneumatic system.

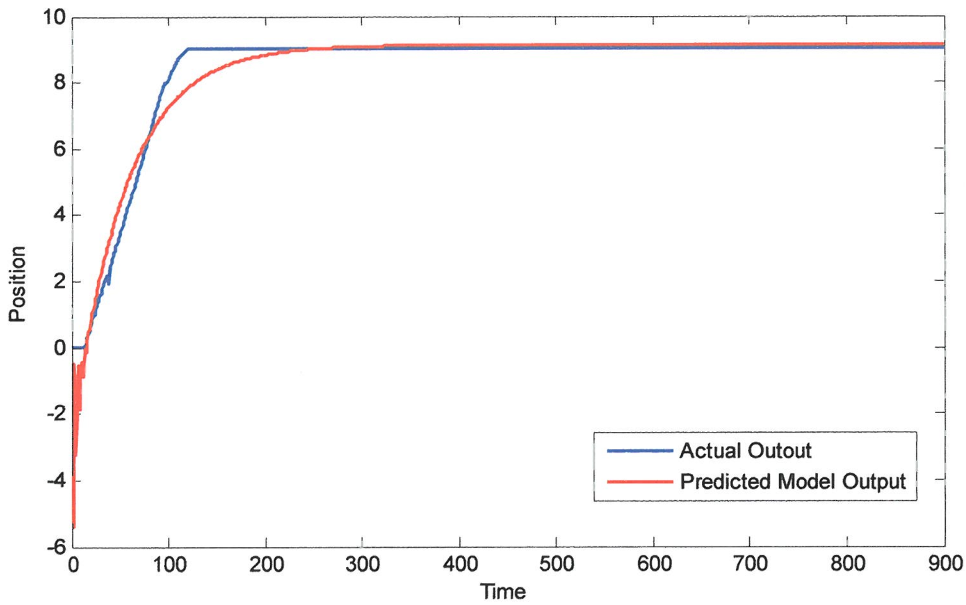

Figure 5.6: Ramp response of the fourth order predicted model output using PSO.

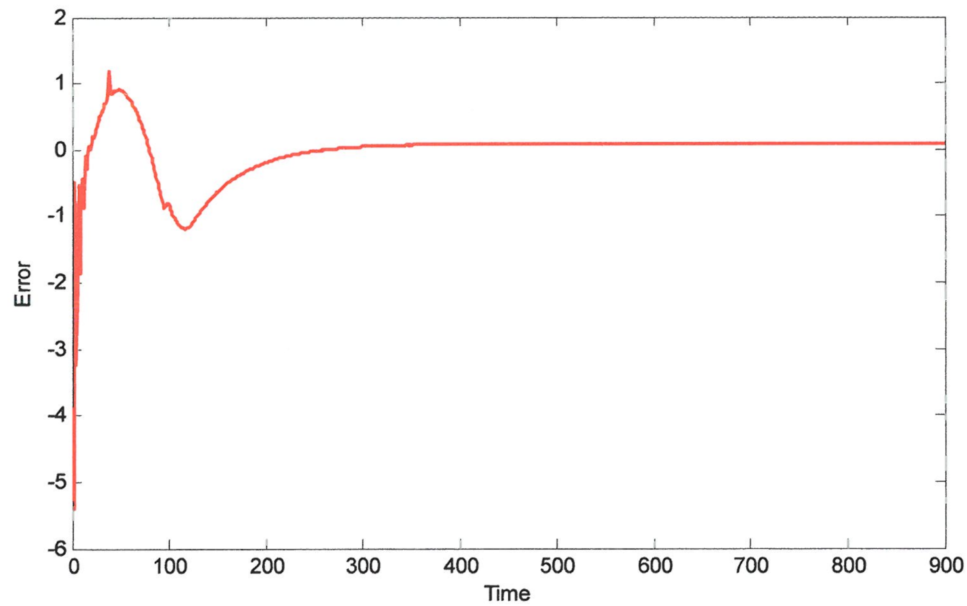

Figure 5.7: Error verses time between predicted model output and actual output.

After comparing the system responses with the parameterized model based on a performance function giving a measure of how well the model response fits the system response. The estimated parameters of the servo pneumatic system using obtained using PSO was:

The identified fourth order-transfer function of ARMA model using PSO algorithm

$$G_{PTrial} = \frac{-0.2996z^4 + 0.3617z^3 + 0.3330z^2 + 0.9908z}{7.7268z^4 + 1.3517z^3 - 2.1390z^2 - 3.7872z - 2.2386}$$

5.5 Comparison between the two transfer functions obtained using RLS and PSO

The identified transfer function using recursive least squares (RLS) algorithm:

$$G1 = \frac{0.0070699z^4 - 0.017738z^3 + 0.019649z^2 + 0.00027783z}{z^4 - 1.9827z^3 + 1.2989z^2 - 0.43504z + 0.12436}$$

The identified transfer function using Particle Swarm Optimization algorithm:

$$G2 = \frac{-0.2996z^4 + 0.3617z^3 + 0.3330z^2 + 0.9908z}{7.7268z^4 + 1.3517z^3 - 2.1390z^2 - 3.7872z - 2.2386}$$

Table5.2 Poles and Zeros locations

Method	Poles	Zeros
RLS	0.9682 0.8193 0.0976 + 0.3837i 0.0976 – 0.3837i	0 1.2615 + 1.1060i 1.2615 – 1.1060i -0.0140
PSO	0.9637 -0.2673 + 0.6528i -0.2673 – 0.6528i -0.6041	0 2.3090 -0.5508 + 1.0625i -0.5508 – 1.0625i

The root locus responses of the systems are shown in Fig 5.8 & 5.9. Which show that the poles in RLS method are closer to the unit circle which means that its relative stability is less than those obtained using PSO technique.

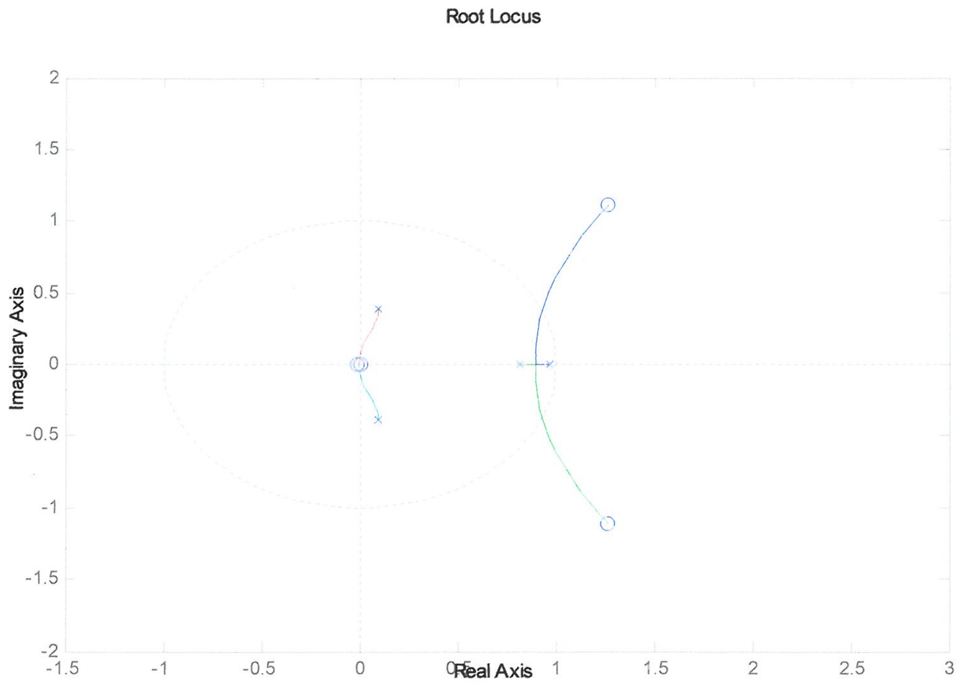

Figure 5.8: Root locus of RLS transfer function.

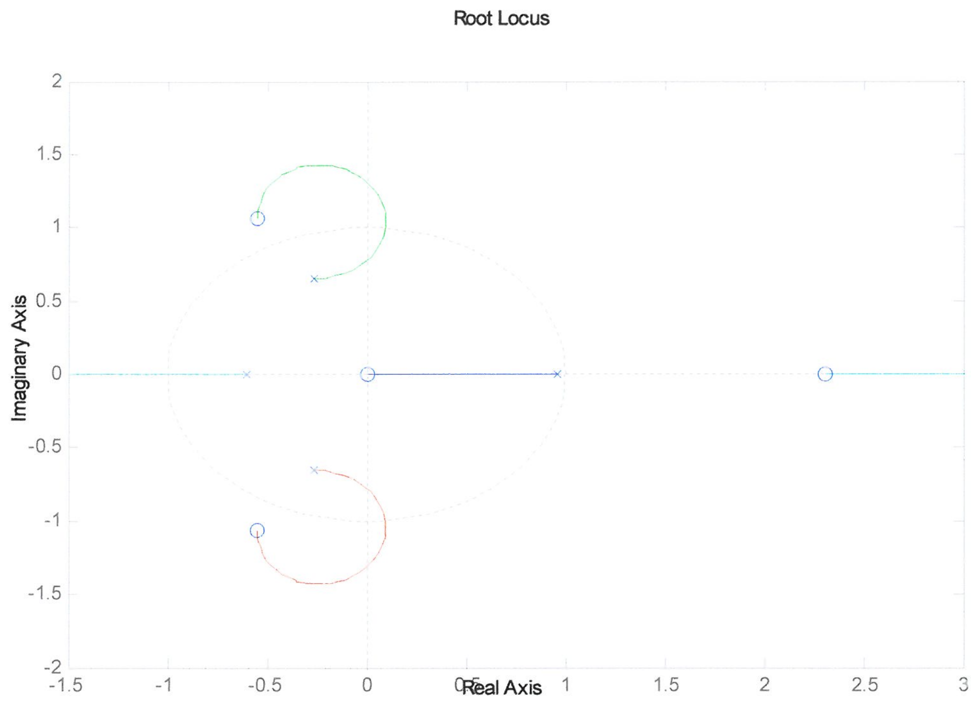

Figure 5.9: Root locus of PSO transfer function

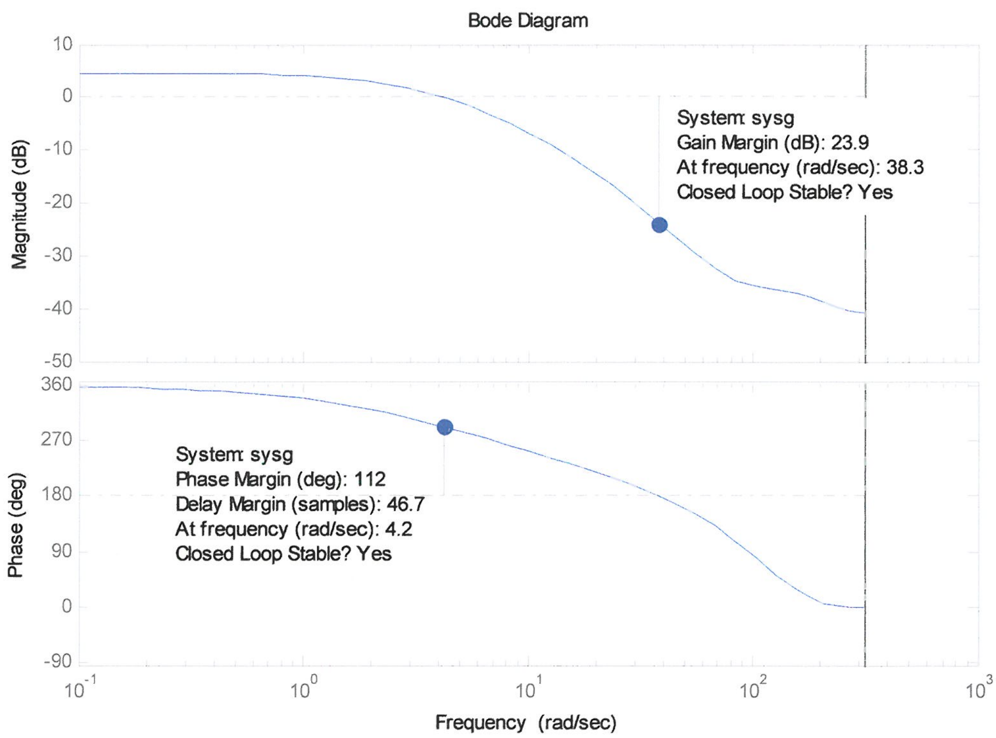

Figure 5.10: Bode plot of 4^{th} order transfer function using RLS algorithm

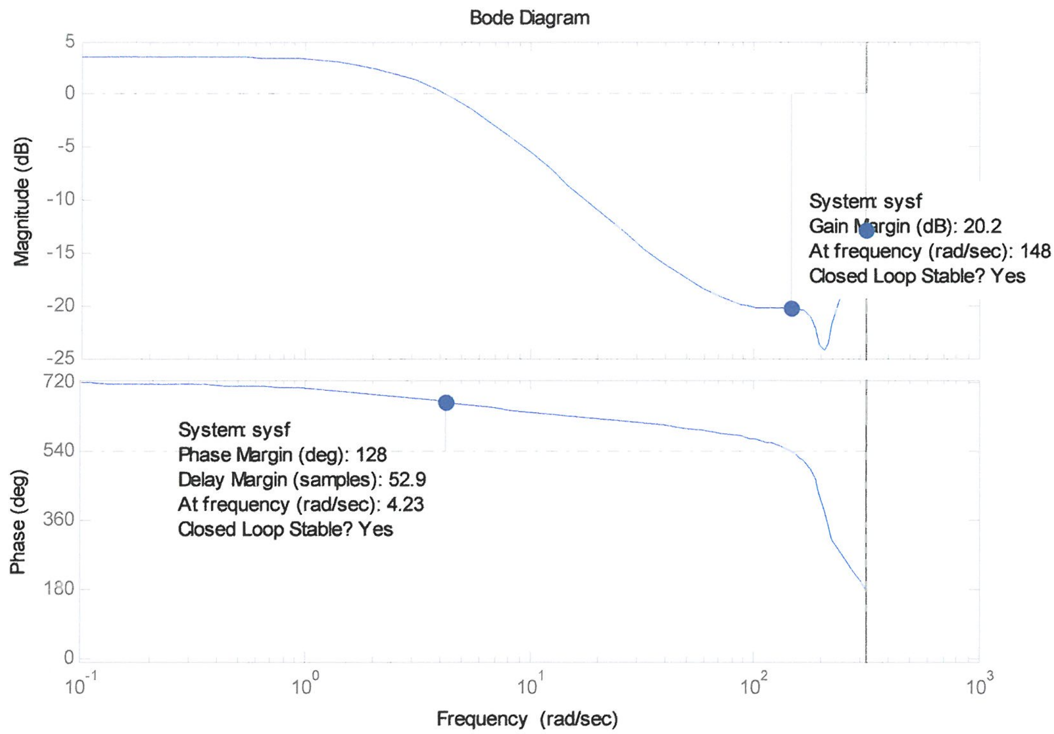

Figure 5.11: Bode plot of 4th order transfer function using PSO algorithm

As shown in figures 5.10 and 5.11, the gain margin of system obtained using PSO was (20.2) , and for system obtained using RLS was (23.9), and phase margin the results were (128) and (112) respectively.

Figure 5.12 shows step input of actual, predicted model using RLS and PSO.

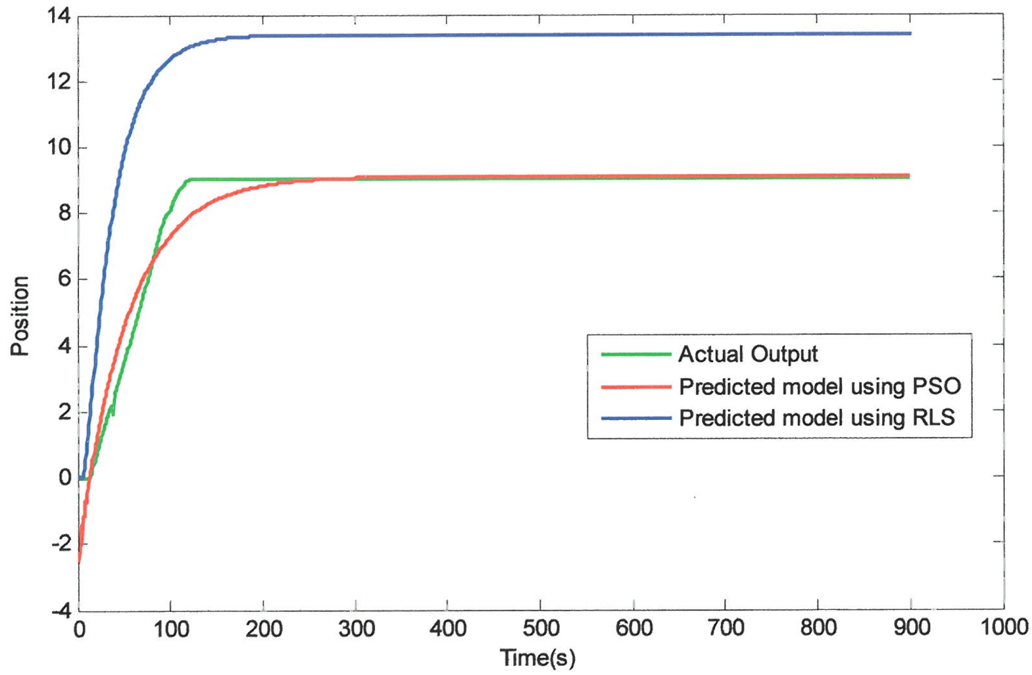

Figure 5.12: Step responses for actual output and predicted using (RLS, PSO) algorithm

5.6 Tuning and Optimization

Tuning work is usually performed manually by trying out different tuning parameter combinations on-line until a satisfactory or at least acceptable results are achieved. This method is laborious, time consuming, unsafe and does not always give the best possible solution, (Saleem, 2006). In this research, PSO has been employed as off-line tuning. After achieving satisfactory results, the optimised controller parameters can be applied to the real system through the proposed HIL concept.

To demonstrate the potentials of the proposed environment (HIL) and PSO as tuning technique, the tuning off-line has been performed to pick the PID parameters. Figures 5.13 & 5.16 show the single PID and cascaded PID controllers respectively that have been adopted in this research.

The following sections demonstrate the feasibility of the PSO in picking up PID parameters offline; it was applied on two structures: single and cascaded PID structures, the results of the two proposed structures were compared. After achieving satisfactory results, the optimised controller(s) can be applied to the real system through the proposed HIL. In this section simulation study is carried out to assess the performance of the proposed models.

5.6.1 Off-Line tuning of single PID structure.

The optimal values of the PID controller parameters K_p, K_i and K_d, is found using PSO. All possible sets of controller parameter values are particles whose values are adjusted so as to minimize the fitness function, which in this case is the error criterion, which is discussed in detail in chapter 4.

To start up with PSO, certain parameters need to be defined. Selection of these parameters decides to a great extent the ability of global minimization. The maximum velocity affects the ability of escaping from local optimization and refining global optimization. The size of swarm balances the requirement of global optimization and computational cost. Initializing the values of the parameters is as per table 5.3.

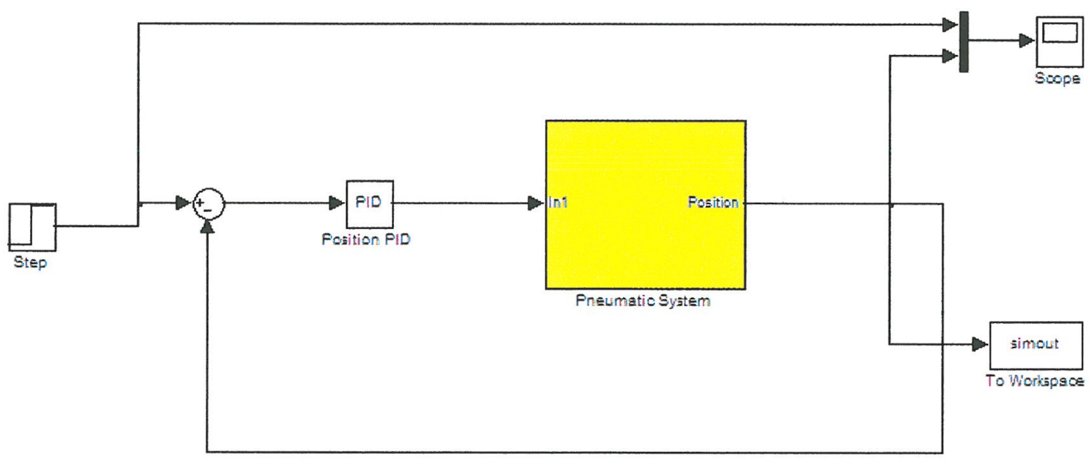

Figure 5.13: Control loop of off-line tuning of single PID using PSO.

Table 5.3: PSO selection parameters of single PID structure.

Population size	200
Number of iterations	5000
Velocity constant,c1	2
Velocity constant,c2	2

All tests for the single PID structure were conducted using the control loop shown in Figure 5.13. Table 5.4 shows the PID gain values obtained off-line for the single PID structure using PSO.

Table 5.4: Off-line of single PID controller Parameters.

Control Method	K_p	K_i	K_d
Single PID using PSO	1.5	4.7568	0.00062219

The results of the simulations in figures 5.14 & 5.15 show the system dynamic response with speed profile and multi speed profile tracking input.

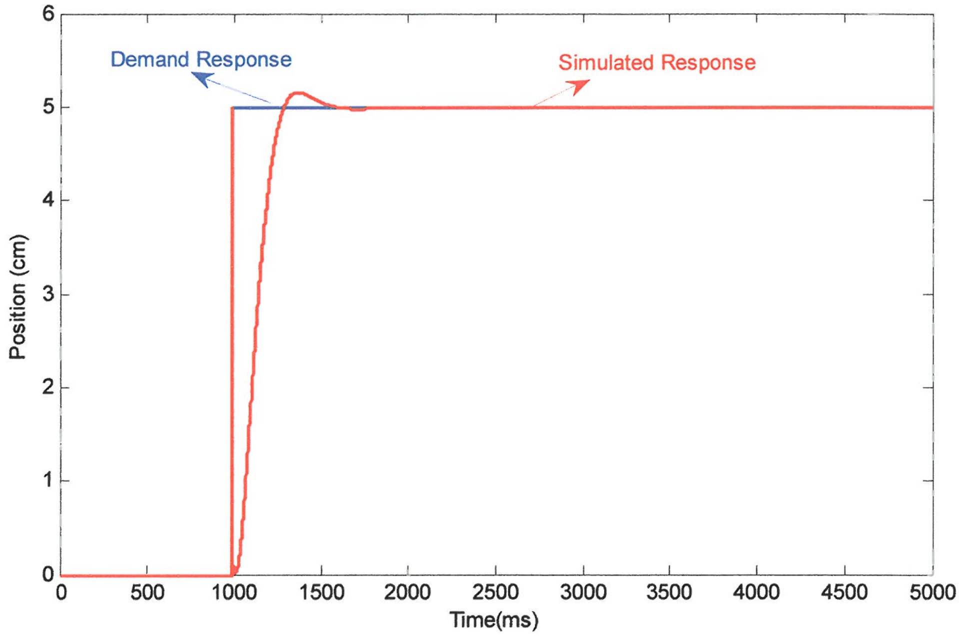

Figure 5.14: Step response of Single PID tuned offline.

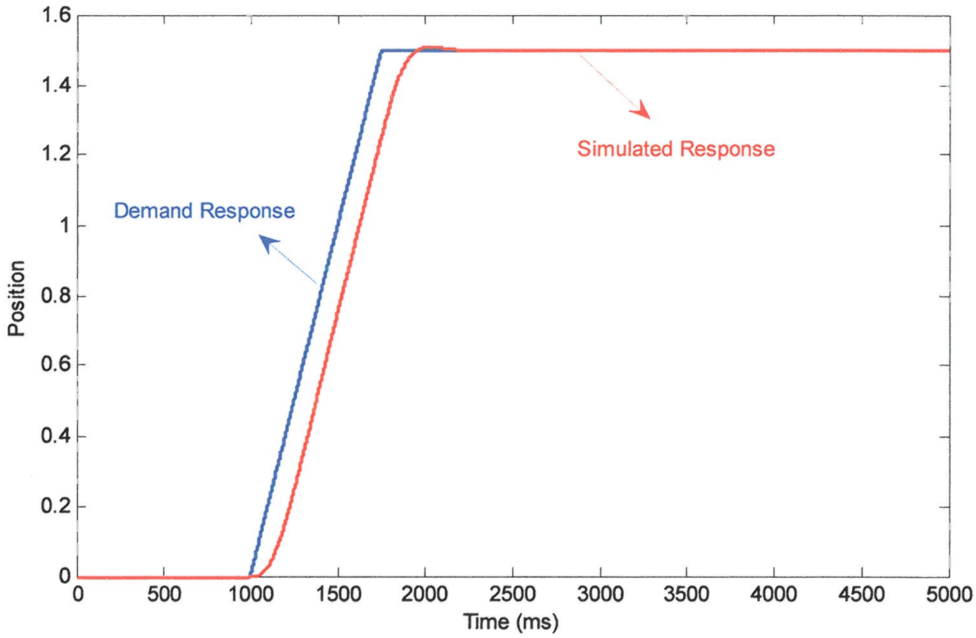

Figure 5.15: Speed Profile response of Single PID tuned offline.

5.6.2 Off-Line tuning of cascaded PID structure.

The easiest approach to applying optimization techniques in a cascade control system to find the parameters of the controllers is to optimize the errors in the loops individually. For this, the inner loop is considered first.

Once the parameters of the inner controller are obtained, the optimization procedure can carried be out to obtain the parameters of the outer loop or the position loop controller. All simulation tests for the cascaded PID structure were conducted using the control loop shown in Figure 5.16. The values of the PSO parameters are as per table 5.5.

Table 5.5: PSO selection parameters of cascaded PID structure.

Population size	500
Number of iterations	6000
Velocity constant,c1	2
Velocity constant,c2	2

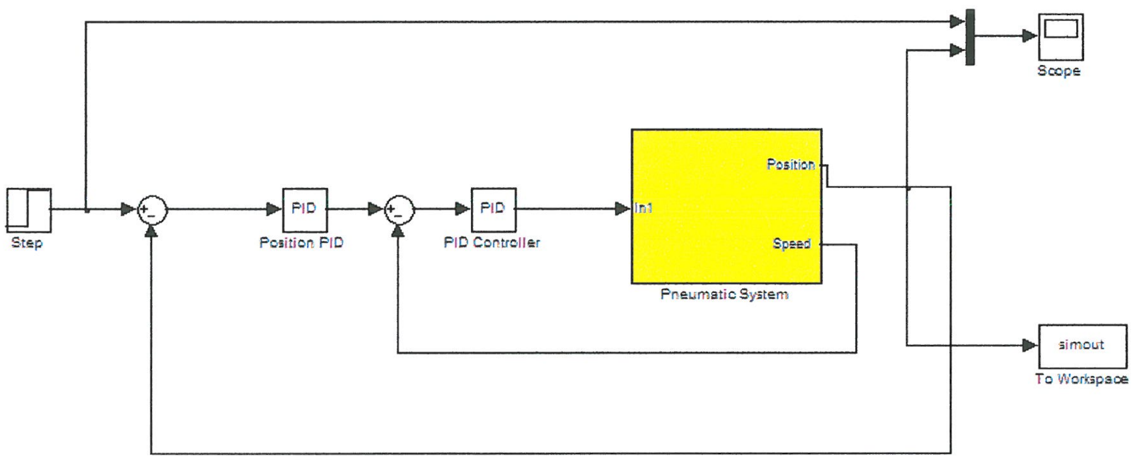

Figure 5.16: Control loop of off-line tuning of cascaded PID using PSO.

Table 5.6 shows the PID gain values for the inner and the outer loop obtained off-line for cascaded PID structure using PSO.

Table 5.6: Off-line of cascaded PID controller Parameters.

Control Method	K_p	K_i	K_d
Cascaded PID(Inner loop, C_s), using PSO	0.321653	0.5442	5.9563e-4
Cascaded PID (Outer Loop, C_p),using PSO	37.993	7.993	0.03777

As shown in figures 5.17 & 5.18 the a good agreement obtained using optimized parameters when they applied on cascaded PID structure with a small time delay and almost zero overshot for both speed and multi speed profiles.

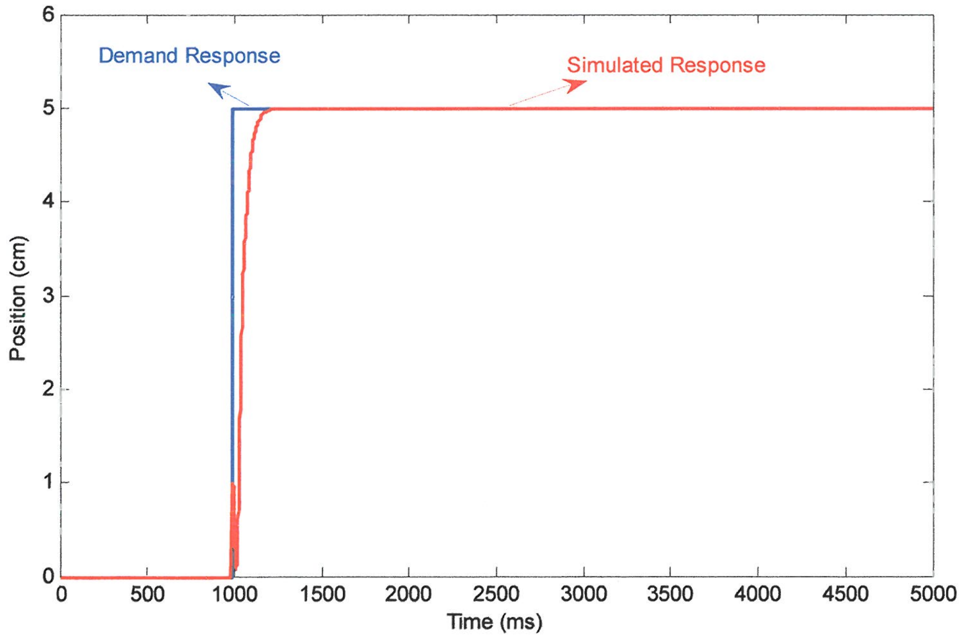

Figure 5.17: Step response of cascaded PID tuned offline.

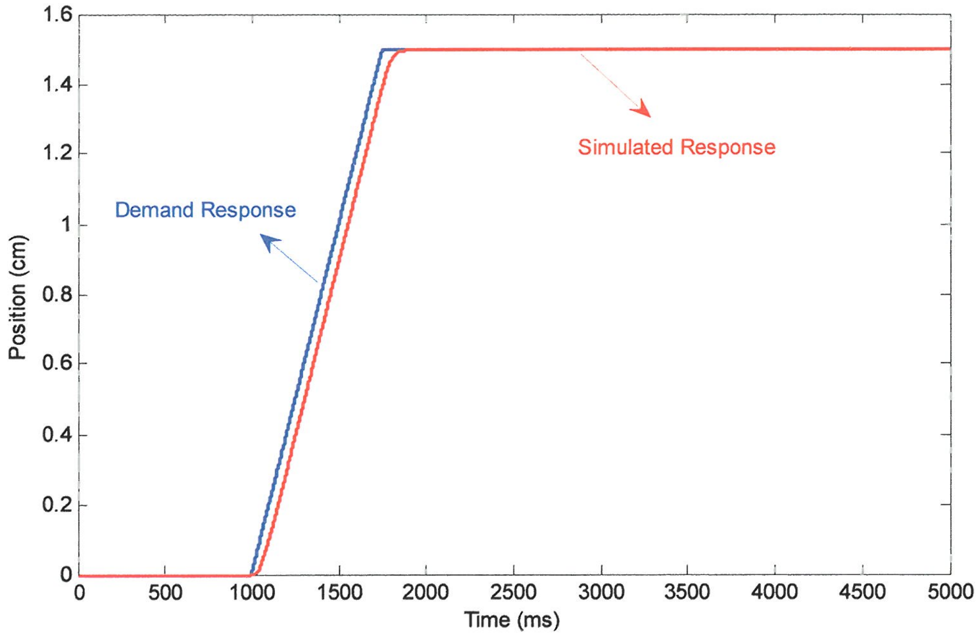

Figure 5.18: Speed Profile response of cascaded PID tuned offline.

5.7 On-line control

The controller with the optimised gain parameters has been applied to the real system through the HIL; the PID parameters were obtained during the off-tuning phase. The online PSO tuning method was compared to ZN results obtained by Saleem,et al (2005). The output of the real system for both ZN and PSO tuning is depicted in the Figures (5.20, 5.24, 5.35, and 5.36).

5.7.1 On-Line control of single PID structure

Figures below show the experimental results of the Servo- pneumatic system. The vertical axes in the figures indicate the target position and the actual piston position, detected by the displacement transducer. It is found that positioning accuracy is about 0.07 mm. Agreement of the trend can be seen from both responses.

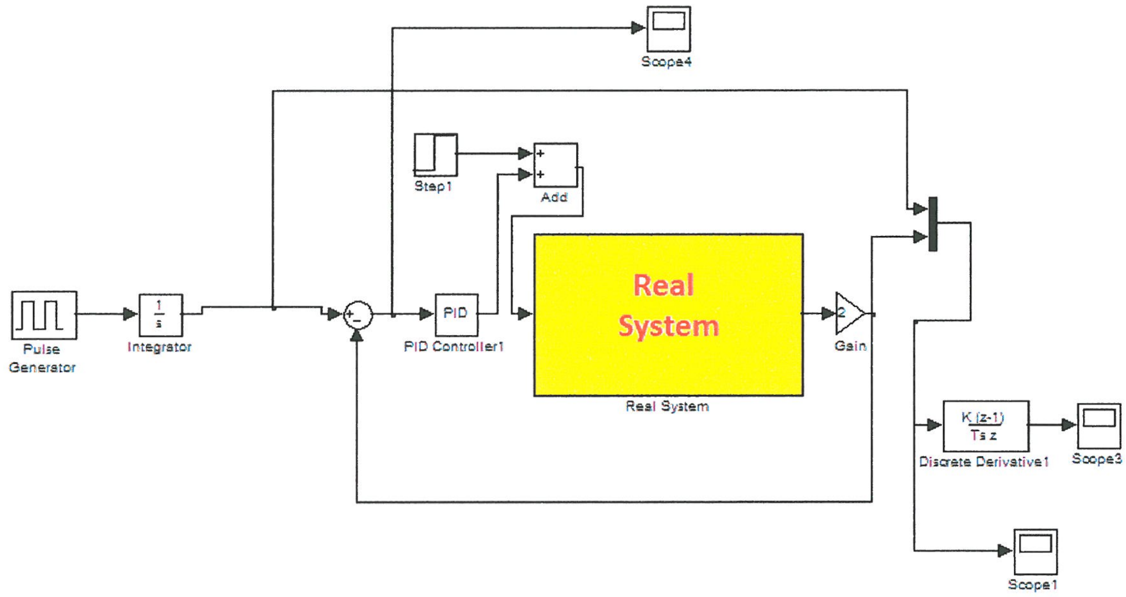

Figure 5.19: Single PID control structure of servo Pneumatic system.

To examine the robustness and effectiveness of the PSO algorithm, different simulations cases with different inputs are considered; speed profile and multi speed profile to ensure the quality of the PSO in finding the optimal solution.

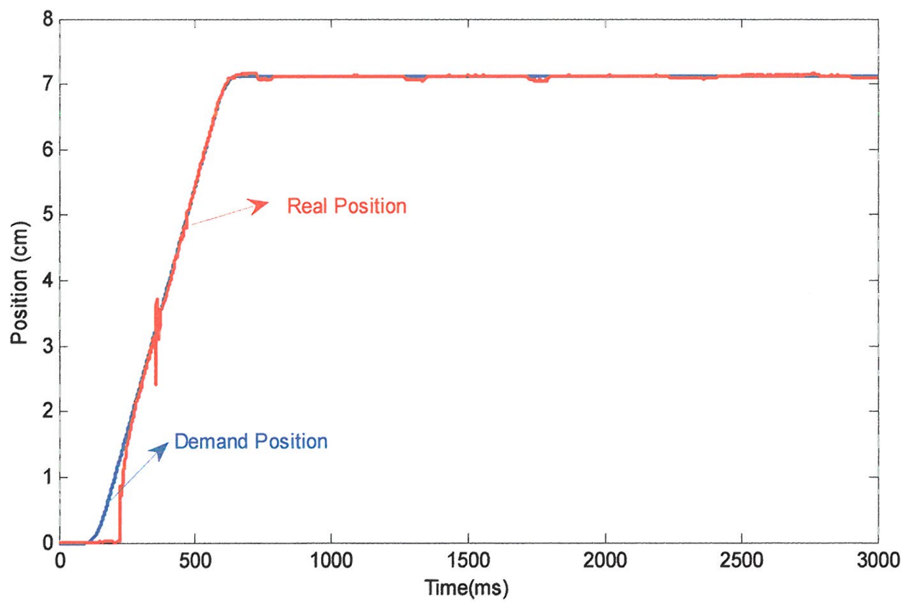

Figure 5.20: Variation of the real and demand positions (cm) verses time (ms).

As shown in figure 5.20 good tracking with small sustained oscillation around the demand position due to pneumatic compressibility.

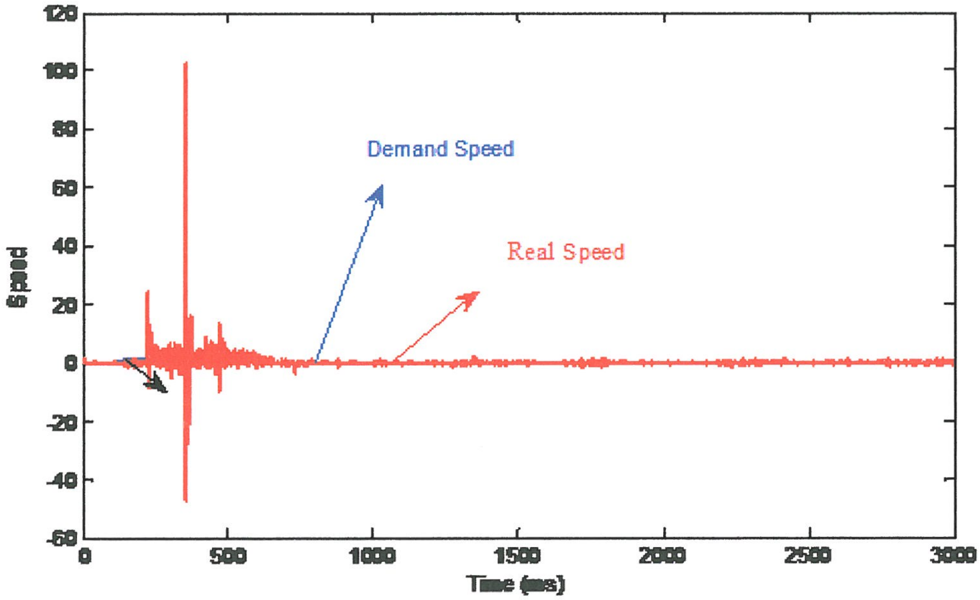

Figure 5.21: Variation of the real and demand speed profiles (cm/ms) verses time (ms).

As shown in Figures 5.20 & 5.21, the responses from the virtual system reasonably agreed with the experimental results. A close agreement was obtained from the demand output and the simulated response with slight time delay due to influence of stiction.

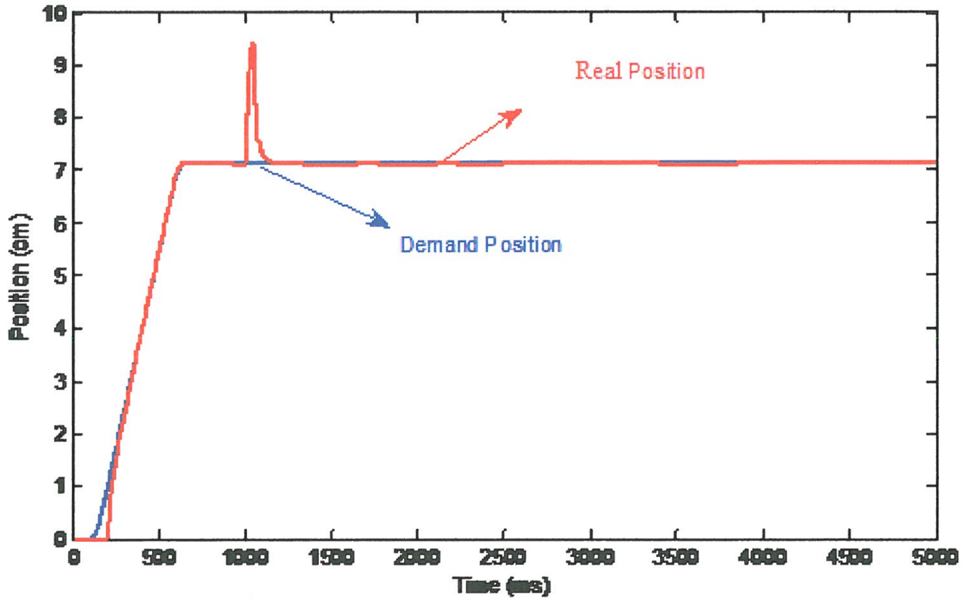

Figure 5.22: Variation of the real and demand positions (cm) verses time (ms) with small disturbance.

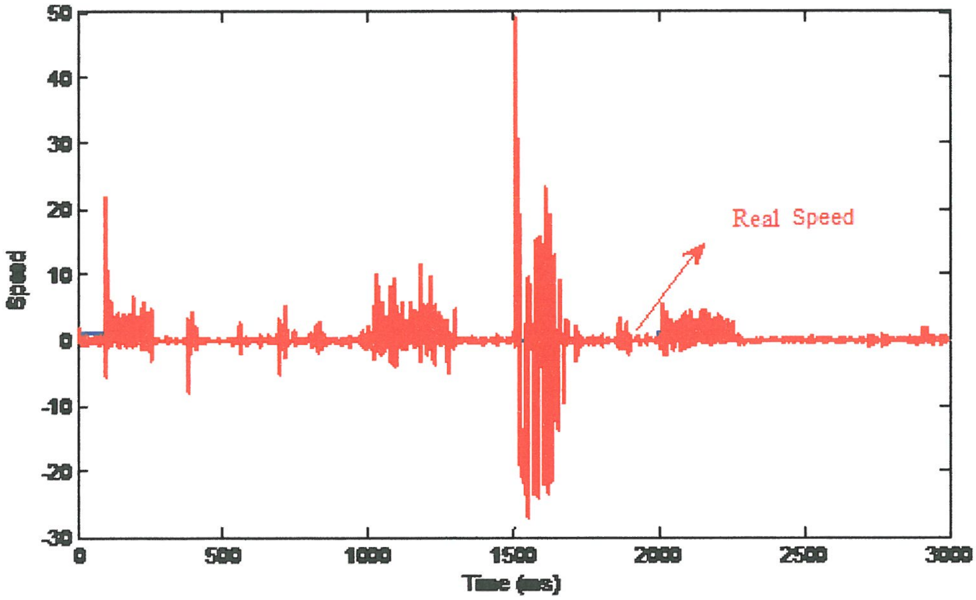

Figure 5.23: Variation of the real and demand speed profiles (cm/ms) verses time (ms) with small disturbance.

In figure 5.22 a small disturbance was applied to test the ability of the controller to reject disturbances. Also the controller with optimized parameters proved good response for disturbance rejection.

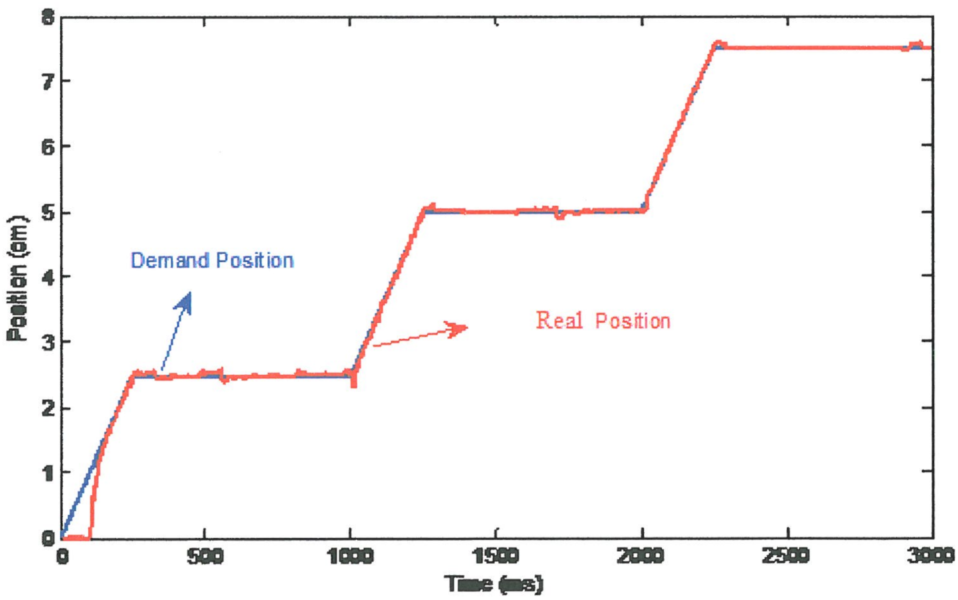

Figure 5.24: Variation of the real and demand positions (cm) verses time (ms).

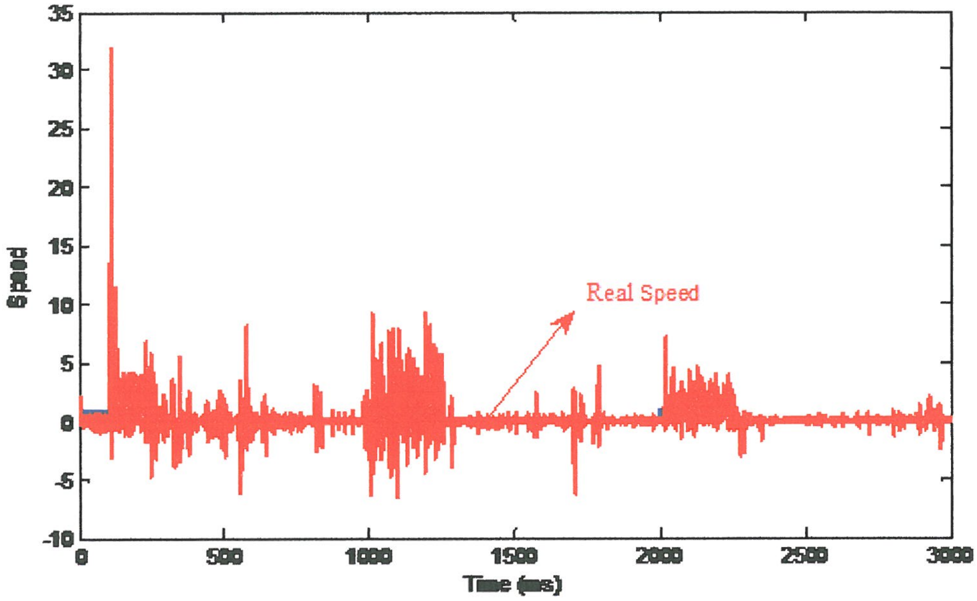

Figure 5.25: Variation of the real and demand speed profiles (cm/ms) verses time (ms).

In figure 5.24 a multi speed profile input was used to test the behaviour of the system, as shown a close agreement was obtained from the demand output and the simulated response with slight time delay due to influence of stiction.

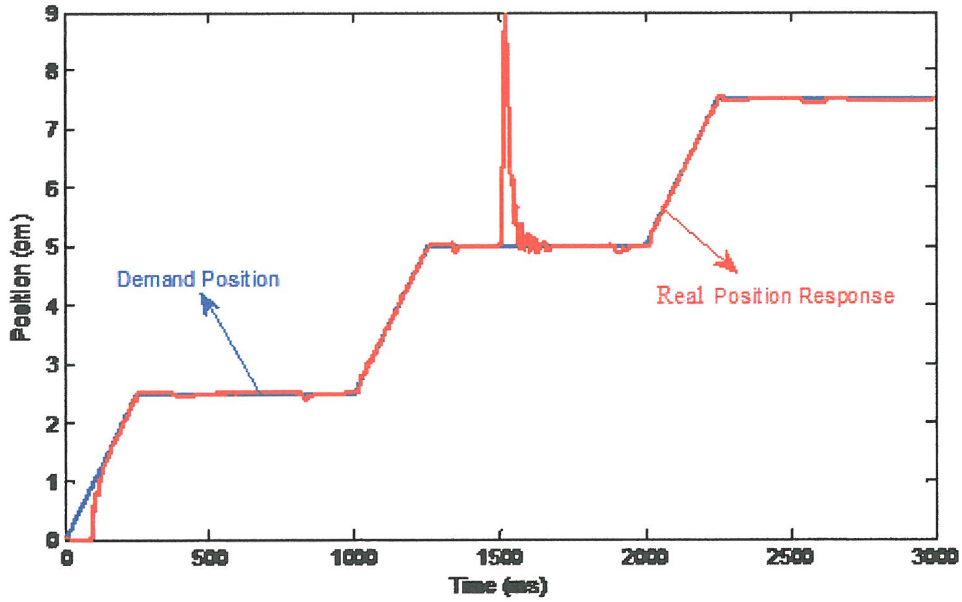

Figure 5.26: Variation of the real and demand positions (cm) verses time (ms) with small disturbance.

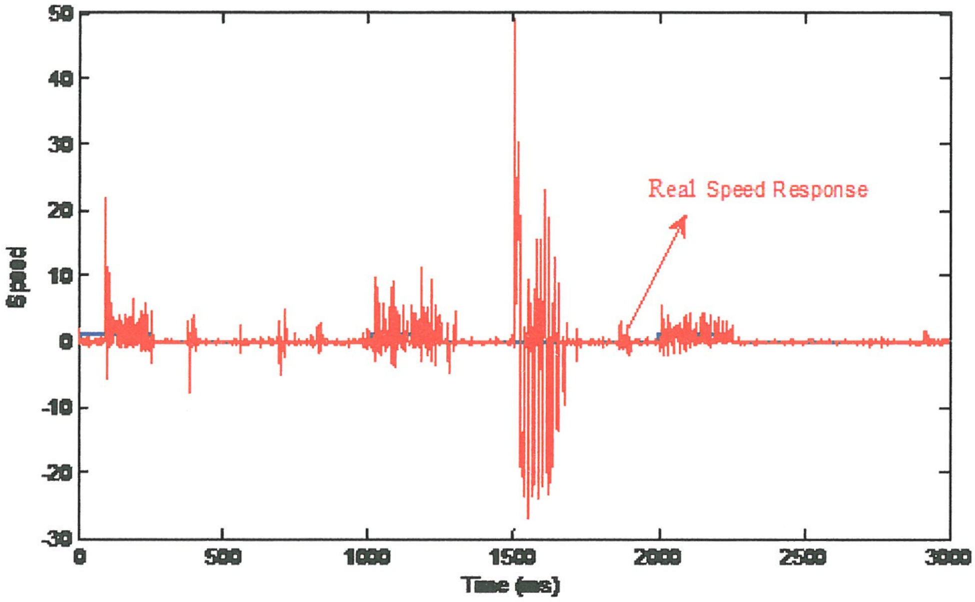

Figure 5.27: Variation of the real and demand speed profiles (cm/ms) verses time (ms).

In figure 5.26 a multi speed profile input was used to test the behaviour of the system, a small disturbance was applied to test also the ability of the controller to reject disturbances. Also the controller with optimized parameters proved good response for disturbance rejection.

5.7.2 On-Line control of cascaded PID structure

All simulation tests for the cascaded PID structure were conducted using the control loop shown in Figure 5.28. To examine the robustness and effectiveness of the PSO algorithm on cascaded PID structure, different simulations cases with different inputs are considered; speed profile and multi speed profile to ensure the quality of the PSO in finding the optimal solution.

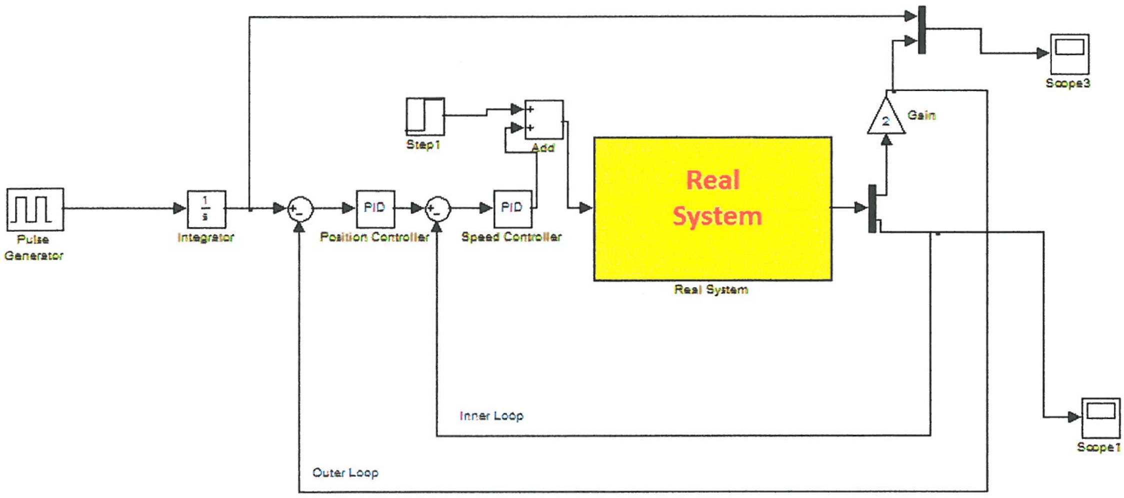

Figure 5.28: Cascaded PID control structure of servo Pneumatic system.

The performances of the cascaded PID control schemes were compared with single PID. The results have shown that the performance of single PID structure was very good when it's compared to cascaded PID structure despite superiority of cascade structure over single loop structure in control many processes.

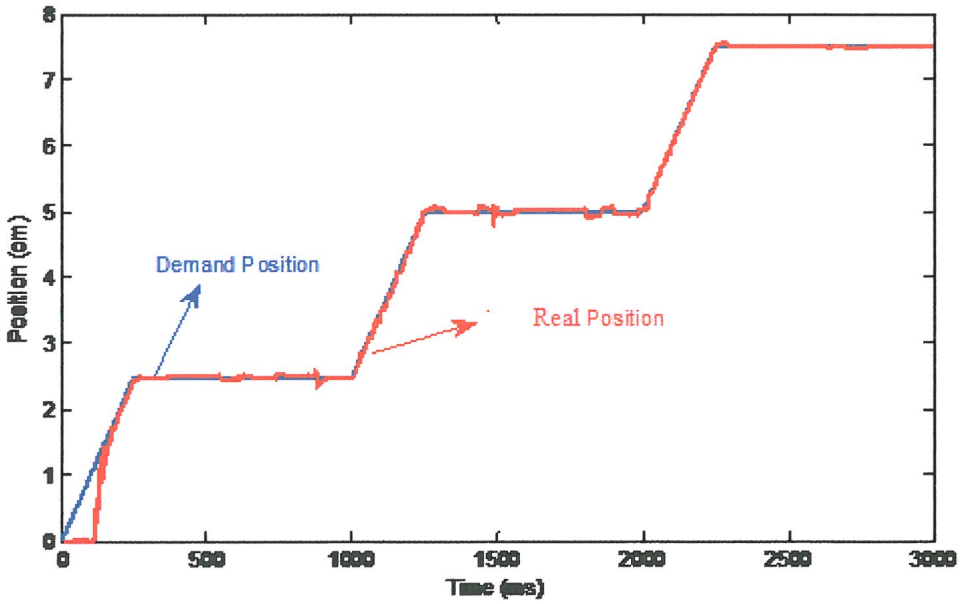

Figure 5.29: Variation of the real and demand positions (cm) verses time (ms).

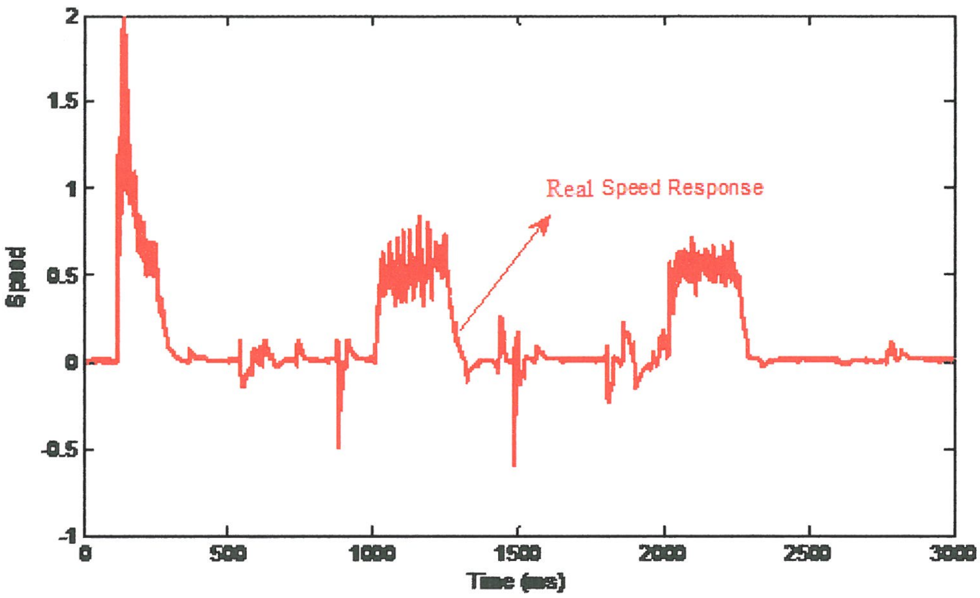

Figure 5.30: Variation of the real and demand speed profiles (cm/ms) verses time (ms).

In figure 5.29 a multi speed profile input was used to test the behaviour of the system, also there were a good agreement between the demand and the simulated response of the system, but the response was a little bit noisy when it is compared with single PID behaviour.

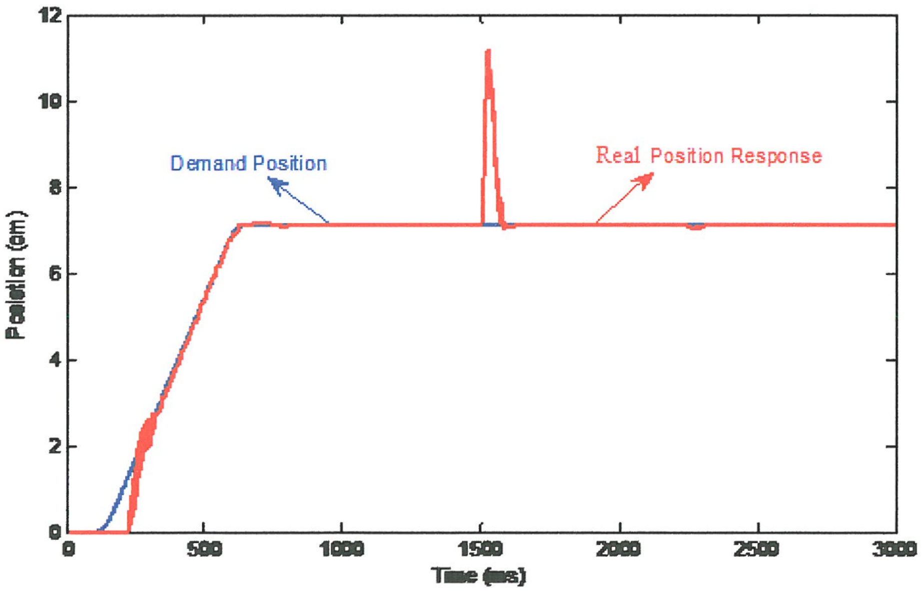

Figure 5.31: Variation of the real and demand positions (cm) verses time (ms) with small disturbance.

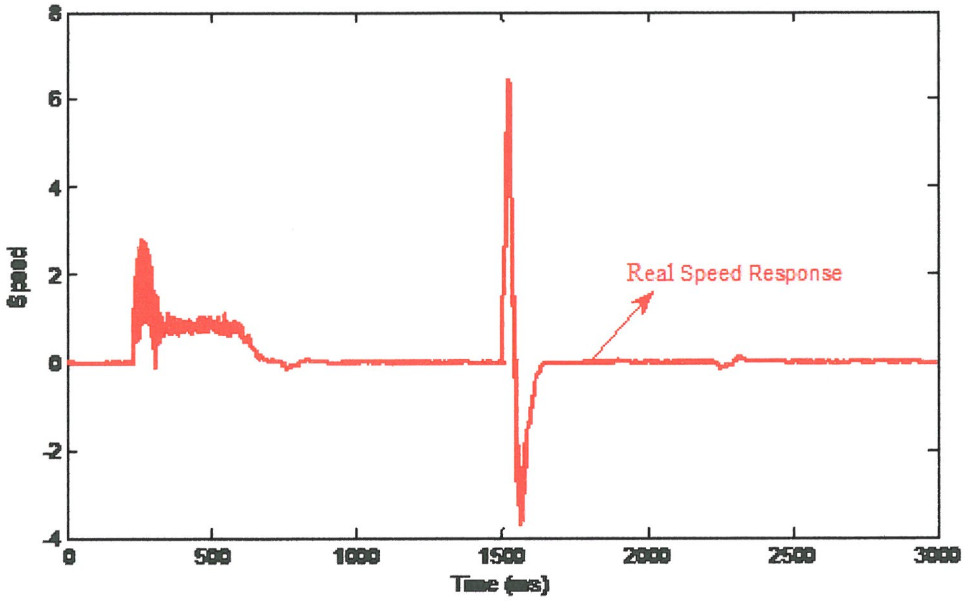

Figure 5.32: Variation of the real and demand speed profiles (cm/ms) verses time (ms).

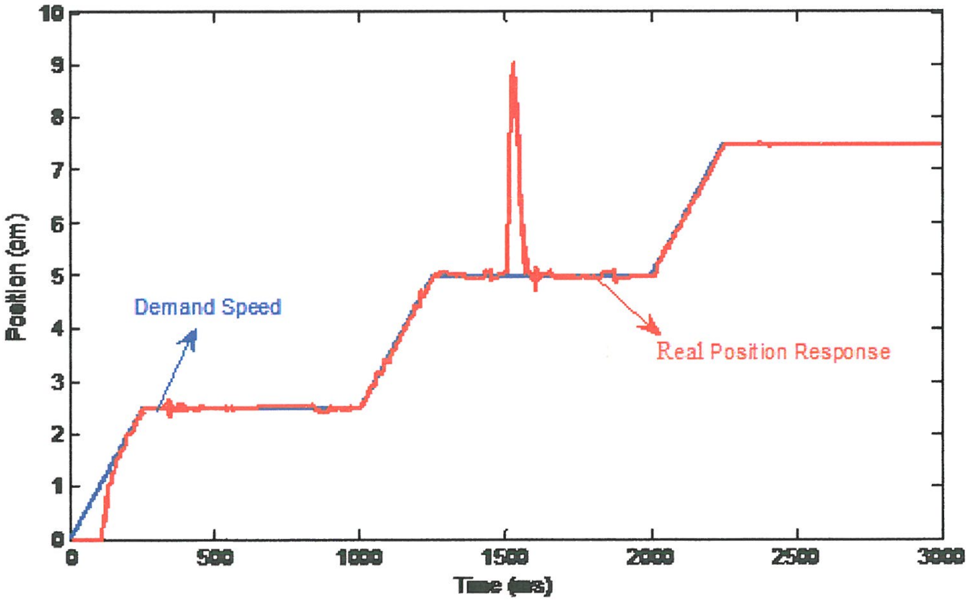

Figure 5.33: Variation of the real and demand positions (cm) verses time (ms) with small disturbance.

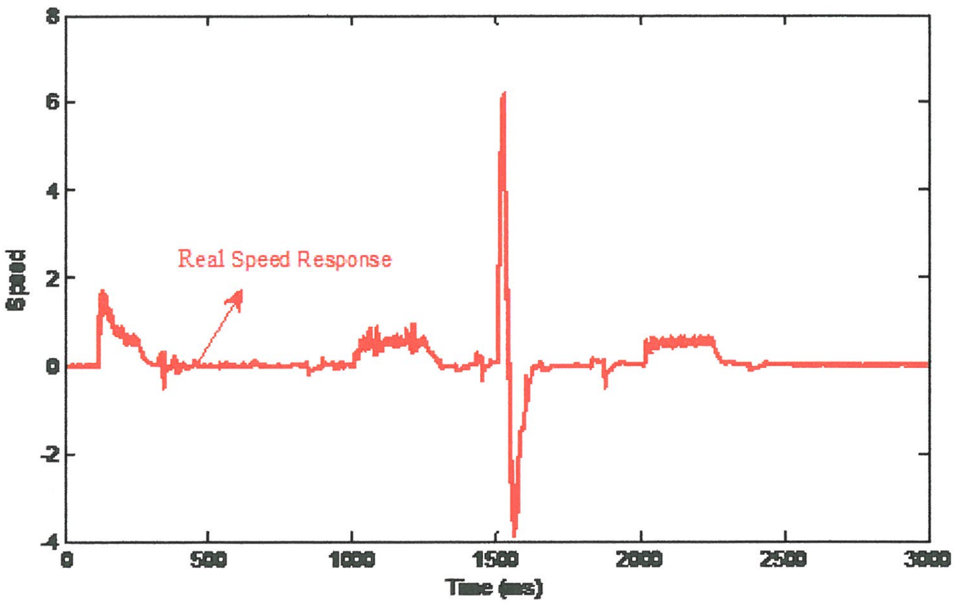

Figure 5.34: Variation of the real and demand speed profiles (cm/ms) verses time (ms).

In figure 5.33 a multi speed profile input was used to test the behaviour of the system, a small disturbance was applied to test also the ability of the controller to reject disturbances. Also the controller with optimized parameters proved good response for disturbance rejection and shows a better response than the single PID in rejection disturbances; this is due to existence of two nested loops that eliminate the error and give good performance with small error.

By using an enhanced option in MATLAB, the simulated and experimental responses were compared statistically by computing the max, mean, median, and standard deviation values of the corresponding responses for different inputs. The results are shown in Table 5.7. The values do indicate the relative accuracy of the simulation output.

Table 5.7: Statistical analysis of both single and cascaded PID controllers.

	Single PID				Cascaded PID			
	Min	Max	Mean	std	Min	Max	Mean	std
Speed profile without disturbance	0.01831	7.173	6.209	2.116	0.01734	7.236	6.130	2.213
Speed profile with disturbance	0.01953	9.436	6.601	1.711	0.0177	11.18	6.245	2.181
Multi speed profile without disturbance	0.01831	7.592	4.662	2.191	0.0177	7.567	4.651	2.201
Multi speed profile with disturbance	0.01953	8.965	4.701	2.204	0.01709	9.011	4.693	2.232

From these results it is evident that the PSO method provides accurate tuning performance. Further, from simulation results and comparisons between single and cascaded PID structures it's observed that the system with single PID structure is more stable and efficient than the one of the cascaded PID structure.

5.7.3 Results of ZN tuning

Hongtao Pan (Pan, 2001) proposed the following tuning strategy, based on Ziegler Nicholas method, for pneumatic drives which has been used in the experiments to tune the gain factors and thus achieve a satisfactory dynamic response. The following steps were applied by Saleem, et al (2009).

Step 1: The tuning was started by applying a small value to the proportional gain (K_p) and then increasing K_p until the system oscillated. Fifteen percent (15%) of the total value of K_p was decreased.

Step 2: Once the proportional gain (K_p) was set, the integral gain (K_i) was increased by a small value until the minimum error was achieved. Twenty five percent (25%) of the total values of K_i and K_p was decreased.

Step 3: After increasing the integral gain K_i, K_p was increased until the system oscillated again to enhance the stability of the system. Then, 15% of the total value of Kp was decreased.

Step 4: Steps 1 to 3 were repeated, adjusting each gain value carefully to achieve better system performance.

Table 5.8 shows the PID gain values obtained Ziegler-Nichols PID Tuning.

Table 5.8: Ziegler-Nichols PID Tuning Values

Control Method	K_p	K_i	K_d
Single PID using ZN	14	6	0.2

Figures 5.35 & 5.36 show the response of the pneumatic controller tuned using ZN methodology, It's obviously shown that results of tuning using ZN suffer from high oscillations when its compared with those obtained using PSO tuning.

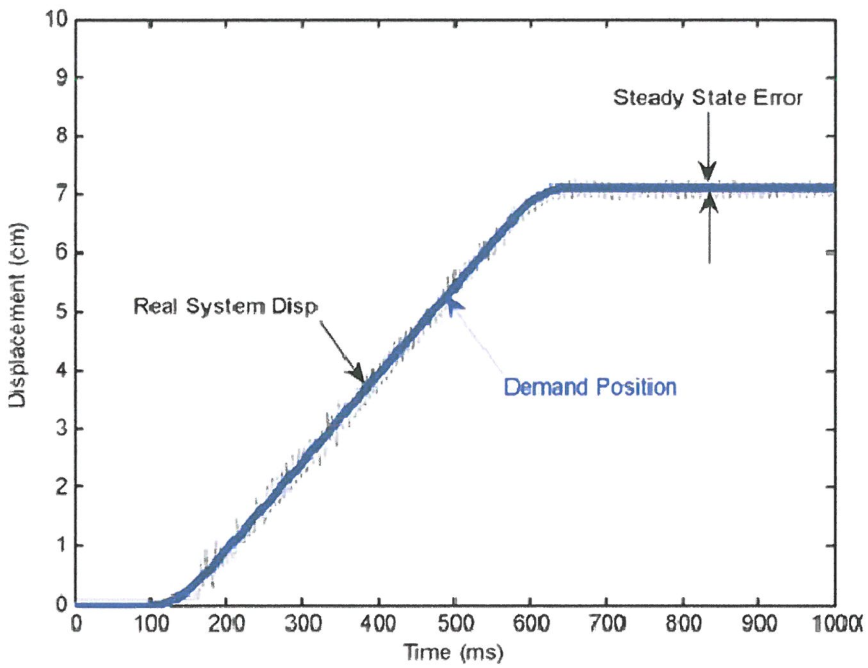

Figure 5.35: Variation of the real and demand displacements verses time at K_p=14, K_i=6, and K_d=0.2. (Saleem, et al 2009)

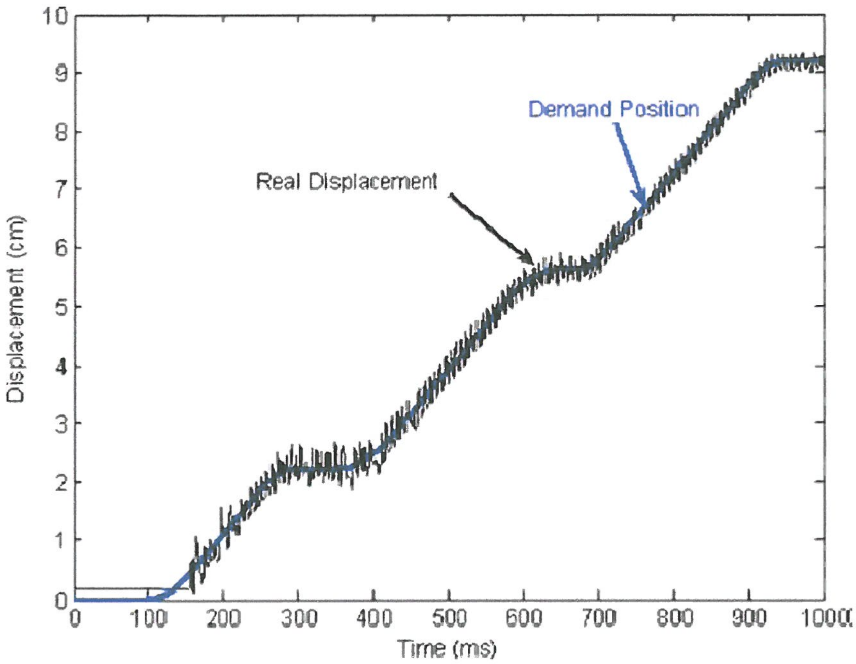

Figure 5.36: Variation of the real and demand positions (cm) verses time (ms). (Saleem, et al 2009).

5.8 Summary

The PID parameters obtained using ZN and PSO applied on different control structures, It's obviously shown that results of tuning using ZN suffer from high oscillations when its compared with those obtained using PSO tuning.

PSO tuned system provides more accuracy and faster convergence speed, stable response with small delay time and very quick settling time; succeeded in tracking the proposed speed profile and multi speed profile well.

PSO algorithms combined with other intelligent techniques, such as neural networks and fuzzy logic control systems open a new way to design and construct intelligence control systems adapted to complex processes.

Experimental results obtained by tuning the controller using PSO applied on single PID structure indicate good improvements compared with cascaded PID structure despite superiority of cascade structure over single loop structure in control many processes.

.

Chapter 6
Conclusions

Introduction

The study has focused on the application of the PSO computational algorithm for identification and tuning controllers of servo-pneumatic systems. The objective has been to improve the performance of pneumatic systems that experience poor control behaviour when tuned using conventional tuning methodologies.

6.1 Research findings

- It has been shown that the obtained model using PSO had captured the dynamic characteristics of the servo-pneumatic system well and has been able to predict the system response to various input signals.
- Two controllers were designed: one using a single PID controller; the other cascaded PID controller. The performances of the control schemes were compared against each other. The results have shown that the performance of single PID structure was very good when it's compared to cascaded PID structure despite superiority of cascade structure over single loop structure in control many processes.

The proposed HIL concept has the following advantages

- Reducing the time of tuning since dealing with virtual system is much faster than real system.
- Increase the safety factor of the process, in some cases tuning can be dangerous on lives and equipments. For example, if the proportional gain has been set to a

high value either by mistake or due to lack of experience, then system will behave aggressively which may harm anyone in the scene or cause a potential damage to the components themselves.

- The system becomes more user-friendly and thus it will reduce the need for specialists to perform system tuning and optimisation.
- Facilitate the implementation of different control strategies/methodologies in order to allow for variation in future. This was proved by implementing two control strategies single and cascaded PID structures.
- Provides the ability to interact with real system, which facilitates the identification of the system parameters and the implementation of controller tuning procedure.

6.2 Future work

1. Blending PSO with the other intelligent optimization algorithms, which combine the advantages of the PSO with the advantages of the other intelligent optimization algorithms to create the compound algorithm that has practical value in identification and tuning servo-pneumatic systems.
2. Different tuning methodologies implementation: tuning methods described in chapter 4, such as gain-scheduling, fuzzy logic, neural network, GA, etc.., can be implemented using the developed simulation environment. It is advisable to spend more efforts on this aspect.
3. Detecting the system's order using PSO as an extension the previous work.
4. Developing a multi axis servo pneumatic system using the proposed method.

References

A. Saleem, C.B. Wong, J. Pu. (2010),**Servo-pneumatic Systems: Component-based modelling, simulation, and control.** (1st ed). LAP LAMBERT Academic Publishing.

A. Saleem,S. Abdrabbo , T. Tutunji. (2009), **On-line identification and control of pneumatic servo drives via a mixed-reality environment**, International Journal of Advanced Manufacturing and Technology, Springer, 40:518–530.

Alireza Alfi, Hamidreza Modares.(2010), **System identification and control using adaptive particle swarm optimization**. Applied Mathematical Modelling Elsevier.

Ali Reza Tavakolpour, Intan Z. Mat Darus, Osman Tokhi, Musa Mailah.(2010). **Genetic algorithm-based identification of transfer function parameters for a rectangular flexible plate system.** Engineering Applications of Artificial Intelligence. (Article In Press).

Angerer BT, Hintz C, Schröder D (2004). **Online identification of a nonlinear mechatronic system**. Control Eng Pract 12:1465–1478.

Åström K., and Hägglund T. (2004). **Revisiting the Ziegler-Nichols Step Response method for PID control.** Journal of Process Control, Vol. 14, pp. 635-650.

Burrows.C. R, Webb. C. R.(1967). **Simulation of an on-off pneumatic servomechanisms**, Journal of mechanical engineering, p 631-641.

Daw N, Wang J, Wu QH (2003). **Parameter identification for nonlinear pneumatic cylinder actuators**. In: Zinober ASI, Owens DH (ed) Nonlinear and adaptive control NCN4. Springer, New York, pp 77–88.

G. Carducci, N.I. Giannoccaro , A. Messina, G. Rollo. (2006), **Identification of viscous friction coefficients for a pneumatic system model using optimization methods,** Mathematics and Computers in Simulation 71 (2006) 385–394.

Gao, Xiang Feng, Zheng-Jin.(2005), **Design study of an adaptive Fuzzy-PDcontrollerfor pneumatic servo system**, Control Engineering Practice, v 13, n 1, January, p 55-65.

H. Pan.(2001). A **further study on high speed pneumatic pick-and-place positioning system.** MPhil thesis. De Montfort University, Leicester 2001.

Hamiti, K., Voda-Besancon, A. Roux-Buisson H. (1996), **Position control of a pneumatic actuator under the influence of stiction**, Control Engineering Practice, v 4, n 8, p 1079-88.

Ljung, L. (1999). **System Identification - Theory for the User**. (2nd ed). Prentice-Hall. Upper Saddle River, N.J.

Mannetje.J.J. (1981), **Pneumatic servo design method improves system bandwidth twenty-fold**, Control Engineering, v 28, n 6, p 79-83.

Michael Brian Thomas. (2003), **Advanced servo control of a pneumatic actuator**. Unpublished Doctoral Dissertation, the Ohio State University.

Moore, P.; Jun Sheng Pu. (1996), **Pneumatic servo actuator technology**, IEE Colloquium on Actuator Technology: Current Practice and New Developments, p 3/1 6.

Nelendran pillay. (2008), **A particle swarm optimization approach for tuning of SISO PID control loops**, unpublished master degree of Durban University of Technology.

Poulin, E.Pomerleau A. (1996**), PID tuning for integrating and unstable processes**, IEE Proc.-Control Theory Appl., Vol. 143, pp. 429-435.

Shearer, J. L. (1956), **Study of pneumatic process in the continuous control of motion with compressed air**, Transaction of ASME, p 233-249.

Sivanandam.S.N, Deepa.S.N. (2008), **Introduction to Genetic Algorithms.** (1st ed.). Springer-Verlag Berlin Heidelberg.

Wang J, Wang JD, Daw N, Wu QH (2004**). Identification of pneumatic cylinder friction parameters using genetic algorithms**. IEEE/ASME Trans Mechatron 9(1):100–107.

Weston.R. H., Moore. P. R., Thatcher.T. W, Morgan, G. (1984), **Computer controlled pneumatic servo drives**, Proceedings of the Institution of Mechanical Engineers, Part B: Management and Engineering Manufacturing, v198, p 275-281.

Ziaei K, Sepehri N (2000). **Modeling and identification of electrohydraulic servos**. Mechatronics, 10:761–772.

التحكم بالانظمة النيوماتيكية بطريقة Hardware In The Loop

إعداد

بشار سعد طه

المشرف

الأستاذ الدكتور أحمد القيسية

المشرف المشارك

الدكتور أشرف اسماعيل سليم

ملخص

تزايد البحث والاهتمام بالاونة الاخيرة بدراسة الانظمة النيوماتيكية وطرق التحكم بها ، ويعود هذا الاهتمام المتزايد الى اهمية هذه الانظمة وما تقدمه من مميزات وحسنات، حيث تعتبر الانظمة النيوماتيكية سهلة التعامل و سهلة الصيانة ومتوافرة باسعار منطقية ، ولكن على الصعيد الاخر يوجد هناك عدة صعوبات تواجه المهندسين عند القيام بالتحكم بالانظمة النيوماتيكية ، من هذه الصعوبات : قابلية الهواء للانضغاط وتسرب الهواء بالاضافة الى عوامل الاحتكاك للاجزاء الداخلية لهذه الانظمة، بالاضافة الى عدم توفر برامج المحاكاة المخصصة لهذه النظمة مما يجعل عملية التحكم اكثر صعوبة.

يهدف هذا البحث الى تطبيق (Particle Swarm Optimization) كطريقة لتعريف النظام النيوماتيكي وتعريف متغيرات التحكم المعروفة ب (PID) وكل ذلك من خلال بيئة محاكاة يمكن من خلالها الربط ما بين اجزاء النظام الحقيقية ونظام التحكم الافتراضي ومحاكاته عن طريق الحاسب حيث تعطينا هذه الطريقة افضلية المحاكاة ومعرفة النتائج من دون اي كلف اضافية اثناء بناء النظام الحقيقي او حتى تعريض اي جزء الى التلف.