POSITION CONTROL OF A PNEUMATIC SYSTEM USING ADAPTIVE INTELLIGENT METHODS

by

BEHRAD DEHGHAN

A thesis submitted to the Department of Mechanical and Materials Engineering
In conformity with the requirements for the degree of Master of Applied Science

Queen’s University
Kingston, Ontario, Canada
(June, 2012)

Copyright ©Behrad Dehghan, 2012
Abstract


A large body of research is devoted to the development of advanced control techniques to improve the positioning performance of pneumatic systems, which are known to be highly nonlinear systems. Although model based controllers show good results, the requirement for a system model makes these methods difficult to implement. So-called intelligent algorithms, such as neural networks and fuzzy rule based controllers, are attractive since they do not require a model. The performance of these controllers can be enhanced by adding an adaptive mechanism to adjust controller parameters in a continuous on-line fashion.

The objective of this thesis was to explore different adaptive intelligent controllers for position control of a pneumatic system. The application was the x-axis and z-axis of a pneumatic gantry robot. They were tested independently for their ability to track step and sine wave trajectories. The rodded x-axis cylinder was an example of a short stroke low friction application. The rodless z-axis cylinder was an example of a long stroke high friction application.

Five different controllers were tested: 1) PID, 2) Fuzzy, 3) PID+Adaptive Neural Network Compensator (ANNC), 4) ANNonly and 5) Fuzzy Adaptive PID (FAPID). Results with FAPID and PID+ANNC showed improvement in tracking performance over PID by 60% for the rodded and 35% for the rodless cylinder. This level of improvement was expected given the adaptive nature of the controller. Unfortunately, both required significant effort to setup and tune.

In order to reduce the tuning effort, a second adaptive mechanism was added to FAPID, to adjust output weights. Results with adaptive PID and modified FAPID (MFAPID) showed further improvement performance over PID by 87% for the rodded and 70% for the rodless cylinder (in addition to being easier to tune). To provide a measure of robustness, experiments were conducted at two supply pressures and three tracking frequencies. The fact that MFAPID was able to improve performance for both cylinders, is considered further evidence of its robustness. MFAPID is considered novel for two reasons: 1) fuzzy rule set is reduced in size relative previous work and 2) addition of an adaptive mechanism for output weights is new.
Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Prof. Brian W. Surgenor, who has supported me throughout my research with patience and diligence. Guiding me with his expansive knowledge and providing me with valuable insights, he also allowed me to work in my own way and maintain a great deal of independence. One could not simply wish for a more supportive and approachable supervisor.

Sasan Taghizadeh was helpful in developing this project and I must thank him for his organized reports and documentation.

I would also like to acknowledge the financial support provided by Queen’s University.

I wish to express my warmest and sincerest thanks to my lovely wife whose generosity and endless love were vital to the success of my thesis. She has been my inspiration in overcoming all the obstacles that existed throughout the completion of this research; to her I dedicate this thesis.

I wish to thank several individuals for helping me get through difficult times. My Grandma, Aziz, my mom and dad, Nahid and Mehdi, and finally my cousin Ashkan, all provided me with invaluable emotional support, caring, understanding and encouragement. Without their support, I would not have been able to achieve my Master’s degree.
# Table of Contents

Abstract ................................................................................................................................. ii  
Table of Contents .................................................................................................................. iv  
List of Figures ....................................................................................................................... vii  
List of Tables ........................................................................................................................ xi  
List of Nomenclature ............................................................................................................ xiii  
Chapter 1 **Introduction** ........................................................................................................ 1  
1.1 Problem Overview ........................................................................................................... 1  
1.2 Objectives ........................................................................................................................ 1  
1.3 Thesis Outline .................................................................................................................. 2  
Chapter 2 **Literature Review** .............................................................................................. 4  
2.1 Comparison of Different Types of Actuators .................................................................... 4  
  2.1.1 Rodless and Rodded Pneumatic Cylinders ................................................................. 5  
2.2 Model Based vs non-Model Based Control ...................................................................... 6  
2.3 Adaptive Neural Networks (ANN) .................................................................................... 7  
  2.3.1 ANN as a Controller .................................................................................................... 9  
  2.3.2 ANN as a Compensator ............................................................................................. 11  
2.4 Fuzzy Logic Control ........................................................................................................ 13  
  2.4.1 Fuzzy Adaptive PID (FAPID) ................................................................................. 14  
2.5 Adaptive Tuning of ANN and FL Controllers ................................................................. 15  
2.6 Comparison of Pneumatic Position Controllers ............................................................. 19  
2.7 Summary .......................................................................................................................... 24  
Chapter 3 **Apparatus and Controller Implementation** .................................................... 26  
3.1 Apparatus Description ..................................................................................................... 26  
  3.1.1 Sensors and Calibration ............................................................................................. 30  
  3.1.2 Data Acquisition ....................................................................................................... 31  
3.2 Open loop Tests ............................................................................................................... 33  
3.3 PID controller .................................................................................................................. 39  
  3.3.1 PID Tuning ................................................................................................................ 40  
3.4 Adaptive Neural Network (ANN) .................................................................................... 41  
  3.4.1 ANN Algorithm ......................................................................................................... 42  
  3.4.2 ANN Implementation ................................................................................................ 45
6.5 Robustness Tests ........................................................................................................... 132
  6.5.1 Added Mass and Change in Pressure for z-axis ...................................................... 133
  6.5.2 Change in Pressure for x-axis ................................................................................. 136
6.6 Performance Comparisons .............................................................................................. 140
6.7 Summary ......................................................................................................................... 144

Chapter 7 Conclusions and Recommendations ................................................................. 145
  7.1 Conclusions .................................................................................................................. 145
  7.2 Recommendations ......................................................................................................... 149

References ............................................................................................................................ 150

Appendix A z-axis HRTextron Valve Tests ........................................................................ 155
List of Figures

Figure 2-1 Dynamic characteristics of different actuators (Thayer, 1984)..........................5
Figure 2-2 Rodded and rodless pneumatic cylinder ..........................................................6
Figure 2-3 General mathematical neuron model (Demuth et al, 2006).............................8
Figure 2-4 Block diagram of NN controller with adaptive learning algorithm (Burton et al, 1999)........9
Figure 2-5 Simulation results for PID position and force controller (Burton et al, 1999).........10
Figure 2-6 Simulation results for NN position and force controller (Burton et al, 1999).........10
Figure 2-7 Block diagram of PID with NN feedback linearization (Choi et al, 1998)..............12
Figure 2-8 PID result with and without NN feedback linearization (Choi et al, 1998)..............13
Figure 2-9 Histogram of 500 random step input test, showing steady state error .................14
Figure 2-10 Step response comparison of the PD-FLC controller (Khan et al, 2008) .................15
Figure 2-11 Basic structure of self-tuning PID (He et al, 1993)........................................16
Figure 2-12 Block diagram of online tuning gain scheduling DNN-PID (Anh and Nam 2011)......17
Figure 2-13 Structure of MLFNN network system used in online tuning DNN-PID ...............17
Figure 2-14 Sinusoidal response of the PAM robot arm – 0.5 kg load (Left), 2 kg load (right) (Anh and Nam 2011).................................................................18
Figure 2-15 Online tuning convergence of DNN-PID controller parameters..........................19
Figure 2-16 Comparison of experimental results for 0.25 Hz sine wave tracking .....................20
Figure 2-17 Comparison of experimental results for 1 Hz sine wave tracking .........................21
Figure 2-18 Fuzzy control with P feedback for sine wave tracking (Chillari et al, 2001)..........22
Figure 2-19 Performance comparison for different controllers and setpoints .........................22
Figure 2-20 PID+ANNC for 0.1 Hz sine wave tracking (Taghizadeh 2010) .........................23
Figure 2-21 RMSE of PID and PID+ANNC for sine wave tracking at different frequencies, (Taghizadeh 2010)........................................................................................................24
Figure 3-1 Gantry robot with axes and valves labeled, note x-axis valve symmetry ..................27
Figure 3-2 Close-up of gantry robot z-axis .............................................................................28
Figure 3-3 Pneumatic circuit used for z-axis cylinder control .............................................29
Figure 3-4 Schematic of rodless x-axis cylinder .....................................................................30
Figure 3-5 Open loop x-axis 0.75 Hz square wave test, start-up with initial step positive ........35
Figure 3-6 Open loop x-axis start-up trend of cylinder peak pressure, differential pressure and velocity (as taken from Figure 3-5) .................................................................36
Figure 3-7 Open loop x-axis 0.25 Hz square wave test, start up with initial step negative ........37
Figure 3-8 Open loop z-axis 1 Hz square wave test, steady state showing drift in position.................. 38
Figure 3-9 Block diagram for PID controller and pneumatic system............................................. 39
Figure 3-10 Flowchart for PID tuning procedure .............................................................................. 41
Figure 3-11 Three layer ANN with one output (Taghizadeh 2010) ............................................. 43
Figure 3-12 Simulink block diagram for ANN controller (Taghizadeh 2010)............................... 46
Figure 3-13 Flowchart for ANN tuning procedure .......................................................................... 49
Figure 3-14 Elements of a fuzzy logic system .................................................................................. 50
Figure 3-15 Fuzzy logic example with 2 inputs, 3 rules, 1 output(adapted from Mathworks 2011)...... 52
Figure 3-16 Fuzzy logic example of Figure 3-15 illustrating role of membership functions .......... 52
Figure 3-17 Flowchart for fuzzy logic controller design ................................................................. 53
Figure 3-18 Alternate arrangements for FAPID controllers (Callai 2005) ...................................... 54
Figure 3-19 Block diagram for implemented FAPID controller ...................................................... 55
Figure 4-1 Low and high frequency tracking to illustrate effect on performance measures .......... 58
Figure 4-2 Ponly oscillatory response for step tracking (Kp=6) ...................................................... 62
Figure 4-3 PID response for ZN tuning for step tracking (Kp=3.6, Ki=0.2, Kd=0.1) ..................... 62
Figure 4-4 PID response for 0.1 Hz sine wave tracking (amp=300 mm) ....................................... 63
Figure 4-5 Simulink model of the x-axis pneumatic cylinder with fuzzy controller ...................... 65
Figure 4-6 Fuzzy rule model for the x-axis pneumatic cylinder ..................................................... 65
Figure 4-7 Fuzzy simulation response for step tracking ................................................................. 66
Figure 4-8 Fuzzy simulation response for 0.16 Hz sine wave tracking .......................................... 66
Figure 4-9 Block diagram for the x-axis fuzzy controller ............................................................... 68
Figure 4-10 Fuzzy rule surface for experimental test ....................................................................... 69
Figure 4-11 Fuzzy membership functions for −es, ve and ut .......................................................... 70
Figure 4-12 Fuzzy controller response for step tracking ................................................................. 71
Figure 4-13 Block diagram for PID+ANNC and pneumatic system ................................................ 73
Figure 4-14 Input vector for ANNC ................................................................................................. 74
Figure 4-15 PID+ANNC response for 0.5 Hz sine wave tracking with tuned ANNC .................... 78
Figure 4-16 PID+ANNC response for 0.2 Hz sine wave tracking with tuned ANNC .................... 79
Figure 4-17 PID+ANNC response for 0.1 Hz sine wave tracking with tuned ANNC .................... 79
Figure 4-18 PID+ANNC response showing transient with events A, B and C highlighted ............ 82
Figure 4-19 Block diagram for ANNonly controller and pneumatic system .................................... 83
Figure 4-20 Input vector for ANNonly controller ............................................................................. 84
Figure 4-21 ANNonly controller response for 0.5 Hz sine wave tracking .................................... 86
Figure 4-22 ANNonly controller response for 0.2 Hz sine wave tracking .................................... 86
Figure 6-9 x-axis MPID for 0.5 Hz sine wave tracking, note $Kd$ is in adaptation phase

Figure 6-10 z-axis untuned PI ($Kp=2$, $Ki=2$) for 0.1 Hz sine wave tracking, adding 12 Kg mass at $t=15s$ ($Ps=500$ kPa)

Figure 6-11 z-axis untuned PI ($Kp=2$, $Ki=2$) + tuned ANNC ($Kv=Kz=\lambda=4.5$) for 0.1 Hz sine wave tracking, adding 12 Kg mass at $t=15s$ ($Ps=500$ kPa)

Figure 6-12 z-axis MFAPID response for 0.5 Hz sine wave tracking ($Ps=500$ kPa)

Figure 6-13 z-axis MFAPID response for 0.5 Hz sine wave tracking ($Ps=280$ kPa)

Figure 6-14 x-axis PID+ANNC response for 0.5 Hz sine wave tracking ($Ps=280$ kPa)

Figure 6-15 x-axis Adaptive PID response for 0.5 Hz sine wave tracking, note $Kp$ in learning, ($Ps=280$ kPa)

Figure 6-16 z-axis bargraph comparison of sine wave tracking results ($Ps=280$ kPa)

Figure 6-17 z-axis bargraph comparison of sine wave tracking results ($Ps=500$ kPa)

Figure 6-18 x-axis bargraph comparison of sine wave tracking results ($Ps=500$ kPa)

Figure A-1 Open loop z-axis 2.5 Hz square wave test with HRTextron valve

Figure A-2 Open loop z-axis 2.5 Hz square wave test with Festo valve

Figure A-3 P-only response for step tracking with HRTextron valve ($Kp=0.6$)

Figure A-4 P-only response for step tracking with HRTextron valve ($Kp=0.6$, $Ki=0.2$)

Figure A-5 PID response for step tracking with HRTextron valve

Figure A-6 Technical specifications of Bimba cylinder
List of Tables

Table 2-1 Tuning methods of PID controller (adopted from Wikipedia, 2012) .................................................. 7
Table 2-2 AVGE values in mm for 0.5 Hz sine wave tracking for various amplitudes ........................................ 12
Table 2-3 AVGE values in mm for sine wave tracking for various frequencies with amp=70 mm (Choi et al, 1998) ................................................................. 12
Table 2-4 Comparison of RMSE in mm values for horizontal moves with the nominal payload .................... 20
Table 3-1 Summary of calibration equations for sensors (Taghizadeh 2010) ................. 31
Table 3-2 Summary of main features of major components used in the apparatus .................. 32
Table 3-3 Channel assignments for output signals .................................................................................. 32
Table 3-4 Channel assignments for input signals .................................................................................. 33
Table 3-5 Heuristic rules of thumb for tuning PID controller (Chong et al, 2005) ................. 40
Table 3-6 ANN model parameters .................................................................................................. 46
Table 4-1 Performance measures for tracking example given in Figure 4-1 .................. 59
Table 4-2 Ziegler/Nichols tuning rules (Ogata 1997) ................................................................. 60
Table 4-3 Summary of performance results for PID tuned with ZN rules for step tracking .... 61
Table 4-4 Summary of performance results for PID tuned by hand for sine wave tracking, .... 63
Table 4-5 Fuzzy rules governing u, based on v, and -e, fuzzy inputs ........................................ 69
Table 4-6 Summary of fuzzy and PID+fuzzy results for sine wave and step tracking .......... 71
Table 4-7 Tuned values of ANNC parameters reported by Taghizadeh (2010) .............. 75
Table 4-8 Effect of changing ANNC parameters from tuned values in Table 4-7 .......... 76
Table 4-9 PID+ANNC tuning results for different frequencies (amp=250 mm), best results in Bold .... 77
Table 4-10 Summary of improvement in PID+ANNC performance relative to .................... 80
Table 4-11 Summary of ANNonly performance results for different sine wave frequencies 85
Table 4-12 Fuzzy rules for PID gains taken from Manjunathm and Janaki (2011) ............ 89
Table 4-13 Fuzzy rules for Kd ........................................................................................................ 92
Table 4-14 Fuzzy rules for Ki ........................................................................................................ 93
Table 4-15 Summary of FAPID performance results for different sine wave frequencies 94
Table 5-1 Summary of ANNC performance results for 0.1 Hz sine wave tracking ................. 107
Table 5-2 Summary of ANNonly performance results for different sine wave frequencies 108
Table 5-3 Summary of FAPID performance for different frequencies with comparison to Manjunathm and Janaki (2011) ................................................................. 114
Table 6-1 Change in x-axis sine wave response compared to Taghizadeh (2010) .......... 118
Table 6-2 Change in $z$-axis sine wave response over 5 months................................................................. 118
Table 6-3 Sign rules for weight update $\Delta w (i)$ .................................................................................................. 125
Table 6-4 $x$-axis comparison of 0.5 Hz sine wave tracking results for two supply pressures..................... 137
Table 6-5 Comparison of best results of different studies for sine wave tracking ........................................... 143
Table A-1 Comparison summary of step tracking performance under two supply pressure for HRTextron and Festo valve ........................................................................................................................................ 159
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>amp</td>
<td>Amplitude of the input signal</td>
</tr>
<tr>
<td>$a_{ii}$</td>
<td>Elements of matrix $A$</td>
</tr>
<tr>
<td>$A$</td>
<td>Sample matrix</td>
</tr>
<tr>
<td>$A_c$</td>
<td>Cross section of the cylinder</td>
</tr>
<tr>
<td>$b_i^2$, $b_i^3$</td>
<td>Bias of node $i$ in the hidden and output layer, respectively</td>
</tr>
<tr>
<td>bias</td>
<td>Activation function bias</td>
</tr>
<tr>
<td>$B^L$</td>
<td>Number of nodes in layer $L$</td>
</tr>
<tr>
<td>$C_{fx}$</td>
<td>Viscous friction coefficient along $x$ axis</td>
</tr>
<tr>
<td>$C_v$</td>
<td>Coefficient of velocity</td>
</tr>
<tr>
<td>$e_x$, $e_z$</td>
<td>Position error of the $x$-axis and $z$-axis, respectively</td>
</tr>
<tr>
<td>$F$</td>
<td>Learning rate of $W$</td>
</tr>
<tr>
<td>$f_{ax}$</td>
<td>Applied force to the $x$ cylinder</td>
</tr>
<tr>
<td>$f_x$</td>
<td>Coulomb friction force along $x$-axis</td>
</tr>
<tr>
<td>$F_x$</td>
<td>Contact force along the $x$-axis</td>
</tr>
<tr>
<td>$F_{wx}$</td>
<td>Position/Velocity sensor wire force along the $x$-axis</td>
</tr>
<tr>
<td>$G$</td>
<td>Learning rate of $V$</td>
</tr>
<tr>
<td>$h$</td>
<td>Sampling time</td>
</tr>
<tr>
<td>$K_p$</td>
<td>Differential pressure gain</td>
</tr>
<tr>
<td>$K_p$, $K_i$, $K_d$</td>
<td>Proportional, integral and derivative gains</td>
</tr>
<tr>
<td>$K_v$, $K_z$</td>
<td>Tunable gains in Neural Network</td>
</tr>
<tr>
<td>$K_u$</td>
<td>Ultimate gain for ZN</td>
</tr>
<tr>
<td>$L$</td>
<td>Layer in NN, where input $L=1$, hidden $L=2$, output $L=3$ for a 3 layered NN</td>
</tr>
<tr>
<td>$m_r$</td>
<td>Manual reset</td>
</tr>
<tr>
<td>$n_h$, $n_i$, $n_o$</td>
<td>Number of nodes in hidden, input and output layers (for Simulink®)</td>
</tr>
<tr>
<td>$net_i^L$</td>
<td>Sum of the inputs to node $i$ in layer $L$</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of samples</td>
</tr>
<tr>
<td>$p_j$</td>
<td>Input $j$ to the hidden layer from the input layer</td>
</tr>
<tr>
<td>$P$</td>
<td>NN input vector</td>
</tr>
<tr>
<td>$P_a$, $P_b$</td>
<td>Absolute pressure of the side “a” and “b” of the cylinder, respectively</td>
</tr>
<tr>
<td>$P_s$</td>
<td>Supply pressure</td>
</tr>
<tr>
<td>$r$</td>
<td>Sensor reading</td>
</tr>
<tr>
<td>ref</td>
<td>Reference</td>
</tr>
<tr>
<td>$Rep$</td>
<td>Repetition of tuning of one gain</td>
</tr>
<tr>
<td>$S$</td>
<td>Slope of sigmoid activation function</td>
</tr>
<tr>
<td>$T$</td>
<td>period</td>
</tr>
<tr>
<td>$T_u$</td>
<td>Ultimate period for ZN</td>
</tr>
</tbody>
</table>
$u_{PID}$  PID control signal  
$u_x, u_z$  Control signal of the $x$-axis and $z$-axis, respectively  
$u_{NN}$  NN output  
$v_x, v_z$  Velocity, $x$-axis and $z$-axis, respectively  
$V, W$  Weight vector for the hidden layer and output layer, respectively  
$V_{Lim}, W_{Lim}$  Limits of $V$ and $W$, respectively  
$V^2_{ij}$  Weight connecting node $i$ in the hidden layer and input $p_j$  
$W_p, W_i, W_d$  Proportional, integral and derivative fuzzy weights  
$W_{3i,j}$  Weight connecting node $i$ in the output layer and node $j$ in the hidden layer  
$x, x_s$  $x$-axis position and position setpoint, respectively  
$z, z_s$  $z$-axis position and position setpoint, respectively  
$Z$  Tunable parameter in Neural Network  

**Acronyms**  

ANN  Adaptive Neural Network  
ANNC  Adaptive Neural Network Compensator  
BPA  Back Propagation Algorithm  
DOF  Degrees Of Freedom  
DZC  Deadzone Compensation  
FF  Feed Forward  
FAPID  Fuzzy Adaptive PID  
FL  Fuzzy Logic  
GA  Genetic Algorithm  
GUI  Graphical User Interface  
I/O  Input/output  
Ionly  Integral Only  
MFAPID  Modified Fuzzy Adaptive PID  
MBPM  Modified Back Propagation Method  
MF  Membership Function  
MNN  Multilayer Neural Network  
NN  Neural Network  
Ponly  Proportional Only  
PFC  Proportional Flow Control  
PID  Proportional Integral Derivative  
PPC  Proportional Pressure Control  
PWM  Pulse Width Modulation  
RMSE  Root Mean Square Error  
SMC  Sliding Mode Control  
ZN  Ziegler-Nichols
Greek Letters

$RMS_i$  Change in the RMSE for case $i$

$P_x, P_z$  $x$-axis, $z$-axis differential pressure, respectively

$\lambda$  Adaptation parameter

$\sigma_i$  Activation function for node
Chapter 1

Introduction

1.1 Problem Overview

Pneumatic actuators are attractive for control applications where there is a need for lower installed cost, longer working life, continuous duty cycles or reduced risk of explosion. But the problem with pneumatic actuators is their accuracy is low relative to electric and hydraulic actuators. A large body of research is devoted to the application of advanced control techniques to servo pneumatics in order to improve their performance (Bone and Ning, 2007). However, nonlinearities continue to limit their performance. As one source of nonlinearity, friction can have a significant effect on tracking performance, especially in applications that use rodless cylinders which have higher Coulomb friction than rodded cylinders. Model-based compensation strategies are difficult to understand and implement, as they need a system model. For this reason, “intelligent” algorithms that don’t use system models are attractive. They possess learning, adaptation, and classification capabilities that improve control performance. Well known types of intelligent controllers employ Neural Network (NN) and Fuzzy Logic (FL) algorithms. NN algorithms capture the parallel processing and learning capability of biological nervous systems. FL algorithms capture the decision making capabilities of human linguistic and cognitive systems. It is believed that NN and FL algorithms have the potential to improve the performance of pneumatic systems as non-model based controllers. FL controllers are rule based controllers that do not require a formal mathematical model of the system to be controlled. NNs provide a means to generate an input-output model of system, again without requiring the user to develop a formal mathematical model. Furthermore, adding an adaptive feature to these intelligent methods can improve performance still further, where the controller gains are constantly updated to account for changes in operating conditions and system parameters.

1.2 Objectives

Taghizadeh (2010) conducted research on position/velocity control of a pneumatic gantry robot apparatus with a PID controller coupled with an Adaptive Neural Network compensator
He implemented his set up on a rodless pneumatic cylinder. Although experimental results were better than with conventional PID control (by 20%), it was concluded that the degree of improvement with ANNC did not warrant the extra effort in tuning and implementation. Also, his original premise for ANNC was that it would be self-tuning due to its adaptive nature. This was judged to be misleading because of the extensive setup and tuning that was required. The underlying purpose of this thesis is to improve on the experiments of Taghizadeh and test other adaptive methods. In addition, effort will be made to make the tuning procedure of the adaptive methods less time consuming.

The first objective of this thesis is to reevaluate the PID+ANNC position controller that was developed by Taghizadeh. Tests will be conducted on both the \textit{x-axis} and \textit{z-axis} of the gantry robot. The former is a long stroke high friction rodless cylinder. The latter is a short stroke low friction rodded cylinder.

The second objective is to evaluate the potential of a fuzzy controller on both axes. The fuzzy adaptive PID controller is the fuzzy version of the traditional PID. The PID gains will be adjusted online with a set of fuzzy rules. This method benefits from the inherent robustness of PID and the adaptive nature of the fuzzy algorithm.

Overall, the objective of this thesis was to explore different adaptive intelligent controllers for position control of a pneumatic system. Comparative results will be given for five different controllers: 1) manually tuned fixed gain PID, 2) adaptive PID (MPID), 3) PID+adaptive neural network compensator (ANNC), 4) fuzzy adaptive PID (FAPID) and 5) modified fuzzy adaptive PID (MFAPID). To provide a measure of robustness, experiments will be conducted at two different supply pressures and three different tracking frequencies.

\textbf{1.3 Thesis Outline}

The organization of the thesis is as follows:

- Chapter 2 presents a literature review on six subjects in pneumatic system position control: 1) comparison of different types of actuators, 2) model based vs. non-model based control, 3) Adaptive Neural Network (ANN) control and compensation, 4) Fuzzy Logic (FL) control and Fuzzy Adaptive PID (FAPID), 5) adaptive tuning of ANN and FL control, and 6) comparison of pneumatic position controllers.
• Chapter 3 provides background on the apparatus including open loop tests of both axes. Details on the ANN, FL control, and FAPID will be given including details on the implementation and tuning.

• In Chapter 4 the x-axis rodless cylinder is used to obtain results for position control. Five different controllers are tested and their performance compared: 1) PID, 2) FL control, 3) PID+ANN, 4) ANNonly control, 5) FAPID control.

• In Chapter 5 the z-axis rodded cylinder is used to obtain results for position control. The differences between x-axis and z-axis two cylinders will be explained and the necessary modifications for the z-axis will be reported. Similar to the previous chapter, a comparison is provided of the performance of: 1) PID, 3) PID+ANN, 4) ANNonly control, 4) FAPID control.

• In Chapter 6 the problem with hand-tuning of FAPID will be discussed and an additional adaptive mechanism will be presented as a solution. The proposed modification will be applied to FAPID and PID. Results will be given to show the robustness of this adaptive approach under different operating conditions.
Chapter 2

Literature Review

Position control of pneumatic cylinders is difficult due to low bandwidth and high nonlinearity due to air compressibility and Coulomb friction effects. Researchers have been struggling to deal with pneumatic nonlinearities since the early 1970’s. This chapter presents a literature review on six subjects: 1) comparison of different types of actuators, 2) model based vs. non-model based control, 3) Adaptive Neural Network (ANN) control and compensation, 4) Fuzzy Logic control (FL) and fuzzy adaptive PID (FAPID), 5) adaptive tuning of ANN and FL control, and 6) comparison of pneumatic position controllers.

2.1 Comparison of Different Types of Actuators

Industrial actuators are either electric, hydraulic or pneumatic. There is no general rule for selection of the type of actuation for a specific application. However, there are some rules of thumb that help engineers in these situations. For example, applications that involve movement of heavy payloads are best handled by hydraulic drives. Pneumatic and electric drives are best suited for medium to light payloads. Figure 2-1 provides a quantitative comparison of power level versus dynamic response of these types of actuators (Thayer 1984). This illustrates that hydraulic actuators are applicable for heavy payloads and pneumatic/electric have similar characteristics.

Another major factor in selection of an actuator is its cost. Initial costs of a pneumatic servo positioning system are approximately 50% lower than that of their electric counterpart, and 25 to 35% less than the equivalent hydraulic system (Burrows 1972, Chapter 1). What Figure 2-1 doesn’t show is the relative accuracy of the systems. Generally hydraulic and electric systems are more accurate than pneumatic systems.
2.1.1 Rodless and Rodded Pneumatic Cylinders

In this thesis the focus is on rodless and rodded pneumatic cylinder position control. Rodless cylinders were originally developed to overcome the space limitations of conventional rodded cylinders. A rodless cylinder, true to its name, has no piston rod, being constructed with a slide table assembly mounted directly above the piston. Thus, the length of the cylinder assembly is fixed. In a rodded cylinder when the rod is fully out stroked, the length of the cylinder assembly will almost double (length of rod plus length of cylinder housing).

Rodless cylinders are more expensive than rodded cylinders due to more complicated construction and costly sealing. In addition, there is another disadvantage due to their special structure and sealing, namely cause more surface friction compared to rodded cylinders. This in turn will increase the amount of nonlinearities associated with the dynamical model of the system. Figure 2-2 shows a typical rodded and a rodless pneumatic cylinder.
2.2 Model Based vs non-Model Based Control

Controllers and compensation techniques are classified into model based and non-model based approaches. Although model-based techniques show relatively good results, the requirement for a system model can make these methods difficult to implement. If a simplified mathematical model is used, then performance is sensitive to uncertainties and parameter variations. This highlights the need for an adaptive controller that is not based on a mathematical model. For this reason, “intelligent” algorithms such as neural networks and fuzzy rule based controllers that don’t use system models are attractive. The adaptive feature in these controllers has improved performance, where the controller gains are constantly updated to account for changes in operating conditions and system parameters. Further details on neural networks and fuzzy control will be given in the next two sections.

The most popular controller is PID. Although its performance is limited by its constant gains, it is relatively easy to tune. In Table 2-1, model-based and non-model based PID tuning procedures are summarized. In this thesis Ziegler-Nichols (ZN) tuning will be used to obtain limited estimates of PID gains.
Table 2-1 Tuning methods of PID controller (adopted from Wikipedia, 2012)

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model based tuning methods</td>
<td>Software tools</td>
<td>Consistent tuning. Online or offline method. May include valve and sensor analysis. simulation.</td>
</tr>
</tbody>
</table>

No matter which of the model-based tuning methods is chosen for the apparatus, one has to consider the ageing, change in the operating conditions, time variance of the system. Thus compensations seem necessary for these variations.

2.3 Adaptive Neural Networks (ANN)

An Artificial Neural Network is a mathematical model or computational model that is inspired by the functional aspects of biological neural networks. In most cases a NN is an adaptive system that updates its structure based on external or internal information, during the learning phase, that flows through the network. Modern neural networks are non-linear statistical data modeling tools which can find complex relationships between inputs and outputs (Spooner et al, 2002).

A typical NN has three layers. The first layer has input neurons (nodes), which send data via synapses (weights) to the second layer of neurons, and then via more synapses to the third layer of output neurons. NN is typically defined by three types of parameters:

- The interconnection pattern between different layers
- The learning algorithm for updating the weights
- The activation function that converts a neuron's weighted input to its output activation
Figure 2-3 shows the mathematical model, where $p_i$ is the input $i$, $V_{1,1}$ is the weight connecting $p_1$ and node 1. $\sigma_1$ and $b_1$ are the activation function and the bias of node 1, respectively, $net_1$ is the sum of the inputs to node 1 and $n_i$ is the number of inputs (Demuth et al, 2006). One of the most common activation functions is the sigmoid function as given by:

$$\sigma_i(net_i) = \frac{1}{1 + e^{-net_i}}$$

(2-1)

![General mathematical neuron model (Demuth et al, 2006)](image)

Networks without feedbacks are commonly called feedforward, because their graph is a directed acyclic graph; i.e. information goes directly from the input layer to the output layer. On the other hand, networks with cycles are commonly called recurrent where there is a synapse from a neuron to itself or to neuron in its layer.

NN is widely used in applications such as function approximation, classification, data processing, systems identification and control. Since nonlinear modeling and learning from input-output examples is in NN nature, NNs have been used, as a non-model-based-approach candidate. It is worthwhile to mention that NN is a tool for adaptive control of nonlinear systems with time varying dynamics (Norgaard et al, 2002).
2.3.1 ANN as a Controller

NNs have been widely applied to system identification and indirect control of nonlinear systems, while less work had been presented about the use of NN in direct controllers (Huang and Lewis, 2003). NN as a controller type usually requires online training. Examples of direct NN are: direct inverse model control, feedforward with NN inverse model control and NN internal model control.

Burton et al (1999) worked on three different applications; covering a wide degree of complexity for neural network control of hydraulic servo-systems. They tested three different controllers for a hydraulic actuator system: 1) quasi-open-loop pattern follower, 2) PID multiple-gain neural controller (NN as a compensator), and 3) a neural-based controller for a very complex multiple-input multiple-output system.

Figure 2-4 illustrates the block diagram of their NN controller. The control parameters for the MIMO system were force and position. They used a mathematical model as a reference model in parallel with the actual model in order to reduce coupling effects. The figure shows that they also added a network learning algorithm to provide updates to the NN controller weights.

![Figure 2-4 Block diagram of NN controller with adaptive learning algorithm (Burton et al, 1999)](image-url)
The NN was trained offline in simulation and were fine-tuned later in the process of experimentation. Figure 2-5 provides experimental results for the PID position and force controller. The setpoint was a 1 Hz sinusoidal signal for both position (-50±250 mm amplitude) and force (1750±1500 N amplitude). Figure 2-6 gives the NN controller result for the same 1 Hz sinusoidal signal on position (±300 mm amplitude) and force (1750±1250 N amplitude). No quantitative performance measures were provided. However, by comparing Figure 2-5 and Figure 2-6 qualitatively, NN controller shows better performance in terms of setpoint tracking and reduced oscillations.

Figure 2-5 Simulation results for PID position and force controller (Burton et al, 1999)

Figure 2-6 Simulation results for NN position and force controller (Burton et al, 1999)
They addressed another important issue in their paper, in which some pre-training must be done to ‘kick start’ the system at an acceptable starting point. Techniques are being developed which facilitate a phase-in process of the neural network control to a standard PID controller.

2.3.2 ANN as a Compensator

Compensation for nonlinearities in pneumatic systems has been a popular area of research in control. Most of the compensation strategies use model based algorithms. Although they show relatively good results, like model reference adaptive controllers (MRAC) and self-tuning regulators, the requirement for model parameter identification has made these methods difficult to implement.

The difficulty of system parameter identification has been the motivation to conduct research on non-model based compensators. Success has been reported with the use of NNs for the compensation of system nonlinearity, including lag in Huang and Lewis (2003) for a simulated electric robot manipulator system.

In pneumatic control, Choi et al (1998) studied feedback linearization by means of a NN for position control. The apparatus consisted of a double-acting rodless cylinder (stroke = 200 mm, diameter = 25 mm) and a five port three way Proportional Flow Control (PFC) valve. The controller had an inner PID pressure controller and an outer PID position controller with indirect NN for feedback linearization. Figure 2-7 gives the block diagram of the controller.

The NN was trained offline with Back Propagation Algorithm (BPA). Figure 2-8 shows the PID controller result with and without NN feedback linearization for a sinusoidal wave at 0.5 Hz. Tables 2-1 and 2-2 provides the Absolute Average Error (AVGE) results for different amplitudes and frequencies. The degree of improvement, as measured by the reduction in average error (AVGE) is noticeable and nominally on the order of 75%.
Figure 2-7 Block diagram of PID with NN feedback linearization (Choi et al, 1998)

Table 2-2 AVGE values in mm for 0.5 Hz sine wave tracking for various amplitudes (Choi et al, 1998)

<table>
<thead>
<tr>
<th>Amplitude (mm)</th>
<th>PID control</th>
<th>PID + Feedback Linearization</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>5.106</td>
<td>1.150</td>
</tr>
<tr>
<td>50</td>
<td>6.320</td>
<td>1.643</td>
</tr>
<tr>
<td>70</td>
<td>8.621</td>
<td>2.239</td>
</tr>
</tbody>
</table>

Table 2-3 AVGE values in mm for sine wave tracking for various frequencies with amp=70 mm (Choi et al, 1998)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>PID control</th>
<th>PID + Feedback Linearization</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2.391</td>
<td>0.798</td>
</tr>
<tr>
<td>0.2</td>
<td>3.305</td>
<td>0.847</td>
</tr>
<tr>
<td>0.5</td>
<td>8.621</td>
<td>2.239</td>
</tr>
</tbody>
</table>
2.4 Fuzzy Logic Control

Ruihua et al (2004) used a multi-region fuzzy algorithm for a pneumatic servo force control system with unknown non-linearity and changing process dynamics. The authors showed better performance with a multi-region approach that with a conventional fuzzy rule approach. They introduced an auxiliary variable to their fuzzy unit, in addition to error and error derivative. They did not compare their method to any other type of controller and did not provide quantitative performance results, as well. Unfortunately, the procedure for attaining the fuzzy maps was not documented so their results could not be reproduced.
2.4.1 Fuzzy Adaptive PID (FAPID)

In Gao and Feng (2005) a new adaptive fuzzy-PD was used for position control of a rodless pneumatic servo system. It was claimed that this method could compensate for the nonlinear and time-varying characteristics of friction of the apparatus. A new term $M_a$ was introduced which made the fuzzy maps adaptive by changing its value during the course of an experiment. For various step size inputs (200, 500, 800 mm), they showed that the responses can reach the set point in less than 1 s without overshoot or with a little overshoot. The steady output error was less than 0.3 mm for all cases. Figure 2-9 tries to illustrate the maximum steady error of position controlling of 500 random step inputs. The figure is a histogram summary of these tests and shows that the maximum error for 500 experiment trials was less than 0.3 mm.

![Figure 2-9 Histogram of 500 random step input test, showing steady state error (Gao and Feng 2005)](image)

Manjunathm and Janaki (2011) designed a two-input and three-output adaptive fuzzy PID controller for a flow control application. Simulation results showed that the fuzzy adaptive controller had reduced overshoot and a faster response compared to conventional PID. The fuzzy rules in this study are used in this thesis as starting point for the fuzzy adaptive PID. However,
they neither explained how they came up with their fuzzy rules nor provided any quantitative results.

2.5 Adaptive Tuning of ANN and FL Controllers

One of the candidates for adaptive tuning of the gains of PID or the membership functions of Fuzzy Adaptive PID is a genetic algorithm (GA) approach.

The design of an optimal Proportional Derivative fuzzy logic controller (PD-FLC) using a GA technique was documented in Khan et al (2008). The application was a temperature control system where fan speed was regulated in response to variations in the ambient system temperature. A GA approach was taken to optimize the inference rules, membership functions and scaling gains. The robustness and adaptability of the PD-FLC control system, was reported as “acceptable”. The performance metrics for the comparison of the conventional PID controller, fuzzy logic PD controller and GA-optimized fuzzy logic PD controller were overshoot, rise time, settling time and steady state error. Figure 2-10 shows the effectiveness of their proposed optimizer in terms of sample step response. Note that only simulation results were reported.

![Figure 2-10 Step response comparison of the PD-FLC controller (Khan et al, 2008)]

The use of any optimization based GA technique, that uses an optimized search method in the envelop of achievable gains, is always risky in experimental applications. Optimization techniques can generate PID gains which when implemented, can cause instability for high
values of $K_p$ and high frequency oscillations for high values of $K_d$. A review of the GA literature also reveals that these methods tend to have long convergence times which make the real-time optimization application not feasible. For example, Wang et al (2004) showed long convergence time problems with GA with a random based optimization process. The application was identification of pneumatic cylinder friction parameters.

Another example of a PID type fuzzy controller with an online autotuner can be found in He et al (1993). The essential idea of their scheme is to parameterize a Ziegler-Nichols-like tuning formula by a single parameter “$\alpha$”, then to use an online fuzzy inference mechanism to self-tune the parameter. Figure 2-11 demonstrates their proposed scheme. It is evident that the fuzzy self-tuning mechanism will generate an $a(t)$ given the instant values of $e(t)$ and $de(t)/dt$ at time “t” using fuzzy logic.

![Fuzzy Self-tuning Mechanism](image)

Figure 2-11 Basic structure of self-tuning PID (He et al, 1993)

A comparison was made between 1) PID, 2) fuzzy PID and 3) fuzzy self-tuning PID in terms of set-point and load disturbance responses. They simulated the responses for generic type of processes and demonstrated the effectiveness of the fuzzy self-tuning approach; but no quantitative conclusions were drawn.

Anh and Nam (2011) proposed a novel online-tuning Gain Scheduling Dynamic Neural Network PID (DNN-PID) Controller using multi-layer neural network with a view of controlling the joint angle of the highly nonlinear pneumatic artificial muscle (PAM) manipulator. They claimed that
the DNN-PID controller has simple dynamic self-organizing structure, fast and flexible online-tuning speed. The PAM manipulator used in this paper is a two-axis, closed-loop activated with 2 antagonistic PAM pairs which are pneumatic driven controlled through 2 proportional valves.

Figure 2-12 depicts the block diagram of tuning gain scheduling DNN-PID position controller. Figure 2-13 provides more details of what is happening inside the controller and how the PID gains are assigned by the help of multi later neural network.

For updating weight values the error defined by the gradient descent method was used, as follows:

\[ E(k) = \frac{1}{2}(y_{ref}(k) - y(k))^2 \]  \hspace{1cm} (2-1)
They utilized two activation functions for updating the NN weights; DNN-PID-SIG and DNN-PID-HYP possessing of Sigmoid and Hyperbolic Tangent activation functions, respectively. They carried out experiments for triangle, trapezoidal, and sinusoidal trajectories with 0.5 and 2 kg loads to examine the robustness and effectiveness of their proposed controllers. Figure 2-14 verifies the effectiveness of the proposed DNN-PID controller with sinusoidal reference of 0.05 Hz in both cases of load. The online tuning of each control parameter (G, Kp, Ki and Kd) in 2 cases of load 0.5 kg and Load 2 kg are also shown in Figure 2-15. Unfortunately, they didn’t provide the results in terms of AVGE or RMSE and just reported the range of the error which is 1 deg for DNN-PID-HYP for both loads and shows noticeable improvement compared to conventional PID controller.

Figure 2-14 Sinusoidal response of the PAM robot arm – 0.5 kg load (Left), 2 kg load (right) (Anh and Nam 2011)
They concluded that the controller had an adaptive control capability and the control parameters were optimized via the back propagation algorithm which does not need any offline training. Also, the proposed DNN-PID-HYP controller was considered to possess better performance than the proposed DNN-PID-SIG one.

2.6 Comparison of Pneumatic Position Controllers

Ning and Bone (2005) conducted an experimental comparison of two servo pneumatic position control algorithms: PVA + feedforward (FF) + deadzone compensation (DZC) and Sliding Mode Control (SMC). They used a rodless cylinder with a Proportional Flow Control (PFC) valve. They investigated different payloads (1.9, 5.8 and 10.8 kg) with amplitudes from 3 to 250 mm. In all cases SMC outperformed PVA+FF+DZC, shown in Table 2-4 with PVA, and on average the RMSE was 59% less. The table provides the RMSE values for the two controllers with no mismatch between the nominal and actual payload masses for 70 mm amplitude sine wave tracking at different frequencies.
Table 2-4 Comparison of RMSE in mm values for horizontal moves with the nominal payload

<table>
<thead>
<tr>
<th></th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>Test 5</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0.25Hz Sine Wave Trajectory</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PVA*</td>
<td>0.840</td>
<td>0.721</td>
<td>0.712</td>
<td>0.738</td>
<td>0.670</td>
<td>0.736</td>
</tr>
<tr>
<td>SMC</td>
<td>0.364</td>
<td>0.333</td>
<td>0.334</td>
<td>0.338</td>
<td>0.340</td>
<td>0.342</td>
</tr>
<tr>
<td><strong>0.5Hz Sine Wave Trajectory</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PVA*</td>
<td>1.017</td>
<td>1.064</td>
<td>1.058</td>
<td>0.994</td>
<td>1.029</td>
<td>1.032</td>
</tr>
<tr>
<td>SMC</td>
<td>0.358</td>
<td>0.379</td>
<td>0.399</td>
<td>0.404</td>
<td>0.401</td>
<td>0.388</td>
</tr>
<tr>
<td><strong>1.0Hz Sine Wave Trajectory</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PVA*</td>
<td>1.659</td>
<td>1.618</td>
<td>1.873</td>
<td>1.78</td>
<td>1.764</td>
<td>1.739</td>
</tr>
<tr>
<td>SMC</td>
<td>0.776</td>
<td>0.725</td>
<td>0.710</td>
<td>0.825</td>
<td>0.733</td>
<td>0.754</td>
</tr>
</tbody>
</table>

They commented that the chatter in their control signal acted as a dither signal, reducing the stick-slip friction problem, and improving the tracking performance. They reported that their cylinder had a static friction force of about 85 N and a dynamic friction force of about 35 N.

In Figures 2-16 and 2-17 one can compare the performance of their proposed control schemes for 0.25 and 1 Hz sine wave tracking, and witness the level of chatter in their control signal. The documentation of how they designed and implemented their proposed SMC was not given. However, since they are using a linearized model of their apparatus, the controller is considered a model based approach.

![Figure 2-16 Comparison of experimental results for 0.25Hz sine wave tracking](Ning and Bone 2005)
Chillari et al (2001) conducted several experiments on pneumatic system control. They examined PID, Fuzzy, Sliding mode and Neuro-Fuzzy controllers. Experimental results for these controllers applied to different setpoint trajectories were presented. Main parts of the apparatus were: rodded pneumatic cylinder (stroke = 200 mm, diameter = 25 mm) and two pairs of on/off solenoid valves.

The controllers were tested on sinusoidal, square, saw-tooth and staircase input signals. In their work, Chillari et al adopted a differential pressure (ΔP) feedback signal in order to compensate for external disturbances and also friction forces that would act against the motion. Figure 2-18 shows the Fuzzy control with ΔP feedback for a sine wave. Unfortunately, they did not present a figure which shows the controller without the ΔP feedback. The model-based reduced order sliding mode controller used estimation of velocity, the derivation of position signal, computed by first order filter.
Figure 2-18 Fuzzy control with P feedback for sine wave tracking (Chillari et al, 2001)

Figure 2-19 presents a quantitative performance comparison of the different controllers based on the standard deviation between the desired and the actual position signal in m. According to the figure, the error increases as the frequency of the signal increases. Also, it is apparent that the sawtooth signal is harder to control. Adoption of the P feedback improved the performance of the fuzzy controller. However, its main drawback is the cost of introducing a pressure sensor to the system. To avoid this extra cost, Neuro-Fuzzy control was implemented in which the pressure signal was estimated with neural network. The performance of the fuzzy controller with the NN estimate of P was comparable to that of the Fuzzy controller with real P feedback.

Figure 2-19 Performance comparison for different controllers and setpoints (Chillari et al, 2001)
The fuzzy+ Controller was found to provide the best performance with a reported root mean square error (RMSE) of 0.3 mm for a 0.2 Hz sinusoidal reference signal with a 70 mm as the amplitude.

Taghizadeh (2010) evaluated experimentally two NN compensators that were originally implemented in simulation by Abu-Mallouh (2008). He applied the approach to position and velocity control of the x-axis of a gantry robot. Performance was reported quantitatively, reported in AVGE and RMSE, and was compared with the performance of a conventional PID controller.

The results of their study were disappointing, with only a 20% improvement in tracking performance when ANN was employed as a compensator for the PID controller. The conclusion was that this degree of improvement with ANN did not warrant the extra effort required for tuning and implementation. Figure 2-20 shows the PID+ANNC controller results for 0.1 Hz sine wave tracking. The ANNC control signal $u_{ANNC}$ is plotted to show its contribution relative to the overall control signal $u$. The high amplitude of the error signal is apparent in this figure.

Figure 2-20 PID+ANNC for 0.1 Hz sine wave tracking (Taghizadeh 2010)

Figure 2-21 provides a comparison of the performance of PID and PID+ANNC for different tracking frequencies, as measured by RMSE. The addition of ANNC to the PID controller is seen
to improve performance, with the drop in \textit{RMSE} ranging from 15\% to 26\%. He reported tuned PID gains, $K_p=2.25$, $K_i=9$, $K_d=0.1$, and the ANNC parameters were kept constant for these tests.

![Bar chart showing RMSE comparison between PID and PID+ANNC at different frequencies](image)

**Figure 2-21** RMSE of PID and PID+ANNC for sine wave tracking at different frequencies, (Taghizadeh 2010)

The key observations from PID+ANNC position control were:

- The plots of the ANNC weights illustrated rapid convergence to their equilibrium values.
- Adding more inputs to ANN did not necessarily improve the performance.
- Subtracting the ANNC signal from the overall control signal gave better performance than adding the signal.
- In step tracking, PID performance was better than PID+ANNC.
- In sinusoid tracking, PID+ANNC performance was better than PID.

2.7 Summary

The following can be presented as the main observations from the literature review:
Compensation for nonlinearities in pneumatic systems has been a popular area of research in pneumatic system control. Model based compensation strategies show relatively good results, although they still suffer from the requirement for model parameter identification.

Little research has been done investigating the use of NNs as direct controllers. Most of the applications of NNs are as indirect controllers or compensators.

Fuzzy logic control is another potential alternative as non-model based controller for nonlinearities of pneumatic systems.

PID controller with fuzzy gains, known as fuzzy PID, has shown noticeable improvement compared to PID with constant gains. However, there is no general rule for assigning the membership function to PID gains. Generating proper fuzzy rules is can be a time-consuming process.

Online adaptive tuning of the membership functions and fuzzy tables can reduce the energy and time required to design the fuzzy controller.

The most common performance measures for sinusoidal inputs are Root Mean Square Error (RMSE) and Average Error (AVGE).

The key observation from the literature review was that a ANN and FL appear to have the best potential to improve the performance of a pneumatic system used as either a controller or as a standalone compensator to a PID controller.
Chapter 3

Apparatus and Controller Implementation

This chapter provides background on the apparatus including sensor calibration. Details on the Adaptive Neural Network (ANN) and Fuzzy Adaptive PID controllers will also be given along with their implementation in Simulink®.

3.1 Apparatus Description

The original mechanical design of the gantry robot was completed by Raoufi (2003) in his Master’s thesis. Some changes were made to the apparatus by Abu Mallouh (2008) in his Doctoral thesis. In particular, the original manual flow control valves were replaced with Proportional Pressure Control (PPC) valves. Taghizadeh (2010) replaced the PPC valves with Proportional Flow Control (PFC) valves. This change was made on the basis of Abu Mallouh’s main recommendation, namely to replace the PPC valves with PFC valves in order to improve the response time of the control valves. The data acquisition system was also rewired to reduce the level of sensor signal noise seen in Abu Mallouh’s results.

Figure 3-1 illustrates the gantry configuration of the pneumatic system, with three DOF labeled as the x-axis, y-axis and z-axis. The y-axis consists of two rodless pneumatic cylinders. The x-axis (bore 32 mm, stroke 1 m) is a single rodless pneumatic cylinder driven by a PFC directional valve with 350 l/min nominal flow rate. The x-axis also acts as the gantry bridge for the two y-axis cylinders. The mechanical coupling of the two y-axis cylinders with the x-axis cylinder was sufficient to avoid the synchronization problems that sometimes arise with this configuration. The z-axis (bore 31 mm, stroke 124 mm) is a rodded pneumatic cylinder with integral linear guides, which is also driven by a PFC directional valve but with 100 l/min nominal flow rate. Note that the PFC valves for both axes are closed-center valves. Thus, cylinder air does not vent to atmosphere when the valve is in its centered position.

As illustrated in Figure 3-2, the end effector of the z-axis is a 2-axis force sensor that ends with a bearing for contact with a workpiece. The z-axis is in the same direction as the y-axis, but it is
labeled this way in order to highlight the availability of a third DOF. The components labeled in the figure are: A = $z$-axis cylinder, B = $x$-axis cylinder, C = payload, D = pressure sensor, and E = position and velocity sensor.

![Gantry robot with axes and valves labeled](image)

Figure 3-1 Gantry robot with axes and valves labeled, note $x$-axis valve symmetry

It is important to mention that in the beginning of this research, the configuration was not as shown Figure 3-1. When an attempt was made to repeat the Taghizadeh experiments, it was found that there was a blockage in the supply pressure airline. After correcting this problem and changing the airlines, it was decided to reconfigure the pneumatic system to improve symmetry and shorten the connecting airlines. In addition, lubrication of the cylinders was considered. However, proportional valves must be not lubricated because oil gums up the spools. Although the rodless cylinder should have been lubricated, it was not done in order to protect the $x$-axis valve.

The input voltage for the valves varies between 0 to 10 V and the valves are centered when the control signals $u_x = u_z = 5$ V. Thus, manual reset ($m_r$) has to be employed to offset the control signals. This bias voltage is $m_r = 5$ V for both valves.
In Figure 3-3, one is able to see the pneumatic circuit used for the *z-axis*. The pneumatic circuit for the *x-axis* is similar. The cylinder is controlled by a 5 port 3 way PFC valve. Position and velocity were measured directly with wire linked potentiometers and tachometers, respectively. Pressure transducers measure the differential air pressure directly across a cylinder. The pressure supply was normally set to 500 kPa (72.5 psi) as regulated by a manual pressure regulator. The Coulomb friction force for the *x-axis* and *z-axis* was determined to be 35 N and 30 N, respectively (Abu Mallouh, 2008). It is important to note the percentage of the friction force with regard to maximum available force, under nominal supply pressure. The available force, $fa$, in N can be calculated as follows:

$$f_a = \max(\Delta P) \times A_c$$

(3-1)

where $A_c$ is the cross sectional area of either of the cylinders and $\Delta P$ is the differential pressure (maximum value 500 kPa). Thus, $fa$ is calculated as approximately 128 N for both cylinders since they have similar bore diameters. This means, the Coulomb friction force is 27% and 23% of the maximum available force for the *x-axis* and the *z-axis*, respectively. In motion control, anything more than 10% is considered a high friction application (Abu Mallouh and Surgenor, 2007).
Figure 3-3 Pneumatic circuit used for $z$-axis cylinder control
3.1.1 Sensors and Calibration

Cable extension transducers measure the cylinder position ($x$) and velocity ($v$). The transducer is a combination of position and velocity transducer. A precision potentiometer provides position feedback while a DC tachometer provides a velocity signal that is proportional to the speed of the cylinder. Rodless and rodded cylinder were connected to position sensors with ranges of 1270 mm and 512 mm.

In addition, two identical differential pressure sensors (a and b) were mounted on both chambers of the cylinders: sensor “a” at the start of the cylinder, where $x=0$ mm ($z=0$ mm) and sensor “b” at the end where $x=1000$ mm ($z=124$ mm). Both have a differential pressure range of 689 kPa (100 psi). A signal conditioner was used to amplify the differential pressure sensor voltage output. Figure 3-4 shows a schematic of the rodless $x$-axis cylinder and the location of the two differential pressure sensors.

![Figure 3-4 Schematic of rodless $x$-axis cylinder](image)

In this thesis the focus is on rodless $x$-axis and rodded $z$-axis pneumatic cylinder position control. Thus, the $y$-axis of the gantry robot was not used. The corresponding calibration equations for the sensors are given in Table 3-1. It is worth mentioning that velocity and position sensors on both axes are identical but they resulted in different calibration equation.
Table 3-1 Summary of calibration equations for sensors (Taghizadeh 2010)

<table>
<thead>
<tr>
<th>Sensors Calibration</th>
<th>Ratio of the Actual Measurement vs. Sensor’s Voltage Reading ($v$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure sensor a</td>
<td>$\Delta P_1 = 64.79^*r - 2.36$</td>
</tr>
<tr>
<td>Pressure sensor b</td>
<td>$\Delta P_2 = 62.66^*r - 3.13$</td>
</tr>
<tr>
<td>x-axis position sensor</td>
<td>$x (mm) = 109.35^*r - 41.6$</td>
</tr>
<tr>
<td>z-axis position sensor</td>
<td>$z (mm) = 37.72^*r - 25.2$</td>
</tr>
<tr>
<td>x-axis velocity sensor</td>
<td>$v_x (mm/s) = 132.2^*r$</td>
</tr>
<tr>
<td>z-axis velocity sensor</td>
<td>$v_z (mm/s) = 323.8^*r$</td>
</tr>
</tbody>
</table>

**3.1.2 Data Acquisition**

Data acquisition and control was PC-based with dSPACE/DSP as the data acquisition hardware/software and MATLAB/SIMULINK as the control software. Sampling time was normally set to 1 msec. The DS1104 R&D controller board is a real-time hardware based on PowerPC technology. There are a total of 8 A/D channels (inputs): 4 are multiplexed with a resolution of 16 bits and conversion time of 2 $\mu$sec and the other 4 are parallel channels with 12 bits resolution and 800 nsec conversion times. There are 8 channels of D/A (outputs) with 16 bits of resolution and a settling time of 10 sec. Table 3-2 gives a summary of the main features of each of the major components used in the apparatus. Table 3-3 and 3-4 show the output and input signal and channel assignments, respectively. The technical information for all components can be found in the appendix section of Taghizadeh (2010).
### Table 3-2 Summary of main features of major components used in the apparatus

<table>
<thead>
<tr>
<th>No.</th>
<th>Component</th>
<th>Model</th>
<th>Axis</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cable extension for position</td>
<td>DV301</td>
<td>x-axis</td>
<td>± 0.1% of full stroke = ± 1.3 mm</td>
</tr>
<tr>
<td>2</td>
<td>Cable extension for position</td>
<td>DV301</td>
<td>z-axis</td>
<td>± 0.1% of full stroke = ± 0.13 mm</td>
</tr>
<tr>
<td>3</td>
<td>Cable extension for velocity</td>
<td>DV301</td>
<td>x-axis</td>
<td>± 3% at the rate 42 mm/sec = ± 1 mm/sec</td>
</tr>
<tr>
<td>4</td>
<td>Cable extension for velocity</td>
<td>DV301</td>
<td>z-axis</td>
<td>± 3% at the rate 42 mm/sec = ± 1 mm/sec</td>
</tr>
<tr>
<td>5</td>
<td>Proportional flow valve</td>
<td>FESTO</td>
<td>x &amp; z axes</td>
<td>Not Available</td>
</tr>
<tr>
<td>6</td>
<td>Differential pressure sensor</td>
<td>PX137</td>
<td>x &amp; z axes</td>
<td>± 0.1% of full scale = ± 0.7 kPa</td>
</tr>
<tr>
<td>7</td>
<td>Differential pressure sensor’s amplifier</td>
<td>DMD-465WB</td>
<td>x &amp; z axes</td>
<td>(Linearity) ± 0.05% of full scale = ± 0.005 V</td>
</tr>
<tr>
<td>8</td>
<td>Controller board &amp; data acquisition</td>
<td>DS1104R&amp;D</td>
<td>x &amp; z axes</td>
<td>Not Available</td>
</tr>
</tbody>
</table>

### Table 3-3 Channel assignments for output signals

<table>
<thead>
<tr>
<th>Output Channel</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch1 (D/A)</td>
<td>z-axis control output ($u_z$)</td>
</tr>
<tr>
<td>Ch2 (D/A)</td>
<td>x-axis control output ($u_x$)</td>
</tr>
</tbody>
</table>
Table 3-4 Channel assignments for input signals

<table>
<thead>
<tr>
<th>Input Channel</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch1 (A/D)</td>
<td>$x$-axis position ($x$)</td>
</tr>
<tr>
<td>Ch2 (A/D)</td>
<td>$z$-axis position ($z$)</td>
</tr>
<tr>
<td>Ch3 (A/D)</td>
<td>$z$-axis differential pressure ($\Delta P_z$)</td>
</tr>
<tr>
<td>Ch4 (A/D)</td>
<td>$x$-axis differential pressure ($\Delta P_x$)</td>
</tr>
<tr>
<td>Ch5 (A/D)</td>
<td>$x$-axis velocity ($v_x$)</td>
</tr>
<tr>
<td>Ch6 (A/D)</td>
<td>$z$-axis velocity ($v_z$)</td>
</tr>
</tbody>
</table>

3.2 Open loop Tests

In order to gain familiarity with the operation of the apparatus hardware, dSPACE and MATLAB software, an initial set of open loop experiments were conducted on the $x$-axis and $z$-axis under 500 and 350 kPa supply pressures. A first-order filter was used to decrease the level of noise in the position signal.

Figure 3-5 gives a sample open loop 0.75 Hz square wave test result for the $x$-axis with the initial step as positive. Note the positive drift in the cylinder pressures. Figure 3-6 plots the peak cylinder pressures, peak differential pressures, and peak velocities, as taken from Figure 3-5. Note that in Figure 3-6, the peak Pb pressure reaches a steady state of 80% of supply pressure ($400 \text{ kPa} = 0.8 \times 500 \text{ kPa}$). This confirms the observation of Pu and Weston (1990) who studied the steady state behavior of pneumatic servo drives driven by a single five-port servo valve. They demonstrated that cylinder pressure reaches 0.8 or 0.2 of supply pressure in the steady state, depending on the sign of the initial step in the control signal. In other words, the cylinder pressures would rise if the open loop test was started with a positive step in control signal, as shown in Figure 3-5.
Figure 3-7 gives a second square wave test result on the $x$-axis, but at 0.25 $Hz$ and with the initial step as negative. Here, one can see the reverse trend for the cylinder pressures which fall to 20% of the supply pressure. This again confirms the behavior noted by Pu and Weston (1990).

Figure 3-8 gives a third square wave test result at 1 $Hz$ for the $z$-axis. This result is taken after the cylinder pressures are allowed to reach steady state. Consequently, the pressures are seen to oscillate about an average value.

It is worthwhile to state that, since the $x$-axis cylinder is rodless, it has an equal bore area. Thus, if drift in position of the $x$-axis is observed, it can only be due to unequal friction given equal differential pressure applied to both sides. This can be witnessed in Figure 3-5 and 3-7 where there is drift of position in the negative direction since the friction is greater in the positive direction (for this particular cylinder). On the other hand, in Figure 3-8 the drift in position for the rodded $z$-axis cylinder is due to its unequal bore area and unequal applied force. It should be drifting in the direction of the rod which is indeed the case for Figure 3-8.
Figure 3-5 Open loop $x$-axis 0.75 Hz square wave test, start-up with initial step positive
Figure 3-6 Open loop x-axis start-up trend of cylinder peak pressure, differential pressure and velocity (as taken from Figure 3-5)
Figure 3-7 Open loop $x$-axis 0.25 Hz square wave test, start up with initial step negative
($amp=2\, V, \, Ps=500\, kPa$)
Figure 3-8 Open loop z-axis 1 Hz square wave test, steady state showing drift in position
(amp=1 V, Ps=350 kPa)
3.3 PID controller

According to Astrom and Hagglund (2001), more than 90% of controllers employ PID algorithms, especially in industrial applications. The simplicity of design and the achievable performance with linear systems are the two main reasons. In Figure 3-9, the PID controller and the pneumatic circuit for the z-axis cylinder are illustrated. Same design applies to the x-axis cylinder. The adopted control law with traditional PID gains was:

\[ u_z = K_p e_z + K_i \int e_z dt + K_d \frac{d}{dt} e_z + m_r \]  

(3-2)

As was explained before, \( m_r \) has been added to the control signal such that when \( e_z = 0 \text{ mm} \), then \( u_z = 5 \text{ V} \) because \( m_r = 5 \text{ V} \). Another experimental concern is integral windup. Integral windup refers to the situation in a PID controller, where the actuator reaches its physical limit and yet the control signal keeps on integrating and eventually saturates. This can result in undesirable control lags (Astrom and Hagglund, 2001). All the controllers used in this thesis, which use an integrator, are equipped with an anti-windup algorithm and will not get trapped in saturation mode. This has been accomplished by limiting the controller output and by using external reset feedback.

![Figure 3-9 Block diagram for PID controller and pneumatic system](image)
3.3.1 PID Tuning

There are a large number of methods for tuning a PID controller. There are theoretical methods but they typically require a dynamic model of the system. Nevertheless, it is essential to understand the effect of different PID gains before applying any of the available tuning methods. Table 3-5 provides heuristic rules of the effects of different gains on system response. By referring to the table, one can see which gain should be adjusted to change values of rise time, overshoot, settling time and steady state error.

Table 3-5 Heuristic rules of thumb for tuning PID controller (Chong et al, 2005)

<table>
<thead>
<tr>
<th>Increase in gain</th>
<th>Rise time</th>
<th>Overshoot</th>
<th>Settling time</th>
<th>Steady-state error</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>Decrease</td>
<td>Increase</td>
<td>Small change</td>
<td>Decrease</td>
<td>Degrade</td>
</tr>
<tr>
<td>$K_i$</td>
<td>Decrease</td>
<td>Increase</td>
<td>Increase</td>
<td>Decrease significantly</td>
<td>Degrade</td>
</tr>
<tr>
<td>$K_d$</td>
<td>Minor decrease</td>
<td>Minor decrease</td>
<td>Minor decrease</td>
<td>No effect in theory</td>
<td>Improve if $K_d$ small</td>
</tr>
</tbody>
</table>

Initially, the rules of Table 3-5 were used to tune the PID controller. However, there were occasions on which hand-tuning of the PID resulted in a better performance.

The underlying assumption of the tuning was that higher values of proportional gain resulted in better performance. Thus, the tuning process starts with increasing $K_p$ until the onset of system instability. Next, $K_d$ has to be increased until the percentage of overshoot (or undershoot) becomes acceptable. Neither $K_p$ nor $K_d$ can compensate for steady state error. Thus, at the third step, $K_i$ has to be increased to decrease the steady state error until it becomes sufficiently small. Figure 3-10 is the flowchart for this PID tuning procedure.
3.4 Adaptive Neural Network (ANN)

Background on NNs in general was given in Chapter 2. In this section, details on an Adaptive Neural Network (ANN) will be given including the method of implementation. For the purposes of this thesis, the formulation will be given assuming a three layered NN as illustrated in Figure 3-11. Some parts of this section are taken from the paper “A Novel Adaptive Neural Network Compensator as Applied to Position Control of a Pneumatic System” (Dehghan et al, 2011).
The algorithm for the Adaptive Neural Network Compensator (ANNC) is based upon the Modified Back Propagation Method (MBPM) originally proposed by Lewis et al (1996). The original MBPM was adapted to real time control applications by Campa et al (2002). They provided a Simulink® block in MATLAB® which models the MPBM of Lewis. Abu Mallouh (2008) in turn took the simulation model of Campa et al and adapted it for the gantry apparatus.

In practice, ANNC provides a feed-forward signal that linearizes the system to enable application of a linear controller to a nonlinear system. The key adaptive parameters are the weights. It is assumed that for every smooth function \( f(x) \), there exists a NN such that:

\[
    f(x) = W^T \sigma (V^T x) + \epsilon
\]

where \( W \) is the weight vector for the output layer, \( V \) is the weight vector for the hidden layer and \( \epsilon \) is the difference between \( f(x) \) and the NN. It is also noted that, in the presence of un-modeled disturbances, the tracking error does not vanish but is bounded. Furthermore, relatively small tracking errors can be achieved with relatively high NN gains. The only drawback is that in the training phase, slow learning rates can cause the NN to oscillate over the local minimum. The advantage of this structure is that the weights can be easily initialized and tuned online. No offline training is required. Lewis et al (1996) demonstrated the viability of the original technique. But they neither addressed key structural issues, such as the effect of the number of nodes, nor provided any experimental results.

### 3.4.1 ANN Algorithm

This section is taken from Taghizadeh (2010). It is repeated here in order to document the ANN algorithm used in this thesis. As illustrated in Figure 3-11, the basic structure for the ANN is that of a three layer neural network. The NN has to be optimized in terms of the number of nodes in both the input and hidden layers. A key design parameter is the nature of the activation functions for each node. For sigmoidal NN’s, the activation function \( \sigma_i^L \) for node \( i \) in layer \( L \) is commonly given as:

\[
    \sigma_i^L = \frac{1}{1 + e^{-\text{net}_i^L}}, \quad i = 1, 2, ..., B^L
\]
where $B_L$ is the number of nodes in layer $L$ and $L=1, 2$ and $3$. The function $net_i^L$ is the sum of the inputs to node $i$ in layer $L$ and is defined for the hidden layer ($L=2$) and output layer ($L=3$) as follows:

$$net_i^2 = \sum_{j=1}^{B^2} V_{i,j}p_j + b_i^2, \ i=1,2,...,B^2 \quad (3-5)$$

$$net_i^3 = \sum_{j=1}^{B^3} W_{i,j}a_j^2 + b_i^3 \quad (3-6)$$

where $V_{i,j}$ is the weight connecting node $i$ in the hidden layer ($L=2$) and input $P_j$ of the input layer, $P_j$ is $j^{th}$ input of the input layer, $w_{1,i}$ is the weight connecting the output node in the output layer and the output of node $j$ in the hidden layer ($a_{j,2}$), $B^2$ is number of nodes in the hidden layer, $b_i^2$ and $b_i^3$ are the bias of node $i$ in the hidden and output layer respectively.

The standard sigmoid activation function given as Equation 3-4 is used as the basis for the ANN. For back-propagation based NN’s, the updates to tuning weights are usually given as:
\[
\dot{W} = F\sigma (V^T P)e
\]  
(3-7)

\[
\dot{V} = GP (W^T \sigma (V^T P)e)^T
\]  
(3-8)

where \( W \) is the weight vector for the output layer, \( V \) is the weight vector for the hidden layer, \( F \) is the learning rate for \( W \) and \( G \) is the learning rate for \( V \), \( P \) is the input vector and \( e \) is the error between the output of the inner layer of the NN and the input vector. The training algorithm for the weights is given as:

\[
\dot{W} = F\sigma (V^T P)e - F\sigma (V^T P)V^T Pe - \lambda FW\|e\| 
\]  
(3-9)

\[
\dot{V} = GP (W^T \sigma (V^T P)e)^T - \lambda GV \|e\| 
\]  
(3-10)

where \( \lambda \) is a small positive tunable parameter whose function is to help deal with un-modeled dynamics (Lewis 1996). Examination of Equations 3-9 and 3-10 reveals that they consist of a standard back-propagation term (1\textsuperscript{st} term in equations), plus the error modification term taken from adaptive control (last term), plus a novel second-order forward-propagation term taken from the back-propagation network (2\textsuperscript{nd} term in Equation 3-9). In the ANNC the neural network output \( u_{NN} \) is given as:

\[
u_{NN} = \sigma_1^3 \left( \sum_{j=1}^{B^2} (W_{l,j} a_j^2) + b_1^3 + D \right)
\]  
(3-11)

where an additional parameter \( D \) has been added to the standard NN output to overcome higher order modeling errors. The parameter \( D \) is given by:

\[
D = -k_\epsilon (\|Z\| + \tilde{Z})e - k_\epsilon e
\]  
(3-12)

where \( \tilde{Z} \) is the maximum expected value of \( Z \), \( K_\zeta \) and \( K_V \) are gain terms and the matrix \( Z \) is given by:
\[ Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \]  \hspace{1cm} (3-13)

In Equation 3-12, \( \|Z\|_F \) denotes the Frobenius norm and for \( Z \) is defined as:

\[ \|Z\|_F = \sqrt{\text{trace}(Z^*Z)} \]  \hspace{1cm} (3-14)

where \( Z^* \) denotes the conjugate transpose of \( Z \). Trace is defined for a square matrix \( A \) as follows:

\[ \text{trace} (A) = \sum_{i=1}^{n} a_{ii} \]  \hspace{1cm} (3-15)

The advantage of this type of ANN is that it is designed specifically for online training. Thus, the weights can be easily initialized and trained online. No offline training is required.

3.4.2 ANN Implementation

This section presents the ANN Simulink® block used in this thesis as originally developed by Campa (2001). Figure 3-12 shows the Simulink® block diagram for the position controller used in Chapter 4 with the ANN input/output (I/O) block highlighted. In this example, the output of the ANN I/O block is added to the output of a PID controller and a \( P \) feedback block. For more details of the ANN input/output block and its Simulink implementation see Taghizadeh (2010).

Table 3-6 gives the ANN model block parameters. In addition to the ANN parameters, the initial weights \((V,W)\) and sampling time \((h)\) must be entered by the user. It was found that the nature and number of the inputs can affect the results. For example, it was found that the number of inputs could be reduced to 10 without losing performance (default value is 30). This improves the computational efficiency. Furthermore, it was found that if the numbers of inputs were greater than 15, then MATLAB® was unable to compile.
Figure 3-12 Simulink block diagram for ANN controller (Taghizadeh 2010)

Table 3-6 ANN model parameters

<table>
<thead>
<tr>
<th>No.</th>
<th>Parameter</th>
<th>Definition</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$n_i$</td>
<td>number of inputs</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>$n_h$</td>
<td>number of nodes in hidden layer</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>$n_o$</td>
<td>number of outputs</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>$G$</td>
<td>learning rate of $V$</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>$F$</td>
<td>learning rate of $W$</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>$\lambda$</td>
<td>adaptation parameter</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>$s$</td>
<td>slope of sigmoid activation function</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>bias</td>
<td>activation function bias</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>$V_{Lim}$</td>
<td>limit of $V$</td>
<td>1e37</td>
</tr>
<tr>
<td>10</td>
<td>$W_{Lim}$</td>
<td>limit of $W$</td>
<td>1e37</td>
</tr>
<tr>
<td>11</td>
<td>$K_z$</td>
<td>tunable parameter</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>$K_v$</td>
<td>tunable parameter</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>$\bar{Z}$</td>
<td>tunable parameter</td>
<td>0</td>
</tr>
</tbody>
</table>

Regarding the implementation of the ANN block, the block is compiled by MATLAB® to generate a C file useable by dSPACE (the controller board). It was found that MATLAB® 2007a was not compatible with dSPACE 6.0 and MATLAB® could not compile the Simulink® file
(result was a running error). This problem was corrected when MATLAB® was updated to 2009a and dSPACE was updated to version 6.3. After these changes, the Simulink® program could be compiled and the respective C file generated.

3.4.3 ANN Tuning Procedure

For the implementation of ANNC, the first question is the number of inputs. The selection of inputs is a key determinant of performance. There are 3 concerns which must be considered:

Inter-dependency of variables: Two or more interdependent variables may carry significant information that a subset would not. Thus, variables cannot be independently selected.

Curse of dimensionality: The addition of an input node to a network adds a dimension to the space and the number of weights increases exponentially. The performance of a network can be improved by eliminating unnecessary inputs. But equally so, performance depends upon having an adequate number of necessary inputs. Unfortunately, there are no rigorous methods of identifying which are “unnecessary” or which are “necessary”.

Redundancy of variables: Different inputs may carry the same information because they are correlated. A subset of uncorrelated inputs can have superior performance relative to a full set of correlated and uncorrelated inputs.

One approach is to use a combination of problem domain knowledge and standard statistical tests to select inputs. A second approach is to experimentally add and remove combinations of inputs, building a new network each time and testing the result. A third approach is to conduct a Sensitivity Analysis to rate the importance of variables with respect to a particular model. For this study, the second approach was used and will be explained more in the next chapter. For the most part, the tuning of a NN is accomplished in an ad hoc fashion. The procedure is illustrated in Figure 3-13 and can be broken into three steps. The variables were previously defined in Table 3-6. This procedure is slightly modified from that originally proposed by Taghizadeh (2010).
Step 1: Initialization

The first step involves initializing those parameters that relate to the basic design of any NN. Given that there is only one output, \( n_o \) is set to 1. The number of inputs (nodes in the input layer) is one of the key design variables for the structure of the NN. For this thesis, \( n_i \) is 12 for the compensator and 15 for the controller. In order to set the number of nodes in the hidden layer, one should increment \( n_h \) initially by 5. Once the change in performance is noticeable, \( n_h \) should be varied by \( \pm 2 \) until the response is satisfactory. Increasing the value of \( n_h \) beyond this point only serves to increase the processing time. In addition, MATLAB® is unable to compile the program if \( n_i \) and \( n_h \) are set too high. The upper limit for compiling was found to be difficult to predict. It was found that \( S \) and \( \text{bias} \) could be set to 1 without affecting the results. The weight limits \( W_{\text{Lim}} \) and \( V_{\text{Lim}} \) were set to 10. These limits can be viewed as analogous to saturation limits for a control signal. Thus, to confirm whether they are initialized correctly, one can monitor \( V \) and \( W \), as the ANN runs to ensure that the weights do not saturate unexpectedly.

Step 2: Coarse Tuning

In the second step, one does coarse tuning which involves \( Z, G, F, S, \text{bias} \). It was found that high \( G \) and \( F \) enables one to use high \( \lambda \). According to the tests, both \( G \) and \( F \) can be set to the same value. Thus, they can be treated as one parameter called \( GF \). It is recommended that changes in \( GF \) be no more than \( \pm 50\% \) from their default value of 1. However, if there is no noticeable change in the tracking, they can be increased up to a value of 3. Once tracking performance is considered reasonably acceptable in terms of the amount of steady-state error, degree of oscillation, and settling time, one can go to the next step.

Step 3: Fine Tuning

In the third and final step, one does fine tuning which involves \( K_v, K_z, \lambda \). It will be shown that on the \( z\)-axis these values become really important and extra attention for tuning is required. This step was called fine tuning because on some occasions slight change causes noticeable improvement in the tracking performance and, compared to coarse tuning; it requires more time to find the tuned values.
In Chapter 2, the benefits of non-model based control were discussed briefly. FL is one example of non-model based control.
FL controllers are applicable and helpful to systems that are:

- too complex to be analyzed by conventional quantitative techniques.
- are interpreted qualitatively, inexactly, or linguistically for their application.
- prone to fault.

A fuzzy logic system contains four basic elements. These elements are the fuzzifier, inference engine, rule base, and defuzzifier, as seen in Figure 3-14.

![Figure 3-14 Elements of a fuzzy logic system](image)

(1) The fuzzifier: transforms crisp input values into fuzzy values

Basically, the fuzzifier takes as input the real number $x$ and normalizes its value as the fuzzifier maps it onto the corresponding position in the defined membership function.

(2) The rule base: contains a knowledge of the application domain

The rule base then examines where the input falls in its membership function and evaluates the output based on a set of predefined rules. These if-then rules assume the form of following equation:

$$\textbf{IF } x \text{ is } F_I, \textbf{ THEN } y \text{ is } G_I$$

(3-16)

(3) The inference engine logic: performs decision-making for fuzzy control actions
The rules process the correct output value for the given input and use the inference engine to create the normalized fuzzy output. For this purpose, Matlab’s Fuzzy Inference System (FIS) will be used.

(4) The defuzzifier: converts the normalized fuzzy values to their real numbers

The defuzzifier then converts the normalized fuzzy output to its real number value and sends it to the controller or its next destination in the fuzzy control system. Most defuzzifiers, including those found in Matlab’s FIS, use a process called centroid defuzzification to formulate the output value. The following equation models this evaluation:

\[
y = \frac{\sum_{i=1}^{M} \tilde{y} \mu_B(\tilde{y})}{\sum_{i=1}^{M} \mu_B(\tilde{y})}
\]  

(3-17)

In Equation 3-17, the value of \( \tilde{y} \) is the center of the fuzzy set where the degree of membership \( \mu_B(y) \) has its maximum value. Defuzzification is based on max, weighted sum, probabilistic OR.

As mentioned, the fuzzy logic portion for the design originates in Matlab’s FIS editor. It can directly draw the membership functions for the multiple fuzzy inputs and outputs and also use a simple interface to create the fuzzy rule base. This simple tool editor allows for great ease and flexibility in testing and tuning the fuzzy logic evaluation in the intelligent adaptive control schemes (Jang 1995).

To illustrate the concept of fuzzy logic, Figure 3-15 is presented. This example comprises two inputs and three set of rules resulting in one output. Figure 3-16 is the same example as in Figure 3-15, but gives more detail on the membership functions, which are triangular. In this example, the arbitrarily chosen input values are activating the rules in the manner shown. Thereafter the implication method finds the consequent output MF based on the antecedents (input MFs). The aggregation method converts these output MFs into a single MF. At the end, it is the defuzzifier’s responsibility to transform this single MF back to a real number.
Figure 3-15 Fuzzy logic example with 2 inputs, 3 rules, 1 output (adapted from Mathworks 2011)

Figure 3-16 Fuzzy logic example of Figure 3-15 illustrating role of membership functions
Designing a FL controller can be sometimes frustrating since it is based on trial and error. There is no offline optimal way to determine the state variable, MFs, or fuzzy rules. Figure 3-17 illustrates one approach to the design of a FL controller.

Figure 3-17 Flowchart for fuzzy logic controller design
3.5.1 Fuzzy Adaptive PID (FAPID)

Conventional PID fuzzy controllers may lead to undesirable performance due to a time varying nonlinear dynamics in a plant. Also, the gains obtained may not be realistic. Based on Astrom and Whittenmark (2001) an adaptive controller has been defined as a controller with adjustable parameters and a mechanism for adjusting the parameters. Thus, it may be necessary to add an adaptive feature to PID to overcome these problems. Regarding the development of Fuzzy Adaptive PID (FAPID) there are a number of different implementations in the literature. These structures are dependent on their applications and the system’s performance criteria. Four different structures are shown in Figure 3-18.

(a) FP+FI+FD – all the control parts are fuzzy; however, design parameters are calculated in an independent way in order to reduce the rule base and the computational effort.

(b) FPD+I – a fuzzy PD controller is added to a conventional I only controller

(c) FI+FPD Incremental – a fuzzy I controller is added to an incremental fuzzy PD controller.

(d) FP+I+D – the P part is calculated in a fuzzy way and adds to D and I parts.

![Figure 3-18 Alternate arrangements for FAPID controllers (Callai 2005)]
It can be inferred that, depending on the application, one can fuzzify the proper PID gains. The more important point is to find a structure that will satisfy the requirements with the fewest rules possible. An increase in the number of rules (or MFs) leads to a higher need for hardware infrastructure and also increases the processing time. One way to lower the complexity of the controller is to make the assignment of the FAPID gains independent of each other. Although it is not appropriate for all applications, this approach appeared to work for the position controller of the pneumatic cylinder. Figure 3-19 illustrates the scheme adopted for this thesis, in the context of the z-axis. The same scheme was applied to the x-axis.

Figure 3-19 Block diagram for implemented FAPID controller

The proposed FAPID method compensates for nonlinearities and uncertainties without requiring a system dynamic model. Adding fuzzy action to the PID gains ensures the adaptation of the controller to unpredicted behaviors and disturbances.

One approach to fuzzy PID controller design is as follows:

- Obtain an estimate for the conventional PID gains;
- Design the Fuzzy Membership functions (MFs) and Rules;
- Substitute PID gains with Fuzzy Adaptive PID Blocks;
- Fine-tune the rules and MFs.
3.6 Summary

This chapter provided the background on the apparatus, including sensor calibration. Details on the ANN and FAPID controller were also provided, including their implementations in Simulink® and tuning procedures as illustrated by flowcharts.

In the next two chapters, the apparatus will be used to obtain results for the case of position control by different adaptive PID algorithm as applied to the two axes of the gantry robot. Chapter 4 will deal with just the $x$-axis. Chapter 5 will deal with just the $z$-axis. Chapter 6 will deal with both.
Chapter 4

Position Control of a Rodless Pneumatic Cylinder (x-axis)

The contents of this chapter are an expansion of the paper “Tuning of an Adaptive Neural Network Compensator for Position Control of a Pneumatic System” (Dehghan, Taghizadeh, and Surgenor, 2011). A set of experiments was conducted on the rodless x-axis cylinder. The following control methods were tested.

1. PID control
2. Fuzzy Logic control
3. PID+ANNC
4. ANNonly control
5. FAPID control

Before presenting the test results a review will be given of the performance measures.

4.1 Performance Measures

As mentioned Chapter 2, the most common performance measures for sinusoidal tracking are Average Error (AVGE) and Root Mean Square Error (RMSE). They are calculated as:

\[
AVGE = \frac{\sum_{i=1}^{N} |error_i|}{N} \quad (4-1)
\]

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{N} (error_i)^2}{N}} \quad (4-2)
\]

Also, the percentage improvement in tracking performance is calculated as:

\[
\Delta AVGE_i = \frac{AVGE_i - AVGE_{ref}}{AVGE_{ref}} \times 100 \quad (4-3)
\]
\[ \Delta \text{RMSE}_i = \frac{\text{RMSE}_i - \text{RMSE}_{\text{ref}}}{\text{RMSE}_{\text{ref}}} \times 100 \]  

(4-4)

Where \( N \) is the number of samples and \( \text{RMSE}_{\text{ref}} \) denotes the RMSE of the reference signal. A negative \( \Delta \text{RMSE} \) means an improvement in performance (i.e. tracking error reduced). It is noted that these performance measures only reflect the output performance. The lack of control signal measure of performance may be misleading in a practical sense. In Figure 4-1, this point is illustrated. In this example, the objective is to track a sine wave with a frequency of approximately 0.16 Hz. Based on the AVGE and RMSE of this example, the high frequency response performed better than the low frequency one, as reported in Table 4-1. But this high frequency oscillatory response is the result of a high frequency control signal. A high frequency control signal, also known as signal with high chatter, results in high wear on a pneumatic control valve.

![Figure 4-1 Low and high frequency tracking to illustrate effect on performance measures](image)

Figure 4-1 Low and high frequency tracking to illustrate effect on performance measures
Table 4-1 Performance measures for tracking example given in Figure 4-1

<table>
<thead>
<tr>
<th>Performance Measures</th>
<th>AVGE (mm)</th>
<th>RMSE (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Frequency Response Fig</td>
<td>128.5</td>
<td>142.3</td>
</tr>
<tr>
<td>High Frequency response Fig</td>
<td>127.7</td>
<td>141.6</td>
</tr>
</tbody>
</table>

In general, the judgment about the quality of performance is application-dependent. For a pneumatic cylinder, highly oscillatory behavior for the control signal may result in lower values of AVGE and RMSE, but it may result in high valve wear. On the other hand, high frequency valve chatter acts like a dither signal, which reduces the effect of Coulomb friction. For example, Ning and Bone (2005) found a certain level of chatter in the control signal acted as a dither signal. This reduced the problem of stick-slip friction, and consequently improved the tracking performance. In other words, having a dither signal may improve the results at the cost of reducing the durability of the valves. To sum up the abovementioned points, these output performance measures provide no information on the nature of control signal. However, for conformity with the literature and to enable one to compare the results with other researchers, AVGE and RMSE were adopted again as the only performance measures.

Also, AVGE or RMSE can be misleading as performance measures as their values are changing during a cycle; i.e. RMSE or AVGE for just a portion of the cycle is not an acceptable performance measure. So it is important to capture the data of one or a complete number of cycles to calculate AVGE or RMSE. However, for conformity with Taghizadeh (2010) and to enable direct results comparison, AVGE and RMSE of 15 s (7.5 cycles of 0.5 Hz, 3 cycles of 0.2 Hz, and 1.5 cycles of 0.1 Hz sine wave tracking) were calculated. Note that this 15 s doesn't negate the validity of the comparative performance measures, but it is recommended that the future work uses a 10 s window in order to provide a better absolute measure of precision.

Finally it should be pointed out that the amplitudes chosen for the sine wave and step tracking experiments were dictated by the need to maximize the movement with respect to cylinder stroke, without hitting the endstops.
4.2 PID Results

An initial set of P-only controller experiments were conducted on the x-axis in order to gain familiarity with the effects of proportional gain. Next, integral and derivative components were added to the controller to gain knowledge about the values of $Ki$ and $Kd$, specifically to observe how they affect the steady state performance and the reduction in overshoot, respectively. As was explained in the previous chapter, the tuning of the PID controller was done manually. Prior to the systematic hand-tuning of the PID gains, the Ziegler Nichols (ZN) approach was tested.

The ZN tuning method is a heuristic method of tuning a PID controller. During ZN tuning, the integral and derivative actions of PID should be initially kept to zero. The proportional gain is then increased until the output oscillates with constant amplitude. The corresponding proportional value is called the ultimate gain. It is then used along with the oscillation period, and ultimate period to calculate the $Kp$, $Ki$, and $Kd$ gains. Table 4-2 gives the formulae for calculating P, PI, PID gains. The required way of conducting this tuning procedure is to test the system in the closed loop with a step input.

For the x-axis the ultimate gain ($K_u$) gain and oscillation period ($T_u$) were found to be 6 and 0.45 s, respectively. In Figure 4-2 one can see the oscillatory response. One sees that the oscillation is not completely sustained over the full cycle because of uneven friction along the axis. The important thing is to not let the system go unstable. In Figure 4-3 one can see the responses for PID with ZN gains conducted with 500 kPa supply pressure.

<table>
<thead>
<tr>
<th>Type of Controller</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>$0.5*K_u$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PI</td>
<td>$0.45*K_u$</td>
<td>$5/6*T_u$</td>
<td>0</td>
</tr>
<tr>
<td>PID</td>
<td>$0.6*K_u$</td>
<td>$0.5*T_u$</td>
<td>$0.125*T_u$</td>
</tr>
</tbody>
</table>

In Table 4-3, numerical performance results for all three ZN tuned controllers are provided. One sees that the responses are comparable with P-only and PID having almost the same AVGEs.
Table 4-3 Summary of performance results for PID tuned with ZN rules for step tracking

<table>
<thead>
<tr>
<th>Controller</th>
<th>RMSE ($mm$)</th>
<th>AVGE ($mm$)</th>
<th>$\Delta$RMSE (%)</th>
<th>$\Delta$AVGE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P only ($K_p=6$)</td>
<td>66.3</td>
<td>22.8</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td>PI ($K_p=2.7$, $K_i=0.4$)</td>
<td>67.3</td>
<td>29.8</td>
<td>1.5</td>
<td>3</td>
</tr>
<tr>
<td>PID ($K_p=3.6$, $K_i=0.2$ $K_d=0.1$)</td>
<td>66.5</td>
<td>22.6</td>
<td>0.3</td>
<td>-0.8</td>
</tr>
</tbody>
</table>

It is interesting to note that when tuned by ZN, all three controllers gave effectively the same performance measures for a step test. However, when these values were tested for sine wave tracking the performance was disappointing. This is understandable, since ZN is intended for step tracking (Microstar Laboratories, 2012). This also raises an interesting view of why fuzzy and NN should be used for control. It is because for a sinusoidal test, it is not clear what the tuning rules are for PID. However, fuzzy and NN algorithms don't require knowledge of the type of input.

Thus, for sine wave tracking, the hand tuning approach presented in Chapter 3 was adopted. The results are summarized in Table 4-4. The tuned gains were $K_p=2.25$, $K_i=9$, $K_d=0.6$ and the resultant response is provided in Figure 4-4. These gains resulted in an acceptable AVGE and RMSE as reported in the table. These are the same PID gains used by Taghizadeh (2010), except $K_d=0.1$. 
Figure 4-2 Ponly oscillatory response for step tracking ($K_p=6$)

Figure 4-3 PID response for ZN tuning for step tracking ($K_p=3.6$, $K_i=0.2$, $K_d=0.1$)
Table 4-4 Summary of performance results for PID tuned by hand for sine wave tracking, \( amp=300 \text{ mm} \)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>180.5</td>
<td>171.2</td>
</tr>
<tr>
<td>0.2</td>
<td>19.9</td>
<td>18.1</td>
</tr>
<tr>
<td>0.1</td>
<td>13.6</td>
<td>11.3</td>
</tr>
</tbody>
</table>

Figure 4-4 PID response for 0.1 Hz sine wave tracking \( amp=300 \text{ mm} \)
4.3 Fuzzy Results

In this section, the performance of the fuzzy controller is examined first in simulation and then in experiment. Simulation objective was to test out the fuzzy controller before application to the apparatus. The simulation process model was taken from Abu-Mallouh’s thesis (2008).

4.3.1 Simulation

The fuzzy controller was implemented in Simulink with application of MATLAB’s fuzzy logic toolbox. The simulation model was based on a force balance of the cylinder, where the applied force \( f_{ax} \) is resisted by the contact force \( f_c \), viscous friction force \( v_x C_{fx} \), the Coulomb friction force \( F_c \), and the wire force \( F_{wx} \). The governing equation for the system can be given as:

\[
f_{ax} - f_x - v_x C_{fx} - F_c - F_{wx} = m_x a_x
\]

(4-5)

where \( m_x \) is the effective mass of the cylinder assembly and \( a_x \) is the end effector acceleration along the \( x\)-axis. Figure 4-5 provides more detail of the Simulink implementation. One aim of this simulation was to make sure that this controller would not result in instability. In addition, it helped to gather useful information about the fuzzy rules and fuzzy MFs. In Figure 4-6 one can see the fuzzy rule surface of the control signal. The fuzzy controller was then evaluated for step and sine wave trajectory following, as can be seen in Figures 4-7 and 4-8, respectively.
Figure 4-5 Simulink model of the $x$-axis pneumatic cylinder with fuzzy controller (adapted from Abu-Mallouh 2008)

Figure 4-6 Fuzzy rule surface for simulation test
Figure 4-7 Fuzzy simulation response for step tracking

Figure 4-8 Fuzzy simulation response for 0.16 Hz sine wave tracking
The main objective of the fuzzy simulation was to get better understanding of the system behavior. It also made the development and tuning of the fuzzy rule table easier. However, simulations are not entirely reliable without exhaustive testing because not all real-life effects could be taken into account. Three major drawbacks of the simulation model were:

- The values for signal noise and disturbances were assumed negligible.
- There were some simplifications in the modeling of the valve and friction.
- Only a small set of operating conditions were tested. For some operating conditions, the fuzzy controller may not perform as well.

The tuning process in simulation examined the variations in MFs, the values of fuzzy outputs and variations in fuzzification or defuzzification techniques. For the coarse tuning only the AVGE and RMSE of one complete period was considered. Although performance was greatly enhanced after this coarse tuning, it was time-consuming. Fine tuning was postponed to the experimental phase in the next section.

4.3.2 Experiment

This section provides the fuzzy controller results as applied to the $x$-axis cylinder. After the simulation phase, the same controller was implemented and built on a dSPACE module. When programming the fuzzy logic block, using the fuzzy logic toolbox (FIS editor), some problems and bugs were encountered. It was found that the toolbox is very basic in terms of naming and connecting the components. In Figure 4-9, the block diagram of the fuzzy control is given. The addition of a Zero-Order Holder (ZOH) block is to remind one that A/D and D/A conversion is taking place.
As was explained, fuzzy control is useful when there is no knowledge about the dynamic model of the system. However, the fine tuning of the fuzzy parameters is time-consuming. Table 4-5 summarizes the fuzzy rules. The table is showing the fuzzy rules based on $v_x$ versus $-e_x$. Table 4-6 provides a summary of fuzzy and PID+ fuzzy results for different sine wave frequencies. For PID+Fuzzy the control signals of PID and fuzzy are combined and summed with equal weights. The integrator component of the PID controller was equipped with anti-windup.

In Figure 4-10 the fuzzy rule surface for the control signal is depicted. Membership function of inputs and output are given in Figure 4-11. Specifically, Figure 4-11 shows the fuzzy inputs $(v_x, -e_x)$ each with 7 MFs which result in 49 rules was given in Table 4-5. Fuzzy output is also comprised of 7 MFs ranging from -5 \( v \) to 5 \( v \). The summed fuzzy control signal is added to $m_r$ and this becomes the output of the fuzzy controller.
Figure 4-10 Fuzzy rule surface for experimental test

Table 4-5 Fuzzy rules governing $u_x$ based on $v_x$ and $-e_x$ fuzzy inputs

<table>
<thead>
<tr>
<th>$v_x$</th>
<th>$-e_x$</th>
<th>V^1 N^2 B^3</th>
<th>NB</th>
<th>NS^4</th>
<th>ZO^5</th>
<th>P^6 L</th>
<th>PB</th>
<th>VPB</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNB</td>
<td>VPB</td>
<td>VNB</td>
<td>PB</td>
<td>PS</td>
<td>PS</td>
<td>ZO</td>
<td>ZO</td>
<td>ZO</td>
</tr>
<tr>
<td>NB</td>
<td>VPB</td>
<td>PB</td>
<td>PB</td>
<td>PS</td>
<td>ZO</td>
<td>ZO</td>
<td>ZO</td>
<td>ZE</td>
</tr>
<tr>
<td>NS</td>
<td>PB</td>
<td>PB</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>ZO</td>
<td>PB</td>
<td>PB</td>
<td>PS</td>
<td>ZO</td>
<td>NS</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
</tr>
<tr>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>ZO</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NB</td>
<td>NB</td>
</tr>
<tr>
<td>PB</td>
<td>PS</td>
<td>ZO</td>
<td>ZO</td>
<td>NS</td>
<td>NS</td>
<td>NB</td>
<td>VNB</td>
<td>VNB</td>
</tr>
<tr>
<td>VPB</td>
<td>ZO</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NB</td>
<td>VNB</td>
<td>VNB</td>
<td>VNB</td>
</tr>
</tbody>
</table>

1. V = Very
2. N = Negative
3. B = Big
4. S = Small
5. ZO = Zero
6. P = Positive
Figure 4-11 Fuzzy membership functions for $-e_x$, $v_x$ and $u_x$. 
Table 4-6 Summary of fuzzy and PID+fuzzy results for sine wave and step tracking

| Controller                                      | Sine wave  
|                                               | amp=250 mm  
|       | 0.5 Hz  | Step tracking  
|       |         | amp=200 mm  |
| PID controller (Kp=2, Ki=3, Kd=0)             | 131.5  | 59  |
| Fuzzy controller                               | 164    | 54  |
| PID+ Fuzzy                                     | 116    | 53  |

The main observation about Table 4-6 is that, there is no general trend which leads to a conclusion about the performance of fuzzy versus PID control. Figure 4-12 shows a fuzzy controller response to a 0.06 Hz step tracking with 200 mm of amplitude.

Figure 4-12 Fuzzy controller response for step tracking
An important observation from these experiments was that the performance improvement for the fuzzy controller was not noticeable relative to untuned PID. Furthermore, when the PID is tuned results are better than with the fuzzy controller. Thus, there is a trade-off between accuracy and tuning time. One can either apply a tuned PID controller after spending time to find tuned PID gains or use fuzzy control with fixed rules and MFs but resulting in lower performance. This trade-off can be resolved if one designs an adaptive PID controller, which will be presented in Chapter 6.

4.4 PID+ANNC

Background on NNs in general was given in Chapter 2. Also in Chapter 3, details of an ANN were covered, including implementation. In this section, ANN will be used as a compensator operating in parallel with a PID controller. The role of the ANN compensator (ANNC) is to help linearize the system in order to improve the performance of the PID controller. Specifically, this online adaptive compensator seeks to negate friction and other nonlinear effects inherent in the pneumatic system (Choi et al, 1998). At first, the NN model and the way it was configured and implemented in MATLAB was studied.

Figure 4-13 illustrates the PID+ANNC controller block diagram. The ANNC block is drawn parallel to the PID block. Equation 4-6 shows that NN signal is being subtracted from the PID signal.

\[ u_x = K_p e_x + K_i \int e_x dt + K_d \frac{de}{dt} - u_{NN} + m_r \]  

\[ (4-6) \]

It was found that subtracting the NN signal gave better performance rather than adding it to the PID signal (Taghizadeh 2010). It can be inferred that by subtracting the output of the NN compensator, the nonlinear terms are negated. The presence of \( m_r \) is to offset the controller signal, which was explained in Chapter 3.
In the next subsection, trial and error experiments are conducted to find tuned values of the ANNC parameters and select the inputs.

4.4.1 ANNC Inputs

As was explained in Chapter 3, there is no general rule for determining the best set of inputs for ANNC. The ANN design reported in Taghizadeh (2010) was considered optimum for the application at hand. Experiments conducted as part of this chapter verify Taghizadeh’s design. Experiments were done at 0.1 Hz, 0.2 Hz, and 0.5 Hz and it was confirmed that these inputs perform better than alternatives, to some extent. However, Taghizadeh did not consider \( u_{NN}, u_x, \) and \( u_{PID} \) in any of his tests. In the next subsection, their influence on performance will be evaluated. It should be noted that the shapes of the actuation functions were not changed during any of tests. Figure 4-14 shows the input vector that was considered to be optimal by Taghizadeh and was verified in this study.
With regard to the inputs shown in Figure 4-14, it is acknowledged that $e_x$ could be viewed as not a unique input because it is calculated from $x$ and $x_s$, which are also inputs. However, as the ANN algorithm is dynamic, $e_x$ is not considered redundant. But this observation does highlight an underlying design issues for NN compensators, namely the challenge in selecting the appropriate inputs.

### 4.4.2 ANNC Tuning

Taghizadeh (2010) reported the values in Table 4-7 as the tuned values of the ANN parameters. His set of parameters was obtained by trial and error, observing system performance when tracking a 0.33 Hz sinusoidal reference signal.
Table 4-7 Tuned values of ANNC parameters reported by Taghizadeh (2010)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_i$</td>
<td>number of inputs</td>
<td>12</td>
</tr>
<tr>
<td>$n_h$</td>
<td>number of nodes in hidden layer</td>
<td>10</td>
</tr>
<tr>
<td>$n_o$</td>
<td>number of outputs</td>
<td>1</td>
</tr>
<tr>
<td>$G$</td>
<td>learning rate of $V$</td>
<td>1</td>
</tr>
<tr>
<td>$F$</td>
<td>learning rate of $W$</td>
<td>1</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>adaption parameter</td>
<td>1.5</td>
</tr>
<tr>
<td>$s$</td>
<td>slope of sigmoid</td>
<td>1</td>
</tr>
<tr>
<td>bias</td>
<td>activation function bias</td>
<td>1</td>
</tr>
<tr>
<td>Lim$V$</td>
<td>limit of $V$</td>
<td>10</td>
</tr>
<tr>
<td>Lim$W$</td>
<td>limit of $W$</td>
<td>10</td>
</tr>
<tr>
<td>$K_z$</td>
<td>tuning parameter</td>
<td>0.5</td>
</tr>
<tr>
<td>$K_v$</td>
<td>tuning parameter</td>
<td>0.5</td>
</tr>
<tr>
<td>$Z$</td>
<td>tuning parameter</td>
<td>0.5</td>
</tr>
</tbody>
</table>

It was intended to redo the tuning experiments and become certain that the tuned parameters had not changed over time. The same coarse tuning procedure was followed and it turned out that Taghizadeh ANN values are still acceptable. Table 4-8 provides a sensitivity measure and shows what happens if one deviates from the tuned values of Table 4-7. The first line in Table 4-8 is the reference case which shows AVGE= 31.0 mm and RMSE= 34.8 mm. Subsequent lines illustrate the change in AVGE and RMSE relative to the reference line for a given parameter change.
Table 4-8 Effect of changing ANNC parameters from tuned values in Table 4-7

<table>
<thead>
<tr>
<th>Changed ANN Parameter</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
<th>RMSE%</th>
<th>AVGE%</th>
</tr>
</thead>
<tbody>
<tr>
<td>with ANNC</td>
<td>34.8</td>
<td>31.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>without ANNC</td>
<td>51.7</td>
<td>45.9</td>
<td>49</td>
<td>44</td>
</tr>
<tr>
<td>$F = 0.05$</td>
<td>37.83</td>
<td>33.0</td>
<td>11</td>
<td>3.9</td>
</tr>
<tr>
<td>$G = 0.05$</td>
<td>35.2</td>
<td>30.8</td>
<td>3.6</td>
<td>-3.0</td>
</tr>
<tr>
<td>$S = 5$</td>
<td>69.7</td>
<td>61.7</td>
<td>105</td>
<td>94</td>
</tr>
<tr>
<td>$\lambda = 5$</td>
<td>49.5</td>
<td>50.5</td>
<td>45.5</td>
<td>59</td>
</tr>
<tr>
<td>$Kv = 0$</td>
<td>54.2</td>
<td>49.3</td>
<td>55</td>
<td>59</td>
</tr>
<tr>
<td>$Kz = 0$</td>
<td>61.6</td>
<td>53.5</td>
<td>77</td>
<td>72</td>
</tr>
</tbody>
</table>

The following observations can be made about Table 4-8:

- Changing learning rates $F, G$ does **not** significantly affect the result (but does affect the time required for the system to get to its steady state behavior)
- Increasing slope significantly degrades performance
- Changing the parameters $Kz, Kv, \lambda$ does significantly affects performance

Note that in the last three rows the parameters of the fine tuning step have been changed. Since the performance shows high sensitivity to these parameters, more rigorous tuning (i.e. fine tuning) is required to find the optimum values of these parameters. The key takeaway from Table 4-8 is that there was no instance of performance significantly improving for the coarse tuning parameters. However, for the fine tuning parameters, the values of $\lambda, Kv, Kz$ did have a significant and were tuned in separate set of experiments.

Concerning the results when using NN compensation, one observes that performance is less than optimum if the ANN is trained and tuned at a high frequency and tested at a low frequency. This is due to the fact that the system is nonlinear across different operating conditions. According to Lewis et al (1999) if the NN parameters are set at one operating condition and tested on another, the NN may not be able to converge when training. Thus, it was important to capture the performance measures corresponding to different parameter values in the main operating
conditions (0.1, 0.2, 0.5 Hz, P=500 kPa, and amp=250 mm) and pick the best values among all the frequencies.

A good rule of thumb, during the manipulation of the ANNC parameters, is that $\lambda$, $K_v$, and $K_z$ are designed to cope with higher-order model errors. These errors seem to be more substantial in 0.5 Hz frequency tests, since higher values of the fine-tuned parameters resulted in better performance at higher test frequencies.

At first, the use of $u_{NN}$, $u_x$, $u_{PID}$ as part of the input vector was thought to have a positive effect. In Table 4-9 the effect of their usage as part of the ANNC input can be witnessed and it can be observed that they did not improve the results in all cases. On other hand, $\Delta P$ was placed in the input vector and as was reported by Taghizadeh, the responses became smoother, but didn’t cause any decrease in AVGE and RMSE. During the tests, $K_v=K_z=2$ was the maximum possible value for these parameters. Increasing them led to high chatter in the control signal and oscillation in the position signal. In order to handle this effect, $P$ was used as an input to the ANNC to decrease the chatter due to high $K_v$ and $K_z$.

Table 4-9 PID+ANNC tuning results for different frequencies ($amp=250$ mm), best results in Bold

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Different values of the ANNC parameters</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>PID</td>
<td>131.5</td>
<td>116.1</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC ($K_v=K_z=0.5$, $\lambda=1.5$)</td>
<td>57.1</td>
<td>51.1</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC ($K_v=K_z=1$, $\lambda=1.5$)</td>
<td><strong>51</strong></td>
<td><strong>45</strong></td>
</tr>
<tr>
<td></td>
<td>PID+ANNC, $u_{NN}, u_x$ added to the input set ($K_v=K_z=1, \lambda=4.5$)</td>
<td>67.8</td>
<td>59.8</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC, $P$ added to the input set ($K_v=K_z=2, \lambda=4.5$)</td>
<td>49.8</td>
<td>43.2</td>
</tr>
<tr>
<td>0.2</td>
<td>PID</td>
<td>16.8</td>
<td>14.7</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC ($K_v=K_z=0.5$, $\lambda=1.5$)</td>
<td>15.6</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC ($K_v=K_z=1$, $\lambda=1.5$)</td>
<td><strong>13.7</strong></td>
<td><strong>12.1</strong></td>
</tr>
<tr>
<td></td>
<td>PID+ANNC, $u_{NN}, u_x$ added to the input set ($K_v=K_z=1, \lambda=4.5$)</td>
<td>14.2</td>
<td>12.6</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC, $P$ added to the input set ($K_v=K_z=2, \lambda=4.5$)</td>
<td>25.8</td>
<td>21.7</td>
</tr>
<tr>
<td>0.1</td>
<td>PID</td>
<td>9.3</td>
<td>7.3</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC ($K_v=K_z=0.5$, $\lambda=1.5$)</td>
<td>7.75</td>
<td>6.4</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC ($K_v=K_z=1$, $\lambda=1.5$)</td>
<td><strong>6.5</strong></td>
<td><strong>5.3</strong></td>
</tr>
<tr>
<td></td>
<td>PID+ANNC, $u_{NN}, u_x$ added to the input set ($K_v=K_z=1, \lambda=4.5$)</td>
<td>6.7</td>
<td>4.9</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC, $P$ added to the input set ($K_v=K_z=2, \lambda=4.5$)</td>
<td>6.3</td>
<td>5.1</td>
</tr>
</tbody>
</table>
According to Table 4-9 with best results for PID+ANNC, $K_v=K_z=1$, $\lambda=1.5$ parameter values were chosen as the best set and were used in the rest of this chapter.

It should be mentioned that Table 4-4 does not convey the whole set of experiments conducted and the list of experiments with different ANN values is much longer. Thus, this extensive tuning procedure emphasizes the necessity of adaptive tuning (or some form of self-tuning) procedure. This self-tuning feature should have a suitable learning routine embedded in it and be adaptive to tune the ANN optimally for any operating conditions. Discussion on adaptive tuning methods will be presented in Chapter 6. In Figures 4-15, 4-16, and 4-17 the best sine wave tracking response of PID+ANNC, from the ones given in Table 4-9, are depicted for 0.5, 0.2, 0.1 Hz, respectively. Note that in the figures, the PID gains were $K_p=2.25$, $K_i=9$, $K_d=0.6$, and ANNC parameters were $K_v=K_z=1$, $\lambda=1.5$. As indicated by the numeric results of Table 4-9, these figures illustrate that performance improves as the tracking frequency is reduced from 0.5 Hz to 0.1 Hz.

![Figure 4-15 PID+ANNC response for 0.5 Hz sine wave tracking with tuned ANNC](image-url)
Figure 4-16 PID+ANNC response for 0.2 Hz sine wave tracking with tuned ANNC

Figure 4-17 PID+ANNC response for 0.1 Hz sine wave tracking with tuned ANNC
4.4.3 Tracking Results

The previous section gave sine wave tracking results as part of the process to tune PID+ANNC. The number of tracking results will be expanded in this section to include two amplitudes, with a direct comparison to the results of Taghizadeh et al (2010). Note that the 15 s time frame is the window over which RMSE and AVGE were calculated for all tests. Experiments were conducted with a supply pressure of 500 kPa. In all cases the PID gains were $K_p = 2.25$, $Ki = 9$ and $Kd = 0.6$.

Table 4-10 gives the performance for different tracking frequencies with PID+ANNC and PID controllers. The relative improvement is shown by RMSE and AVGE and the change is benchmarked against PID without ANNC. The improvement is significant, between 45% and 66%, and is much better than the 13% to 26% reported in Taghizadeh et al (2010).

<table>
<thead>
<tr>
<th>Frequency in Hz</th>
<th>amp in mm</th>
<th>PID Only</th>
<th>PID + ANNC</th>
<th>Percent change for this thesis</th>
<th>Percent change in Taghizadeh et al</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RMSE (mm)</td>
<td>AVGE (mm)</td>
<td>RMSE (mm)</td>
<td>AVGE (mm)</td>
</tr>
<tr>
<td>0.5</td>
<td>300</td>
<td>180.5</td>
<td>171.2</td>
<td>67.5</td>
<td>58.5</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>131.5</td>
<td>116.1</td>
<td>51</td>
<td>45</td>
</tr>
<tr>
<td>0.2</td>
<td>300</td>
<td>19.9</td>
<td>18.1</td>
<td>9.7</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>16.8</td>
<td>14.7</td>
<td>13.7</td>
<td>12.1</td>
</tr>
<tr>
<td>0.1</td>
<td>300</td>
<td>13.6</td>
<td>11.3</td>
<td>7.9</td>
<td>6.1</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>9.3</td>
<td>7.3</td>
<td>6.5</td>
<td>5.3</td>
</tr>
</tbody>
</table>

The results are comparable to Choi et al (1998) who reported a 74% improvement over PID. At the same time, recall that the NN of Choi et al was offline and had to be trained for each operating condition. The inherent advantage of this ANNC is that it is an online NN and consequently retrains itself as the operating condition change. Finally, although accuracy is on the order of ± 6 mm which seems poor, as a percentage of amplitude this equates to ± 2%, which
is comparable to the percent accuracy reported by Choi et al (1998) and Bone et al (2007). It is concluded that the amount of percentage improvement justifies the efforts taken in implementing the ANNC.

4.4.4 Transient Behavior of ANNC

A series of additional tests were conducted at 0.5 $Hz$ with 250 $mm$ amplitude, to study the transient behavior of ANNC and to confirm that it was indeed “adapting”. Figure 4-18 illustrates system response when the ANNC is turned on and off as the compensator for the tuned PID. There are three events to highlight:

Event A (10 s) – ANNC learning starts ($u_x = u_{PID}$)
Event B (20 s) – ANNC connected ($u_x = u_{PID} - u_{NN}$)
Event C (30 s) – ANNC disconnected ($u_x = u_{PID}$)

The active adaptive nature of ANNC is clearly visible and the result for this particular case is a 61% reduction in AVGE. The weights reach their new values in just one cycle ($W_{15}$). Note that weighted sum is the sum of the elements of $W$ and thus exceeds the individual limit of 10 that was seen in Table 4-7.
Figure 4-18 PID+ANNC response showing transient with events A, B and C highlighted
As the summary of this section, it was found that by combining ANNC with a traditional PID controller, tracking performance could be improved on the order of 45% to 66%. This level of performance was achieved after careful tuning of both the ANNC and PID components. A key observation is that tuning of ANNC requires no more effort than the tuning of PID, in that they both require the user to find values for 3 important parameters.

4.5 ANNonly Results

In Taghizadeh (2010) some work was conducted on NN as a velocity controller. He had not used an ANNonly controller for position control of the cylinder. In Figure 4-19 the structure of an ANNonly position controller is shown.

![Figure 4-19 Block diagram for ANNonly controller and pneumatic system](image)

Taghizadeh used \( n_h = 10 \) in his thesis (number of nodes in hidden layer). However, preliminary tests showed that selecting higher number for \( n_h \) will improve the response of the ANNonly controller. On the other hand, there are hardware limitations in increasing this value, since it makes the program bigger. This increase in program size can be compensated by deleting some of the inputs, and \( n_h \) was consequently increased to 12. After some trial and error, the inputs in Figure 4-20 were selected. Adding \( u_{NN} \) helped improve the result noticeably, which was not the case for PID+ANNC.
Next step was to adjust the ANN model parameters. As was presented in Chapter 3, the update rule for the first layer of weights and the calculation of parameter $D$ is as follows:

$$
\dot{V} = G [ P (W^T \hat{\sigma} (V^T P) e)^T - \lambda V \|e\|_F ] 
$$

(4-7)

$$
D = -K_z (\|Z\|_F + \bar{Z}) e - K_v e
$$

(4-8)

In these equations $\lambda$ is a small positive tunable parameter whose function is to help deal with unmodeled dynamics and $D$ is designed to overcome higher order modeling errors (Lewis et al 1999). During experiments, it was found that an increase in $\lambda$ gave a smoother response. However, there was no noticeable improvement to the performance measures (AVGE and RMSE) which again addresses the frequency tracking issue discussed in Section 4.1.

Regarding $D$, it consists of a standard back propagation term ($1^{st}$ term in the Equation 4-8), plus the error modification term taken from adaptive control (last term). It is a given that an increase
in any of the parameters in Equation 4-7 will increase the absolute value of D. It is assumed that when the tracking frequency is greater the higher order modeling errors become more substantial; i.e. a greater absolute value of D and a less value of $\lambda$ would give a better response compared to PID results. This is confirmed in Table 4-11, where better results with higher values of $K_v$, $K_z$ for 0.5 Hz are observed relative to tuned PID (AVGE from 116 to 46). Note that the results are obtained for 250 mm sine wave tracking under 500 kPa supply pressure.

Table 4-11 Summary of ANNonly performance results for different sine wave frequencies

<table>
<thead>
<tr>
<th>Controller</th>
<th>ANN parameters</th>
<th>Frequency (Hz)</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANNonly controller</td>
<td>$K_v=K_z=2.2$, $Z=1$, $\lambda=4.5$</td>
<td>0.5</td>
<td>54.7</td>
<td>46.2</td>
</tr>
<tr>
<td></td>
<td>$K_v=K_z=1.7$, $Z=1$, $\lambda=9$</td>
<td>0.2</td>
<td>31.2</td>
<td>27.8</td>
</tr>
<tr>
<td></td>
<td>$K_v=K_z=1.7$, $Z=1$, $\lambda=9$</td>
<td>0.1</td>
<td>18.6</td>
<td>15.5</td>
</tr>
<tr>
<td>Tuned PID gains</td>
<td>$K_p=2.25$, $K_i=9$, $K_d=0.6$</td>
<td>0.5</td>
<td>131.5</td>
<td>116.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td>16.8</td>
<td>14.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1</td>
<td>9.3</td>
<td>7.3</td>
</tr>
</tbody>
</table>

Figures 4-21, 4-22, 4-23 demonstrate the response with ANNonly. As can be seen in the figures the response is different from Figures 4-15, 4-16, 4-17. The amount of oscillation in the position signal with ANNonly is higher in the lower tracking frequencies compared to PID+ANNC.
Figure 4-21 ANNonly controller response for 0.5 Hz sine wave tracking

Figure 4-22 ANNonly controller response for 0.2 Hz sine wave tracking
In conclusion, the ANNonly controller results for the \(x\)-axis were disappointing except at high frequencies. At the same time, the implicit objective of this research is to reduce the tuning effort. The proposed methods, ANNC and ANNonly, require extensive tuning which is even more complicated than PID tuning.

### 4.6 FAPID

To further explore the potential of adaptive control methods, the implementation of a Fuzzy Adaptive PID (FAPID) controller on the \(x\)-axis is presented in this section. The adopted FAPID controller is a variable gain PID controller whose gains are adjusted online with a set of fuzzy rules. In Figure 4-24 the Simulink implementation of the proposed method is depicted. The background of this controller was provided in Chapter 3. Details on input and output MFs and fuzzy rules are given in this section.
For the initial design of the FAPID controller, Manjunathm and Janaki (2011) fuzzy rules were used. These rules are given in Table 4-12. One implementation problem was the inability of dSPACE to compile the full set of rules. Hardware constraints required that the compiled program have a limited size. Using 49 rules for each of the fuzzy PID gains was found to be not possible and the number of rules had to be reduced to 15 for each of the PID gains. The selected portions of the fuzzy control rules are highlighted in Table 4-12. Note that this rule set still relates all the fuzzy PID gains to $e_x$ and $v_x$. 

Figure 4-24 Simulink block diagram for the $x$-axis FAPID controller
Table 4-12 Fuzzy rules for PID gains taken from Manjunathm and Janaki (2011)

\[ \text{Kp Fuzzy Control Rule} \]

\[ \text{Kd Fuzzy Control Rule} \]

\[ \text{Ki Fuzzy Control Rule} \]

\[ \text{Z} \]

7 M= Medium
The FAPID controller was implemented using the FIS editor of the MATLAB fuzzy toolbox and the following parameters were used as the fuzzy set operation for the fuzzy Mamdani type:

And Method = min  Implication = min  Aggregation = max  Defuzzification = centroid

Initial tracking results were not good relative to PID, as reported in previous sections of this chapter. One of the reasons for this poor performance was thought to be the resolution of the MFs. As the number of the MFs had to be limited for compatibility issues with dSPACE, the resolution of the MFs was decreased so they could cover the whole range of the input. Therefore the design procedure, originally given as Figure 3-17 in Chapter 3, had to be reconsidered and the fuzzy blocks retuned.

On reflection, the assignment of $K_d$ and $K_i$ to $e_x$ and $v_x$ was not justified, at least for this application. Following a series of trial and error tests, it was confirmed that relating the PID gains to the terms for which they are most dependent, would result in better adaptation. Thus, it was decided to associate $K_i$ to $\dot{e}_x$ and $K_d$ to $v_x$. In the new version for the fuzzy structure, $K_i$ and $K_d$ become 1 input and 1 output fuzzy systems. This means that the number of rules reduces noticeably and consequently the number of MFs assigned to $K_i$, $\dot{e}_x$ or $K_d$, $v_x$ can be increased; i.e. the resolution and precision of the MFs becomes higher. However, the $K_p$ fuzzy control rules were kept as before and the reduced number of $K_p$ rules in Table 4-12 was used. The surface corresponding to the three dimensional rules of $K_p$ is given as Figure 4-25.

Figure 4-25 FAPID surface rule view of proportional gain $K_p$ with $v_x$ and $-e_x$ as inputs
Figure 4-26 shows the adopted triangular MFs for inputs, $e_x$ and $v_x$, and output $Kp$ of this fuzzy unit which were implemented in the FIS editor. Triangular MFs are the simplest type available in the FIS editor. More complex MFs like Gaussian and Sigmoidal MFs were found to be not compatible with MATLAB real-time workshop software. The same Mamdani fuzzy set operations (default set) were selected.

![Figure 4-26 FAPID membership functions for $Kp$, $-e_x$, $v_x$](image)

Table 4-13 illustrates the 7 rules that replaced the 49-rule table of Manjunathm and Janaki. Figure 4-27 shows the MFs of $Kd$ and $v_x$, respectively. One notes that $Kd$ is bigger for smaller
absolute values of velocity. The idea behind these proposed rules was that the derivative control action which is $Kd^h v_x$ has to be kept at approximately the same level for different values of $v_x$.

Table 4-13 Fuzzy rules for Kd

<table>
<thead>
<tr>
<th>$V_x$</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZO</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Kd$</td>
<td>VS</td>
<td>S</td>
<td>M</td>
<td>B</td>
<td>M</td>
<td>S</td>
<td>VS</td>
</tr>
</tbody>
</table>

Figure 4-27 FAPID membership functions for derivative gain $Kd$ and $v_x$

Table 4-14 gives the fuzzy rules for $Ki$ and Figure 4-28 plots the MFs of $Ki$. Once again, the number of rules has decreased to 7 and the resolution of the MF has increased. The same concept as with $Kd$ is applied for determination of the fuzzy $Ki$ rules; i.e. high values of $Ki$ for low absolute values of $\dot{v}_x$. In this way, low steady state errors would cause large integration action
due to high value of $Ki$. This in turn would force the controller to follow the trajectory with higher precision.

Table 4-14 Fuzzy rules for Ki

<table>
<thead>
<tr>
<th>$\delta_X$</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZO</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Ki$</td>
<td>VS</td>
<td>S</td>
<td>M</td>
<td>B</td>
<td>M</td>
<td>S</td>
<td>VS</td>
</tr>
</tbody>
</table>

![Figure 4-28 FAPID membership functions for integral gain $Ki$ and $\delta_X$.]

4.6.1 Tracking Results

In the previous section, the structure and different elements of FAPID was presented. It was observed that with simpler fuzzy rules for $Ki$ and $Kd$, better results could be obtained. Table 4-15 reports the sine wave tracking results for Manjunathm and Janaki (MJ) fuzzy rules versus the reduced set of fuzzy rules, presented in previous section (BD rules).
Table 4-15 Summary of FAPID performance results for different sine wave frequencies

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Controller</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
<th>ΔRSME %</th>
<th>ΔAVGE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>PID</td>
<td>132</td>
<td>116</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>MJ fuzzy rules</td>
<td>91</td>
<td>79</td>
<td>-30</td>
<td>-31</td>
</tr>
<tr>
<td></td>
<td>BD fuzzy rules</td>
<td>89</td>
<td>73</td>
<td>-32</td>
<td>-34</td>
</tr>
<tr>
<td>0.2</td>
<td>PID</td>
<td>16.8</td>
<td>14.7</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>MJ fuzzy rules</td>
<td>32</td>
<td>28</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>BD fuzzy rules</td>
<td>21</td>
<td>18.3</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>0.1</td>
<td>PID</td>
<td>9.3</td>
<td>7.3</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>MJ\textsuperscript{8} fuzzy rules</td>
<td>14.4</td>
<td>11.5</td>
<td>54</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>BD\textsuperscript{9} fuzzy rules</td>
<td>9.9</td>
<td>8</td>
<td>5</td>
<td>9.5</td>
</tr>
</tbody>
</table>

The key observation for these results is that performance was improved by reducing the number of rules and changing the governing rules for $Ki$ and $Kd$ (BD results compared to MJ results). However, FAPID needs extensive MF tuning to be able to improve the results for any given frequency. In this table it seems that the assigned MFs are only well suited for 0.5 Hz, with 35% reduction in RMSE and AVGE. But PID performed better at 0.2 Hz and 0.1 Hz. Figures 4-29 and 4-30 depict the FAPID response with MJ and BD set of fuzzy rules for 0.1 Hz sine wave tracking. As can be seen the error with BD rules is lower. It is evident in the figures that fuzzy PID gains are adapting. In addition, one can see the range and shape of the fuzzy inputs, $\xi$ and $\nu$, in the figures.

\textsuperscript{8}MJ=Manjunathm and Janaki

\textsuperscript{9}BD = Behrad Dehghan
Figure 4-29 FAPID response with MJ fuzzy rules for 0.1 Hz sine wave tracking
Figure 4-30 FAPID response with BD fuzzy rules for 0.1 Hz sine wave tracking
4.7 Summary

Experimental results were collected for the tracking of a 0.5 Hz sine wave. In Figure 4-31 RMSE and AVGE performance results for the different controllers studied in this chapter are compared. It noted that FAPID results were not as good as ANNonly and PID+ANNC. The reason thought to be improper assignment of MFs for FAPID controller.

![Bar graph comparison of 0.5 Hz sine wave tracking performance results](chart.png)

In this chapter, implementation of PID, Fuzzy, PID+ANNC, ANNonly, and FAPID position controllers were explained and evaluated. All controllers were carefully tuned to provide a fair comparison. The advantage of any adaptive method is that it provides robustness against changes to operating conditions. More specifically, the advantage of FAPID and PID+ANNC techniques is that once they are setup and tuned, one can apply them to any PID controller and be assured that performance will improve. Improvement is expected given the adaptive nature of both methods. However, both adaptive methods required fairly significant effort to setup and tune. The tuning procedure for the fuzzy controller is judged to be more intuitive in nature.
Chapter 5

Position Control of a Rodded Pneumatic Cylinder (z-axis)

The contents of this chapter are an expansion of the paper “Comparison of Fuzzy and Neural Network Adaptive Methods for the Position Control of a Pneumatic System” (Dehghan and Surgenor, 2012b). In this chapter the position controllers that were tested on the x-axis in Chapter 4 have been modified for application to the rodded z-axis cylinder. Since, the fuzzy controller results on the x-axis were not satisfactory this method was not used on the z-axis. The following control methods were tested:

1. PID control
2. ANNonly control
3. PID+ANNC
4. FAPID control

A series of open loop and PID tests were first conducted to gain familiarity with the z-axis.

5.1 Open Loop and PID Tests

An initial set of PID experiments were conducted on the z-axis of the gantry apparatus in order to gain familiarity with the tuning of the z-axis. The main problem with the z-axis in comparison to the x-axis is the short stroke of the cylinder that restricts the amount of movement. On the other hand, the advantage is that the friction is less, since the cylinder is rodded, in comparison to the high friction rodless cylinder.

The original installed cylinder on the z-axis was a BIMBA with bore diameter of 0.01905 m and stroke of 0.1016 m. The interpretation of the initial open loop tests was that the valve was oversized for the cylinder; i.e. the nominal flow rate of the z-axis Festo valve, which is 100 l/min, is more than what was necessary. As it can be seen in the PI response of Figure 5-1, the
maximum control signal reaches only 15% of its range. This mismatch can also be interpreted from the low values of the PID gains; higher values would increase the overshoot and lead the system to instability.

As a result, modification to the z-axis cylinder of the apparatus seemed to be required. It was necessary to either change the valve and replace it with a smaller valve having lower nominal flow, or replace the cylinder with a higher volume cylinder. The Festo valve was first replaced with a HRTextron valve, since it was available in the lab. The results of the HRTextron open loop and PID tests are provided in Appendix A. As explained in the appendix, the results with the HRTextron valve were no better, as they were roughly the same size. Next, an effort was made to look for a smaller replacing valve. However, it appeared that the Festo and HRTextron valves were the smallest proportional valves available, commercially. Thus, the final decision was to find a new larger cylinder that still fit in the available space in the gantry robot.

Figure 5-1 PI response for original cylinder for step tracking ($K_p=0.7, K_i=0.2$)
To select a new cylinder, it was assumed that sinusoidal tests up to 1 Hz with 40 mm amplitude would be conducted on the z-axis. This means that the cylinder had to translate 160 mm in 1 s, or in other words, have 0.16 m/s of average velocity. Assume that the rod velocity would reach a maximum of 0.3 m/s at max. Given the Festo valve 100 l/min nominal rate (0.0016 m³/s) the cylinder area has to be 0.0033 m². An area of 0.0033 m² equates to a cylinder with a bore diameter of 65 mm. However, a cylinder with this bore would not fit in the available space of the z-axis. Instead a BIMBA cylinder with a bore diameter of 31 mm and stroke of 124 mm was selected. Note this was the largest cylinder that would fit in the available space.

Figure 5-2 illustrates the open loop results for 1 Hz 1 v square wave test under 350 kPa supply pressure with the original cylinder. Cylinder velocity exceeds 0.5 m/s in this case. Figure 5-3 shows the open loop test with the new larger cylinder under the same supply pressure. In comparing Figure 5-3 to 5-2, one sees that although the control signal is larger, the velocity is lower, as would be expected with a larger cylinder. Figure 5-4 gives the tuned PID response for the new cylinder for step tracking with 500 kPa as the supply pressure. As it can be seen the PID controller is doing a good job in following the square wave trajectory. Although the velocity in Figure 5-1 and 5-4 are almost the same, the larger controls signal in Figure 5-4 means a higher degree of control signal resolution and consequently better controllability.
Figure 5-2 Open loop for original cylinder for 1 Hz square wave test ($amp=1$ v)

Figure 5-3 Open loop for new cylinder for 1 Hz square wave test ($amp=1.5$ v)
After getting satisfactory results with the step tracking responses of the new cylinder, tuning of the PID controller was conducted for sine wave tracking. Following the same process mentioned in Chapter 3, $K_p=4$, $K_i=4$, and $K_d=0.6$ were selected as the tuned PID gains. In Figure 5-5, the 0.5 Hz sine wave tracking result for the tuned PID controller is given. The experiment took place under $P_s=500 \text{ kPa}$ and for 40 mm sine wave amplitude. As in Chapter 4, the chosen amplitude for the $z$-axis was dictated by the need to maximize the movement with respect to available cylinder stroke, without hitting the endstops.
5.2 Parameter Modifications for z-axis

The cylinder properties were provided in the previous section. Due to differences in the nature of the z-axis and x-axis, it is assumed that the controller parameters for Chapter 4 would have to be modified. In other words, the difference in size and type is more than can be handled by the adaptive nature of the FL or the ANN. In this section, the parameters that need to be modified are explained and the tuned process for the z-axis is explained.
5.2.1 PID+ANNC

Since the cylinder characteristic is different, the ANN tuning procedure explained in Chapter 3, had to be redone. Another objective of this subsection is to distinguish the level of improvement when ANNC is applied to untuned PID. For this reason $K_p=K_i=2$, $K_d=0$ were arbitrarily selected to play the untuned PID role. Figure 5-6 gives the response of the PID controller for 0.1 $Hz$ sine wave tracking with amplitude of 35 $mm$. It illustrates the transient response of ANNC when it is added to the untuned PID, at 15 $s$ into the test.

![Figure 5-6 Untuned PI+untuned ANNC (at t=15 s) for 0.1 Hz sine wave tracking](image)

After an extensive set of trial and error tests were conducted, the same input vector as shown in Figure 4-14 Chapter 4 was chosen. In addition the coarse tuning procedure resulted in the same values as reported in Table 4-8 Chapter 4. However, it was found that performance was more sensitive to the value of $D$ in ANNC, compared to the $x$-axis. This may come from the difference in stroke and friction of the cylinders. The coarse tuning parameters reported in Chapter 4 were kept constant and the best values found for ANN fine tuning parameters were $K_v=K_z=\lambda=4.5$. Figure 5-7 plots untuned PID+tuned ANNC, where the ANNC parameters have
been modified to give better performance. Note that better results in Figure 5-7 are due to modified ANNC. Numerical performance results are given in Table 5-1. The error decreases from over 10 mm to less than 5 mm with ANNC. The ANNC weights adapt in less than one period of tracking for both cases (10 s). However, both experiments suffer from the ANNC initialization and the resultant “jerk” in the control signal, when turned on. A timer can be applied to avoid this phenomenon and give a smoother start to ANNC.

Figure 5-7 Untuned PI+ tuned ANNC (at t=15 s) for 0.1 Hz sine wave tracking (Kp=2, Ki=2)

Figure 5-8 and 5-9 are very important as they demonstrate the value of ANNC. The hypothesis was that adding an adaptive extension to an untuned PID controller would result in better performance than a tuned PID controller. In this approach, instead of finding optimal values for the PID gains, which is time-consuming, one can apply the ANNC to an untuned PID controller and be certain that it will result in same or even better performance. Figure 5-8 shows the performance of tuned PID while Figure 5-9 depicts the performance of untuned PID+ANNC.
Figure 5-9 is in fact showing what is happening after ANNC settles in Figure 5-7. As reported in Table 5-1 the tracking error of untuned PID+ANNC is less than that for tuned PID.

Note that the control signal in Figure 5-9 shows considerably less chatter than that one seen in Figure 5-8; less chatter means less wear on the valve.

Figure 5-8 Tuned PID response for 0.1 Hz sine wave tracking ($K_p=4$, $K_i=4$, $K_d=0.6$)

Figure 5-9 Untuned PI+ modified ANNC for 0.1 Hz sine wave tracking ($K_p=2$, $K_i=2$, $K_v=K_z=\lambda=4.5$)
Table 5-1 summarizes the results taken from initial and tuned ANNC which are added to untuned and tuned PIDs. The table shows the 0.1 Hz sine wave tracking results with 40 mm amplitude. Note that the results are given in ranked order. The percentage of improvement is given and the best result is taken from tuned PID + tuned ANNC. The relative percentage of improvement is 71.5% referenced against the performance of PID with untuned gain settings.

Following points summarize the important details of Table 5-1:

- Improvement when ANNC was added was greater for untuned PID (RMSE 4.8 to 1.25) than for tuned PID (RMSE 2.9 to 1.2).
- Untuned PID + tuned ANNC shows better results than tuned PID alone (58.5% to 75%), which highlights that ANNC can be added without prior tuning of PID and overall will result in better performance.
- Tests were also conducted with tracking frequencies of 0.2 Hz and 0.5 Hz, and the conclusions with respect to relative performance were the same.
- The relative error of best performance, Tuned PID+Tuned ANNC, with respect to z-axis stroke and sine wave amplitude is $1.1/124=0.88\%$ and $1.1/40=2.75\%$, respectively.

<table>
<thead>
<tr>
<th>Controller</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
<th>RMSE (%)</th>
<th>AVGE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuned PID+ tuned ANNC ($K_v=K_z=\lambda=4.5$)</td>
<td>1.2</td>
<td>1.1</td>
<td>-75</td>
<td>-71.5</td>
</tr>
<tr>
<td>Untuned PID + tuned ANNC</td>
<td>1.25</td>
<td>1.15</td>
<td>-74</td>
<td>-69.5</td>
</tr>
<tr>
<td>Tuned PID+ANNC</td>
<td>2.0</td>
<td>2.0</td>
<td>-58.5</td>
<td>-47</td>
</tr>
<tr>
<td>Tuned PID ($K_p=K_i=4, K_d=0.6$)</td>
<td>2.9</td>
<td>2.1</td>
<td>-45</td>
<td>-52.5</td>
</tr>
<tr>
<td>Untuned PID+ untuned ANNC ($K_v=K_z=0.5, \lambda=4.5$)</td>
<td>3.8</td>
<td>3.5</td>
<td>-21</td>
<td>-7.9</td>
</tr>
<tr>
<td>Untuned PID ($K_p=K_i=2, K_d=0$)</td>
<td>4.8</td>
<td>3.8</td>
<td>ref</td>
<td>ref</td>
</tr>
</tbody>
</table>
5.2.2 ANNonly

The ANNonly controller showed acceptable results in Chapter 4, when it was tested with the \( x-axis \) cylinder. The same design was implemented on the \( z-axis \) without further tuning of the ANNC parameters. In other words, the ANNonly controller was applied to the \( z-axis \) with the values from PID+ANNC in Section 5.2.1. So, once again, the important values of the ANN were \([K_v K_z \lambda] = [4.5 4.5 4.5]\)

According to Table 5-2, the higher order modeling errors become more substantial when the frequency is higher or when the friction is less (rodded cylinder). It is because in these two cases the term associated with these errors has become more significant in comparison to the other errors in the system. Therefore, as shown in the table, high values of \( K_v \) and \( K_z \) led to better performance. Table 5-2 compares the tuned PID \((K_p=4 K_i=4 K_d=0.6)\) and ANNonly controller \((K_v=K_z=4.5, \lambda=4.5)\) responses for different sine wave frequencies. Figure 5-10 plots the ANNonly response for 0.5 Hz sine wave tracking.

Table 5-2 Summary of ANNonly performance results for different sine wave frequencies

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Controller</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
<th>RMSE %</th>
<th>AVGE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>PID</td>
<td>8.8</td>
<td>8</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>ANNonly</td>
<td>4.85</td>
<td>4.5</td>
<td>-44</td>
<td>-43</td>
</tr>
<tr>
<td>0.2</td>
<td>PID</td>
<td>5.7</td>
<td>4.7</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>ANNonly</td>
<td>2.8</td>
<td>2.7</td>
<td>-50</td>
<td>-42.5</td>
</tr>
<tr>
<td>0.1</td>
<td>PID</td>
<td>2.9</td>
<td>2.1</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>ANNonly</td>
<td>1.9</td>
<td>1.8</td>
<td>-34</td>
<td>-14</td>
</tr>
</tbody>
</table>
Since improvement was witnessed in all frequencies with respect to tuned PID, it was concluded that ANNonly controller works better on the \textit{z-axis} rather than it does on the \textit{x-axis} cylinder.

### 5.2.3 FAPID

After implementing the FAPID on the rodless \textit{x-axis} cylinder, it was decided to use the same fuzzy blocks, inputs and rules for the \textit{z-axis} cylinder. However, the fuzzy MFs should be adjusted to fit the properties of \textit{z-axis}. This modification was not as hard as in Chapter 4, as the tuning procedure was now well established. Similar to the previous chapter, the rules were implemented using the FIS editor of MATLAB and the default system operations were adopted for setting up the fuzzy algorithm within FIS editor. MFs associated with the \textit{z-axis} FAPID controller are presented in the following plots. Hence, controller inputs are error in position and velocity of the cylinders and the output is a voltage that is fed to the PFC valve. The inputs of the fuzzy block are $e_z$ and $v_z$, $\int v_z$ for $Kp$, $Ki$, and $Ki$.

Figure 5-11 plots symmetrical fuzzy rule surface of $Kp$ versus $v_z$ and $z$. 

---

Figure 5-10 ANNonly response for 0.5 Hz sine wave tracking ($\lambda=K\nu=Kz=4.5$, $Z=0.5$)
Figure 5-12 and 5-13, show the triangular MFs of the $\hat{v}_z$, $Ki$ and $v_z$ and $Kd$, respectively. The overall range of the MFs was selected after conducting a number of trial and error experiments.
Figure 5-12 Fuzzy membership functions for $e_z$ and $Ki$, respectively

Figure 5-13 Fuzzy membership functions for $v_z$ and $Kd$
FAPID sample results for 0.5 and 0.1 Hz sine wave tracking are given as Figures 5-14 and 5-15, respectively. In both cases, the controller was required to track a sine wave setpoint signal with 40 mm amplitude under 500 kPa supply pressure. The tracking error, cylinder position and control signal, in addition to fuzzy PID gains, are plotted. One can see that the fuzzy PID gains are adapting themselves as the test progresses. It is evident that the method is doing a good job for 0.1 Hz, in Figure 5-15. On the other hand there is relatively high tracking error for 0.5 Hz in Figure 5-14. This is due partly to limited cylinder band width and partly because of the valve, which apparently cannot deliver on demanded values for the control signal. Performance measures regarding these figures are provided in the next section.

![Figure 5-14 FAPID response with BD fuzzy rules for 0.5 Hz sine wave tracking](image-url)
5.3 Performance Comparisons

After running the system with different sine wave frequencies, performance measures are reported in Table 5-3 for constant amplitude of 40 mm. It has to be noted that the fuzzy rules of Manjunathm and Janaki (2011) were also tested (designated as MJ rules) and the associated responses were noticeably worse than with the set of fuzzy rules adopted in this thesis (designated as BD rules).
Table 5-3 Summary of FAPID performance for different frequencies with comparison to Manjunatham and Janaki (2011)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Controller</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
<th>ΔRMSE %</th>
<th>ΔAVGE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PID</td>
<td>16.1</td>
<td>13</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>MJ(^{10}) fuzzy rules</td>
<td>11.8</td>
<td>10.4</td>
<td>-26.7</td>
<td>-20</td>
</tr>
<tr>
<td></td>
<td>BD(^{11}) fuzzy rules</td>
<td>10</td>
<td>9.5</td>
<td>-37.8</td>
<td>-26.9</td>
</tr>
<tr>
<td>0.5</td>
<td>PID</td>
<td>8.8</td>
<td>8</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>MJ fuzzy rules</td>
<td>9.55</td>
<td>8.4</td>
<td>8.5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>BD fuzzy rules</td>
<td>8.1</td>
<td>7.1</td>
<td>-8</td>
<td>-11</td>
</tr>
<tr>
<td>0.2</td>
<td>PID</td>
<td>5.7</td>
<td>4.7</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>MJ fuzzy rules</td>
<td>5</td>
<td>3.9</td>
<td>-12</td>
<td>-17</td>
</tr>
<tr>
<td></td>
<td>BD fuzzy rules</td>
<td>4.1</td>
<td>3.4</td>
<td>-18</td>
<td>-27.6</td>
</tr>
<tr>
<td>0.1</td>
<td>PID</td>
<td>2.9</td>
<td>2.1</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>MJ(^{12}) fuzzy rules</td>
<td>2.45</td>
<td>1.8</td>
<td>-15</td>
<td>-14.2</td>
</tr>
<tr>
<td></td>
<td>BD(^{13}) fuzzy rules</td>
<td>1.8</td>
<td>1.37</td>
<td>-40</td>
<td>-34.6</td>
</tr>
</tbody>
</table>

Table 5-3 verifies that the BD reduced set of fuzzy rules show better results than the MJ set of rules, even though the number and complexity of the rules has been decreased.

\(^{10}\) MJ = Manjunatham and Janaki

\(^{11}\) BD = Behrad Dehghan
Figure 5-16 summarizes the results of Chapter 5. It also verifies that the advantage of both adaptive techniques (FAPID and ANNC). Namely, that once they are setup, one can apply them to any PID controller (tuned or untuned) and be assured that performance will improve.

![Figure 5-16 Bargraph comparison of 0.1 Hz sine wave tracking performance results](image)

**5.4 Summary**

The main conclusion of this chapter is that both FAPID and PID+ANNC methods can significantly improve tracking performance. PID+ANNC found to improve tracking performance over untuned PID by upwards of 71.5% (AVGE from 3.8 to 1.1 mm). FAPID was found to improve tracking performance over fixed gain PID by upwards of 63% (AVGE from 3.8 to 1.4 mm). Improvement is expected given the adaptive nature of both methods. However, the level of improvement is considered significant, with only a 30% improvement expected.
Both adaptive methods required fairly significant effort and tuning to set up. However, the tuning procedure for the fuzzy controller is judged to be more intuitive in nature. Thus, adaptive fuzzy is considered to be more practical than the ANNC approach. A major contribution of this chapter is demonstration that a novel fuzzy rule set that is reduced in size with rules modified from those used in previous studies (i.e. Manjunathm and Janaki), can improve performance and reduce the computation overhead of the control algorithm.

In the next chapter, the addition of a second adaptive mechanism is considered to reduce the amount of effort required to setup FAPID.
Chapter 6

Modified Fuzzy Adaptive PID Position Control

The contents of this chapter are an expansion of the paper “Comparison of Intelligent PID Position Controllers with Autotuners for a Pneumatic System” (Dehghan and Surgenor, 2012a). In Chapter 5, it was shown that FAPID control produced satisfactory results in controlling the z-axis cylinder. For adapting the MFs, which were tuned for a specific frequency, fuzzy weights were designed. Arriving at the final FAPID weights was achieved manually. It was hypothesized that these weights could be obtained online with an optimization method.

In this chapter the optimization method is implemented and designed to arrive at a final set of weights. The role of these weights and how they shift the MFs will be discussed in detail. A similar adaptive technique will be applied to the PID controller to replace the systematic hand-tuning presented in Chapter 3. Results will be presented for both the x-axis and z-axis.

6.1 Need for Adaptive PID

The need for an adaptive mechanism is illustrated by noting the change in performance of both cylinders over a period of months. The changes as benchmarked with a PID controller are summarized in Table 6-1 and 6-2. With the PID gains unchanged, there is a noticeable change in performance for both axes, especially the x-axis. As was reported in Chapter 3, relocation of valves, and fittings did affect the responses of the x-axis. However, the performance changes on the z-axis are more attributed to a change in friction, air supply quality and seal characteristics due to mechanical wear. Note that these z-axis results are for the new (bigger) cylinder.
Table 6-1 Change in x-axis sine wave response compared to Taghizadeh (2010),
\((Kp=2.25, Ki=9, Kd=0.6, amp=300 \text{ mm})\)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Time of experiment</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Taghizadeh</td>
<td>35</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Initial (Apr 2011)</td>
<td>13.6</td>
<td>11.3</td>
</tr>
<tr>
<td>0.2</td>
<td>Taghizadeh</td>
<td>51</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>Initial (Apr 2011)</td>
<td>19.9</td>
<td>18.1</td>
</tr>
<tr>
<td>0.5</td>
<td>Taghizadeh</td>
<td>122</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>Initial (Apr 2011)</td>
<td>180.5</td>
<td>171.2</td>
</tr>
</tbody>
</table>

Table 6-2 Change in z-axis sine wave response over 5 months

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>PID gains</th>
<th>Time of experiment</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>(Kp=4) () (Ki=4) (Kd=0)</td>
<td>Initial (June 2011)</td>
<td>3.2</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>After 5 Months</td>
<td>2.9</td>
<td>2.1</td>
</tr>
<tr>
<td>0.5</td>
<td>(Kp=4) () (Ki=4) (Kd=0.6)</td>
<td>Initial (June 2011)</td>
<td>8</td>
<td>7.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>After 5 months</td>
<td>8.8</td>
<td>8</td>
</tr>
</tbody>
</table>

Variation in responses over time can be also illustrated by open loop tests. Open loop tests were conducted and compared to the same open loop tests of 5 months earlier. Main difference between these two set of tests was a change of change in the cylinder velocity. Table 6-1 and 6-2 demonstrate that the difference between tests over a period of months can be significant and the PID gains must be retuned. By contrast, difference between tests on a given day is less than
0.1 mm in AVGE. This small difference, with respect to the setpoint amplitude, for same-day tests is probably due to random behavior of friction and variations in supply pressure.

In this chapter, the tuning process for the PID gains and FAPID weights has been modified and an adaptive mechanism has been incorporated into the design. This adaptive mechanism will adjust the controller parameters to become independent of the operating conditions.

### 6.1.1 Adaptive Method to Minimize AVGE

The proposed adaptive mechanism sets out to search through a 3-dimensional space which includes all possible alternatives in order to minimize the AVGE by adjusting PID gains in case of adaptive PID (MPID) or the weights in case of Modified FAPID (MFAPID). A genetic algorithm (GA) approach for this minimization algorithm was initially considered due to its success reported by other researchers (Khan et al, 2008). However, the use of a GA based optimization technique, which is an optimized search method in the space of achievable gains, is considered risky for experimental work. For example, random based optimization techniques, like GA, can generate PID gains which when used, can cause instability or oscillatory responses for high values of $K_p$. A review of the GA literature, also, reveals that these methods tend to have long convergence times which make the real-time optimization application not feasible (Wang et al, 2009). Furthermore, the (Genetic Algorithm toolbox) GADS toolbox does not support Embedded MATLAB function block, and integration between MATLAB and Simulink is not supported for Real-Time code generation. The only way to overcome this problem is to bring everything into Embedded MATLAB function block, i.e. without using extrinsic functions. In other words, writing an equivalent GA in an Embedded MATLAB function block is the only solution. Due to the complexity involved in writing such a program, it was decided to take a less sophisticated approach to design the adaptive mechanism. In Section 6.2 this approach will be discussed for modified FAPID and the same mechanism then will be applied to PID in Section 6.3.

### 6.2 Modified FAPID (MFAPID)

After a large number of experiments with FAPID, it was found out that the degree of improvement is different with same the MFs for different input frequencies. Therefore, the need
for an adaptive mechanism to handle this effect became more apparent. As was explained in Chapter 5 (Section 5.3), one approach is to introduce weights to the output of the fuzzy algorithm to mimic changes in the MFs. The adaptive mechanism would manipulate the weights in an online fashion to decrease the AVGE in series with the adaptive nature of the fuzzy rule-based controller. Figure 6-1 is similar to the block diagram given as Figure 3-19 in Chapter 3, except that it has added weights, that are tuned manually (i.e. fixed weights)

![Block diagram for FAPID controller with embedded weights added](image)

**Figure 6-1** Block diagram for FAPID controller with embedded weights added

Figure 6-2, gives the FAPID response with manually tuned weights for the *x-axis* tracking a 0.5 Hz sine wave. The fuzzy weights are *Wp*=2, *Wi*=0.8, *Wd*=0.3. RMSE and AVGE of this plot are 43.5 and 36, respectively. These values when compared to the original FAPID results given in Table 4-15 (RMSE=89, AVGE=73) signify a noticeable improvement in performance, 50%. This demonstrates the effectiveness of using embedded weights and the fact that the MFs in FAPID controller, presented in Chapter 4, were not adequately tuned.

The weights used for Figure 6-2 were tuned for 0.5 Hz sine wave tracking with P=500 kPa supply pressure. These weights would not remain tuned if one of the operating conditions such as supply pressure, frequency, amplitude or type of the input signal changed. This observation, once again, points out the importance of an additional adaptive mechanism to change the weights in an online fashion.
6.3 Optimization of Fuzzy Weights

The adopted FAPID controller used in Chapters 4 and 5, is a variable gain PID controller whose gains are adjusted online with a set of fuzzy rules. Figure 6-3 is same as Figure 6-1, except that it has modified with an added adaptive mechanism for online tuning (of the weights). Note that the adaptive mechanism block comes after the fuzzy PID gain blocks. Both the fuzzy gain blocks and the adaptive mechanism block work to minimize the AVGE.
In essence, the adaptive mechanism applies $W_p$, $W_i$ and $W_d$ as fuzzy weights to the proportional ($K_p$), integral ($K_i$) and derivative ($K_d$) gains. The process used to adjust these weights is illustrated in Figure 6-4.
Figure 6-4 Flowchart of adaptive mechanism process
Inspection of Figure 6-4 shows that the fuzzy weights are “learned” in sequence, with $W_p$ first for 15 times, followed by $W_i$ for 10 times and finally $W_d$ for 10 times. It makes sense to make $W_p$ learn first, given the importance of the proportional action for a PID controller. However, the next step is not as obvious. So both ways were tried: i) learn $W_i$ then $W_d$ and ii) learn $W_d$ then $W_i$. It was found that neither way was significantly better than the other.

The basic update equation for the weights is:

$$W(i) = W(i-1) + \Delta W(i)$$  \hspace{1cm} (6-1)

Any change in the sign of $W$ is given by the rules of Table 6-3, which in turn are embedded in the following update equation:

$$\Delta W(i) = -\frac{\text{sign}(\Delta \text{AVGE}) \left(\frac{|\Delta \text{AVGE}|^\zeta}{\text{Max}}\right) \Delta W(i-1)}{\Delta W(i-1)}$$  \hspace{1cm} (6-2)

where $\text{Max}$ is the estimated maximum of $|\Delta \text{AVGE}|^\zeta$ and is in place to ensure stability and guarantee that $W_p(i) < 1$. Note the value of $\text{Max}$ increases with increase in input frequency. $\zeta$ is a damping factor that determines how fast one gets to the global optimum and it varies between 0 and 1. Picking low values of $\zeta$, i.e. high damping factor, may result in very slow convergence or even finding a local optima. Note that Equation 6-2 is based roughly on the method of gradient descent. Adapted initial values were $W_p=W_d=W_i=0$ and $W_p=W_i=1$ and $W_d=0.1$. A counter is used to track the number of times the update is applied.

The “learn” procedure for the weights is thus as follows:

1) Proportional weight is first, initially $W_p=0$ and $W_p=1$. Then $W_p$ is updated by Equations 6-1 and 6-2. The update process is cycled 15 times.

2) Integral weight is second, initially $W_i=0$ and $W_i=1$. As with $W_p$, Equations 6-1 and 6-2 are applied to $W_i$. The update process is cycled 10 times. In this step, the integral of the error should decrease noticeably.
### Table 6-3 Sign rules for weight update $\Delta w (i)$

<table>
<thead>
<tr>
<th>Sign of $\Delta$AVGE</th>
<th>positive $\Delta W_p$ or $\Delta W_i$ or $\Delta W_d$ (i-1)</th>
<th>negative $\Delta W_p$ or $\Delta W_i$ or $\Delta W_d$(i-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive $\Delta$AVGE</td>
<td>$\Delta W(i)$ is negative</td>
<td>$\Delta W(i)$ is positive</td>
</tr>
<tr>
<td>negative $\Delta$AVGE</td>
<td>$\Delta W(i)$ is positive</td>
<td>$\Delta W(i)$ is negative</td>
</tr>
</tbody>
</table>

3) Derivative weight is third, initially $W_d = 0$ and $W_d = 0.1$. As with $W_p$, Equations 6-1 and 6-2 are applied to $W_d$. The update process is cycled 10 times. This step reduces the overshoot and can significantly reduce AVGE.

The period of a cycle is the period of the tracking reference signal. Thus, for a 0.5 $Hz$ reference signal, the 15 cycles for $W_p$ takes 30 $s$. The above three steps can then be repeated.

For a repeat, the algorithm starts from the beginning, by using the values of the $W_p$, $W_i$, $W_d$ and $W_p$, $W_i$, $W_d$ from the last repeat. The number of repetitions is called $rep$. As the value of $rep$ is increased, the degree of change in the weights is decreased. The process can be reset in the middle of a test.

The program has the capability of being reset and it also has a learning enable feature. Thus, the learning process can be restarted whenever necessary as well as being stopped altogether. When satisfactory performance has been obtained and the cylinder will continue working with the fixed gains that it already has. This capability is demonstrated in Figure 6-4 with the stop learning decision box.

Figure 6-5 and 6-6 show the MFAPID response, including plots of the final fuzzy weights, and time varying fuzzy gains for 0.1 $Hz$ sinusoidal tracking on the $x$-axis and the $z$-axis respectively. Tracking error, cylinder position, control signal are also plotted. The fuzzy weights plotted in these figures show how far the implemented MFs were from the optimum MFs. One can compare Figure 4-30 with Figure 6-5 and Figure 6-6 with Figure 5-15 to see the improvement in the results caused by adding the adaptive mechanism to FAPID.
Note that the adaptive mechanism has been divided into two phases: learning and adaptation phases. The same program is used for both of these phases. However, the first three reps has been named learning phase, where the gains start to change from zero to find their tuned values. In this phase the decrease in AVGE is more noticeable than in the adaptation phase, which happens after learning. In the adaptation phase the gains have already settled to their tuned values and AVGE is almost constant. This continues until the routine senses some changes in AVGE, i.e. minimization function. Thus, the adaptive mechanism comes into action and tries to adapt the gains to either a new operating condition or to reduce the effect of a disturbance.
Figure 6-5 $x$-axis MFAPID response for 0.1 Hz sine wave tracking
Figure 6-6 z-axis MFAPID response for 0.1 Hz sine wave tracking
6.4 Adaptive PID (MPID)

In order to provide a fair comparison with MFAPID, it was decided to apply the same adaptive mechanism to the gains of a PID controller. This adaptive PID controller will be referred to as Modified PID (MPID). The algorithm for MPID is essentially the same as for MFAPID; only replacement of $W_p$, $W_i$, $W_d$ as they appear in Equations 6-1 and 6-2 and Table 6-4, with $K_p$, $K_i$, $K_d$, respectively, is needed. Figure 6-7 shows the block diagram of MPID and illustrates the application of the adaptive mechanism.

![Figure 6-7 Block diagram for implemented MPID controller](image)

Figure 6-8 gives the $z$-axis MPID transient response for 0.5 Hz sine wave tracking. In addition to the tracking error and control signal, the PID gains are plotted. From 0 to 30 s one sees the change in $K_p$, from 32 to 52 s the change in $K_i$ and from 54 to 74 s the change in $K_d$. This is considered as the first cycle and the same thing happens in subsequent cycles. The change in $K_p$ is seen to have the biggest effect on the error.
Figure 6-8 z-axis MPID transient response for 0.5 Hz sine wave tracking, showing gains adapting. Figure 6-9 depicts the MPID response for the x-axis for 0.5 Hz sine wave tracking after 3 cycles of learning (3 reps). One notes the change in $K_d$ value which demonstrates that gains are adapting.
The adaptive nature of FL, embedded in FAPID, helps to avoid an unstable response at high tracking frequencies. However, this was not the case for the MPID and more rigorous tuning of damping factor in the program was required to make sure about stabilization of the method.

Figure 6-9 x-axis MPID for 0.5 Hz sine wave tracking, note $K_d$ is in adaptation phase

6.4.1 Operating Experience with MPID and MFAPID

The adaptive mechanism takes a noticeable time to minimize AVGE and this is problematic as time increases with decrease in tracking frequency.
AVGE or RMSE can be misleading as performance measures as their values are changing during a cycle; i.e. RMSE or AVGE for just a portion of the cycle is not an acceptable performance measure. Thus, these values may show good performance for a portion of cycle while in fact they are showing degradation for the entire cycle.

Poor initialization in the learning phase can lead to a local minima or even poor convergence. It was investigated that for the start of the process the precondition is the fixed value of the minimization function, AVGE. If the cylinder was in motion, the fitness function was changing and this would “fool” the algorithm and consequently incorrect actions were taken by the algorithm. Solution is to take the cylinder to one of its ends and let the fitness functions stabilize to some values before resuming motion.

Good initial guess for the values of the MPID gains or MFAPID weights seems to be very helpful as it decreases the convergence time and prevents the method from failing to converge.

Results and plots for MFAPID and their associated AVGE and RMSE are reported only when the apparatus had stabilized after three repetitions. This was thought to give a more meaningful measure of performance as it was unfair to penalize each method until it had been given an opportunity to stabilize measure. However, PID+ANNC has an advantage over MFAPID is that it is efficient and takes about one cycle to converge.

6.5 Robustness Tests

As was explained in previous chapters, the intent of the adaptive mechanism was to maintain performance in the face of changing operating conditions.
6.5.1 Added Mass and Change in Pressure for z-axis

As one source of disturbance, a change in the payload mass was tested. Since at high frequencies this was not physically possible; tests were conducted at only 0.1 Hz. Figure 6-10 shows the performance of an untuned PI controller ($Kp=2, Ki=2$) on the z-axis where 12 kg mass is added to the end effector at $t=15$ s. Figure 6-11 demonstrates the same disturbance for the untuned PID ($Kp=4, Ki=4$) plus tuned ANNC. Note that in Figure 6-10 no noticeable change in performance is observed, which was unexpected since the controller was only a typical PID, while in Figure 6-11 the error chatters more after 15 s, when mass is added, but good tracking can still be observed.

Since adding weights on the payload as the disturbance did not result in a significant change of the behavior on the z-axis, it was decided that a change in supply pressure would be more effective. At first, the intent was to change the supply pressure online by forcing a step change. Since with the available pressure regulator a step change was not possible, instant closing of the supply line was considered. However, during the tests, it was observed that the reservoir was too big for the z-axis and the effect was minimized. Hence, Supply reservoir is illustrated in Chapter 3 Fig 3-3, acts to modulate the air supply. After that, it was attempted to apply the same procedure to the x-axis, but it was unsuccessful again. The reason this time was the relatively small size of the reservoir with respect to the x-axis. After closing the valve it was a matter of few cycles for the cylinder to stop. One of the disadvantages of the proposed adaptive mechanism is that it is slow. Thus, these few cycles were not long enough for the adaptive mechanism to produce acceptable results and demonstrate its degree of robustness. Thus, a sudden change in supply pressure was not possible. Instead, two set of experiments were conducted independently with 280 and 500 kPa supply pressures.

Figures 6-12 and 6-13 illustrate the response of MFAPID for 0.5 Hz sine wave tracking under 500 and 280 kPa supply pressure, respectively. One can see that in Figure 6-12, $Wi$ is in adaptation phase while in Figure 6-13 $Wd$ is in the adaptation phase to decrease AVGE. It is important to point out that the controller needs a large control signal to generate the same velocity for $Ps=280$ kPa relative to 500 kPa, which can be verified by comparing Figures 6-11 and 6-14. This observation is more apparent on the x-axis and will be explained in more detail in
Section 6.4.2. Other controllers were also tested under these two supply pressures and the results are reported in summary bargraphs of Section 6.5.

Figure 6-10 $z$-axis untuned PI ($K_p=2, K_i=2$) for 0.1 Hz sine wave tracking, adding 12 Kg mass at $t=15s$ ($P_s=500$ kPa)

Figure 6-11 $z$-axis untuned PI ($K_p=2, K_i=2$) + tuned ANNC ($K_v=K_z=\lambda=4.5$) for 0.1 Hz sine wave tracking, adding 12 Kg mass at $t=15s$ ($P_s=500$ kPa)
Figure 6-12 $z$-axis MFAPID response for 0.5 Hz sine wave tracking ($P_s=500$ kPa)
6.5.2 Change in Pressure for x-axis

The degree of robustness was also tested on the x-axis; but this time just for 0.5 Hz sine wave tracking where changes in performance are more noticeable. Table 6-4 conveys the results for PID, PID+ANNC, and MFAPID for 500 kPa and 280 kPa supply pressures. MFAPID outperforms the other controllers for both supply pressures. By decreasing the supply pressure the performance of all the controllers is seen to degrade. This observation verifies the pneumatic
rule of thumb, which states that better results will be obtained with higher supply pressure, where air is less compressible. The poor performance with 280 kPa supply pressure can be attributed to lack of air flow for driving the cylinder at the high velocity required for 0.5 Hz tracking. This also decreases the effectiveness of the adaptive methods. It is mostly due to their demands for a high flow rate that exceeds capacity of the valve, to compensate for the lower supply pressure. It is unattainable in practice owing to the valve’s saturation limit. These can be verified by comparing the PID+ANNC control signals of Chapter 4 (Figure 4-15) and Figure 6-14 which were tested under 500 and 280 kPa, respectively. Figure 6-14 shows that control signal reaches the saturation limit and the performance degrades; producing high-amplitude error. Same conclusion can be drawn by comparing Figure 6-9, 500 kPa, and Figure 6-15, 280 kPa.

Table 6-4 x-axis comparison of 0.5 Hz sine wave tracking results for two supply pressures

<table>
<thead>
<tr>
<th>Supply Pressure</th>
<th>Controller</th>
<th>RMSE (mm)</th>
<th>AVGE (mm)</th>
<th>ΔRSME %</th>
<th>ΔAVGE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 kPa</td>
<td>PID</td>
<td>131.5</td>
<td>116</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC</td>
<td>51</td>
<td>45</td>
<td>-61</td>
<td>-61</td>
</tr>
<tr>
<td></td>
<td>MFAPID</td>
<td>44</td>
<td>35.3</td>
<td>-66.5</td>
<td>-69.5</td>
</tr>
<tr>
<td>280 kPa</td>
<td>PID</td>
<td>172</td>
<td>159</td>
<td>ref</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td>PID+ANNC</td>
<td>129</td>
<td>117</td>
<td>-25</td>
<td>-27</td>
</tr>
<tr>
<td></td>
<td>MFAPID</td>
<td>115</td>
<td>103</td>
<td>-33</td>
<td>-35</td>
</tr>
</tbody>
</table>
Figure 6-14 x-axis PID+ANNC response for 0.5 Hz sine wave tracking ($P_s=280$ kPa)
Figure 6-15 $x$-axis Adaptive PID response for 0.5 Hz sine wave tracking, note $K_p$ in learning, ($P_s=280$ kPa)
6.6 Performance Comparisons

The ability of five controllers to track a sine wave reference signal at different frequencies and different operating pressures was tested experimentally. The five controllers were: 1) manually tuned fixed gain PID, 2) MPID, 3) PID+ANNC, 4) FAPID and 5) MFAPID. Sampling time was 5 ms. The amplitude of the sine wave was 40 mm for the z-axis and 250 mm for the x-axis. The controllers were all equipped with PID anti-windup. The manually tuned PID gains were $K_p=4$, $K_i=4$ and $K_d=0.6$ for the z-axis while they were $K_p=2.25$, $K_i=9$, and $K_d=0.6$ for the x-axis. The z-axis AVGE for the controllers are summarized in Figures 6-16 and 6-17, for two supply pressures (280 kPa and 500 kPa) and three frequencies (0.1, 0.2 and 0.5 Hz). Figure 6-18 provides AVGE numerical values of different controllers, for the x-axis, in three frequencies.
Figure 6-16 z-axis bargraph comparison of sine wave tracking results ($P_s=280$ kPa)

Figure 6-17 z-axis bargraph comparison of sine wave tracking results ($P_s=500$ kPa)
Figure 6-18 *x-axis* bargraph comparison of sine wave tracking results ($P_s = 500 \text{ kPa}$)

Inspection of Figure 6-16 shows that the 280 kPa results were truly promising for both MFAPID and MPID (79% and 67% drop in AVGE for 0.1 Hz tracking, respectively). When the pressure was increased to 500 kPa, Figure 6-17, the MFAPID controller was able to increase its advantage still further (87% drop in AVGE for 0.1 Hz tracking). These figures confirm that MFAPID has the lowest AVGE, independent of supply pressure and tracking frequency or type of the cylinder.

The important points conveyed in the bargraphs are as follows:

- MFAPID does a better job under all tested operating conditions, for AVGE and RMSE.
- Same trend was witnessed with RMSE for all controllers (i.e. the ranking of the controllers in different operating conditions)
Table 6-5 Comparison of best results of different studies for sine wave tracking

<table>
<thead>
<tr>
<th>Type of cylinder</th>
<th>Researcher and type of controller</th>
<th>Frequency (Hz)</th>
<th>Ratio of PM to amplitude (%)</th>
<th>Performance measure (PM)</th>
<th>Percentage improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rodless</td>
<td>This Thesis MFAPID</td>
<td>0.1</td>
<td>1.7</td>
<td>AVGE</td>
<td>70% over PID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Choi et al (1998) PID+NN feedback linearization</td>
<td>0.1</td>
<td>1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td>1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ning and Bone (2005) SMC</td>
<td>0.2</td>
<td>0.5</td>
<td>RMSE</td>
<td>75% over PID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rodded</td>
<td>This Thesis MFAPID</td>
<td>0.2</td>
<td>1.8</td>
<td></td>
<td>59% over PVA+FF+DZC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>6.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chillari et al (2001) Fuzzy+ P</td>
<td>0.2</td>
<td>0.5</td>
<td></td>
<td>75% over PID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>1</td>
<td></td>
<td>60% over PID</td>
</tr>
</tbody>
</table>

Table 6-5 summarizes the performance measures seen in this thesis and compares them to those of other researchers. Both AVGE and RMSE are used as performance measures (PMs). The ratio of the PM to the amplitude of the sine wave being tracked is used in the table in order to try and normalize the results and provide a fairer comparison, given that the researchers used different types of apparatus and different pneumatic cylinders. As can be seen in the table, other researchers obtained better or lower PM ratios. Ning and Bone obtained 0.5% for their rodless cylinder, compared to 1.7% in this study. Chillari et al obtained 1.0% for their rodded cylinder, compared to 1.8% in this study. The reason behind the lower performance (and higher ratios) is thought to be due to: 1) saturation of the control signal at 0.2 and 0.5 Hz (for rodless) and 0.5 Hz (for rodded), 2) high payload (7.3 kg) for the rodless relative to others, 5.8 Kg for Ning and Bone and 2.7 kg for Choi et al, and 3) large amplitude (250 mm) for the rodless relative to others (amp=70 mm). Finally, it is believed that the friction of the rodless (x-axis) cylinder has increased over time, in part due to wearing of the seals. If the x-axis is replaced with a newer cylinder, it is expected that performance would improve significantly. More importantly, Choi et al and Ning and Bone methods are both model based and higher precision is expected, given the complex time-consuming process of the implementation.
6.7 Summary

In Chapters 4 and 5, experiments with FAPID and PID+ANNC controllers showed improvement in tracking performance over fixed gain PID, between 45 to 52%. But, the results of FAPID for the \textit{x-axis} were not satisfactory. An online adaptive mechanism was developed to reduce tuning effort and improve the ease of FAPID implementation. Subsequently, comparative results were given for five different controllers: 1) manually tuned fixed gain PID, 2) FAPID, 3) PID+ANNC, 4) MPID and 5) MFAPID. To provide a measure of their robustness, experiments were conducted at two different supply pressures and three different tracking frequencies. It was shown that the MFAPID and MPID controllers improved performance over tuned PID, up to 87\% and 56\% for \textit{z-axis} (up to 40\% and 24\% for \textit{x-axis}) 0.1 Hz sine wave tracking, respectively.
Chapter 7

Conclusions and Recommendations

The objective of this thesis was to explore different adaptive intelligent controllers for position control of a pneumatic system. The application was the x-axis and z-axis of a pneumatic gantry robot. Use of non-model based algorithms to compensate the nonlinearities and uncertainties of the pneumatic system was the underlying strategy for all the controllers. Accordingly, the following controllers were tested:

1. PID+Adaptive Neural Network Compensator (ANNC)
2. Fuzzy Logic (FL) controller
3. ANNonly controller
4. Fuzzy Adaptive PID (FAPID)
5. Adaptive PID (MPID)
6. Modified FAPID (MFAPID)

Performance of these controllers was reported quantitatively and compared with the performance of a conventional PID controller.

7.1 Conclusions

Chapter 2 presented a literature review on six subjects: 1) comparison of different types of actuators, 2) model based vs. non-model based control, 3) Adaptive Neural Network control (ANN) and compensation, 4) Fuzzy Logic control and Fuzzy Adaptive PID, 5) adaptive tuning of ANN and FL control, and 6) comparison of pneumatic position controllers. The following can be presented as the main observations from Chapter 2:

- Among the nonlinearities in a pneumatic system, friction can have a significant effect on tracking performance, especially in applications that use rodless cylinders which have higher Coulomb friction than rodded cylinders.
• Little research has been done investigating the use of NNs as direct controllers. Most of the applications of NNs are as indirect controllers or compensators.

• Fuzzy Logic control is another potential alternative as a non-model based controller for nonlinearities of pneumatic systems.

• PID control with fuzzy gains, known as fuzzy PID, has shown noticeable improvement compared to PID with constant gains. However, there is no general rule for assigning the MFs to PID gains. Generating proper fuzzy rules is can be a time-consuming process.

• Online adaptive tuning of the MFs and fuzzy tables can reduce the energy and time required to design the fuzzy controller.

The key observation from the literature review was that a NN and FL control appear to have the potential to improve the performance of a pneumatic system either as a controller or a compensator.

In Chapter 4, the thesis examined the application of different techniques for the position control of the rodless $x$-axis cylinder of the gantry robot. Experimental results confirmed that performance could be noticeably improved by reconfiguring the hardware and retuning a PID+ANNC controller, as compared to the results reported by Taghizadeh (2010). In order to evaluate the performance as applied to the $x$-axis, five different controllers were tested and their performance was compared: 1) PID, 2) Fuzzy, 3) PID+ANNC, 4) ANNonly, and 5) FAPID. Highlights from Chapter 4 can be stated as:

• Z/N PID tuning method did not provide satisfactory results for sinusoidal tracking.

• Plots of the ANNC weights illustrated rapid convergence to their equilibrium values. The degree of PID+ANNC improvement in AVGE with respect to tuned PID varied from 18% to 61%.

• Adding more inputs to ANN did not improve the performance

• In sine wave tracking, PID+ANNC performance was better than FAPID. According to Chapter 6, the reason may be improper tuning of the MFs.
• All adaptive methods required fairly significant effort to setup and tune.
• The tuning procedure for the fuzzy controller was judged to be more intuitive in nature and hence, more practical than that for ANNC.

The best achieved AVGE, obtained by PID+ANNC, for 0.1 Hz sine wave tracking was equal to 5.3 mm for the x-axis, which corresponds to 2.1 % of the sine wave amplitude.

In Chapter 5, the position controllers that were tested on the x-axis were modified for application to the rodded z-axis cylinder of the gantry robot. Since, the fuzzy controller results on the x-axis were not satisfactory this method was not tested on the z-axis. After re-tuning the controllers, step and sine wave tracking for different frequencies were tested. Highlights from Chapter 5 can be stated as:

• Increasing D, the non-standard term added to $u_{NN}$, gave better performance for PID+ANNC on z-axis
• ANNonly control responses showed improvement in all frequencies compared to tuned PID; this was not the case for the x-axis.
• In sine wave tracking, PID+ANNC performance was better than FAPID. According to Chapter 6, the reason may be improper tuning of the MFs.
• Due to the nature of FL, the adoption of the x-axis MFs by the z-axis was easily accomplished by simply downsizing the MFs from Chapter 4.

One of the important objectives of this thesis was concluded in this chapter. It was shown that untuned PID+ANNC resulted in better performance than tuned PID. In other words, adding an adaptive extension to a PID controller can compensate for untuned PID gains and can improve the results to a better degree than with tuned PID. A major contribution of Chapter 4 and 5 was the demonstration that a novel fuzzy rule set that is reduced in size with rules modified from those used in previous studies (Manjunathm and Janaki, 2011), can improve performance and reduce the computation overhead of the control algorithm. Best achieved AVGE, obtained by PID+ANNC, for 0.1 Hz sine wave tracking was equal to 1.1 mm for the z-axis, which corresponds to 2.75 % of the sine wave amplitude.
Finally, in Chapter 6, an additional adaptive tuning mechanism was incorporated in the tuning procedure of FAPID and PID. This step was necessary because the original premise for an adaptive method was that it would be self-tuning due to its adaptive nature. This was judged to be misleading because of the extensive setup tuning that was required in Chapter 4 and 5. Highlights from Chapter 6 can be stated as:

- The adaptive routine takes a significant time to optimize and this problem increases with a decrease in tracking frequency. However, good initial guesses for the gains or weights are helpful in decreasing the convergence time.

- The relatively poor performance of the controllers on the $x$-axis for high frequencies was likely due to control signal saturation limit.

- MFAPID had the best performance under all tested operating conditions.

The best reported value of AVGE on the $z$-axis was equal to $0.28 \text{ mm}$ for $0.1 \text{ Hz}$ sine wave tracking which was obtained MFAPID. This value as a ratio of the sine wave amplitude, $40 \text{ mm}$, was $0.70\%$. This value is comparable to what is claimed in Ning and Bone (2005), Choi et al (1998), and Gao and Feng (2005). The best reported value of AVGE on the $x$-axis was equal to $4.4 \text{ mm}$ for $0.1 \text{ Hz}$ sine wave tracking which was also obtained by MFAPID. This value as a ratio of the sine wave amplitude $250 \text{ mm}$, was $1.76\%$. This degree of precision is comparable to what is claimed in Chillari et al (2001).

In summary, the objective of this thesis was to explore different adaptive intelligent controllers for position control of a pneumatic system. In order to reduce the tuning effort, a second adaptive mechanism was added to FAPID, to adjust output weights in a continuous fashion. Results with MPID and MFAPID showed further improvement in tracking performance over PID by $87\%$ for the rodded and $70\%$ for the rodless cylinder (in addition to being easier to tune). To provide a measure of robustness, experiments were conducted at two different supply pressures and three different tracking frequencies. MFAPID showed the best results in this thesis and is considered novel for two reasons: 1) fuzzy rule set is reduced in size relative previously published work and 2) addition of an adaptive mechanism for output weights is new.
7.2 Recommendations

The following are recommendations for further work in this area:

- Trying a different NN structure with two hidden layers. ANNC and ANNonly were based on Modified Back Propagation Network (MBPN), which has only one hidden layer.

- In the literature of pneumatic servosystems Sliding Mode Controllers (SMC) has shown a great performance for position control. Consideration should be given to evaluating SMC on the gantry robot.

- Addition of the adaptive mechanism to tune ANNonly and PID+ANNC could be considered as the next phase to reduce the tuning efforts for these controllers.

- The application of a neural network was not examined for the tuning of FAPID and PID. It might be worthwhile to optimize the gains using a neural network. The inputs of this neural network could be Ki (or Wi), Kp (or Wp), Kd (or Wd), the amplitude and frequency of the desired input, and the operating supply pressure. The output could be the AVGE or RMSE of one complete sine wave cycle. The neural network is then programmed to go under a minimization process of AVGE to find the best values of PID gains (or FAPID weights) and decrease the AVGE value for any given operating conditions. This approach would be similar to that taken by Anh and Nam (2011).

- In this study both AVGE and RMSE were used as performance measures. During the experiments it was observed that they showed a similar trend in the results. Thus, it is recommended to record only AVGE, as a more direct measure of performance than RMSE for future experiments. In addition, definition of a new performance measure that is sensitive to oscillation in the position signal and chatter in the control signal is recommended.
References


Appendix A

z-axis HRTextron Valve Tests

This appendix evaluates the performance of a HRTextron valve on the z-axis original cylinder. The appendix demonstrates quantitatively that HRTextron performance is similar to Festo valve and the mismatch problem of valve and cylinder cannot be solved by using the HRTextron valve.

As was explained in Chapter 5, there was a mismatch between the original z-axis cylinder and the Festo valve. Since there was a HRTextron (27A10F) available in the lab, a set of step tracking tests were conducted to assess the applicability of the HRTextron valve. The specifications of the valve are provided at the end of this appendix. According to the specifications, the valve has 4 scfm flow rate at 100 psi; i.e. 113 l/min at 689 kPa. Since the valve nominal flow rate is higher than the installed Festo valve (100 l/min) no improvement in results was expected. However, by testing the HRTextron valve, an attempt was made to make sure the abovementioned problem is solely due to a size mismatch and not a malfunctioning of the Festo valve.

Figure A-1 and A-2 show 2.5 Hz square wave open loop test conducted on z-axis with HRTextron and Festo valve under 350 kPa supply pressure, respectively. A close look at the figures illustrates that the characteristics of the valves are similar and in particular the velocity reaches 1 m/s for both tests. In addition to open loop experiments, a set of step tracking tests were conducted with two supply pressures. Figures A-3, A-4, A-5 plot the performance of P-only, PI, PID controllers under 500 kPa supply pressure. One can observe the low gains of the controllers in the captions and the similarity of P and PI performance.
Figure A-1 Open loop $z$-axis 2.5 Hz square wave test with HRTextron valve

Figure A-2 Open loop $z$-axis 2.5 Hz square wave test with Festo valve
Figure A-3 P-only response for step tracking with HRTextron valve ($K_p=0.6$)

Figure A-4 P-only response for step tracking with HRTextron valve ($K_p=0.6, K_i=0.2$)
Table A-1 provides the results of P-only, PI, PID controllers for step tracking. Performances are reported for both valves under two supply pressures. The objective was to find the best result for each controller independently. Thus, the PID gains of controllers are different for each supply pressure and each valve.

The only deduction that can be made out of the table is the improvement in results with HRTextron valve for $P_s=500 \text{ kPa}$. However, as can be observed in the figures the control signal is covering a short range and any increase would lead to instability of the system.
**Table A-1** Comparison summary of step tracking performance under two supply pressure for HRTextron and Festo valve

<table>
<thead>
<tr>
<th>Supply pressure</th>
<th>Controller</th>
<th>RMSE (mm) Festo</th>
<th>RMSE (mm) HRTextron</th>
<th>AVGE (mm) Festo</th>
<th>AVGE (mm) HRTextron</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 kPa</td>
<td>P-only</td>
<td>9.39</td>
<td>9.4</td>
<td>5.23</td>
<td>4.45</td>
</tr>
<tr>
<td></td>
<td>PI</td>
<td>10</td>
<td>9.54</td>
<td>6</td>
<td>4.97</td>
</tr>
<tr>
<td></td>
<td>PID</td>
<td>10</td>
<td>8.62</td>
<td>7.51</td>
<td>5.28</td>
</tr>
<tr>
<td>350 kPa</td>
<td>P-only</td>
<td>10</td>
<td>9.56</td>
<td>5.97</td>
<td>5.33</td>
</tr>
<tr>
<td></td>
<td>PI</td>
<td>10.5</td>
<td>10</td>
<td>5.72</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>PID</td>
<td>7.54</td>
<td>8.15</td>
<td>4.15</td>
<td>4.75</td>
</tr>
</tbody>
</table>

By checking the open loop and closed loop results it can be verified that the HRTextron valve is also oversized for the z-axis cylinder. Thus, as was explained in Chapter 5, the original cylinder was replaced by a new smaller cylinder.
## Technical Specifications of Bimba Cylinder

<table>
<thead>
<tr>
<th>Model</th>
<th>27A10</th>
<th>27A50</th>
<th>27C00</th>
<th>27C20</th>
<th>27E50</th>
<th>27G50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Pressure (PSI)</td>
<td>0-300</td>
<td>0-5000</td>
<td>0-300</td>
<td>0-3000</td>
<td>0-5000</td>
<td>0-5000</td>
</tr>
<tr>
<td>Rated Flow (± 10%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic GPM @ 1000 PSI</td>
<td>0.18</td>
<td>5.0</td>
<td>10.0</td>
<td>20.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>8.0 (4)</td>
<td>15.0</td>
<td>30.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>10.5 (7)</td>
<td>40.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
<td>60.0 (8)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pneumatic SCFM @ 100 PSI</td>
<td>0.45</td>
<td>11</td>
<td>23</td>
<td>46</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>18</td>
<td>34</td>
<td>68</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>24</td>
<td>92</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td></td>
<td>136</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal Leakage (Maximum)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPM @ 1000 PSI</td>
<td>0.02 (1)</td>
<td>.08</td>
<td>0.2</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCFM @ 100 PSI</td>
<td>0.26 (5)</td>
<td>0.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chip Shear (lbs.)</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>50</td>
<td>83</td>
</tr>
<tr>
<td>Weight (lbs.)</td>
<td>0.8</td>
<td>0.8</td>
<td>1.4</td>
<td>1.4</td>
<td>5.1</td>
<td>10.7</td>
</tr>
<tr>
<td>Maximum Steady State Current (amps.)</td>
<td>0.10</td>
<td>0.45 (2)</td>
<td>0.15</td>
<td>0.90</td>
<td>0.60</td>
<td>0.90</td>
</tr>
<tr>
<td>Electrical Interface</td>
<td>4 wires – 2 Power and 2 Command</td>
<td>CF3102C-14S-2P (6) Connector</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

1) 0.30 GPM for 3.5 GPM version.
2) 1.0 A for 3.5 GPM version.
3) Higher temperature operation available. Consult Womack sales office.
4) Supply pressure 2000 PSI maximum.
5) 0.39 SCFM @ 100 PSI for 8 SCFM version.
6) Mates with MS3106-14S-2S.
7) Supply pressure 1000 PSI maximum.
8) Supply pressure 3000 PSI maximum.

---

Figure A-6 Technical specifications of Bimba cylinder