

**MODEL DEVELOPMENT AND CONTROLLER DESIGN FOR AN
ELECTROHYDRAULIC SERVO ACTUATOR**

by

Eric J. Swahlstedt

**A thesis submitted to the
Department of Mechanical Engineering
in The Graduate School of
Bradley University in partial fulfillment
of the requirements for the
Degree of Master of Science.**

Peoria, Illinois

2013

UMI Number: 1554022

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 1554022

Published by ProQuest LLC (2014). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code

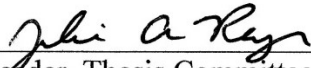


ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

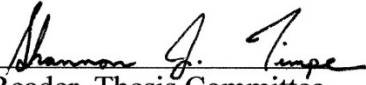
This Thesis for the M.S. Degree
by
Eric J. Swahlstedt
has been approved
November 25, 2013



Chairman, Thesis Committee



Reader, Thesis Committee



Reader, Thesis Committee



Chair, Dept. of Mechanical Engineering

ACKNOWLEDGEMENTS

In order to make this thesis possible I would like to thank some people. First I would like to thank my parents for their patience and guidance. Additionally, I would like to thank them for the many countless revisions that they have suggested for this thesis.

Secondly, I would like to thank Dr. Dean Kim for the opportunity to work on this project. Furthermore, I would like to thank Dr. Kim for the mentoring he has provided throughout the years not only as my graduate advisor but also when I was an undergrad.

I would like to also extend a thank you to Dr. Patrick Opdenbosch of Caterpillar for helping me get started with the modeling and providing many tips and techniques for modeling hydraulic systems in Simulink. Also, I would like to thank him for providing me with some useful material including the hydraulic library that he developed and allowed use of in this thesis. The saturation limits used in this thesis were created by Dr. Patrick Opdenbosch as part of his hydraulic library.

Finally, I would like to thank Dr. Julie Reyer and Dr. Shannon Timpe for taking time out of their schedule to provide me with input on areas of improvement for this thesis.

Eric J. Swahlstedt

Bradley University, Peoria, IL

November 2013

BRADLEY UNIVERISTY

ABSTRACT

MODEL DEVELOPMENT AND CONTROLLER DESIGN FOR AN
ELECTROHYDRAULIC SERVO ACTUATOR

By Eric J. Swahlstedt

Chairman of the Supervisory Committee: Dr. Dean Kim

Dept. of Mechanical Engineering

This thesis presents an analysis of an electrohydraulic servo actuator. A fully nonlinear model of an electrohydraulic actuator has been created in Simulink to fully model the dynamics of the entire system. The entire system was modeled and variables were estimated, due to the inability to accurately measure the actual values. Validation steps including step response, frequency response, and literature verification were performed on the model to verify the accuracy of the model. The model provides the unique advantage of being able to capture the nonlinearity of an electrohydraulic system including the servovalve whose

dynamics usually are ignored or approximated. This system is dependent on the values chosen and exhibits typical nonlinearities, which allows this model to be used for other electrohydraulic actuators after tuning the variables to match the response of a system. The full nonlinearity and modeling approach allows the response of any aspect of the system to be analyzed, increasing the value this model has over other model examined in the literature review. Three controllers were analyzed; a PID with anti-windup, full state feedback with state estimators, and a hybrid PI full state feedback, with regards to the nonlinear system. Due to the large order of magnitudes found in the transfer function of the plant, additional procedures had to be developed to analyze and design controllers with the state space representation. Equations were developed to create state feedback and estimator gains based off of scaled step responses. These equations can be used when numerical issues arise from traditional pole placement techniques. In addition to improving the response time by looking at the settling time, the three controllers that were also analyzed based on the resulting input signal to the plant. Lastly the three controllers are also based on their ability to reject a pulse disturbance added after the controller.

Table of Contents

List of Figures	viii
List of Tables	xi
Chapter 1 Introduction and Literature Review	1
1.1 Introduction.....	1
1.2 Literature Review.....	4
1.2.1 Efficient Controller Design.....	5
1.2.2 Numerical Modeling.....	13
1.3 Problem Statement.....	21
Chapter 2 Modeling of Electrohydraulic Servo Actuator	23
2.1 Chapter Overview	23
2.2 Nomenclature.....	24
2.3 Defining Equations	26
2.3.1 Torque Motor Stage	28
2.3.2 Flapper-Nozzle Stage.....	30
2.3.3 Spool Force Analysis	31
2.3.4 Area Gradient.....	33
2.3.5 Flow Continuity Through Actuator	35
2.3.6 Actuator Force Balance.....	35

2.4	Linearized Transfer Functions	36
Chapter 3	Simulink Modeling.....	38
3.1	Chapter Overview	38
3.2	Simulink Modeling	38
3.3	Simulink Model Justification	48
3.3.1	Time Response Parameter Estimation	49
3.3.2	Frequency Justification	54
3.3.3	Variable Justification	60
3.4	Numerically Solving	65
3.4.1	Stiff Differential Equations.....	65
Chapter 4	Controller Design	67
4.1	Chapter Overview	67
4.2	PID Controller.....	68
4.2.1	Step Response and Input Signal Analysis.....	68
4.2.2	Disturbance Rejection.....	76
4.3	Full State Feedback.....	78
4.3.1	Step Response and Input Signal Analysis.....	78
4.3.2	Disturbance Rejection.....	93
4.4	PI-Full State Feedback.....	95
4.4.1	Step Response and Input Signal Analysis.....	95

4.4.2	Disturbance Rejection.....	102
4.5	Controller Comparison.....	104
Chapter 5	Conclusions and Recommendations.....	111
5.1	Chapter Overview.....	111
5.2	Simulink Modeling.....	112
5.3	Controller Designs.....	113
5.4	Recommendations.....	115
5.4.1	Simulation.....	115
5.4.2	Controllers.....	116
5.5	Conclusion.....	117
Appendix A	Nonlinear and Linear Step Responses.....	119
Appendix B	Scaling factor and full state feedback.....	121
REFERENCES.....		139

List of Figures

Fig. 1.1: Six Valve Energy Recovery System [5].....	7
Fig. 1.2: Modified Tustin Unit Circle [8]	11
Fig. 1.3: Linearized Simulink Model for Pressure Relief Valve [10].....	14
Fig. 1.4: Servo-proportional Valve [12]	16
Fig. 1.5: Simulink Model of Servo-Proportional Valve [12].....	17
Fig. 1.6: Two Spool Flow Control Servovalve [14]	19
Fig. 1.7: S-Function Simulink Model [14].....	20
Fig. 2.1: Electrohydraulic Servo Actuator Schematic	27
Fig. 2.2: Servovalve Operation [16]	28
Fig. 2.3: Area Gradient Comparison Linear vs Nonlinear.....	34
Fig. 3.1: Closed Loop Nonlinear System.....	39
Fig. 3.2: Main Nonlinear Plant Model.....	40
Fig. 3.3: Derivative Filter Utilized in Simulink Model	41
Fig. 3.4: Saturation Limits	43
Fig. 3.5: Conditional Statement Modeling.....	46
Fig. 3.6: Linearized vs Nonlinear Step Response at $2.54e-4$ [m]	50
Fig. 3.7: Step Input Response of $1.27e-2$ [m]	52
Fig. 3.8: Frequency Response Nonlinearities [18].....	53
Fig. 3.9: Magnitude Phase Comparison of po8n Model	55
Fig. 3.10: Magnitude Phase Comparison of Spool vs Current Input.....	56

Fig. 3.11: Magnitude Phase Comparison of Spool Input vs Actuator	58
Fig. 3.12: Jet Flow Angle [20].....	63
Fig. 3.13: Flow Angle as Function of Spool Gap [20].....	64
Fig. 4.1: Reference Input and Output Simulated Response for Actuator Position in Relation to Input Spool Signal for $2.54e-4$ [m] Step Input.....	70
Fig. 4.2: Input and Output Simulated Response for Actuator Position in Relation to Input Spool Signal for $2.54e-3$ [m] Step Input.	71
Fig. 4.3: Block Diagram Representation of PID with Derivative Filter	72
Fig. 4.4: PID Controller with Anti-Windup via Tracking and Derivative Filter ..	73
Fig. 4.5: Disturbance Rejection for PID with Anti-Windup Narrow Pulse	76
Fig. 4.6: Disturbance Rejection for PID with Anti-Windup Wide Pulse.....	77
Fig. 4.7: Block Diagram Representation of Full State Feedback	80
Fig. 4.8: Full State Feedback and State Estimator	84
Fig. 4.9: Full State Feedback with Nonlinear Model.....	88
Fig. 4.10: Step Response Characteristics for Full State Feedback and Estimator at $2.54e-4$ [m].....	91
Fig. 4.11: Step Response Characteristics for Full State Feedback and Estimator at $2.54e-3$ [m].....	92
Fig. 4.12: Disturbance Rejection for Full State Feedback Narrow Pulse	93
Fig. 4.13: Disturbance Rejection for Full State Feedback Wide Pulse.....	94
Fig. 4.14: PI-Full State Feedback with Estimator	95

Fig. 4.15: PI-Full State Feedback for Step of $2.54e-4$ [m]	99
Fig. 4.16: PI-Full State Feedback for Step of $2.54e-3$ [m]	100
Fig. 4.17: Disturbance Rejection for PI-Full State Feedback Narrow Pulse	102
Fig. 4.18: Disturbance Rejection for PI-Full State Feedback Wide Pulse.....	103
Fig. 4.19: Controller Comparison Actuator Response at $2.54e-4$ [m] Step.....	106
Fig. 4.20: Controller Comparison Actuator Response at $2.54e-3$ [m] Step.....	107
Fig. 4.21: Disturbance Rejection Responses Narrow Pulse.....	108
Fig. 4.22: Disturbance Rejection Responses Wide Pulse	109
Fig. A.1: Step Input Response of $2.54e-3$ [m]	119
Fig. A.2: Step Input Response of $1.27e-3$ [m]	120
Fig. B.1: Step Response of $H(s)$	123
Fig. B.2: Bode Plot of $H(s)$	124
Fig. B.3: Step response of $H(s')$	125
Fig. B.4: Bode Plot of $H(s')$	126
Fig. B.5: Step Response of $\rho\delta n(s)$ Transfer Function.....	129
Fig. B.6: Block Diagram Representation of Full State Feedback.....	129
Fig. B.7: Full State Feedback and State Estimator	134
Fig. B.8: Step Response $H(s)$ with Full State Feedback Gain	137
Fig. B.9: Step Response of $H(s')$ with Full State Feedback Gain	138

List of Tables

Table 3.1: Nonlinear-Linear Frequency Comparison	59
Table 3.2: Variable Values	60
Table 4.1: PID Current Comparison	75
Table 4.2: PID Actuator Position Comparison	75
Table 4.3: Full State Feedback Pole Placement.....	87
Table 4.4: Full State Feedback Current Comparison.....	89
Table 4.5: Full State Feedback Actuator Position Comparison.....	90
Table 4.6: PI-Full State Feedback Pole Locations.....	98
Table 4.7: PI-Full State Feedback Current Comparison.....	101
Table 4.8: PI-Full State Feedback Actuator Position Comparison.....	101
Table 4.9: Current Comparison of Controller Designs	104
Table 4.10: Actuator Comparison of Controller Designs	105
Table 4.11: Controller Comparison for Disturbance Rejection	110
Table 5.1: Current Comparison of Controller Designs	117
Table 5.2: Actuator Comparison of Controller Designs	118
Table B.1: Comparison of Transfer Function with Scaling Factor.....	126
Table B.2: Scaling Factor Full State Feedback.....	133
Table B.3: Scaling Factor Full State Estimator	136
Table B.4: Full State Feedback Gain Comparison.....	138

Chapter 1 Introduction and Literature Review

1.1 Introduction

In control systems, typically the primary objective is to improve the response of a certain task. This can be accomplished through better tracking or a quicker response time. In most control systems, having a complete understanding of the system or plant that is being controlled is of the utmost importance. Typically, the equations that define the system are put into either state space form, or as a transfer function. For both cases, the dynamic equations need to be linear or linearized to be put into the forms mentioned above. In addition, the dynamic equations need to accurately define the system.

The controller, typically, is designed and then digitally implemented. If any unexpected behavior occurs during the digital implementation, the system is manually tuned to adjust for the uncertainties in the original system equations developed. To improve the usefulness of the controller designed for the given plant equations, robustness can be introduced into the controller design. This can

be in the form of a H_{∞} controller or through introducing uncertainty into the system equations [1]. In addition to modeling uncertainties, problems can also become evident when the continuous controller is converted into a digital controller in the form of continuous to discrete conversions.

For the modeling and control of an electrohydraulic servo actuator system both, of these are common problems. Most hydraulic systems are highly nonlinear, usually due to the equations that define the system. In addition, the servovalve dynamics are usually modeled with different degrees of complexity. Some authors assume the servovalve to be fast enough, such that, the actual dynamics of the servovalve can be ignored or approximated by a second or third order system [2–4].

This paper will demonstrate the Simulink implementation of the electrohydraulic servo actuator system equations, as well as, controller models. The controllers are designed around improving the efficiency of the step response by looking at the input signal for a plant from a controller, while at the same time attempting to decrease the settling time of the system. The percent overshoot of the controller designs are also investigated. The Simulink model will also make use of the

nonlinear equations and will not use linearized equations or transfer functions as approximations.

Chapter 1 provides a literature review of efficient controller design. The design of controllers and efficiency of hydraulic systems is reviewed. Numerical simulation is also discussed in detail as it pertains to hydraulic systems which are typically, by nature, stiff nonlinear systems.

Chapter 2 discusses the equations used to model the electrohydraulic actuator. The equations for each section of the electrohydraulic actuator are presented with a discussion of the variables used. Improvements to the equations are also presented for use later on in the modeling of the equations.

Chapter 3 introduces the Simulink model produced for this thesis. The Simulink model is presented along with an explanation of different aspects of the model. The nonlinear model is also verified using three methods; frequency analysis, step input response comparison, and variable verification based on literature sources. Finally, a discussion on techniques for solving differential equations especially pertaining to nonlinear and stiff differential equations is explored.

Chapter 4 discusses the different types of controllers designed. Three different controllers were designed; PID with anti-windup, full state feedback, and a hybrid PI-full state feedback. All controller designs are evaluated in the Simulink nonlinear model created. A discussion of the different controller types and their input signal is discussed. Furthermore, equations are developed to aid in the creation of full state and state estimator gains for scaled systems that fall symptom to numerical issues during calculation. The controllers mentioned are also analyzed with regards to pulse disturbances added after the controller.

Chapter 5 provides an overview of what is presented in the thesis. Additionally, the areas for improvement are discussed. Areas for future study are also presented.

1.2 Literature Review

This section will highlight literature sources that focus on the modeling and efficiency analysis of hydraulic systems. While the approach that is being taken

in this paper differs from the research reviewed it is important to have an understanding of other means of efficient performance.

1.2.1 Efficient Controller Design

There are many different routes that can be taken to increase the efficiency of a fluid power system. Different methods include efficient controller designs and possibly extra equipment to allow regeneration. Troxel and Yao [5], looking at the velocity control of a hydraulic cylinder, utilized separate spool valves to regulate the cylinder chamber pressures. Typical hydraulic cylinders have one spool valve which makes regulating each cylinder chamber pressure impossible. By utilizing separate spool valves, regeneration becomes much more possible. Regeneration in this context refers to utilizing fluid from one chamber and using it in another chamber by recycling the fluid. The purpose of the study was to reduce the energy by means of regulating the flow rate and pressure. Both of which cannot be done independently by a typical spool valve configuration. Regeneration could be implemented whenever the pressure is higher in one chamber than the other, with regards to a fluid with a flow rate. This is typical in instances of deceleration or when lowering a heavy load. In the case of a single

rod cylinder, regeneration may also be possible when extending a rod due to the difference between the head and rod cross sectional areas.

Oil flowing through a valve dissipates power, based on the product of the pressure drop and the flow rate. By this relationship, the most efficient flow comes from a fully open valve which would produce a very small pressure drop. Troxel and Yao utilized a multi valve system and controller design to improve the efficiency of a hydraulic cylinder. The six valve configuration can be seen below as Fig. 1.1.

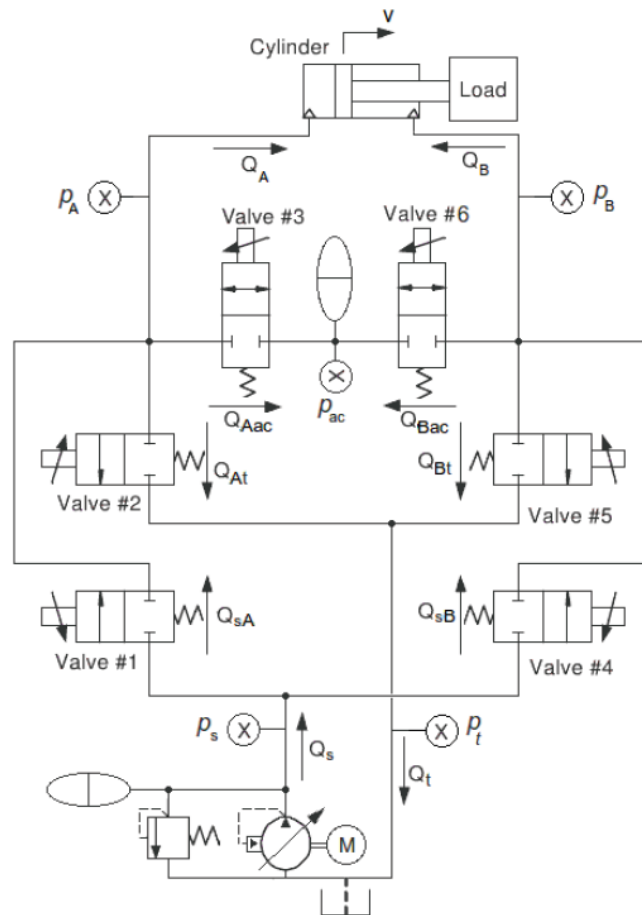


Fig. 1.1: Six Valve Energy Recovery System [5]

This concept of using separate valves has been studied quite a bit by other researchers. With this configuration, regeneration can be utilized. In order to properly utilize this efficient design, a low-level adaptive robust controller and high level logic had to be implemented. Additionally, the design allowed independent control of cylinder pressures.

Almost all systems especially fluid power systems have saturation limits that prevent control systems from operating to their fullest. This is due to the general inability to easily model systems and implement controllers that know about the saturation limits. This can also cause problems with leading to less than ideal efficient designs. One simple way of working around this issue is to implement anti-windup, which is regularly used with integrator anti-windup methods. There can be issues with a system that has more than one saturation limit, such as a spool with known limits that drives the placement of an actuator.

In addition to this, the control scheme can also suffer degradation by sudden lack of change from the plant due to the saturation. Doyle et. al [6] has shown that in single input single output (SISO) systems, Internal Model Control (IMC) for a given stable plant can guarantee closed loop stability even with saturation. Traditional anti-windup works by comparing the signal before and after the saturation point. The signal is then multiplied by a gain block and added back into the loop. This is usually performed and found to be effective. This method is used extensively in industry, especially with traditional PID schemes. However, the traditional anti-windup can cause instability given the right set of conditions.

Other methods have also been developed utilizing minimum energy control. Chen et. al [7] compared different minimum energy methods for a mass spring damper system. Four different methods were compared. The traditional minimum control energy method via Hamiltonian function and optimal control by Riccati equation are compared. Additionally, an optimal control with exact solution and a minimum control energy based on a PID controller utilizing particle swarm optimization approach was also analyzed. From the analysis, the PID method developed from particle swarm optimization yielded the best results in terms of control energy input. All of these methods were validated by numerical simulation. It should be noted that while optimal is generally used to define the best solution, the PID from particle swarm optimization actually yielded the best results. As such it should be noted that other forms than the optimal solution should be developed and tried out on a system.

Additionally, the efficiency of the controller can also be improved during the conversion from a continuous controller to the implementable digital controller. Kim et. al [8] introduced a modified Tustin transformation to improve the conversion from continuous to discrete. Traditionally, the standard Tustin transformation is most commonly used to convert the continuous controller design to the digital version. As known all poles that are stable are correctly transferred

to inside the unit circle of the digital z plane by the Tustin transformation as shown by Eq. (1.1).

$$z = \frac{1 + (\frac{T}{2}s)}{1 - (\frac{T}{2}s)} \quad (1.1)$$

Other methods of transformation include the forward rectangular method, which overestimates the digital pole locations. This method can lead to instability, due to the fact that this method can place stable continuous poles outside of the unit circle, thus, leading to an unstable system that has been digitally implemented. The backward rectangular method places all the poles in the unit circle between $z=[0,1]$. This can lead to poles not being utilized to their fullest, by means of the placement in the continuous design

Kim et. al [8] developed a method that allows the digital pole locations of an already stable system to be altered based on a scaling factor. This modified Tustin transformation has the ability to move the poles that are close to $z=-1$, closer to the origin, i.e. away from the $z=-1$, as shown in Fig. 1.2. It is known that poles near $z=-1$ tend to introduce more oscillation into a system as $z=-1$ is an area of high frequency on the unit circle. This can lead to an increase in extra energy

and, especially for electrohydraulic systems, can cause excess wear on the equipment.

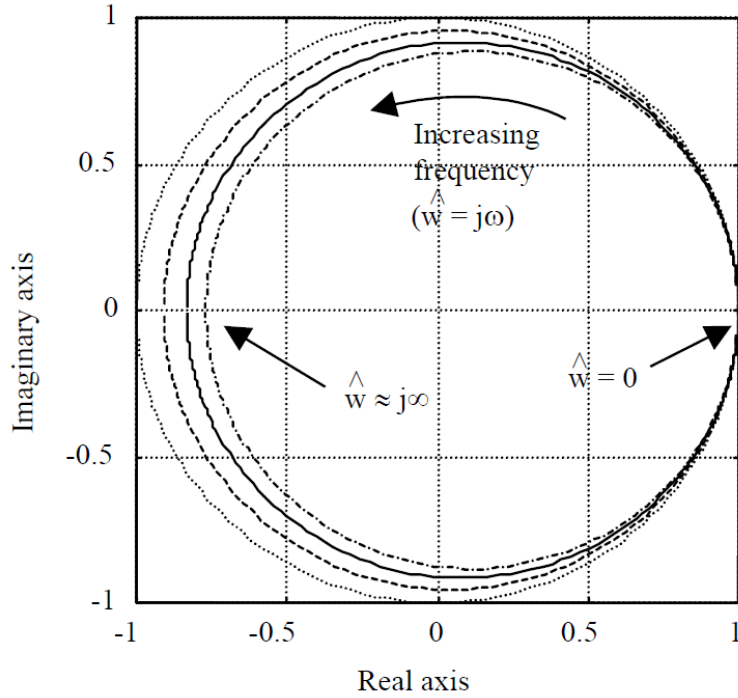


Fig. 1.2: Modified Tustin Unit Circle [8]

The modified Tustin transformation utilizes the \hat{w} domain. To begin, first the plant is converted to the digital z domain utilizing the standard Tustin transformation. Then, the modified Tustin is performed by Eq. (1.2),

$$z = \frac{1 + \left(\frac{T}{2}\hat{w}\right)}{1 - f\left(\frac{T}{2}\hat{w}\right)} \quad (1.2)$$

where T is the time step and f is the scaling factor to be determined. Through trial and error, the f value is varied until an acceptable response is found. Then the standard Tustin transformation is used to convert back into the z domain. Due to the direct replacement of variables there is no approximation and, as such, the mapping is correctly mapped back to the z domain from the \hat{w} domain. This has been suggested as an improvement over simply using the standard Tustin transformation, due to the approximation that is performed from the continuous to digital domains.

By making the scaling factor, f , in Eq. (1.2) greater than unity, this causes the unit circle to be skewed further away from the $z=-1$, while still maintaining the $z=1$ side of the unit circle. This is very useful when designing a controller in the continuous state and then converting to the digital state, only to find that the digital poles are very close to the $z=-1$ side of the unit circle. This provides a quick and simple way to make the digital poles more favorable. Kim et. al [8] also performed an analysis of an electrohydraulic actuator and was able to reduce the energy input signal. This also reduced the system overshoot along with the settling time. All of which is very beneficial to an engineer.

1.2.2 Numerical Modeling

There are many examples where models of fluid power systems are numerically simulated. However, not very many papers go into detail describing how models are implemented, or if nonlinear models are used.

Chatzakos and Papadopoulos [9] created a Simulink model of a similar servovalve used in this analysis. To implement their design in Simulink, linear graph theory was used. The simulated model was then used to judge the ability of controllers to position and force tracking. Chatzakos and Papadopoulos ignored the effects of the torque motor. This was due to the fact that the natural frequency was much larger than the desired closed loop bandwidth. The orifice equations were, however, taken into account. Rather than simply using an nth order transfer function to model specific aspects of the electrohydraulic actuator, separate linearized equations were used. This helps the modeling process by not solely relying on transfer functions that may miss important aspects not developed in the transfer function. At the same time the linear equations reduce the complexity of the model both in terms of computation difficulty and possibly the accuracy of the solution.

Maiti et. al [10] developed a dynamic model based off of a pressure relief valve. The dynamic equations were then linearized and the model was created in Simulink for further analysis. Due to the fact that the system was linearized, allowed for a much easier implementation. Additionally, all values were already known. The complete model utilizing the built-in Simulink blocks can be seen below as Fig. 1.3.

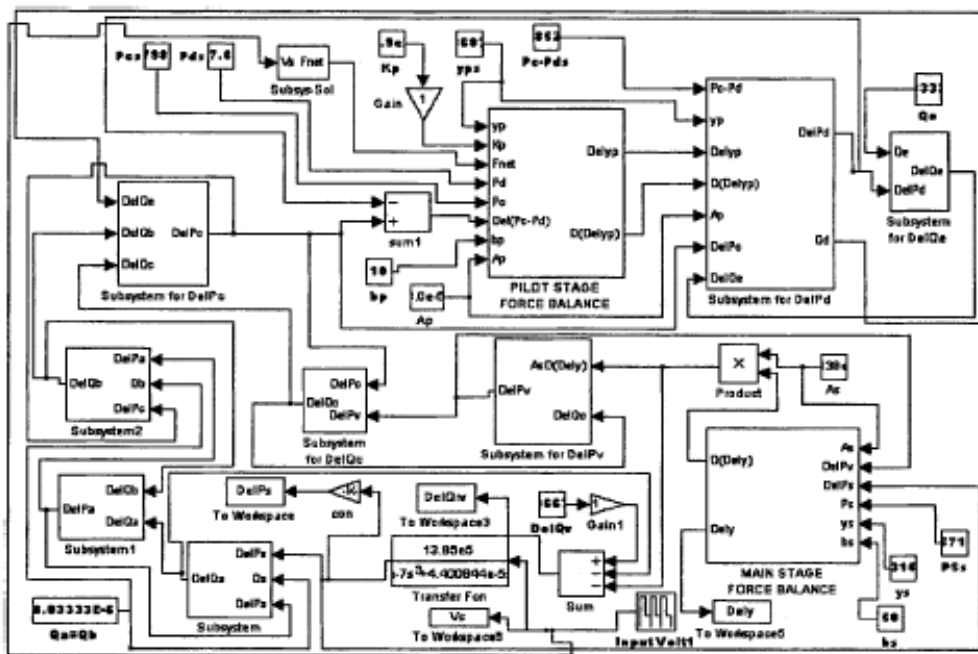


Fig. 1.3: Linearized Simulink Model for Pressure Relief Valve [10]

The Simulink model also consists of subsystems containing other parts of the linearized dynamic equations. Maiti et. al were able to produce accurate results from different parts of the linearized model. Both the experimental and simulated

results matched closely. Validation was performed at different step inputs and the different pressures in the model were verified.

Numerical models have also been created for specific parts of a hydraulic system. Gordic et. al [11] created a Simulink model for only the servovalve and torque motor. The flapper assembly was also analyzed in detail. Gordic et. al was able to create an accurate mathematical model that described the behavior of electrohydraulic servovalve by implementing the modeling in Simulink. Additionally according to the article linearization techniques were not performed on the developed equations used for simulation.

Fang et. al [12] performed a simulation based on a three way servo-proportional valve. The servo-proportional valve used for the simulation is shown below as Fig. 1.4.

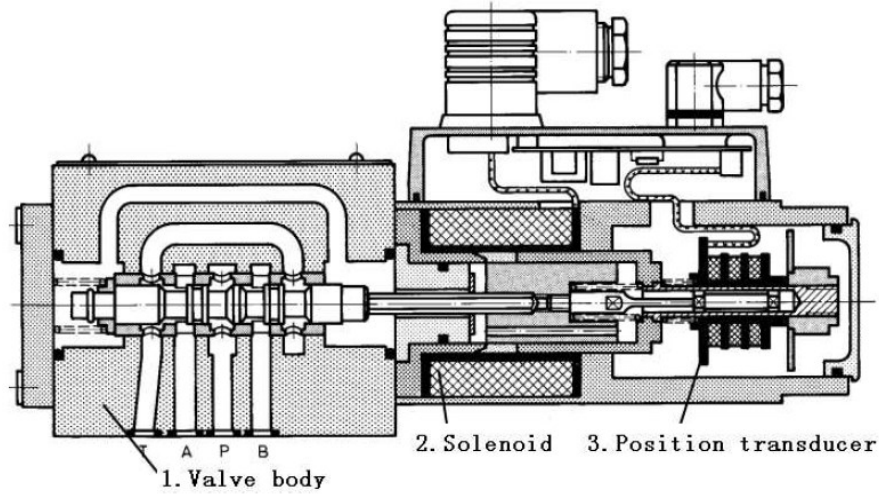


Fig. 1.4: Servo-proportional Valve [12]

The spool valve does not have the traditional setup where the solenoid or torque motor is positioned above the spool, but is rather in line with the spool. This provides a more instantaneous relation because the spool movement and is not regulated by a pressure difference, but rather directly from the solenoid. The spool valve is then solely responsible for regulating the flow into and out of the valve.

Based on experimental data, the parameters were determined and used in a Simulink model as shown in Fig. 1.5

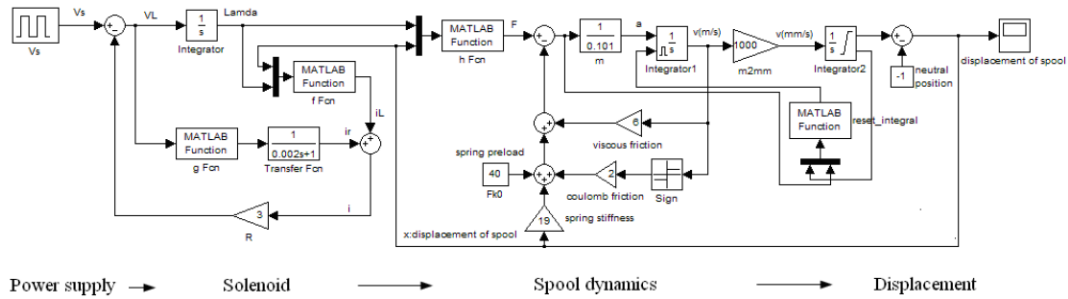


Fig. 1.5: Simulink Model of Servo-Proportional Valve [12]

From the Simulink model, it can be seen that, the model has been simplified with the use of transfer functions and MATLAB Function Blocks. The use of linear transfer functions decreases the complexity of the model however; the use of the MATLAB Function Blocks can lead to an increase in computational time due to having to compile the MATLAB code for the simulation. Additionally the spool is linearized and the solenoid is approximated by a first order system.

As discussed, fluid power systems are typically highly nonlinear systems that are hard to completely model by a transfer function or state space model. Additionally, fluid power systems are traditionally stiff systems in terms of the differential equations. Aman [13] dissertation explains ways to improve upon the simulation of fluid power systems. Aman indicates that in addition to using a stiff ODE solver, if a fixed step rather than a variable step solver is used will cause stability problems. This is especially true when pressures in small volumes, such

as a spool or actuator, are integrated. By using a variable step, the step size will decrease as the volume decreases.

Aman also discussed many other methods for improving the dynamic simulation ability of a fluid power system. There are two main ways of improving numerical solutions of stiff systems. The first method as mentioned above is to develop or use a solver specifically designed to handle stiff systems. The other approach is to use approximations for various stiff parts, such as transfer functions or other approximations. The latter method is shown to be a common approach throughout the literature sources in this section.

Anderson and Li [14] created a Simulink model of a two spool flow control servovalve with a pressure controlled pilot. The main purpose of this model was to model the valves nonlinearity with Simulink. The valve is shown below as Fig. 1.6.

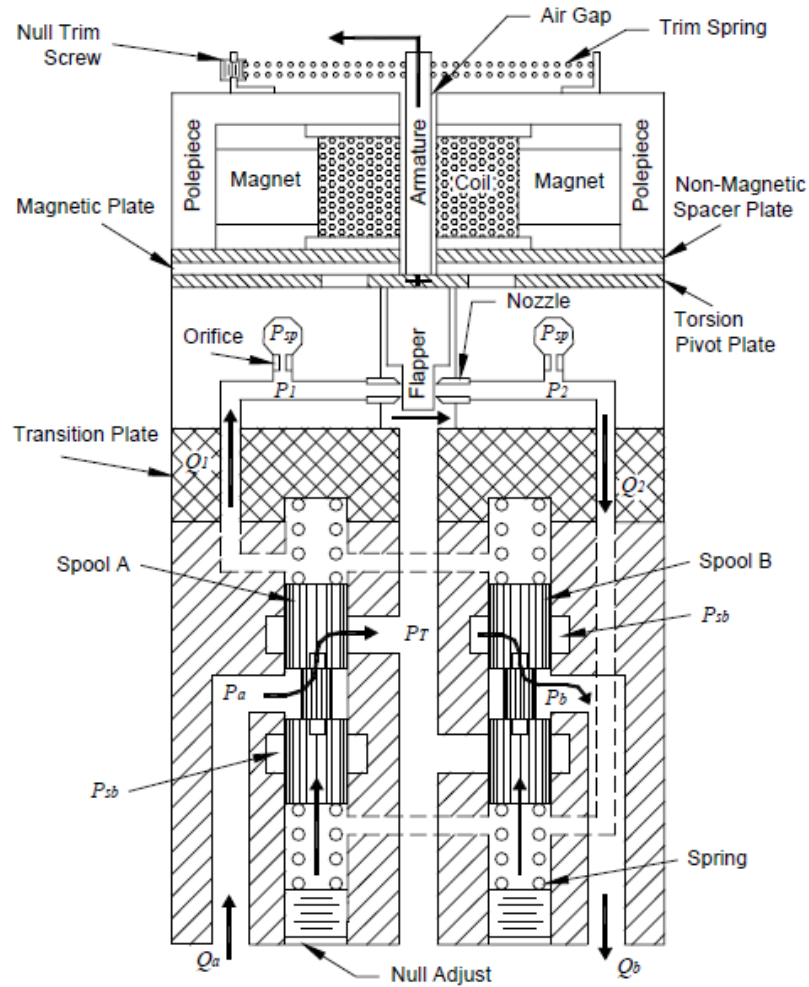


Fig. 1.6: Two Spool Flow Control Servovalve [14]

The equations were then presented for description of the system based on force and flow analysis. After a few simplifications a Simulink diagram was created.

This is shown below as Fig. 1.7.

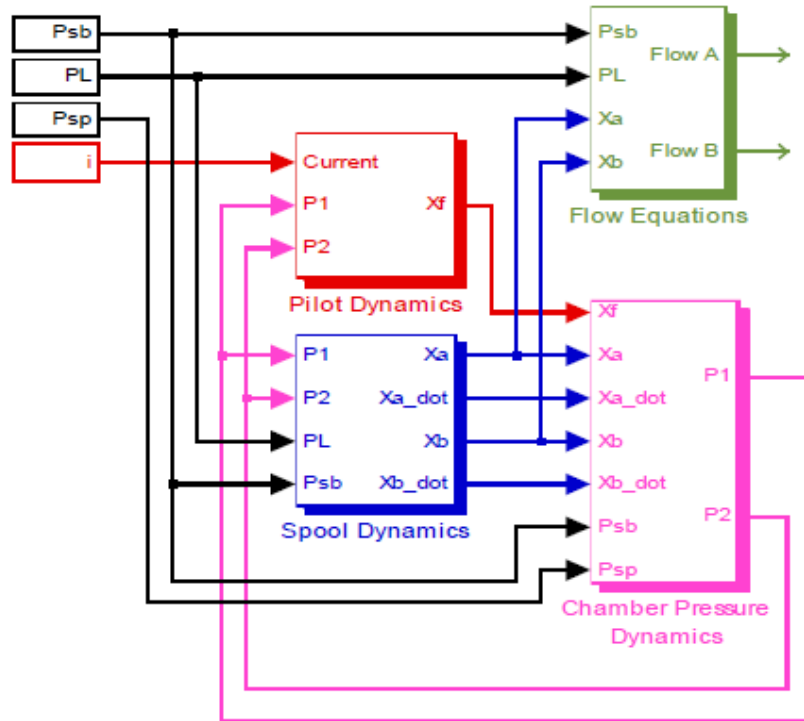


Fig. 1.7: S-Function Simulink Model [14]

The Simulink model utilized S-Functions to simplify the creation of the model. S-Functions were used instead of typical graphical blocks due to the complexity of the system as illustrated by the equations shown by Anderson and Li. While the S-Function does simplify the building of the model, it does reduce the ability to readily look at any value of interest in a model, which can be seen by using the graphical block diagram approach.

To verify the accuracy of the model, the steady state response was evaluated for the flow versus current and pressure against experimental data. The step response was also compared against experimental data. Finally, a swept sinusoidal current was applied to produce a frequency response characterization of the model to compare against experimental data.

All of these sources explain the importance of accurate modeling due to the high nonlinearity associated with hydraulic systems. The ultimate test for any set of equations is experimentally validating them. Additionally the references brought up in these sources indicate the degree of difficulty associated with modeling a set of nonlinear equations to the degree presented in this thesis.

1.3 Problem Statement

The objective of this thesis is to model the nonlinearities of electrohydraulic servo actuator in Simulink. Additionally controllers were designed and implemented in the nonlinear Simulink model to improve the settling time. The percent overshoot was also evaluated. The controller designs were also gauged on their energy effectiveness by looking at the input signal. As a final check the controllers were

also analyzed for disturbance rejection. A pulse disturbance was applied after the controller and the actuator positions were then compared to determine which controller rejected the disturbance the best. A literature review was performed that evaluated different techniques of energy recovery for fluid power systems. Additionally, due to the degree of Simulink modeling performed, different models were analyzed to show the level of detail presented in this model which is not usually shown in literature.

Due to the fact that not all the values that defined the set of equations were available or known parameter estimation had to be performed to determine the values. The actuator step response was compared against the accepted eighth order closed loop transfer function that defined the electrohydraulic servo actuator. The step response was compared at $2.54e-4$ [m], which is considered to be within the linear region of the model. The frequency response of the overall system and the subsystems were compared against their respectable transfer functions. The step response at larger step inputs represented typical nonlinear behavior for hydraulic systems outside the linear range. The final variables were also justified against accepted literature values.

Finally three controller designs were implemented; a PID with anti-windup, full state feedback, and a PI full state feedback. The eighth order transfer function has large values which can lead to numerical issues. This became apparent with the full state feedback approach which required pole placement techniques to be implemented. To remedy this, special equations were developed to determine the feedback gains from feedback gains that were developed using scaled transfer functions. The scaled transfer function reduces the large values and allows calculations of the feedback gains while preserving the response in the scaled time domain without numerical issues.

Chapter 2 Modeling of Electrohydraulic Servo Actuator

2.1 Chapter Overview

This chapter discusses modeling of an electrohydraulic actuator. The chapter is broken down into the different equations used to describe the electrohydraulic actuator along with an explanation of the equations. The primary objective of this section is to explain the equations already developed [4] and how they will be used in the subsequent chapter for modeling in Simulink. Important

modifications are made to make the model exhibit more nonlinearities and make it easier for the solver to solve the stiff differential equations.

2.2 Nomenclature

This section presents the variables, units, and a description of the variables used to explain the system presented below.

<u>Variable</u>	<u>Units</u>	<u>Description</u>
θ_f	[rad]	angle at which fluid jet leaves spool chamber
M_s	[kg]	mass of spool
C_{qo}	[-]	orifice flow coefficient
C_{qn}	[-]	nozzle flow coefficient
C_q	[-]	flow coefficient through spool port into chamber
J	[kg-m ²]	moment of inertia of torque motor
A_o	[m ²]	cross sectional area of orifice
D_n	[m]	diameter of nozzle
K_a	[N-m/rad]	rotational stiffness of flexure tube
A_s	[m ²]	area of spool valve
R	[m]	distance from nozzle to pivot point of flexure tube
B	[m]	distance from nozzle to spool
B_s	[N/(m/s)]	damping coefficient of servovalve system
ρ	[kg/m ³]	density of hydraulic fluid
L_1	[m]	axial length between 'P _s ' port and input port of actuator

L_2	[m]	axial length between 'P _e ' port and input port of actuator
A_1	[m ²]	effective area of double-ended piston
W	[m ² /m]	area gradient
M_t	[kg]	mass of actuator
β_e	[N/m ²]	compressibility of hydraulic oil (Bulk Modulus)
P_s	[N/m ²]	supply pressure equivalent to 2700 [psi]
K	[N-m/m]	net stiffness of cantilever feedback spring connected to flapper
B_v	[N-m/(rad/sec)]	damping coefficient of torque-motor
K_t	[N-m/A]	torque motor gain
X_{fm}	[m]	maximum flapper displacement
V_{ao}	[m ³]	enclosed volume on each side of actuator when $x_a=0$
B_a	[N/(m/s)]	damping coefficient of actuator
K_m	[N-m/rad]	electromagnetic rotational stiffness
F_d	[N]	disturbance force input on actuator
V_{so}	[m ³]	enclosed volume on each side of spool when $x_s=0$
f_r	[N]	flow reaction force
θ	[rad]	angular position of armature
Ω	[rad/s]	angular velocity of armature
p_{s1}	[N/m ²]	pressure on one end of spool
p_{s2}	[N/m ²]	pressure on one end of spool
x_s	[m]	spool position
v_s	[m/s]	spool velocity
p_{a1}	[N/m ²]	pressure on one side of actuator
p_{a2}	[N/m ²]	pressure on one side of actuator
x_a	[m]	actuator position
v_a	[m/s]	actuator velocity
i	[A]	input current to torque-motor

2.3 Defining Equations

The equations that define the system are nonlinear and deserve a closer examination to understand the assumptions made for the correct analysis of the system. Shown below, Fig. 2.1 is the schematic for the basis of the electrohydraulic servo actuator. This model is based around the Moog 760 Direct-Operated Servovalve line. The system consists of a two-stage flow control servovalve with a double-ended actuator. The servovalve is broken down into two stages; the current to angle stage and the angle to spool position. The four way sliding spool valve is in a closed-center position, as is typical for a spool valve used in conjunction with a double acting cylinder.

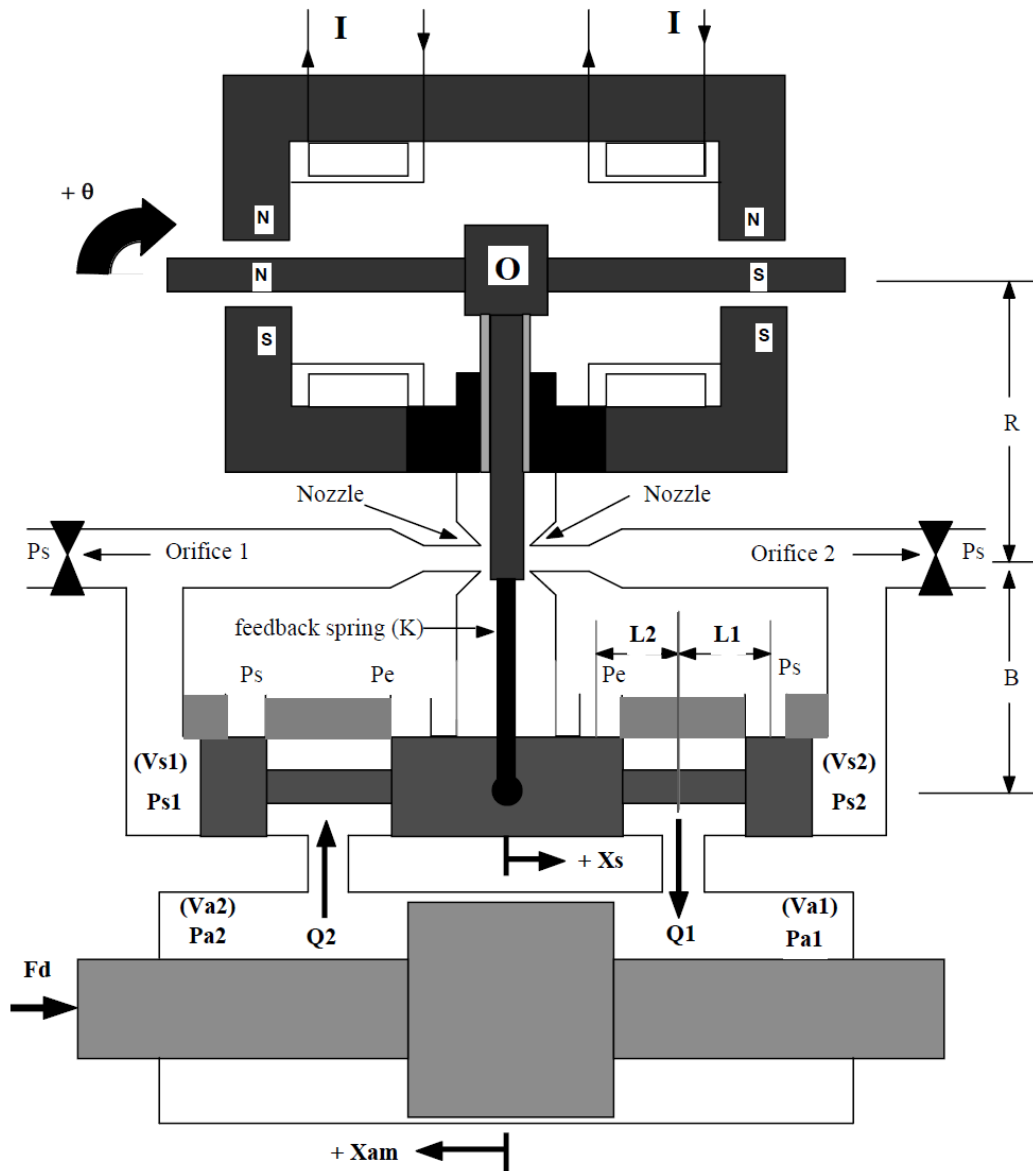


Fig. 2.1: Electrohydraulic Servo Actuator Schematic

2.3.1 Torque Motor Stage

The torque motor stage is the top half of the electro hydraulic actuator as shown in Fig. 2.1. The torque motor stage takes in a current, typically from a voltage supply. The current that is supplied to the armature inside a permanent magnet induces a torque causing rotation of the armature. This in turn causes the flapper to move and allows a pressure difference on either side of the spool valve. This difference in pressure causes a force to be acted on the spool, allowing it to move in one direction or the other as shown in Fig. 2.2 [15].

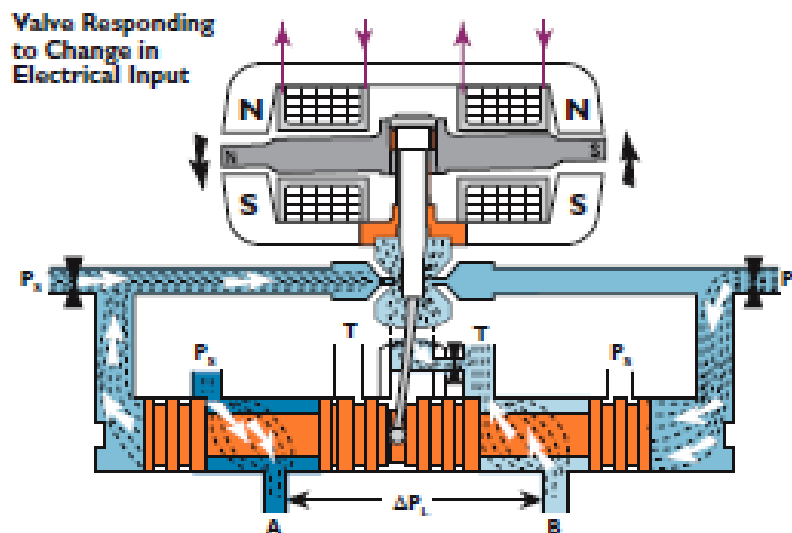


Fig. 2.2: Servovalve Operation [16]

Spool valves usually have a spring to help center the spool. Two configurations are common. The first configuration consists of two springs of equal quality on either end of the spool. The second configuration is a flexure tube that extends from the flapper and acts as a cantilever spring. The latter configuration described will be used throughout the analysis.

The torque motor stage is represented by

$$\begin{aligned} \dot{\theta} &= \Omega \\ \dot{\Omega} &= \frac{1}{J} \left\{ K_t i - (K_a - K_m)\theta - K[x_s + (R + B)\theta] - B_v \Omega \right. \\ &\quad \left. - (p_{s1} - p_{s2}) \frac{\pi D_n^2}{4} R - 4\pi C_{qn}^2 R [(X_{fm} - R\theta)^2 p_{s1} \right. \\ &\quad \left. - (X_{fm} + R\theta)^2 p_{s2}] \right\} \end{aligned} \quad (2.1)$$

from summing the torques about the center, O, of the torque motor in Fig. 2.1. From Eq. (2.1), it can be seen, that the small angle theorem is assumed. Also, the predominant torque comes from the first term, the torque motor gain. The second and third terms are the rotational stiffness. The remaining terms are rotational damping, and the flow and pressure forces exhibited at the nozzle. R and B are the corresponding distances from O, the center, to the nozzle and from the nozzle to the center of the spool respectively.

2.3.2 Flapper-Nozzle Stage

The flapper nozzle stage consists of the reactions at, and around the nozzle. This is an important part of the system, due to the fact that the pressure is regulated at this point and thus affects how the spool will move and ultimately the actuator. As such, the flow forces and position of the flapper needs to be accurately modeled.

The flapper nozzle stage is defined as

$$\dot{p}_{s1} = \frac{\beta_e}{V_{s1}} \left\{ C_{qo} A_o \sqrt{\frac{2(P_s - p_{s1})}{\rho}} - C_{qn} a_{cx} \sqrt{\frac{2(p_{s1} - p_{atm})}{\rho}} - A_s v_s \right\} \quad (2.2)$$

$$\dot{p}_{s2} = \frac{\beta_e}{V_{s2}} \left\{ C_{qo} A_o \sqrt{\frac{2(P_s - P_{s2})}{\rho}} - C_{qn} a_{cy} \sqrt{\frac{2(p_{s2} - p_{atm})}{\rho}} + A_s v_s \right\} \quad (2.3)$$

where:

$$a_{cx} = (X_{fm} - R\theta)\pi D_n \quad (2.4)$$

$$a_{cy} = (X_{fm} + R\theta)\pi D_n \quad (2.5)$$

$$V_{s1} = V_{s0} + A_s x_s \quad (2.6)$$

$$V_{s2} = V_{s0} - A_s x_s \quad (2.7)$$

In Eqs. (2.2) and (2.3) the first two terms are the flows through the orifice and nozzle respectively, where s_1 and s_2 indicate either side of the system. The last terms in Eqs. (2.2) and (2.3) are equivalent to the flow rate exhibited on the spool.

2.3.3 Spool Force Analysis

To include an analysis of the spool, a force balance had to be conducted to properly model the spool dynamics. The spool is allowed to move horizontally in the chamber and has five different forces acting on it.

The force balance on the spool is

$$\begin{aligned} \dot{x}_s &= v_s \\ \dot{v}_s &= \frac{1}{M_s} \left\{ (p_{s1} - p_{s2})A_s - \frac{K[x_s + (R + B)\theta]}{R + B} - B_s v_s - f_r \right. \\ &\quad \left. - [\rho L_2 \dot{q}_1 - \rho L_1 \dot{q}_2] \right\} \end{aligned} \quad (2.8)$$

where:

$x_s > 0$:

$$f_r = 2C_q W x_s \cos(\theta_f) (P_s - p_{a1} + (p_{a2} - p_{atm})) \quad (2.9)$$

$$q_1 = C_q W x_s \sqrt{\frac{2(P_s - p_{a1})}{\rho}} \quad (2.10)$$

$$q_2 = C_q W x_s \sqrt{\frac{2(p_{a2} - p_{atm})}{\rho}} \quad (2.11)$$

$x_s = 0$:

$$f_r = 0 \quad (2.12)$$

$$q_1 = 0 \quad (2.13)$$

$$q_2 = 0 \quad (2.14)$$

$x_s < 0$:

$$f_r = 2C_q W x_s \cos(\theta_f) (P_s - p_{a2} + (p_{a1} - p_{atm})) \quad (2.15)$$

$$q_1 = C_q W x_s \sqrt{\frac{2(p_{a1} - p_{atm})}{\rho}} \quad (2.16)$$

$$q_2 = C_q W x_s \sqrt{\frac{2(P_s - p_{a2})}{\rho}} \quad (2.17)$$

In Eq. (2.8), the first term is the pressure difference multiplied by the cross sectional area of the spool. The second term is the force from the flexure spring, and the third term is the damping force of the spool. The last two terms are the flow reaction force and the transient flow force. Due to the fact that the spool is in a closed-center position requires the conditional statements for $x_s > 0$, $x_s < 0$, and $x_s = 0$.

2.3.4 Area Gradient

Eqs. (2.9)-(2.17) make use of a linear area gradient, W , to compute the ratio of the spool area that is open in relation to the position of the spool. This is acceptable for square ports, however, with circular ports, as was used in this model, this can lead to an overestimation of the true area opening of the port.

As an improvement, the area gradient can be rewritten as a function of the radius and spool position, shown below as Eq. (2.18).

$$W = \begin{cases} \frac{\left[2 \cos\left(\frac{r - |x_s|}{r}\right)^{-1} - \sin\left(2 \cos\left(\frac{r - |x_s|}{r}\right)^{-1}\right) \right] \frac{r^2}{2}}{|x_s|} & x_s < 2r \\ \frac{\pi r^2}{2r} & x_s \geq 2r \end{cases} \quad (2.18)$$

With the addition of Eq. (2.18), a more accurate Simulink model can be developed to model the circular ports more accurately. This nonlinear equation better approximates the nonlinearities associated with a circular port. By performing a linear fit on the area gradient, as shown below in Fig. 2.3, it can be seen that by assuming a constant area gradient of 0.0071, the constant area gradient both under estimates and over estimates the actual area. When the spool port is first opened the assumption of the constant area gradient can cause an error

of approximately 95% as compared to the expected response between the port area approximations.

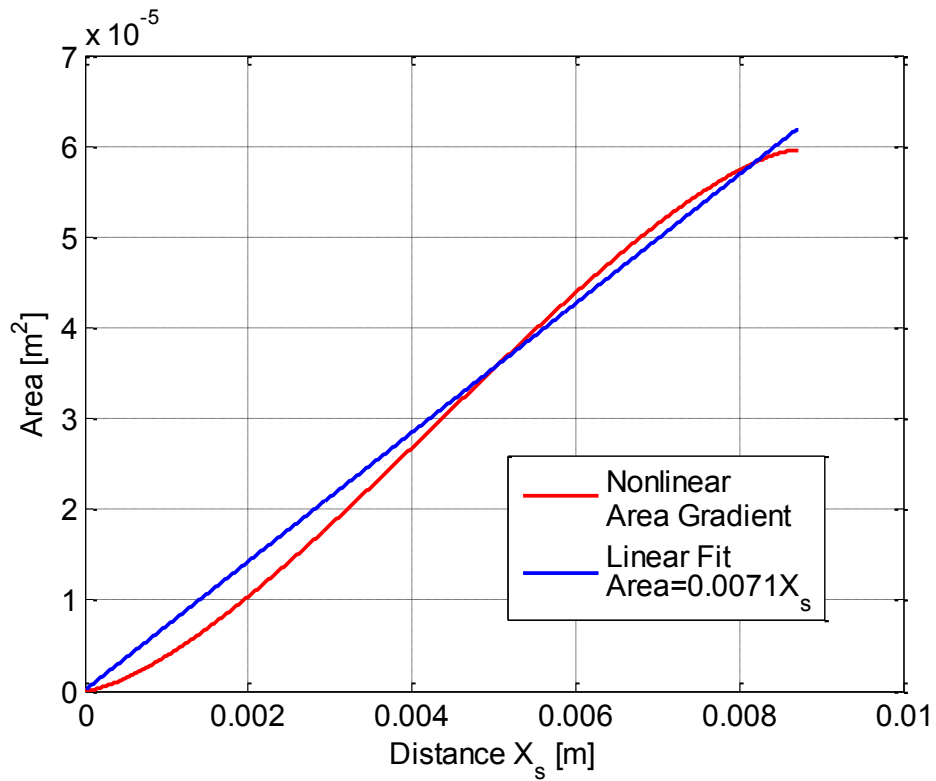


Fig. 2.3: Area Gradient Comparison Linear vs Nonlinear

2.3.5 Flow Continuity Through Actuator

The flow continuity through the actuator is defined as,

$$\dot{p}_{a1} = \frac{\beta_e}{V_{a1}} (q_1 - A_1 v_a) \quad (2.19)$$

$$\dot{p}_{a2} = \frac{\beta_e}{V_{a2}} (A_1 v_a - q_2) \quad (2.20)$$

where

$$V_{a1} = V_{a0} + A_1 x_a \text{ and } V_{a2} = V_{a0} - A_1 x_a$$

Eqs. (2.19) and (2.20) account for the difference between the flow rate and the velocity of the actuator.

2.3.6 Actuator Force Balance

To model the dynamics of the actuator in response to a pressure change, the dynamic equations needed to be developed. The force balance on the actuator is,

$$\begin{aligned} \dot{x}_a &= v_a \\ \dot{v}_a &= \frac{1}{M_t} [(p_{a1} - p_{a2})A_1 - B_a v_a - f_d] \end{aligned} \quad (2.21)$$

where the primary force comes from the pressure difference shown as the first term. The second term is the damping of the actuator and f_d is a supplied force on the actuator, which could possibly be used as a disturbance force [4].

2.4 Linearized Transfer Functions

From these equations, two linearized transfer functions were created, one for the current to spool and one for the spool to actuator [4]. Not all of the variable values were known so the creation of the transfer functions also relied on frequency plots of real data to fit the transfer function to the experimental data. Equation (2.18) is disregarded during the linearization and is only included in the Simulink model. The current to spool transfer function is defined as,

$$\frac{x_s}{i} = \frac{-2.09e14s + 1.25e18}{s^5 + 9900s^4 + 6.89e7s^3 + 3.09e11s^2 + 6.57e14s + 3.39e17} \quad (2.22)$$

while the spool to actuator transfer function is defined as,

$$\frac{x_a}{x_s} = \frac{7.46e8}{s^3 + 1190s^2 + 1.57e7s} \quad (2.23)$$

From Eq. (2.23) there is a pole at zero which would lead the open loop system to being unstable. To improve the stability of the plant, two transfer functions are

cascaded and placed in a closed loop system to stabilize the system. The transfer function that results is an eighth order system shown as

$$po8n(s) = \frac{num8}{den8} \quad (2.24)$$

where num8 is

$$num8 = -1.559e23s + 9.325e26 \quad (2.25)$$

and den8 is

$$\begin{aligned} den8 = & s^8 + 11090s^7 + 9.638e7s^6 + 5.464e11s^5 + 2.106e15s^4 \\ & + 5.972e18s^3 + 1.072e22s^2 + 5.166e24s \\ & + 9.325e26 \end{aligned} \quad (2.26)$$

If the po8n model was not designed as discussed then the spool would stay in an open position causing the actuator to move whenever an input signal was applied to the plant. It would become much more difficult to control the placement of the actuator without the negative feedback loop.

Chapter 3 Simulink Modeling

3.1 Chapter Overview

The purpose of this chapter is to discuss the steps that went into modeling the electrohydraulic actuator from the equations presented earlier. Additionally, the model is verified through three different means to determine if the model can be used for further analysis. Finally, a discussion on numerically solving differential equations is presented in the context of stiff and nonlinear differential equations. A discussion on zero-crossing detection is also presented

3.2 Simulink Modeling

The equations used earlier to define the system are implemented into a full nonlinear Simulink model. The closed loop model is implemented with a subsystem as shown in Fig. 3.1.

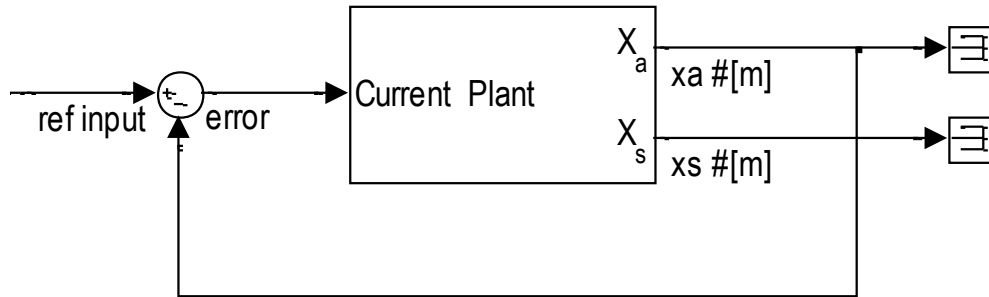


Fig. 3.1: Closed Loop Nonlinear System

The subsystem mentioned in Fig. 3.1 is shown as Fig. 3.2.

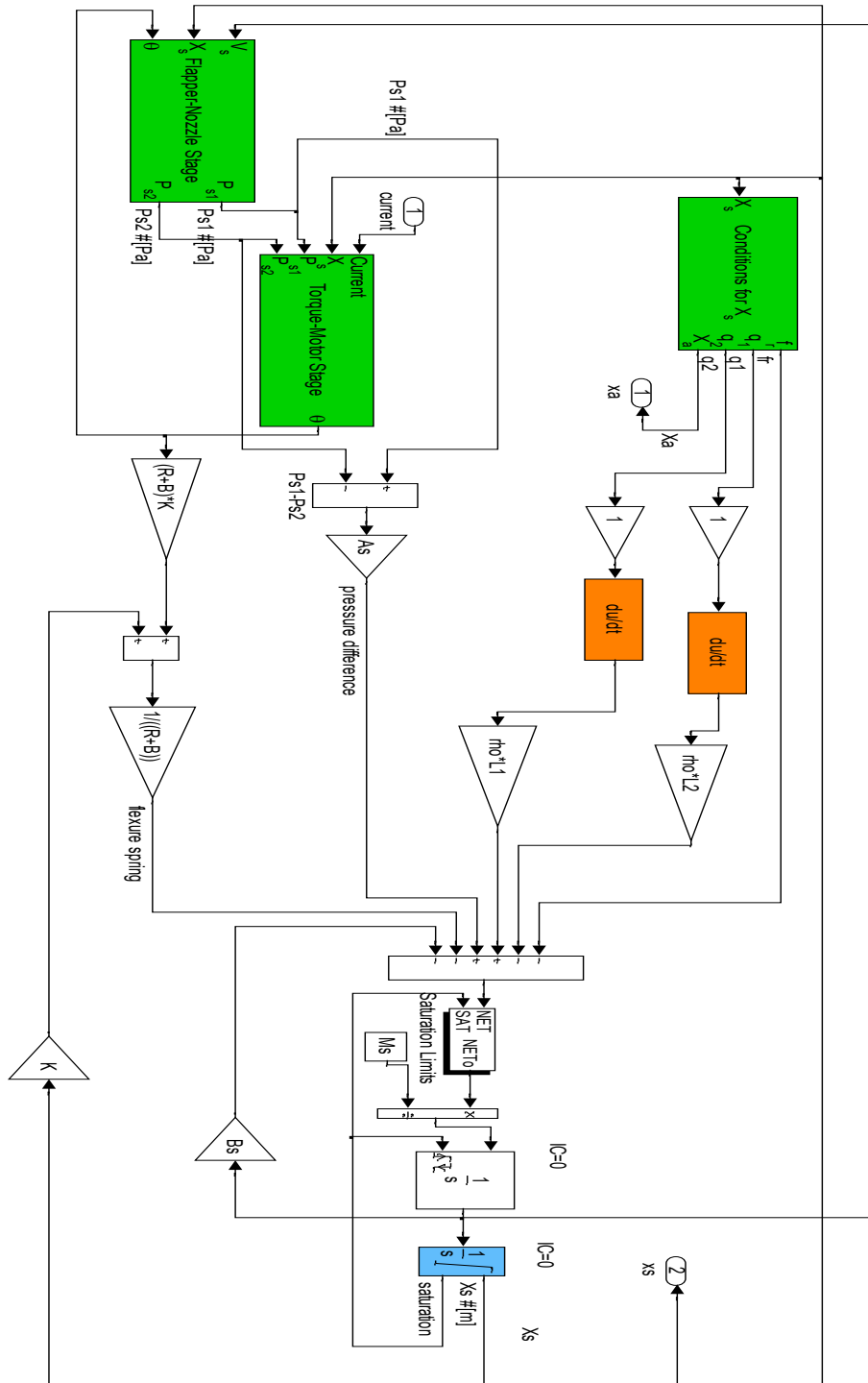


Fig. 3.2: Main Nonlinear Plant Model

From this model, the system is further broken down into different sections such as the Flapper-Nozzle Stage, Torque Motor Stage, and Conditions for x_s . All of these sections are based on the nonlinear equations developed earlier.

Within the Simulink model, the force balance equations on the spool contain two sets of values that require the calculation of the flow rate utilizing the derivative of the flow rate into and out of the actuator from the spool, see the last two terms in Eq. (2.8). In order to numerically solve this in Simulink, a Derivative Filter block was created, Fig. 3.3, which included a derivative, s term (zero), and a First-Order Low Pass Filter.

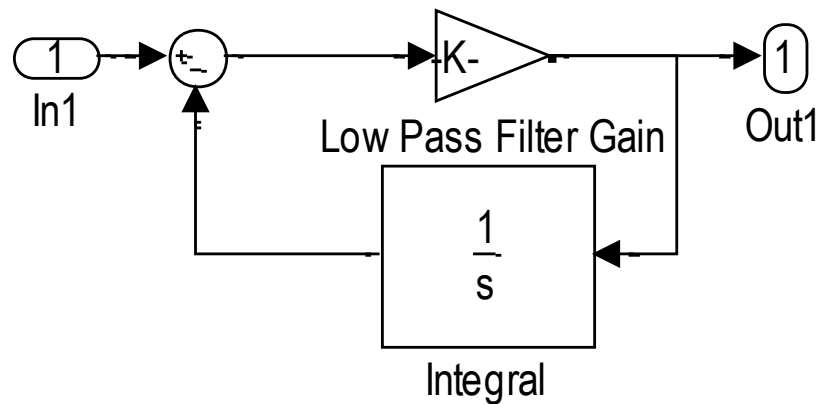


Fig. 3.3: Derivative Filter Utilized in Simulink Model

From this block diagram the transfer function can be shown as,

$$\frac{K}{1 + \frac{K}{s}} \quad (3.1)$$

where K, is the low pass filter gain, and Out1 and In1, are the output and input of the block system, respectively.

Rearranging Eq. (3.1), the derivative and low pass filter can more easily be seen,

$$\frac{s}{\frac{s}{K} + 1} \quad (3.2)$$

With this form in the s-domain the derivative, s, and the low pass filter, K, can be easily understood. In the form of this equation the low pass filter gain value should be inputed as $1/f_{\text{break}}$, where f_{break} is the break frequency. For this simulation, the break frequency was set to be 100 Hz. This was as an appropriate value to prevent noise from causing the simulation to become numerically unstable and was determined through multiple iterations at different values.

As with any system, there are limits to which a system can be expected to operate. This system has three main limits. The first is the limit of the input current which drives the system. For the actual system the input current is limited to ± 40 [mA]; anything outside of that range could cause the torque motor to burn up.

Additionally, the spool is limited to $\pm 8.71e-3$ [m] and the actuator is limited to $\pm 2.54e-2$ [m] per the actual limits on the physical system.

In order to accurately model the spool and actuator saturation limits, a series of blocks were used to either allow or not allow the simulation to calculate values based on the placement of the spool and actuator. To accomplish this in Simulink, Fig. 3.4 was implemented.

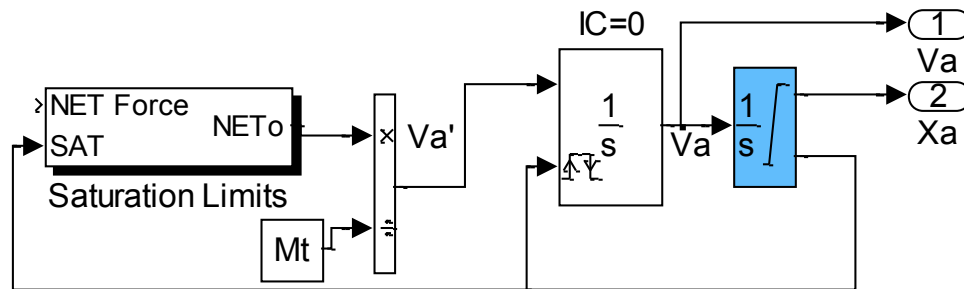


Fig. 3.4: Saturation Limits

The integrator for the velocity to displacement term has saturation limits along with a saturation port. The saturation port will supply a +1 or -1 if the upper or lower saturation limit has been reached respectively. The saturation signal is then fed to the acceleration to velocity integral which will reset back to the initial condition when there is a sign change. The saturation signal is also fed into the saturation limit block. This block inputs the net force and the saturation binary

value. Depending on the saturation signal and the sign of the force, the block will either output zero or the force value. This allows the system to stop calculating the acceleration, velocity, and displacement when the saturation limit is reached, and then, begin calculating again once the displacement signal is less than the saturation limit. This setup was implemented for both the spool and actuator.

The set of nonlinear equations used to define the system are in absolute pressure. This presents the unique opportunity in Simulink to allow saturation limits to be defined. The saturation limits on the integrator block for calculation can be limited to zero and infinity. This helps with computation time since using absolute pressure calculations will never allow a value below zero units of pressure to exist.

With that being said, there is still a possibility, especially, when calculating the difference between pressures, that the net difference will be less than zero due to numerical simulation techniques. This can cause problems, especially for the flapper nozzle equations that deal with the turbulent flow equations through an orifice. Due to the fact the square root will be taken of the pressure difference, a negative pressure difference will result in a complex value. To remedy this

problem a saturation block was implemented at those problem points. This limits the signal to be between zero and infinity, allowing for a resilient simulation model.

To model the spool dynamics, conditional statements had to be implemented as shown in Eq. (2.9)-(2.17). The model is broken down into three separate sections to accurately handle the equations as shown in Fig. 3.5.

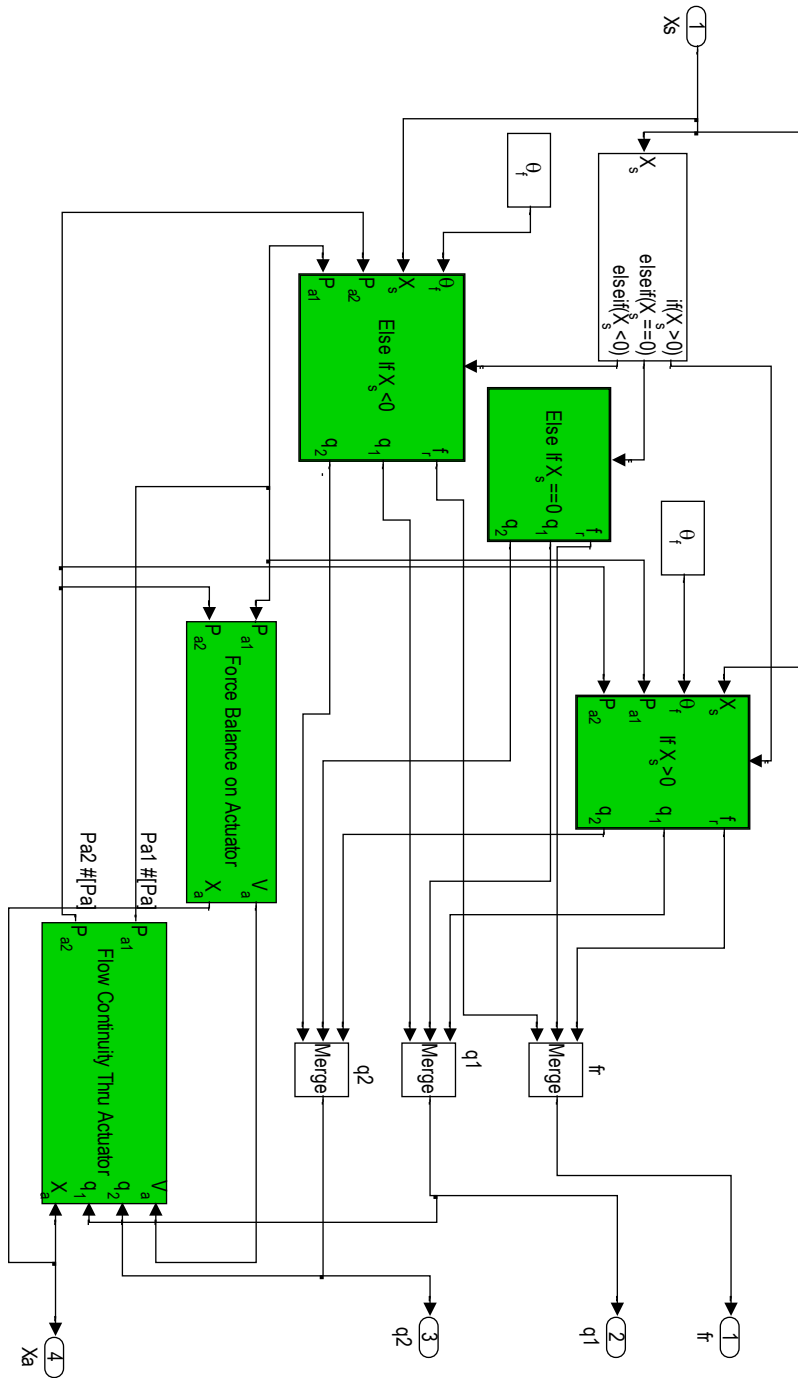


Fig. 3.5: Conditional Statement Modeling

The conditional was setup by first using an if/elseif conditional block. Depending on the input of the block, x_{ss} , the output would then be sent out and enable the corresponding subsystem. Additionally, a merge block had to be used for F_r , Q_1 , and Q_2 , due to the fact that there were three potential outputs. The merge block worked by combining or merging the signals from the separate points of origination to create one cohesive signal.

Throughout literature most models approximate the system of an electrohydraulic actuator through the use of linearized equations [17]. While this is useful, a more complete model can be implemented by utilizing the nonlinear equations into a Simulink model. With the addition of the nonlinear model the nonlinear nature of the model can be better analyzed, especially, when controllers need to be implemented or even for design modifications. Typically, the servovalve dynamics are usually neglected or approximated by first, second, or third order equations [11].

3.3 Simulink Model Justification

In order to justify the Simulink model created and the solutions presented, three methods were chosen. The first method was a frequency justification using bode plots of different aspects of the closed loop system. Additionally, the step input responses of the transfer function was compared against the nonlinear model at acceptable linear ranges. The step responses at larger distances were also used to gauge the extent of the nonlinearity as well as the believability of the Simulink nonlinear model. Finally, the values that were unknown had to be estimated via parameter estimation. The values that were estimated were then compared against known acceptable values to determine the validity of the estimation. The final model offers the unique advantage over other models, especially, models that are modeled with transfer functions or S-functions or MATLAB function blocks that allows an individual to look at the values all throughout the model. This does not just apply to the input and output responses, but also pressures, flow rates, and a whole host of other values of interest.

3.3.1 Time Response Parameter Estimation

The step input response was evaluated at different step inputs. The data was originally fit to the transfer function step input of $2.54e-4$ [m] as a starting point. This was done to accurately use the linearized po8n model for the nonlinear model. Knowing that there was only a limited range for which the linear po8n would accurately predict the nonlinear response, a step input of $2.54e-4$ [m] was used. For the actuator to spool transfer function the same sinusoidal input signal was applied to the current, but then the spool and actuator resulting sinusoidal response was gathered. The parameter estimation was estimated at this point because it was deemed a safe region where the true model was known to exhibit linear traits, as seen in the linearized eighth order transfer function. By fitting the unknown parameters to the linearized transfer function output, the following figure was created.

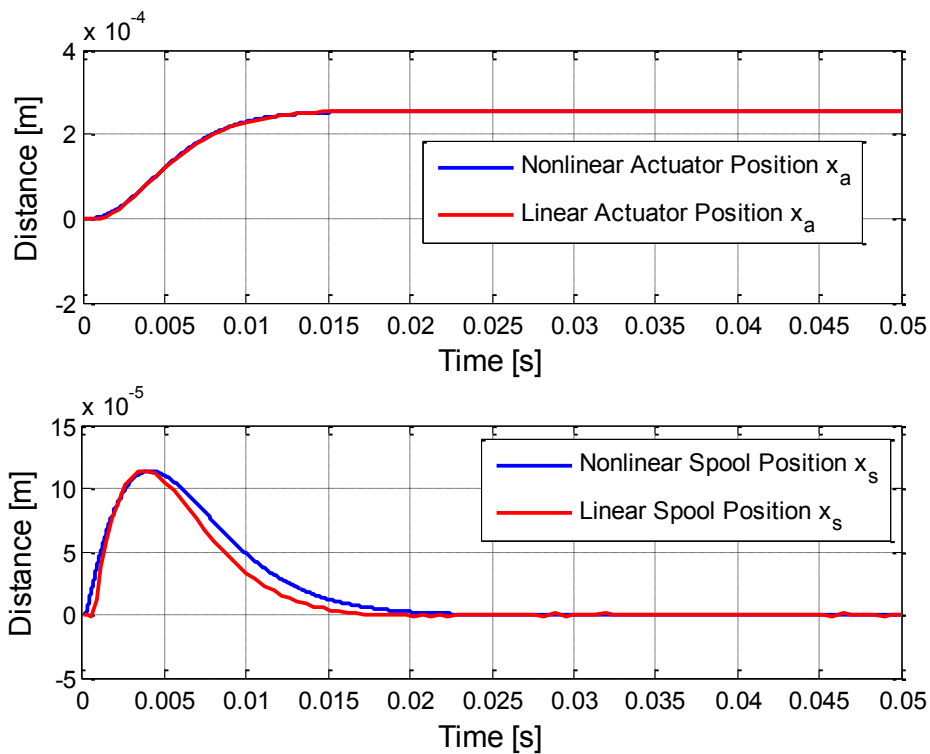


Fig. 3.6: Linearized vs Nonlinear Step Response at 2.54×10^{-4} [m]

After many iterations of parameter estimation, the best parameter values were used to create the response shown in Fig. 3.6. In addition, the estimated parameters also fit the spool response well. The largest variation from the transfer function response is 4.53×10^{-6} [m] for the actuator response. Due to the original transfer functions for the spool and actuator being developed based on a voltage signal, it was difficult to determine the actual amplitude of the spool. This being the case, the actuator and spool response both could not be used in parameter estimation. To scale the linear spool response down to the same level as the nonlinear response, the linear was scaled by 0.1503. The peak magnitudes of the

spool responses, both linear, and nonlinear were used to develop the 0.1503 factor. The effect of the spool value can be seen in the bode plot for the spool, Fig. 3.10 and Fig. 3.11.

While experimental data was not available for large step inputs as shown in Fig. 3.7, the nonlinear Simulink model was still able to accurately reproduce nonlinear results for both the spool and actuator. This is in contrast to other models, especially the references, which typically approximate the servovalve, and the combination of the spool and actuator combination into one model.

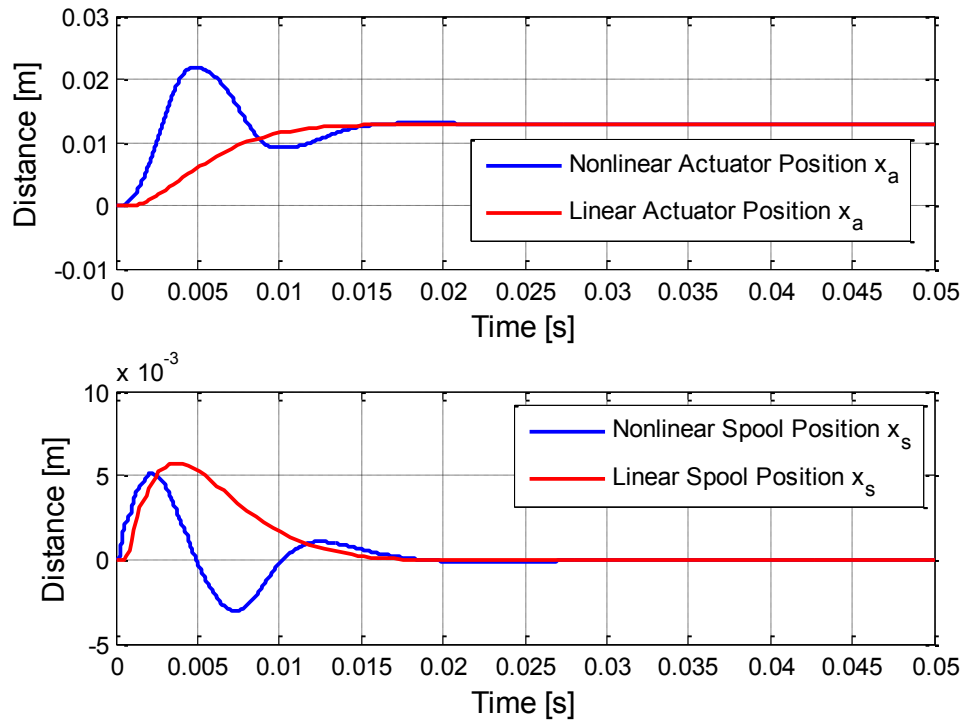
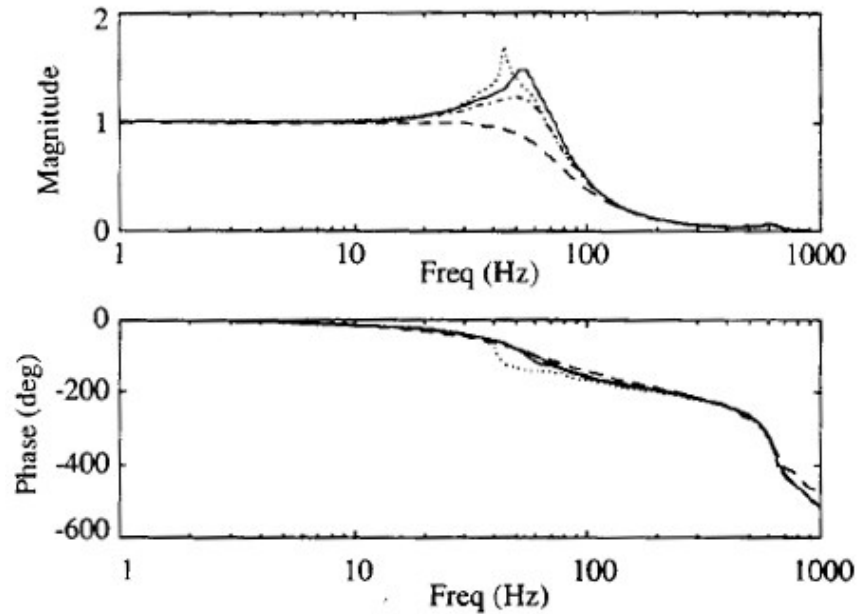


Fig. 3.7: Step Input Response of $1.27\text{e-}2$ [m]

The actuator and spool both provide responses that would typically be seen at larger step inputs. This is especially true with the overshoot and undershoot seen by the actuator and spool responses. Additionally, more plots at various step amplitudes in between $1.27\text{e-}3$ [m] and $2.54\text{e-}4$ [m] can be seen in Appendix A. The nonlinearities, especially with the larger step inputs can be verified by the frequency plot shown in Fig. 3.8. The larger amplitude deviates significantly from the smaller amplitudes. The larger magnitude peaks near the natural frequency for larger ranges of motion is equivalent to a lower damping ratio,

which results in a larger overshoot with a step response. A similar response can be seen from the larger step inputs of the closed loop model.



Frequency Responses of Plant: 0.5 mm input (dash),
2.0 mm (dash-dot), 3.0 mm (solid), 5.0 mm (dot).

Fig. 3.8: Frequency Response Nonlinearities [18]

The nonlinear effects in electrohydraulic systems are well known and can cause response problems which is apparent as mentioned above [19].

Additionally a similar system could be tuned for different variable values and then be used to show the nonlinearities of that system. Ultimately this allows an

individual to explore different aspects of a system with nonlinearities, rather than a pseudo-nonlinear system or entirely linear system.

3.3.2 Frequency Justification

In the Simulink model created, both the step and bode plots were compared against the linear transfer functions. As shown below, Fig. 3.9 shows the comparison between the nonlinear Simulink model at various reference sinusoid input amplitudes versus the accepted eighth order po8n model.

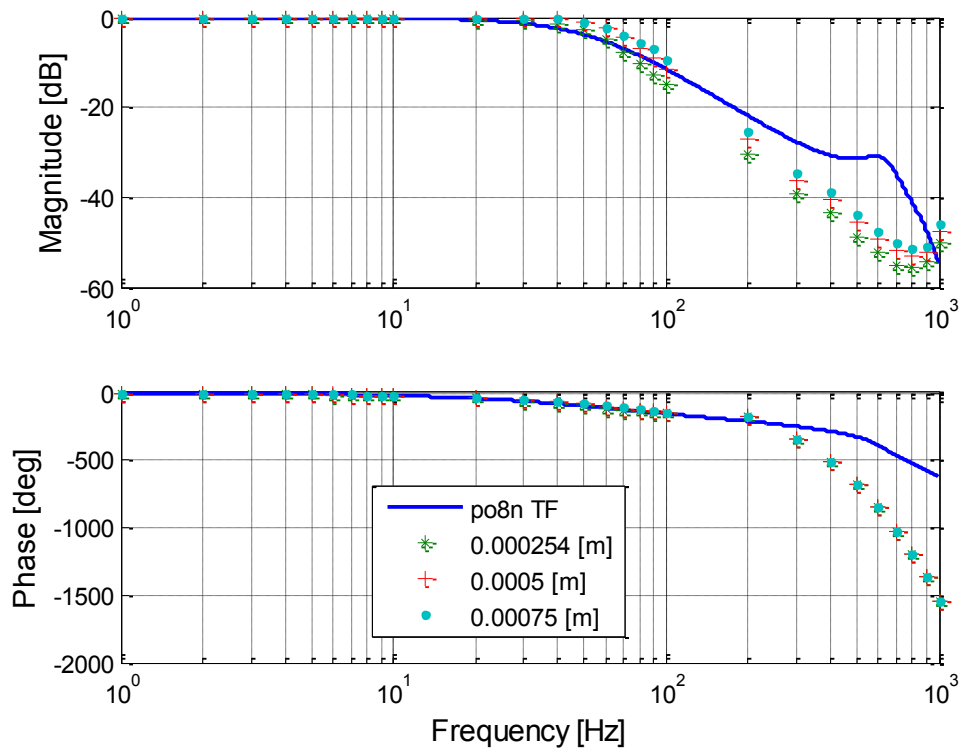


Fig. 3.9: Magnitude Phase Comparison of po8n Model

From the Fig. 3.9, it can be seen, that the nonlinear model matches the accepted linear transfer function, Eq. (2.24), up to the 200-300 [Hz] range. This is well past the bandwidth of the system, assuming a -3 [dB] drop from the initial position. The bandwidth for the closed loop system is approximately 45 [Hz], which shows that the model works well past that frequency. The simulation data is determined by running the simulation at the various sinusoidal amplitudes as indicated by the legend for the input signal. The output, actuator position, and input, reference signal, were then compared to produce the plot above.

Shown below, Fig. 3.10 shows the current input versus spool response.

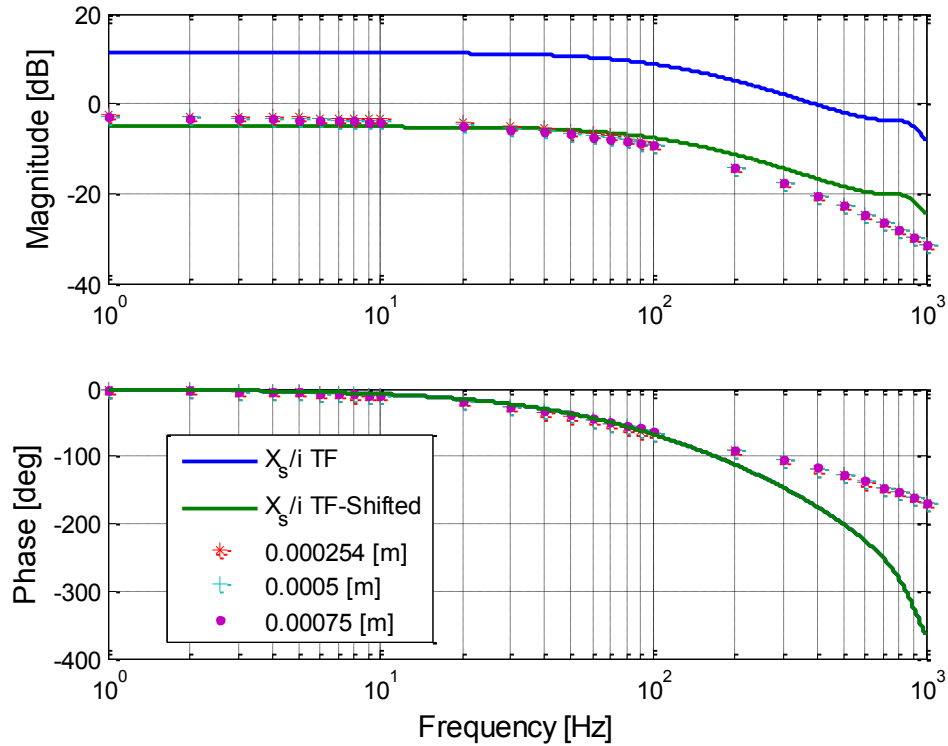


Fig. 3.10: Magnitude Phase Comparison of Spool vs Current Input

From Fig. 3.10, it can be seen, that magnitude and phase both closely follow the accepted fifth order transfer function, Eq. (2.22). The magnitude values for the nonlinear model are offset from the linearized transfer function, due to the fact that, the nonlinear model uses the pure distance to calculate the magnitude. Additionally the scaling value used to scale the spool linear response of 0.1503

when converted to [dB] results in the magnitude shift in the transfer function as shown in Fig. 3.10. From this explanation it can be seen that a correlation has been made linking the two justification methods for this particular set of variable values. In addition, the linearized transfer function is based off of a voltage from the lvdt used to measure the displacement of the spool. This accounts for the offset shown above. The simulation data is determined by running the simulation at the various sinusoidal amplitudes as indicated by the legend for the current input signal. The output, spool position, and input, current, were then compared to produce the plot above. This bode plot also shows that high speed dynamics has been accurately captured by the nonlinear model given that the bandwidth of x_s/i transfer function is approximately 115 [Hz]. This bandwidth is more than twice the bandwidth of the closed loop system, showing that while it is not entirely necessary for modeling purposes, it can still provide useful information about the dynamics of the spool.

Finally, Fig. 3.11 below shows the comparison of the spool to actuator for the nonlinear and linear versions Eq. (2.23). The magnitude plot has the same offset as Fig. 3.10, does and follows the same general trend as the linear transfer function. The data for Fig. 3.11 was calculated the same way the other plots

were, by applying a constant sinusoidal input and then comparing the response of the spool and actuator.

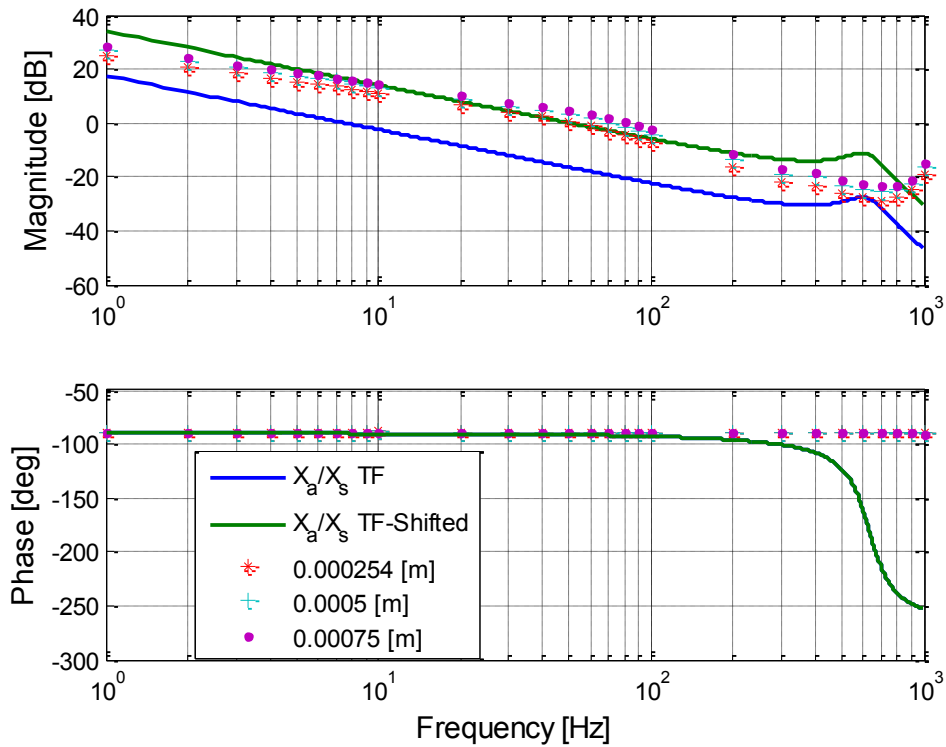


Fig. 3.11: Magnitude Phase Comparison of Spool Input vs Actuator

The nonlinear plant follows the linear transfer function until approximately 500 [Hz] for both the phase and magnitude plots. Once again the shifted magnitude response of the transfer function is determined based on the same factor that was used for the 2.54×10^{-4} [m] step response, 0.1503. This further shows the relationship between the two justification methods and based on the parameters chosen provide an accurate correlation, accounting for each other discrepancy.

There is a slight disagreement between the simulated data and the transfer function at low frequencies; however this is typical for hydraulic systems. Additionally the difference in magnitudes between the spool and actuator can also be credited with the error between the transfer function and simulation values.

Table 3.1 below shows the bandwidth for the corresponding system and the agreement of the nonlinear Simulink model to the linear transfer function.

Table 3.1: Nonlinear-Linear Frequency Comparison

<u>Model Subsection</u>	<u>Bandwidth [Hz]</u>	<u>Nonlinear Model Effective Range [Hz]</u>
po8n (closed loop model)	45	0-200
X_s/I (Current and Spool)	115	0-400
X_a/X_s (Spool and Actuator)	1.4	0-500

From the table it can be seen that the nonlinear model follows the linearized transfer function well past the bandwidth of the system. The bode plots and Table 3.1 show that the nonlinear model developed in Simulink is a good representation of the electrohydraulic actuator. Additionally not just the initial reference signal and actuator position signal, but also the transitional parts of the system have good correlation with the appropriate linear model.

3.3.3 Variable Justification

In order get the correct response, variables had to be estimated, due to not knowing the actual value. Table 3.2 shows the final values used for the simulation.

Table 3.2: Variable Values

Variable	Units	Value	Estimated/Known
A _l	[m ²]	3.613e-4	Measured
A _o	[m ²]	2.241e-7	Estimated
A _s	[m ²]	1.677e-5	Measured
B	[m]	2.267e-2	Measured
B _a	[N/(m/s)]	1.068e-2	Estimated
B _s	[N/(m/s)]	5.621e-4	Estimated
B _v	[N-m/(rad/sec)]	7.161e-2	Estimated
C _q	[-]	5.726e-1	Estimated
C _{qn}	[-]	5.901e-1	Estimated
C _{qo}	[-]	6.730e-1	Estimated
D _n	[m]	6.972e-4	Estimated
F _d	[N]	0	Variable
J	[kg-m ²]	8.866E-08	Estimated
K	[N-m/m]	7.916e2	Estimated
K _a	[N-m/rad]	8.582e2	Estimated
K _m	[N-m/rad]	4.402e-4	Estimated
K _t	[N-m/A]	1.318e3	Estimated
L ₁	[m]	1.2e-2	Measured
L ₂	[m]	1.067e-2	Measured
M _s	[kg]	2.52e-2	Measured
M _t	[kg]	9e-1	Measured
P _s	[N/m ²]	1.860e7	Measured
R	[m]	1.067e-2	Measured
V _{ao}	[m ³]	9.177e-6	Measured
V _{so}	[m ³]	1.461e-7	Estimated
X _{fm}	[m]	3.115e-5	Estimated
β _e	[N/m ²]	1.400e9	Literature Value

θ_f	[rad]	1.124	Estimated
ρ	[kg/m ³]	800	Measured

Table 3.2 also shows which values were estimated and which values were physically measured or determined from known sources. The values that had to be estimated are more difficult to measure. Several of the estimated values used in the final model are discussed for justification.

The A_o value, which is the cross sectional area of the orifice was estimated, due to the fact that actual orifice cross sectional area was not measured and not known. Gordic et. al [11] for the orifice area used $2.5447e-8$ [m²] in there servovalve model. This value is closer to the final value used in this model as opposed to the initial value used.

K_t is defined as the torque motor gain. This is the value that amplifies the current input. K_t becomes a major factor in determining the response of the position of the spool. This causes the K_t value to become much larger than the initial value that was previously given. Additionally, since the original K_t value did not produce a response with magnitudes close to the supposed linear transfer function

response, the damping and spring constants for the torque motor, spool, and actuator all needed to be estimated to produce the correct response.

Initially in the Simulink model the spool was only allowed to open a percentage of the circular port total diameter, which allows oil to enter and exit the spool and flow into the actuator. This caused a major restriction on how fast, ultimately, the actuator could respond, due to not having the full circular port to work with. Knowing the diameter of the circular port and that the spool should be able to, at the very least, be able to full open the circular port it was decided to increase the spool limit to be $\pm 8.71e-3$ [m], which is equivalent to the diameter of the spool. V_{so} is the enclosed volume on each side of the spool when the spool is at $x_s=0$. Knowing the cross sectional area of the spool and the overall length the spool can move gives the final value shown in Table 3.2 for V_{so} .

θ_f is defined as the angle at which the fluid jet leaves the spool chamber. Merritt [20] determined the angle to be $69^\circ=1.2043$ [rad]. θ_f is defined as the angle between the spool and the port as shown by θ in Fig. 3.12

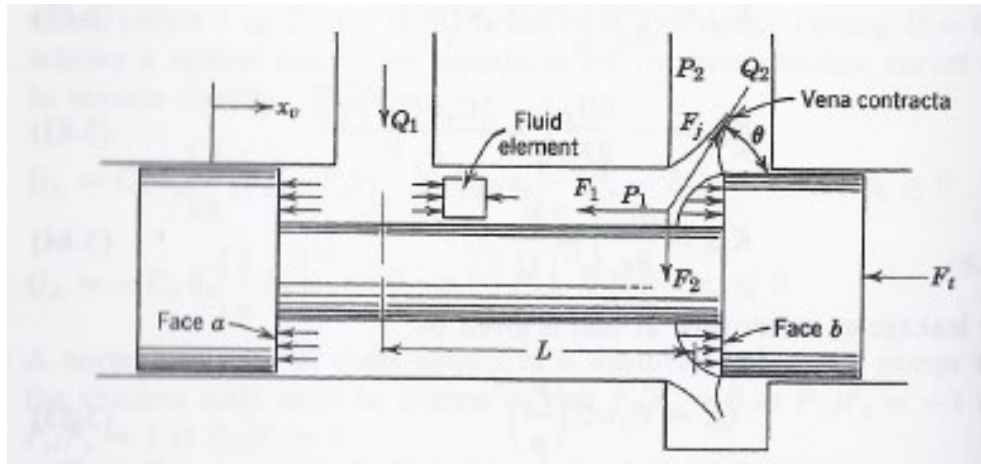


Fig. 3.12: Jet Flow Angle [20]

From the parameter estimation, the angle was actually determined to be $64.4^\circ = 1.1244$ [rad], which is slightly smaller than the original value. Merritt assumed that the variable orifice was rectangular. The orifices in the spool were not entirely rectangular shaped but rather the port was circular and could account for the reason why there was a slight difference between values.

Additionally, Fig. 3.13 shows the relationship that was developed between the vertical gap between the spool and chamber by Merritt [20]. The figure shows the relationship between the horizontal and vertical spacing of the spool. There could have also been a gap between the spool and chamber. This could also help account for the difference between the accepted 69° and the parameter estimation of 64.4° .

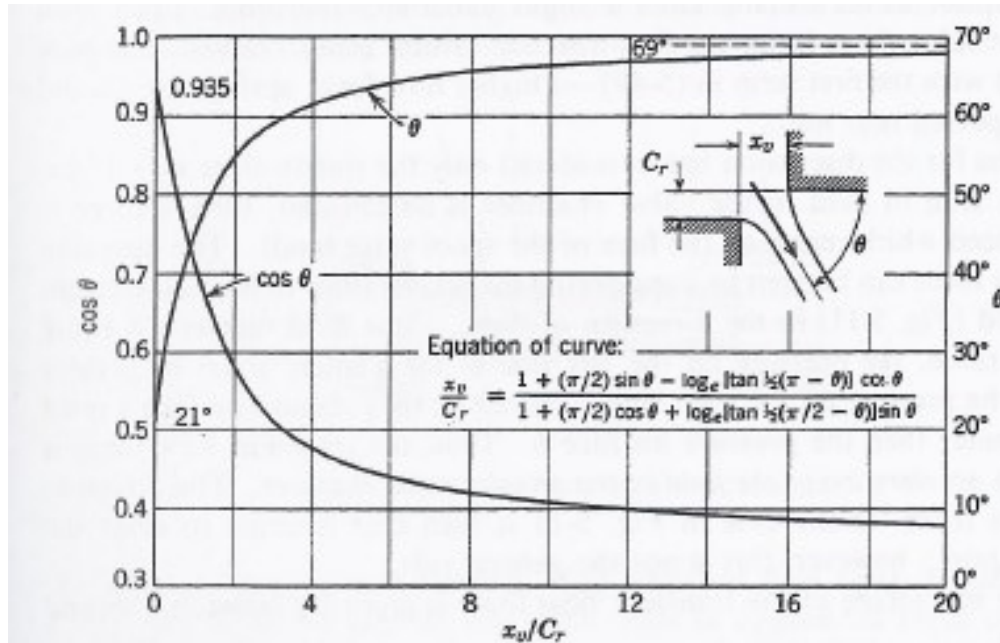


Fig. 3.13: Flow Angle as Function of Spool Gap [20]

Due to the fact that the step response was of a closed loop model that stabilized the plant, meant that the spool position varied. The value produced from the parameter estimation can be thought of more as an average θ_f . If an even more detailed response was necessary and data was available to validate, the equation of the curve shown in Fig. 3.13 would be used. The equation would be solved for θ_f , and C_r would be chosen to be negligible in relation to the overall distance the spool moves.

3.4 Numerically Solving

Solving differential equations are usually accomplished by solvers designed for specific purposes. Hydraulic systems are typically stiff systems which require special attention. There are different methods that can be taken to improve the simulation time and ability to solve a large number of differential equations.

3.4.1 Stiff Differential Equations

One of the major obstacles with running the Simulink model was to be able to solve the many differential equations quickly and efficiently. Hydraulic systems are typically stiff in nature and, as such, have to be approached differently to allow for quick simulation times. Stiff differential equations are not easily perceived as being stiff. The definition of a stiff differential system is, in itself, still a place of continued discussion. Generally speaking a stiff system consists of both slow and quickly changing variables [21].

Typically, if a system cannot be solved in a reasonable amount of time by using traditional solvers such as ODE45, then a stiff solver such as ODE23S or

ODE15S is recommended. The standard solver used by MATLAB is the ODE45 solver. The ODE45 solver is based on the Runge-Kutta(4,5) function. The ODE45 solver is a one-step solver, in the sense that, it only needs the solution at the previous time point [22]. While the ODE45 could potentially solve a stiff differential equation, the potential trade off would be that it would be difficult for the solver to solve and could take a large amount of time to solve.

The solution to this dilemma is to use a stiff solver, i.e. a solver designed to handle stiff differential equations. The ODE23S is specifically designed for solving stiff differential equations as indicated with the 'S'. The ODE23S solver is based on a Modified Rosenbrock formula of second order [22]. As is the ODE45 solver, the ODE23S is also a one-step solver. Utilizing the one step solver can make the solver more efficient than the other stiff solver, such as the ODE15S.

The ODE15S solver was also used in this simulation. One of the main differences between the ODE15S and ODE23S solver is that the ODE15S is a multistep solver. The solver can use multiple previous points in time to compute the current solution. This differs from the ODE23S which only needs the previous point in

time. The ODE15S also uses a numerical differentiation formula as opposed to the Modified Rosenbrock [22].

With the PID controller design mentioned later on all three solvers worked just as effectively in terms of time to solve. The full state feedback controller worked for both the ode45 and ode15s solver. The ode23s solver did not however work as efficiently. In eight hours approximately 3% of the entire time had been simulated. A similar behavior was experience with the PI full state feedback controller. All simulations were run under default solver configurations.

Chapter 4 Controller Design

4.1 Chapter Overview

This chapter discusses the different type of controllers designed for use with nonlinear plant. All results have been simulated using the nonlinear plant model explained earlier. An analysis of both the step response and the current input signal is discussed.

4.2 PID Controller

4.2.1 Step Response and Input Signal Analysis

The PID controller method is used extensively throughout industry for its ease of use, both from the standpoint of implementation and determining the characteristic values.

A standard controller has the form

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (4.1)$$

where K_p is the proportional (P) gain, K_i is the integral (I) gain, and K_d is the derivative (D) gain. There are many variations of this PID controller that are used; one variation adds a low pass (first order) filter to the derivative term. The filter is used in conjunction with the derivative term to attempt to reduce the amount of noise produced from the derivative portion of the PID. The modified equation is shown below as Eq. (4.2)

$$G_{cf}(s) = K_p + \frac{K_i}{s} + K_d s \left[\frac{N}{s + N} \right] \quad (4.2)$$

where N is the filter coefficient and is responsible for the location of the filter pole.

Combining into one transfer function G_{cf} becomes,

$$G_{cf}(s) = \frac{(K_p + K_d N)s^2 + (K_p N + K_i)s + K_i N}{s^2 + Ns} \quad (4.3)$$

The best PID that was determined based on tuning and the Ziegler-Nichols step response formula in the SISOTOOL from MATLAB was determined to be,

$$G_{cf\,values}(s) = \frac{34.93s^2 + 3.749e4s + 1.053e7}{s^2 + 1.152e4s} \quad (4.4)$$

From Eq. (4.4) the K_p , K_i , K_d , and N values are 3.175, 914.365, 0.002757, and 11518.605 respectively. With these values, Fig. 4.1 is shown below.

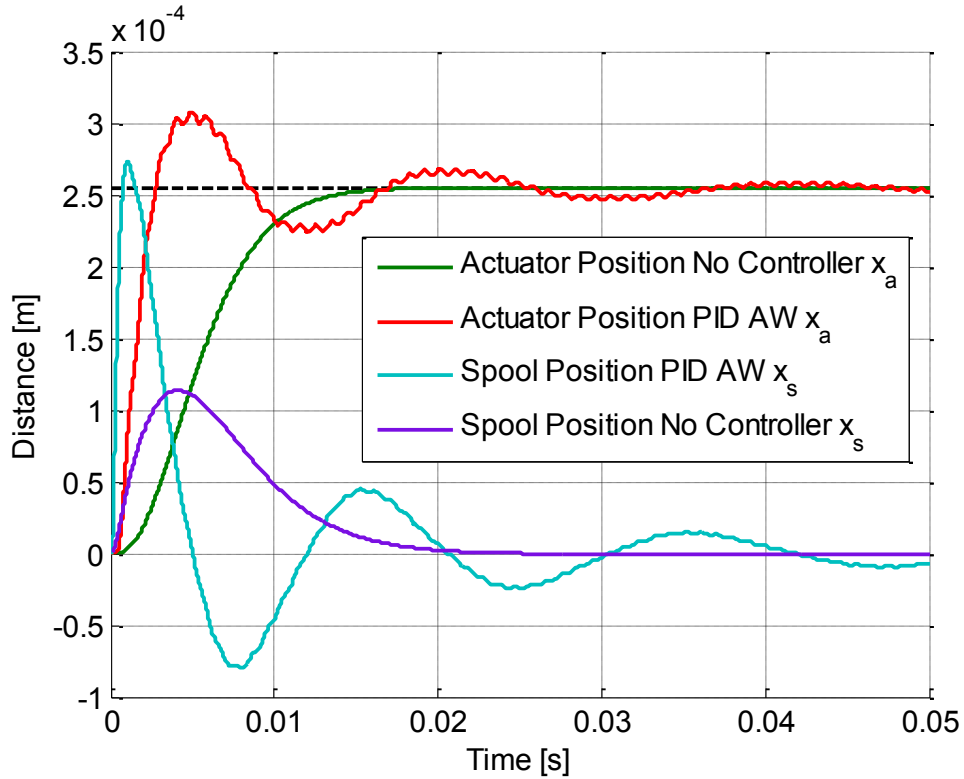


Fig. 4.1: Reference Input and Output Simulated Response for Actuator Position in Relation to Input Spool Signal for $2.54e-4$ [m] Step Input.

As can be seen, the values were chosen to increase the response time of the actuator system. The other stipulation of this system was to attempt to decrease the input signal while not losing the response time. The compromise was found with the above controller design. Due to the lack of information available from the current input signal plot, the relevance of the signal will be presented in a

meaningful way later on. Additionally, when the step is increased the controller is still able to track even with the existence of the nonlinearities from the larger step input. This is due in part to the anti-windup introduced that is only enabled at larger step inputs.

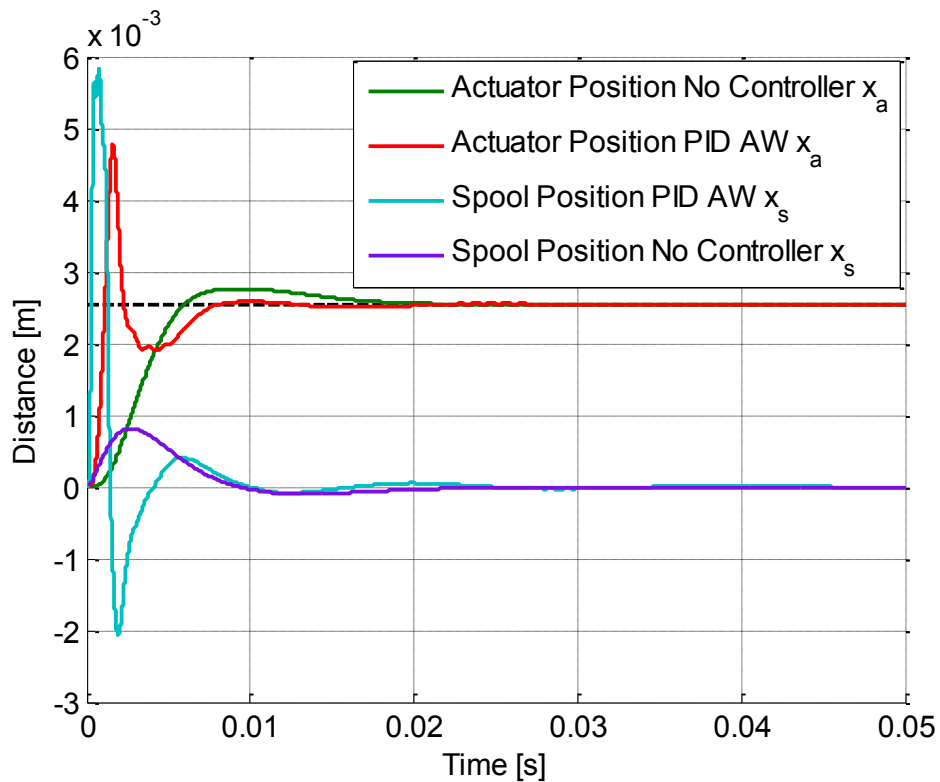


Fig. 4.2: Input and Output Simulated Response for Actuator Position in Relation to Input Spool Signal for $2.54e-3$ [m] Step Input.

A block diagram representation of the PID with derivative filter can be seen below in Fig. 4.3.

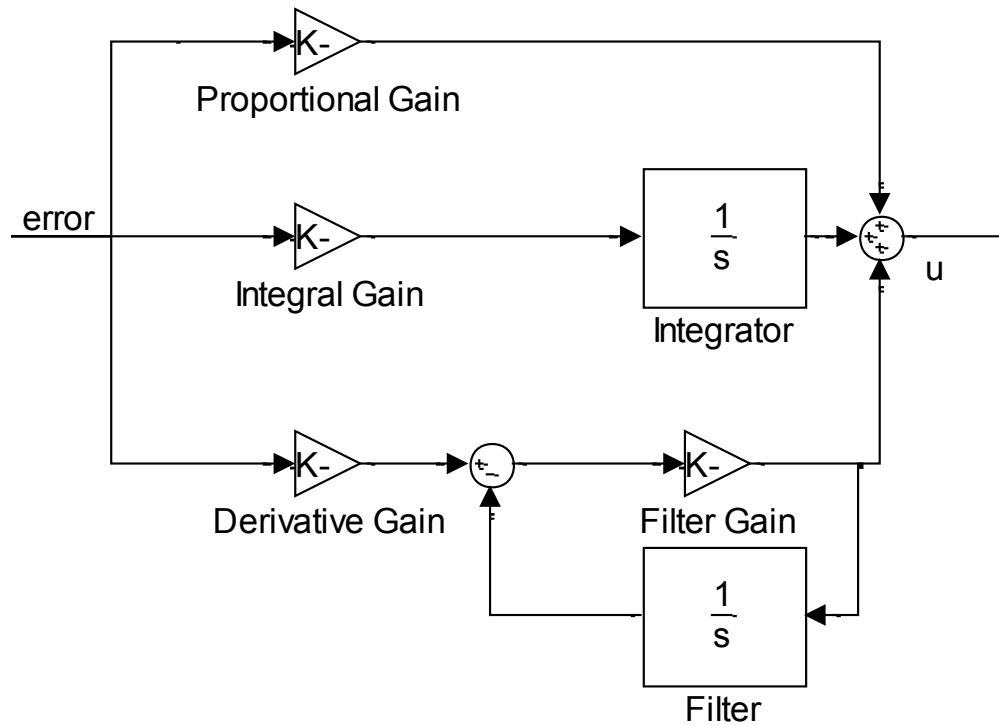


Fig. 4.3: Block Diagram Representation of PID with Derivative Filter

In addition to the derivative filter attached to the derivative gain, an anti-windup technique can also be applied to the integral portion of the PID [23]. Tracking is a type of anti-windup that keeps the integral at a proper value to readily account for the error signal that is fed back into the closed loop system. As shown below in Fig. 4.4, the signal fed into the saturation block is compared to the output of the saturation block. This comparison is then multiplied by a gain of $\frac{1}{T_t}$, which determines the extent of the tracking based on the T_t value given.

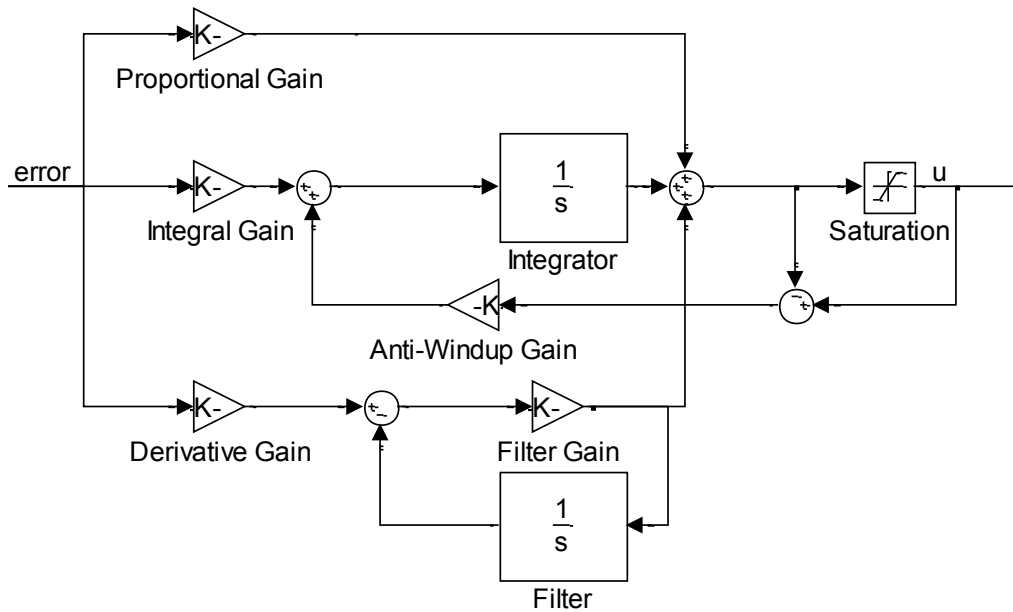


Fig. 4.4: PID Controller with Anti-Windup via Tracking and Derivative

Filter

With this error added back to the integral signal, the integral signal can be reduced to acceptable values.

For the actual nonlinear model the anti-windup method utilized a slightly different approach. Due to the stable plant consisting of a feedback loop, the actual signal to the electrohydraulic actuator is within the plant the controllers are designed around. Also, the electrohydraulic actuator only allows ± 40 [mA], so a saturation limit block was placed before the input to the physical system. The current was then compared before and after the saturation block. This signal was then fed

back into the integral signal of the PID controller, but before multiplying by a gain of 1/5. 1/5 was determined by manually tuning the system.

Near a $2.54e-4$ [m] step input, the saturation inputs did not play apart, however, at larger inputs around $2.54e-3$ [m] the saturation limit did. Rather than simply disregarding the input, the signal was fed back into the PID controller, which in turn, helped increase the response time at larger step inputs as seen in Fig. 4.1 and Fig. 4.2.

Table 4.1, below, shows the comparison of the current input signal to two step inputs with regards to different controllers. The term large step will be used to define the step of $2.54e-3$ [m] and small step will be used to define the step of $2.54e-4$ [m]. The current area refers to the area under the curve of the input signal, where all points are considered positive. The current area is defined by the integral of the absolute current from $[t_0 t_1]$ as shown in Eq. (4.5).

$$Ca = \int_{t_0}^{t_1} |i(t)| dt \quad (4.5)$$

This serves as a way to gauge the effectiveness of the controller to the plant in terms of energy efficiency. All analysis was performed from 0 to 0.05 seconds to be consistent.

Table 4.1: PID Current Comparison

Controller Type	Current Area [A-s]		Max Current [A]	
	Small step	Large step	Small step	Large step
Plant	1.48e-6	1.00e-5	2.54e-4	2.5e-3
PID	3.81e-6	4.06e-5	89.0e-4	88.7e-3
PID with Anti-windup	3.81e-6	3.51e-5	89.0e-4	40.0e-3

Table 4.2 shows the actuator response to the various controllers and non-controller designs. From the table, it can be seen that the anti-windup does not negatively impact the standard PID controller in anyway. The addition of the anti-windup actually decreases the settling time of the actuator for a step input of 2.54e-3 as compared to the open loop or standard plant design.

Table 4.2: PID Actuator Position Comparison

Controller Type	Actuator Percent Overshoot [%]		Actuator Settling Time [s]	
	Small step	Large step	Small step	Large step
Plant	0.144	8.615	1.35e-2	17.2e-3
PID	22.170	86.407	4.59e-2	15.0e-3
PID with Anti-windup	22.170	87.598	4.59e-2	7.5e-3

4.2.2 Disturbance Rejection

In addition to the step response of the controller design, the disturbance rejection ability was also analyzed. The response for a reference input of zero, is shown with the disturbance in Fig. 4.5. The disturbance is added after the controller, but before the input to the closed loop plant.

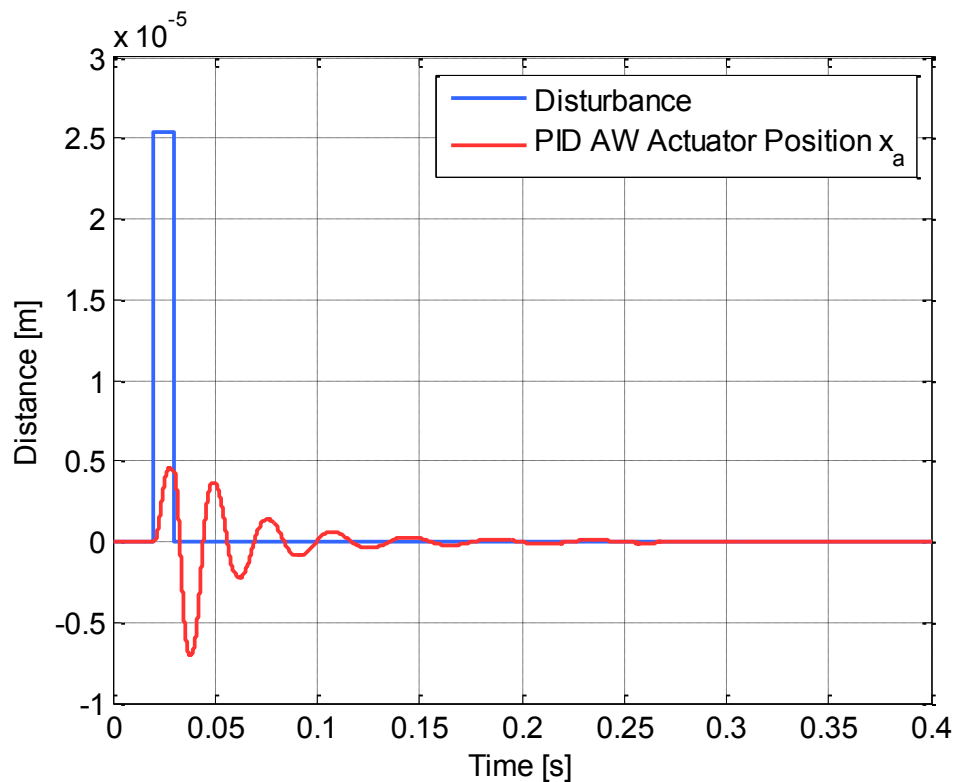


Fig. 4.5: Disturbance Rejection for PID with Anti-Windup Narrow Pulse

From the figure it can be seen that it takes 0.263 [s] for the disturbance to be removed from the output signal by the controller. The disturbance consists of a

pulse, $1e-2$ [s] wide and $2.54e-5$ [m] tall. The disturbance begins at $2e-2$ [s]. The disturbance also causes a fair amount of oscillation in the actuator position. A second disturbance was also analyzed with a pulse twice as wide as the first pulse mentioned. The $1e-2$ [s] pulse will be referred to as the narrow pulse and the $2e-2$ [s] pulse will be referred to as the wide pulse.

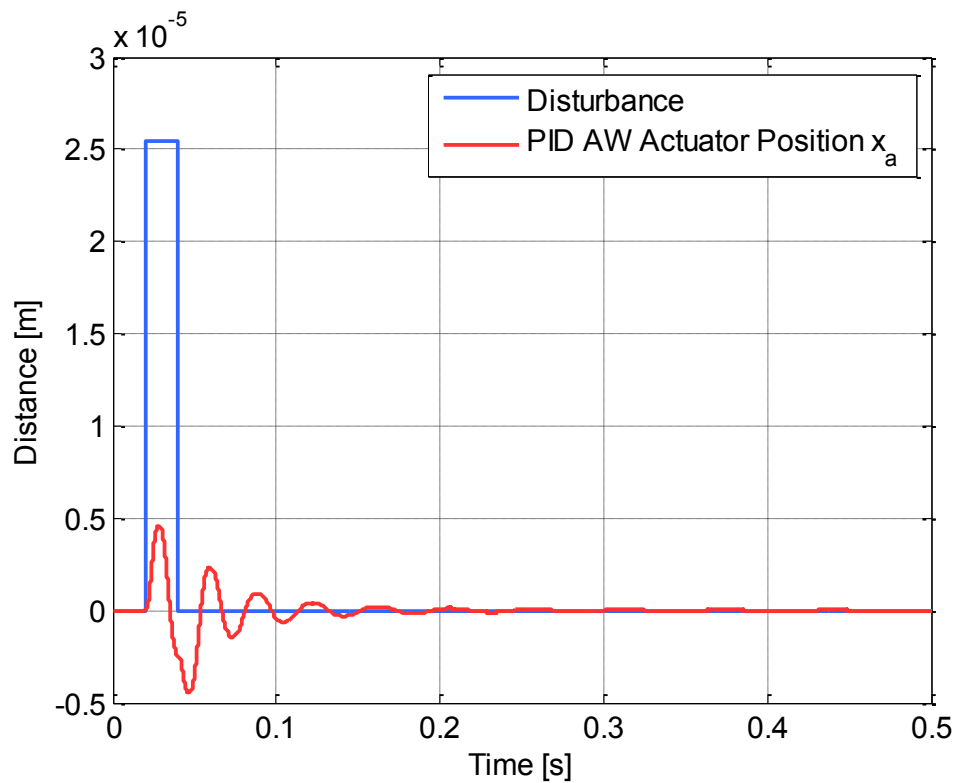


Fig. 4.6: Disturbance Rejection for PID with Anti-Windup Wide Pulse

With the wider pulse disturbance, Fig. 4.6, the settling time becomes 0.293 [s].

The wider pulse still creates multiple oscillations before settling to the reference

input of zero. Additionally the peak value has decrease with the wider pulse disturbance.

4.3 Full State Feedback

4.3.1 Step Response and Input Signal Analysis

Full state feedback allows one to move the poles of a system to a better utilized position. This is utilized by using gains that are multiplied by each state variable, which have the ability to make a system perform better or possibly become stable.

The standard state space is defined as

$$\begin{aligned}\dot{x} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}\tag{4.6}$$

where x is the state variable matrix, u is the input matrix, and \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} are the matrices used to define the system of interest.

For simplification the \mathbf{D} term will be neglected from Eq. (4.6). With state feedback the input u is given by

$$\mathbf{u} = -\mathbf{K}\mathbf{x} = -[k_1 \ k_2 \ k_3 \ \dots \ k_n]\mathbf{x} \quad (4.7)$$

where k_i is the feedback gain for each state variable x required to move the poles of the system to a specific location.

With all this being said, the system must, first and foremost, be controllable if the state feedback system design is to be progressed. The controllability is determined to see if the controllability matrix has full rank where

$$\mathbf{C} = [\mathbf{B} \ \mathbf{A}\mathbf{B} \ \mathbf{A}^2\mathbf{B} \ \mathbf{A}^3\mathbf{B} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}] \quad (4.8)$$

is the controllability matrix \mathbf{C} , and n is the number of state variables. For full state feedback the observability matrix,

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} \quad (4.9)$$

where n is length of the \mathbf{A} matrix, must have full rank in order to utilize full state feedback. If this is not the case, then the states that are not observable can be estimated with a state estimator.

With both of these events fulfilled, the full state feedback can be defined in such a way to increase the response of a system (moving the poles further away from the origin), or moving the poles to a location to influence the characteristics of the response, such as the percent overshoot, rise time, damping, natural frequency,...

The feedback gains can be determined by using the MATLAB commands place or acker depending on the system.

The standard form for determining the desired closed loop poles μ_i with full state feedback is defined as,

$$|sI - A + BK| = \prod (s - \mu_i) \quad (4.10)$$

where \mathbf{K} is the full state feedback gain.

The block diagram state space representation of full state feedback is shown in Fig. 4.7 below.

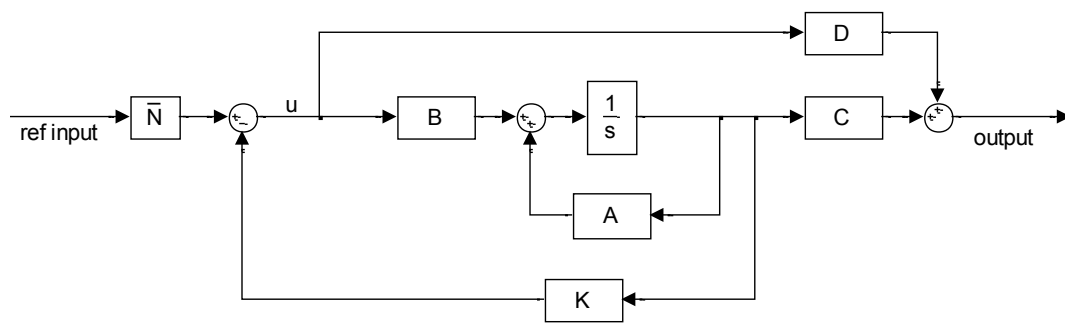


Fig. 4.7: Block Diagram Representation of Full State Feedback

\mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} are the matrices that define the system, \mathbf{K} is the full state feedback gain, and \bar{N} is the reference input gain. Due to the fact that the full state feedback is, in effect, moving the poles of the system and affecting the physical outcome of

the system, the reference input gain needs to be included. This is primarily used to accurately track the supplied reference input.

The reference input is defined as [24],

$$\bar{N} = N_u + \mathbf{K}N_x \quad (4.11)$$

where

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \quad (4.12)$$

The reference input gain produces a response with no steady state error.

The inverse matrix in Eq. (4.12) will from now on be referred to as \mathbf{W} . From the Cayle-Hamilton theorem, \mathbf{W} , has the characteristic equation of the form

$$\lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_1\lambda + a_0 = 0 \quad (4.13)$$

where a is the coefficient of the polynomial function determined from the roots which are the eigenvalues of \mathbf{W} . The inverse matrix can then be written and solved as

$$\mathbf{W}^{-1} = \left(\frac{-1}{a_0} \right) (\mathbf{W}^{n-1} + a_{n-1}\mathbf{W}^{n-2} + \dots + a_1\mathbf{I}) \quad (4.14)$$

This is useful for working between the scaled state space model and the full scale state space model.

Thus, Eq. (4.12) can then be rewritten as

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = W^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4.15)$$

A more complete explanation is shown in Appendix B.

Looking at the closed loop transfer function for the electrohydraulic actuator [4],

$$po8n(s) = \frac{num8}{den8} \quad (4.16)$$

where num8 is

$$num8 = -1.559e23s + 9.325e26 \quad (4.17)$$

and den8 is

$$\begin{aligned} den8 = & s^8 + 11090s^7 + 9.638e7s^6 + 5.464e11s^5 + 2.106e15s^4 \\ & + 5.972e18s^3 + 1.072e22s^2 + 5.166e24s \\ & + 9.325e26 \end{aligned} \quad (4.18)$$

this equation does have large terms on the order of 10^{27} that could potentially cause problems during simulation. This was fixed by using $s = 10000s'$ with a scaling factor of $sf=10000$, as explained in Appendix B. The scaled equation becomes,

$$po8n(s') = \frac{num8(s')}{den8(s')} \quad (4.19)$$

where num8 is

$$num8(s') = -155.9s' + 932.5 \quad (4.20)$$

and den8 is

$$\begin{aligned} den8(s') = & s'^8 + 11.09s'^7 + 96.38s'^6 + 546.4s'^5 + 2106s'^4 \\ & + 5972s'^3 + 1.072e4s'^2 + 5166s' + 932.5 \end{aligned} \quad (4.21)$$

The main issue with using Eq. (4.16), the non-scaled transfer function, is that errors will occur when converting to state space, and trying to determine the controllability and observability of the system. Using Eq. (4.19), the controllability and observability matrices can be determined to have full rank. This demonstrates that the system is fully stable and all the states are fully controllable and observable.

The full state feedback gain scaling can be determined by using Eq. (4.22) as shown below

$$K_{1,i} = K_{s1,i} sf^i \quad (4.22)$$

where i is the index of the feedback gain matrix starting with 1, K_s is the scaled feedback gain calculated with `acker` or `place` MATLAB function, and sf is the scaling factor used. The scaled feedback gain K_s is determined by determining

the ideal placement of the poles in the scaled system knowing the poles are scaled by the scaling factor originally chosen. A more detailed explanation can be found in Appendix B.

For instances where the states are not able to be sensed, a state estimator can be developed. The state estimator is shown below in Fig. 4.8

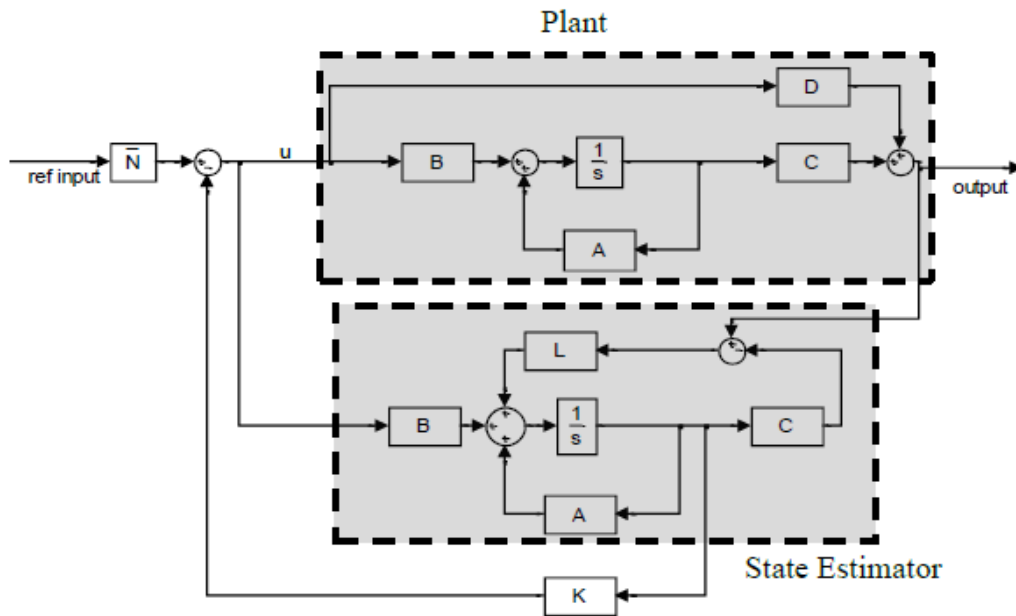


Fig. 4.8: Full State Feedback and State Estimator

To determine the state estimator gain, L , the error between the estimated state variable \hat{x} and x is used. \hat{x} is defined as the state variable estimate of x . The state estimator works by estimating the state based on the difference between the actual

plant output and the state estimator. The estimator gain is used to position the poles of the subsystem so that they are faster than the actual system.

Defining the state estimator as

$$\dot{\hat{x}} = \mathbf{A}\hat{x} + \mathbf{B}u + \mathbf{L}(y - \mathbf{C}\hat{x}) \quad (4.23)$$

The error equation becomes,

$$e = x - \hat{x} \quad (4.24)$$

taking the derivative with respect to time becomes,

$$\dot{e} = \dot{x} - \dot{\hat{x}} \quad (4.25)$$

Inserting the appropriate values,

$$\dot{e} = [\mathbf{A}x + \mathbf{B}u] - [\mathbf{A}\hat{x} + \mathbf{B}u + \mathbf{L}(y - \mathbf{C}\hat{x})] \quad (4.26)$$

After simplifying Eq. (4.26),

$$\dot{e} = (\mathbf{A} - \mathbf{LC})(x - \hat{x}) = (\mathbf{A} - \mathbf{LC})e \quad (4.27)$$

Equation (4.27) governs the error in the estimation. In order for the error to go to zero the characteristic equation,

$$|s\mathbf{I} - (\mathbf{A} - \mathbf{LC})| = 0 \quad (4.28)$$

needs to have all the roots or the eigenvalues of $\mathbf{A}-\mathbf{LC}$, in the left-plane. Typically, the poles (eigenvalues) are chosen to be three to five times faster than the closed loop poles of the plant resulting from the full state feedback. This allows the states to be estimated faster than the plant itself.

Additionally, due to the similarities between Eqs. (4.28) and (4.10) the MATLAB place or acker functions can be used with A' and C' in place of A and B , respectively, to calculate the state estimator gain L .

The state estimator gain can be calculated similarly to the full state feedback gain for scaled systems as,

$$L_{i,1} = L_{s,i,1} s f^{-(i-1)} \quad (4.29)$$

where L is the state estimator and L_s is the scaled state estimator. L_s is calculated from using the place or acker function in MATLAB with the transposed scaled A and C matrices. A more detailed explanation of how this equation was developed and an example can be found in Appendix B.

The full state feedback and estimator are designed in the scaled system. Knowing that the un-scaled poles will be faster by a factor of the scaling factor used, the system can be easily designed in the scaled state. Then, the full state feedback gain and estimator gain can be calculated. To make this work with the un-scaled system Eqs. (4.22) and (4.29) are then used to scale up the full state feedback gain and state estimator gain. Both of these equations have applications outside of this project and can be used under similar circumstances when numerical issues appear.

Implementing the above procedure for the po8n model, Eq. (4.16), using full state feedback, the modified poles are shown below in Table 4.3.

Table 4.3: Full State Feedback Pole Placement

<u>Original Full Scale Poles</u>	<u>Moved Full Scale Poles</u>
-1.068e3±5.658e3i	-1.068e3±5.658e3i
-6.057e2±3.939e3i	-7.057e2±3.939e3i
-3.570e3±1.290e3i	-3.570e3±1.291e3i
-3.013e2±1.791e2i	-6.813e2±1.091e2i

Ideally, if this was a 2nd order system one could look at a root locus plot that indicates the damping and natural frequency lines to achieve a designated response. In addition, if there was not an importance on efficiency the poles that were far away from the origin would be placed further away from the origin to reduce overall impact the poles have on the system. By moving the poles further away from real axis into the left half plane, the response of the poles will cause the system to become faster thus reducing the overall effect they have on the system [24]. As can be seen in Table 4.3 the poles are already very large. From the movement of the poles to the final location for the closed loop response, the full state feedback gain becomes,

$$\mathbf{K}=[960.0 \ 1.058e7 \ 8.864e10 \ 4.863e14 \ 1.762e18 \ 4.575e21 \ 7.620e24 \ 2.710e27]$$

as can be seen these are very large values.

As Eq. (4.16) shows there is no pure integrator in the closed loop transfer function. As is with the typical nature of integrators, the addition of an integrator to the system can help aid in better tracking of the system.

With Fig. 4.9 below, the design of the full state feedback with full state estimator can be shown combined with the nonlinear plant.

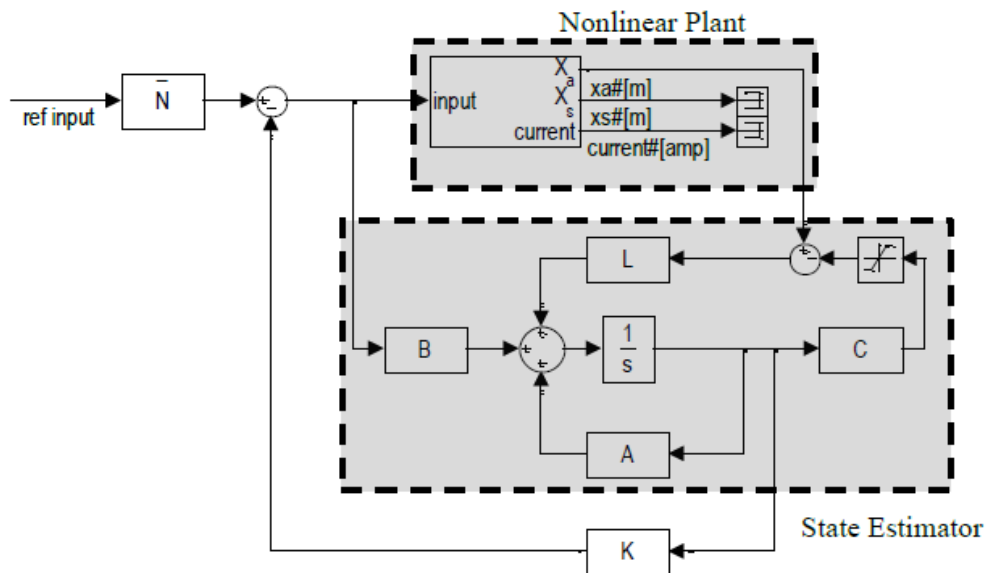


Fig. 4.9: Full State Feedback with Nonlinear Model

A saturation block was added to the output of the estimator, to eliminate the chance that the estimator would produce an actuator position that was larger than the limits of the model specified earlier.

After the full state feedback poles were tuned to the best possible placement, based on the nonlinear simulation, Table 4.4 and Table 4.5 were produced. There were some simulation instances, which prevented some pole placements from being tried due to numerical simulation errors. The best pole location was also used for the estimator gain calculation, however, the poles were moved by a factor of three.

Table 4.4: Full State Feedback Current Comparison

Controller Type	Current Area [A-s]		Max Current [A]	
	Small step	Large step	Small step	Large step
Plant	1.48e-6	10.0e-6	2.54e-4	2.5e-3
PID with Anti-windup	3.81e-6	35.1e-6	89.0e-4	40.0e-3
Full State Feedback	1.49e-6	6.69e-6	9.922e-4	9.900e-3

From Table 4.4 the current area is almost equivalent to the system without a controller for the smaller step size. However, comparing the current area with the PID controller design, the input current is smaller with the full state feedback for all estimator pole locations. This primarily comes from the fact that, the reference input gain is used to remove the steady state error. Additionally, at the 2.54e-3 step the current is reduced even more, to the point, where it is less than the open loop or standard plant design.

Table 4.5: Full State Feedback Actuator Position Comparison

Controller Type	Actuator Percent Overshoot [%]		Actuator Settling Time [s]	
	Small step	Large step	Small step	Large step
Plant	0.144	8.615	13.5e-3	17.2e-3
PID with Anti-windup	22.170	87.598	45.9e-3	7.5e-3
Full State Feedback	1.250	0.256	8.5e-3	5.9e-3

Table 4.5 shows that the full state feedback method offers much better response times than the standard plant. Additionally, the percent overshoot is decreased dramatically from the standard plant.

The plots of the spool and actuator to the step reference input is shown below as Fig. 4.10 and Fig. 4.11. Ideally, if the numerical model could simulate the response, the spool response would be increased to try and induce a faster response time. This would also allow the circular port on the spool to be opened further, thus, allowing the efficiency of the flow of oil to increase.

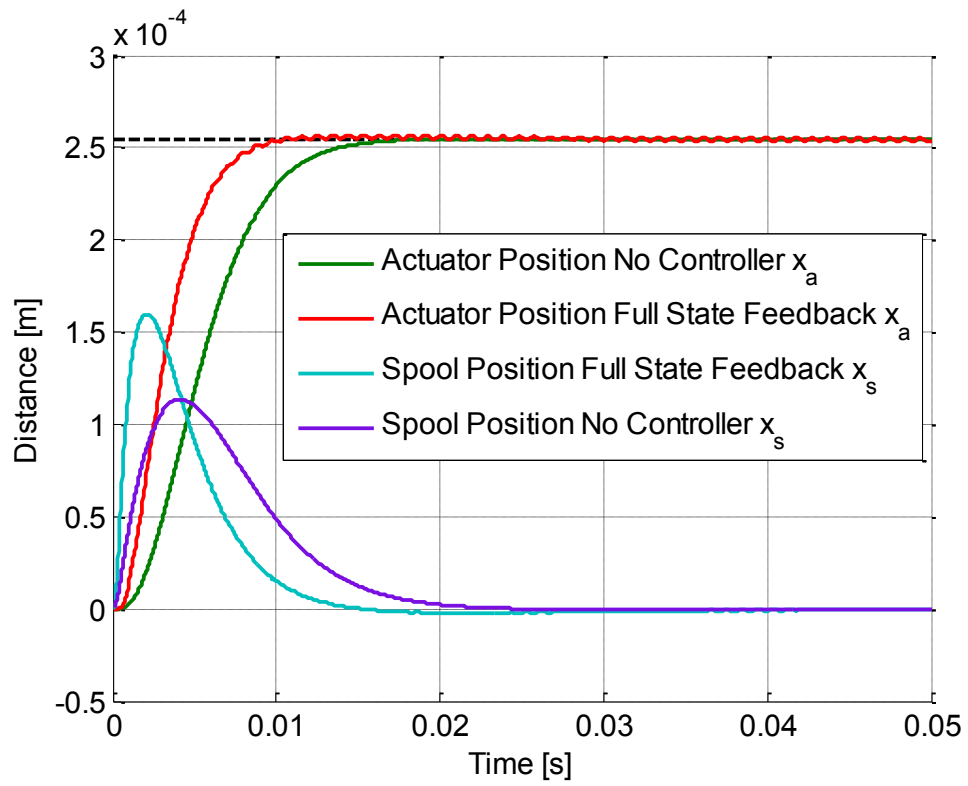


Fig. 4.10: Step Response Characteristics for Full State Feedback and Estimator at 2.54×10^{-4} [m]

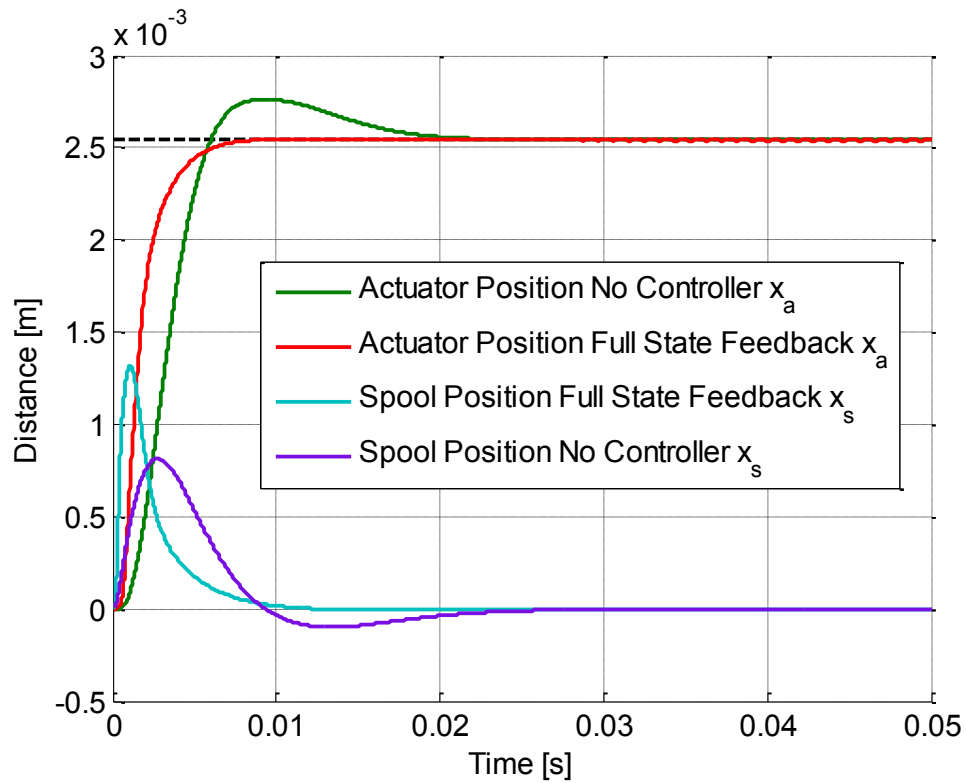


Fig. 4.11: Step Response Characteristics for Full State Feedback and Estimator at 2.54×10^{-3} [m]

From Fig. 4.10 and Fig. 4.11, the spool response time is increased along with the maximum spool movement. Both of these aspects allow the actuator to move faster and as can be seen provide minimal overshoot.

4.3.2 Disturbance Rejection

The disturbance rejection is again checked for the full state feedback controller, and the response is shown below as Fig. 4.12. The same pulse response is used for comparison between the different controllers.

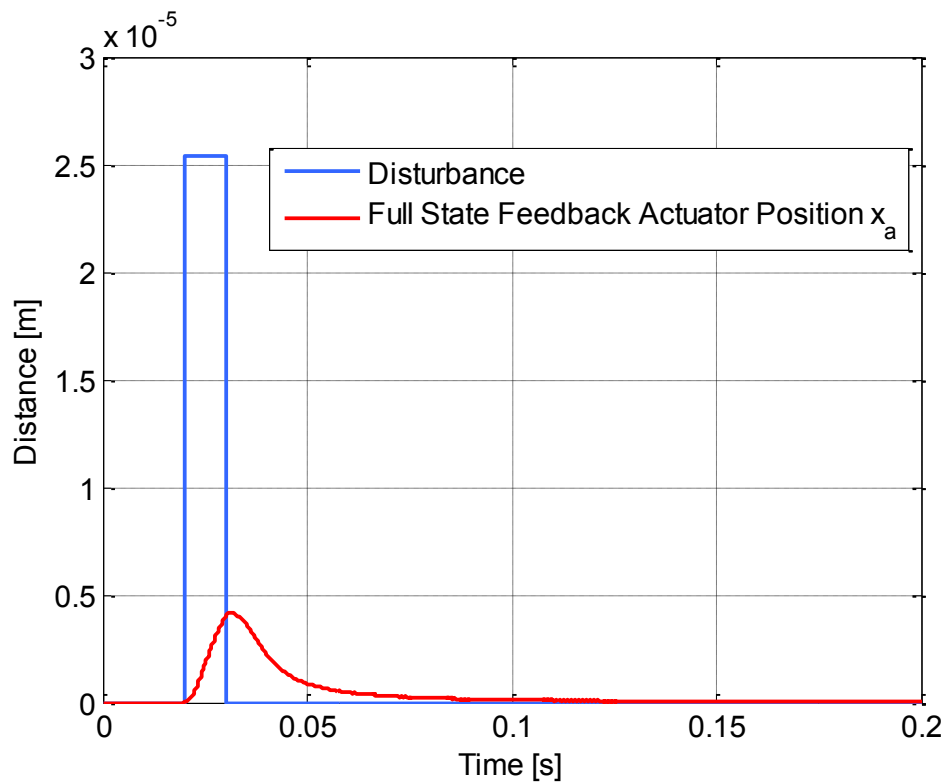


Fig. 4.12: Disturbance Rejection for Full State Feedback Narrow Pulse

From the figure, the disturbance does not cause any large oscillations, additionally the settling time is approximately 0.152 [s]. The wide pulse disturbance is shown in Fig. 4.13.

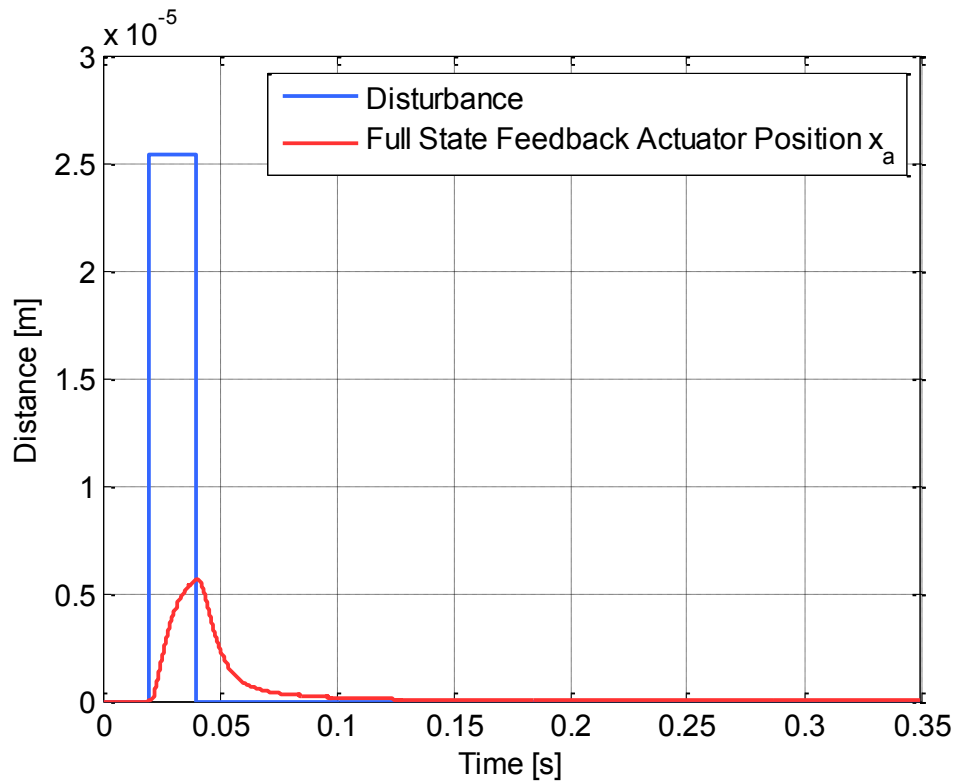


Fig. 4.13: Disturbance Rejection for Full State Feedback Wide Pulse

With the wider pulse disturbance the settling time increased to 0.154 [s]. Additionally the peak value increase slightly due to the wider pulse disturbance. Both disturbances cause similar responses.

4.4 PI-Full State Feedback

4.4.1 Step Response and Input Signal Analysis

With the success of the full state feedback controller, the next logical step to improve the tracking, is by using an augmented PI-full state feedback model. The controller design is shown below in Fig. 4.14.

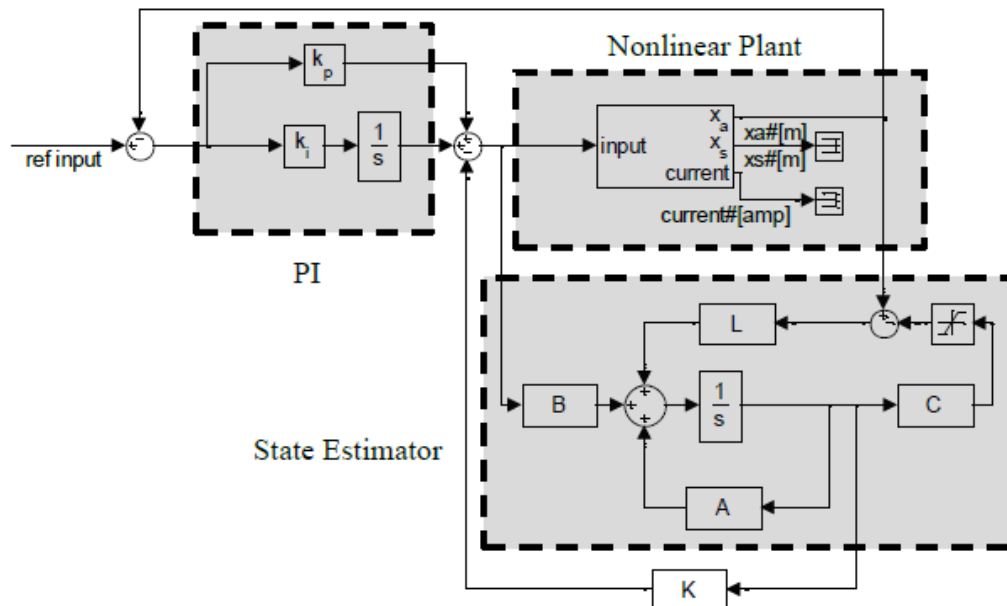


Fig. 4.14: PI-Full State Feedback with Estimator

Jelali and Kroll [2] developed this augmented PI-state feedback system. Jelali and Kroll presented two tuning conditions,

$$K_p = -(CA^{-1}B)^{-1} \quad (4.30)$$

and

$$K = K_I - K_p C \quad (4.31)$$

This eliminates the need for determining the pole placement. The placement of the poles becomes dependent on the only tuning variable, K_I .

The state estimator design is calculated along the same basis as with the full state feedback design. First, the equations from the block diagram are developed where,

$$\bar{u} = K_p u + \frac{K_I}{s} u + K \hat{x} \quad (4.32)$$

and defining

$$y = Cx \quad (4.33)$$

$$\dot{x} = Ax + B\bar{u} \quad (4.34)$$

$$\hat{y} = C\hat{x} \quad (4.35)$$

$$\dot{\hat{x}} = A\hat{x} + B\bar{u} + L(y - \hat{y}) \quad (4.36)$$

Inserting and simplifying

$$\dot{x} = Ax + B[K_p u + \frac{K_I}{s} u + K\hat{x}] \quad (4.37)$$

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\left[K_p u + \frac{K_I}{s}u + \mathbf{K}\hat{\mathbf{x}}\right] + \mathbf{L}(y - \hat{y}) \quad (4.38)$$

Since the error estimate needs to go to zero as quickly as possible, the error equation is defined as

$$\dot{e} = \dot{x} - \dot{\hat{x}} \quad (4.39)$$

Inserting the appropriate values,

$$\begin{aligned} \dot{e} = \mathbf{A}\mathbf{x} + \mathbf{B}\left[K_p u + \frac{K_I}{s}u + \mathbf{K}\hat{\mathbf{x}}\right] - \mathbf{A}\hat{\mathbf{x}} + \\ \mathbf{B}\left[K_p u + \frac{K_I}{s}u + \mathbf{K}\hat{\mathbf{x}}\right] + \mathbf{L}(y - \hat{y}) \end{aligned} \quad (4.40)$$

After simplifying

$$\dot{e} = (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{x} - (\mathbf{A} - \mathbf{L}\mathbf{C})\hat{\mathbf{x}} \quad (4.41)$$

$$\dot{e} = (\mathbf{A} - \mathbf{L}\mathbf{C})(\mathbf{x} - \hat{\mathbf{x}}) \quad (4.42)$$

This is the same equation used in the full state feedback example for determining the state estimator gain. As such, there are not any additional values that need to be taken into account during the calculation of the state estimator gain \mathbf{L} . The pole locations, after taking into account the state feedback gain, can be calculated by looking at the eigenvalues of the $\mathbf{A}-\mathbf{BK}$ matrix. The stipulation on the state estimator is that the poles of the state estimator need to be faster than the poles of the closed loop system, to allow accurate quick estimation of the state variables. During the calculation of the estimator poles based on the full state feedback gain, both the acker and place functions were unable to accurately produce an estimator

gain with poles that were three times faster than the closed loop poles. Due to the estimator poles only needing to be faster than the closed loop poles of the system, the estimator poles were then chosen to be five times faster than the original stable pole locations. This satisfied the criteria that the estimator poles must be faster than the closed loop poles.

From the tuning conditions the proportional gain, K_p , calculated to be 1, and tuning the integral gain, K_i , the best gain was $\frac{1}{4}$. This resulted in the poles being moved to the location shown in Table 4.6.

Table 4.6: PI-Full State Feedback Pole Locations

<u>Original Full Scale Poles</u>	<u>Moved Full Scale Poles</u>
-1.068e3±5.658e3i	-1.066e3±5.664e3i
-6.057e2±3.939e3i	-5.950e2±3.918e3i
-3.570e3±1.290e3i	-3.592e3±1.267e3i
-3.013e2+1.791e2i	-7.295e2
-3.013e2-1.791e2i	0

From these pole locations the following step input responses were simulated at $2.54e-4$ [m] and $2.54e-3$ [m]. Both are shown below as Fig. 4.15 and Fig. 4.16.

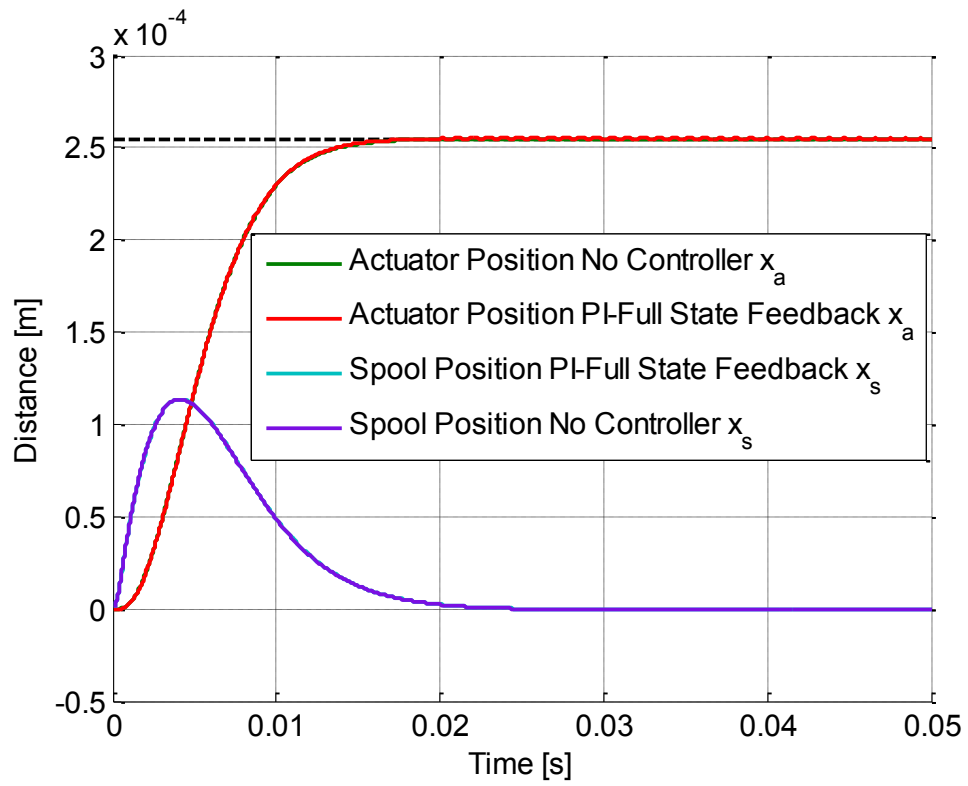


Fig. 4.15: PI-Full State Feedback for Step of 2.54e-4 [m]

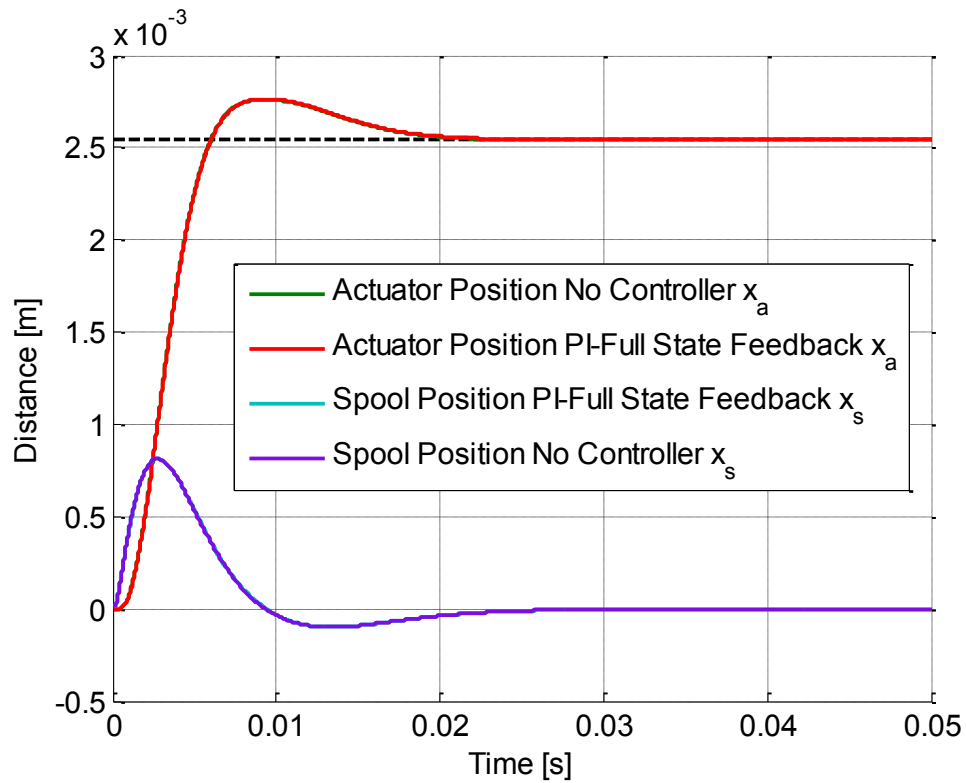


Fig. 4.16: PI-Full State Feedback for Step of $2.54e-3$ [m]

From both Fig. 4.15 and Fig. 4.16 it can be seen that the PI-Full state feedback is nearly identical to the standard plant model. With this criteria, the comparison between the controller design and the open loop/no controller design is shown below as Table 4.7 and Table 4.8.

Table 4.7: PI-Full State Feedback Current Comparison

Controller Type	Current Area [A-s]		Max Current [A]	
	Small step	Large step	Small step	Large step
Plant	1.48e-6	10.0e-6	2.54e-4	2.5e-3
PID with Anti-windup	3.81e-6	35.1e-6	89.0e-4	40.0e-3
Full State Feedback	1.49e-6	6.69e-6	9.922e-4	9.900e-3
PI-Full State Feedback	1.48e-6	10.00e-6	2.54e-4	2.5e-3

Table 4.8: PI-Full State Feedback Actuator Position Comparison

Controller Type	Actuator Percent Overshoot [%]		Actuator Settling Time [s]	
	Small step	Large step	Small step	Large step
Plant	0.144	8.615	13.5e-3	17.2e-3
PID with Anti-windup	22.170	87.598	45.9e-3	7.5e-3
Full State Feedback	1.250	0.256	8.5e-3	59.0e-3
PI-Full State Feedback	0.179	8.644	13.5e-3	17.2e-3

From the tables, it can be seen that the PI-full state feedback, with the aforementioned tuning condition, does not produce results that, in any way, increase the response time or decrease the input current. Additionally, the use of anti-windup is not necessary in this case because the integrator signal does not produce a large enough signal that would warrant the use of an anti-windup scheme.

4.4.2 Disturbance Rejection

The disturbance rejection is again analyzed for the final controller design, the PI-full state feedback. The actuator position response is shown below in Fig. 4.17.

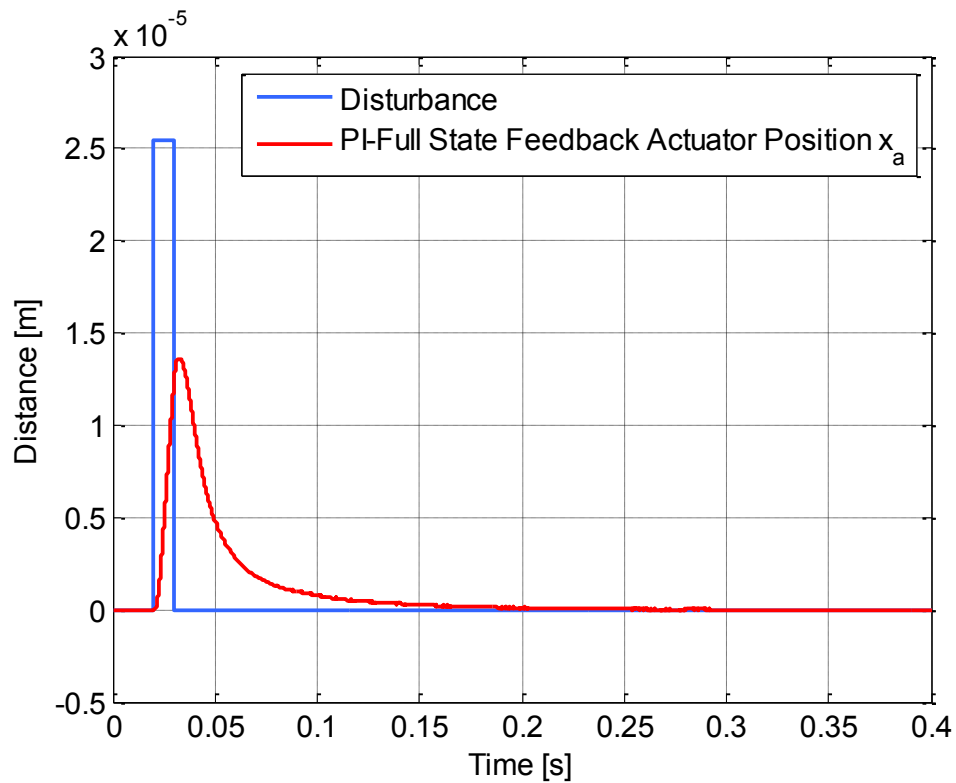


Fig. 4.17: Disturbance Rejection for PI-Full State Feedback Narrow Pulse

From Fig. 4.17, the actuator response has a large peak value and subsequently a long settling time of approximately 0.219 [s]. The wider pulse disturbance is shown in Fig. 4.18.

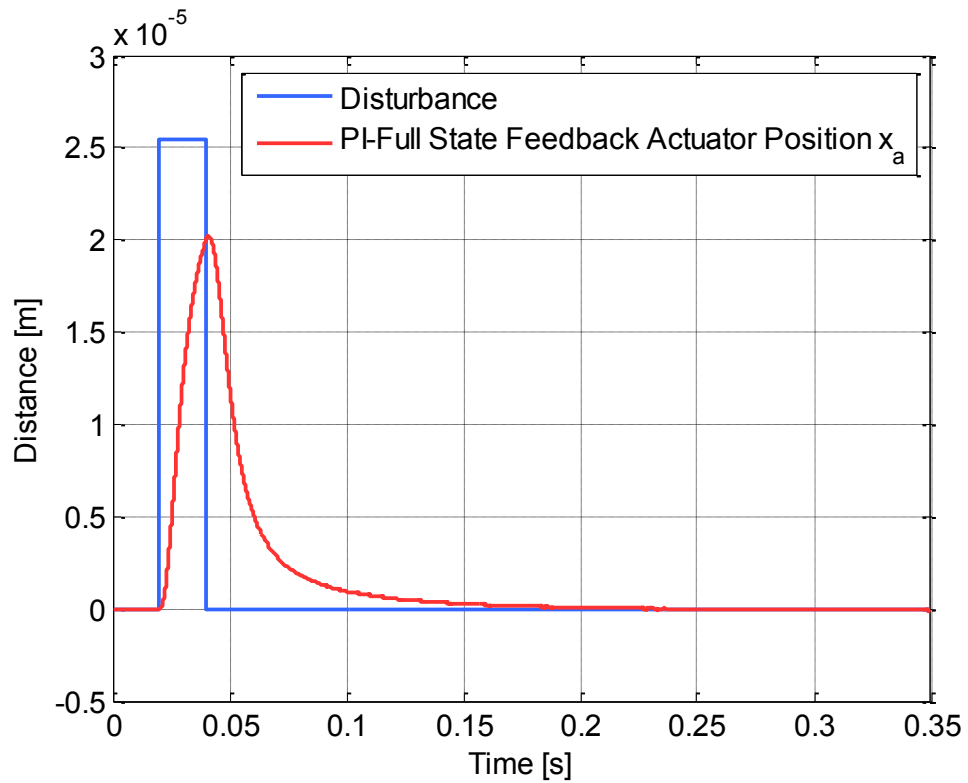


Fig. 4.18: Disturbance Rejection for PI-Full State Feedback Wide Pulse

The actuator position for the wide pulse disturbance produces a larger peak value however the settling time decrease to 0.197 [s]. Additionally the response is similar for both disturbance lengths.

4.5 Controller Comparison

As discussed above, all the controller designs have pros and cons. The PID and PI-full state feedback have the additional advantage of having better tracking ability due to the addition of the integrator term. This, however, is based on the responses causes a larger current area. Presented again below, Table 4.9 and Table 4.10 show the comparison of the different controllers versus the open loop or standard plant design.

Table 4.9: Current Comparison of Controller Designs

Controller Type	Current Area [A-s]		Max Current [A]	
	Small step	Large step	Small step	Large step
Plant	1.48e-6	10.0e-6	2.54e-4	2.5e-3
PID with Anti-windup	3.81e-6	35.1e-6	89.0e-4	40.0e-3
Full State Feedback	1.49e-6	6.69e-6	9.922e-4	9.900e-3
PI-Full State Feedback	1.48e-6	10.00e-6	2.54e-4	2.5e-3

From Table 4.9 it can be seen that in terms of current area, the full state feedback design is the smallest for the 2.54e-3 [m] step input, but is only slightly larger than the open loop or standard plant current area for the 2.54e-4 [m] step.

Table 4.10: Actuator Comparison of Controller Designs

Controller Type	Actuator Percent Overshoot [%]		Actuator Settling Time [s]	
	Small step	Large step	Small step	Large step
Plant	0.144	8.615	13.5e-3	17.2e-3
PID with Anti-windup	22.170	87.598	45.9e-3	7.5e-3
Full State Feedback	1.250	0.256	8.5e-3	59.0e-3
PI-Full State Feedback	0.179	8.644	13.5e-3	17.2e-3

In addition, Table 4.10 shows the full state feedback produces the smallest percent overshoot and settling time. This is true for all cases except for the percent overshoot with a step of $2.54e-4$ [m], which has a percent overshoot of 1.25%. Overall the full state feedback provides the best mix of response time and current input reduction. The only downside to the full state feedback is that the controller has difficulty tracking any inputs other than a step input. Additionally if the filter gain \bar{N} is not used there will be a steady state error associated with the actuator position.

Of the three different controllers tested, the step responses are shown below for the step input response of $2.54e-4$ [m], Fig. 4.19. At this step response, the nonlinear model still exhibits the linear response from the po8n transfer function.

This provides a good starting point to determine whether or not the controller designs will be feasible with the nonlinear model.

From Fig. 4.19, it can be seen that the PID with anti-windup offers the quickest rise time, however, the percent overshoot is the largest of all the controller designs. The full state feedback controller offers the next quickest rise time and a very small percent overshoot. Finally, the PI-full state feedback controller, closely follows the standard nonlinear model.

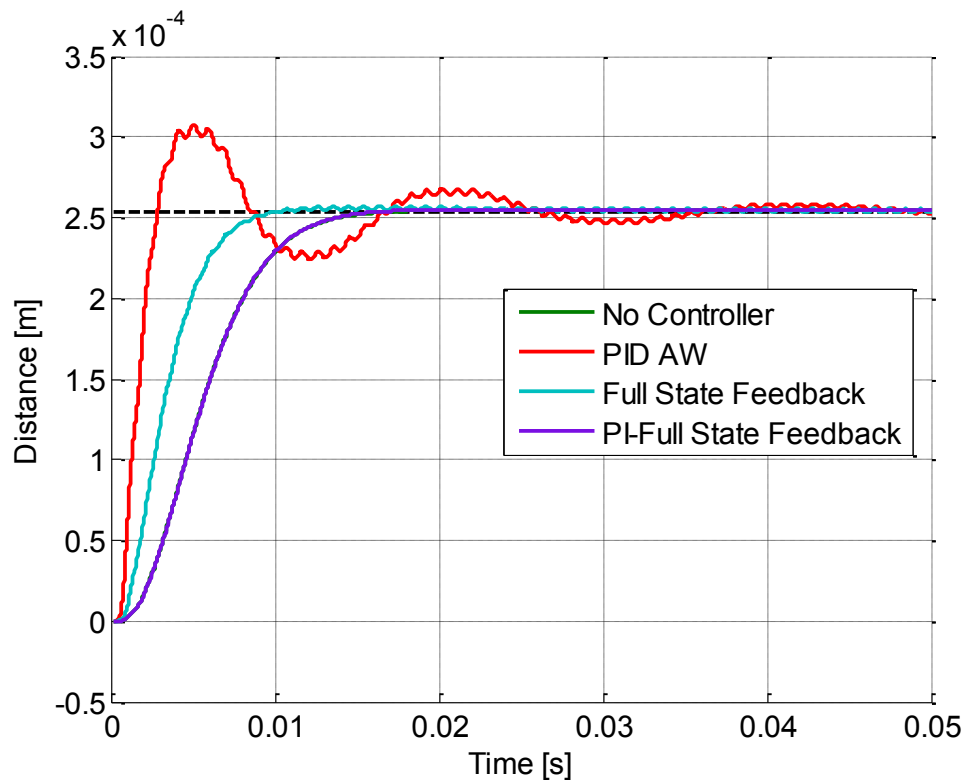


Fig. 4.19: Controller Comparison Actuator Response at $2.54e-4$ [m] Step

The step response was also compared at $2.54e-3$ [m], as shown in Fig. 4.20. The PID with anti-windup at $2.54e-3$ [m] provides an overshoot that is larger than any other controller design tested. This unsatisfactory behavior necessitates looking into other controller designs.

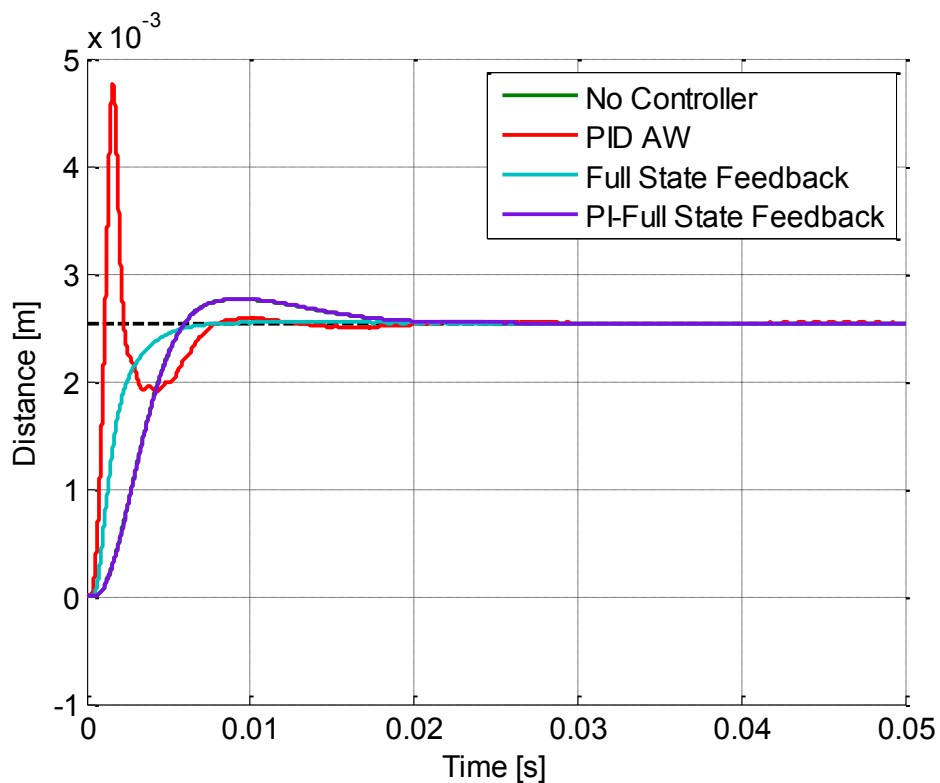


Fig. 4.20: Controller Comparison Actuator Response at $2.54e-3$ [m] Step

The full state feedback controller continues to offer the second fastest rise time and also has the lowest percent overshoot of all the controller designs. The PI-full state feedback continues to follow the nonlinear model. As can be seen, the

change from the linear region to the nonlinear region shows that the full state feedback offers the best response time, even with the existence of nonlinearities not modeled by the po8n model.

In addition to the step response comparison, the disturbance response is also compared in Fig. 4.21 for the narrow pulse disturbance.

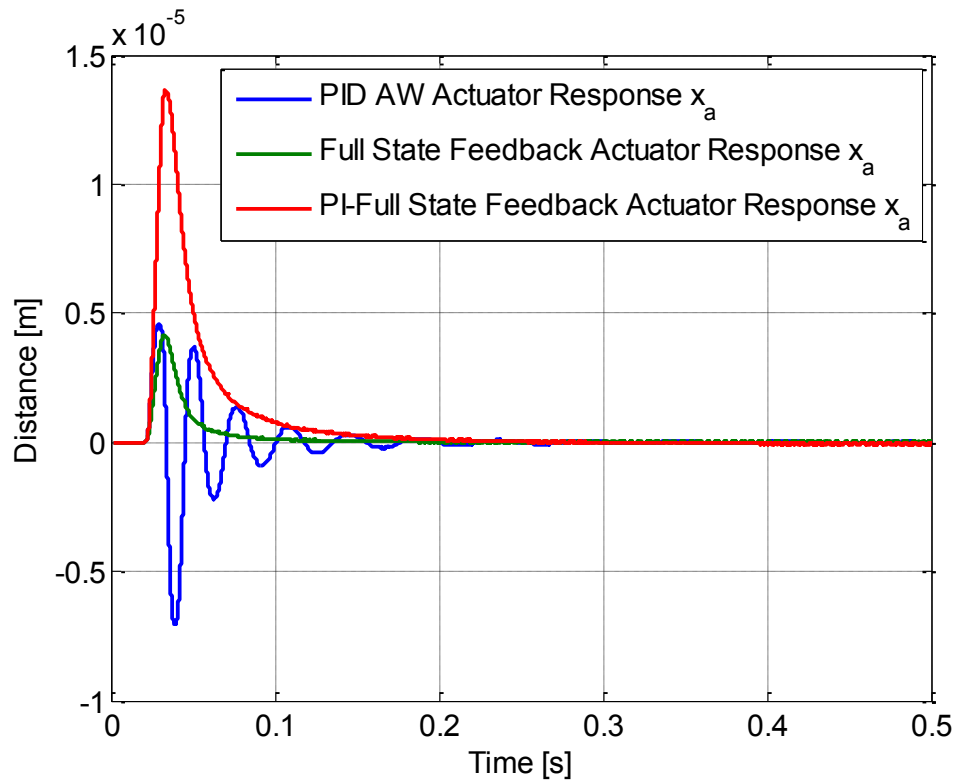


Fig. 4.21: Disturbance Rejection Responses Narrow Pulse

From Fig. 4.21 it can be seen that the full state feedback offers the best disturbance rejection due to the fact that the position signal reaches the reference

input the fastest and also has the smallest peak value. Finally there is not any oscillation present in the full state feedback as compared to the PID with anti-windup response. The actuator position response for the wide pulse disturbance is shown in Fig. 4.22.

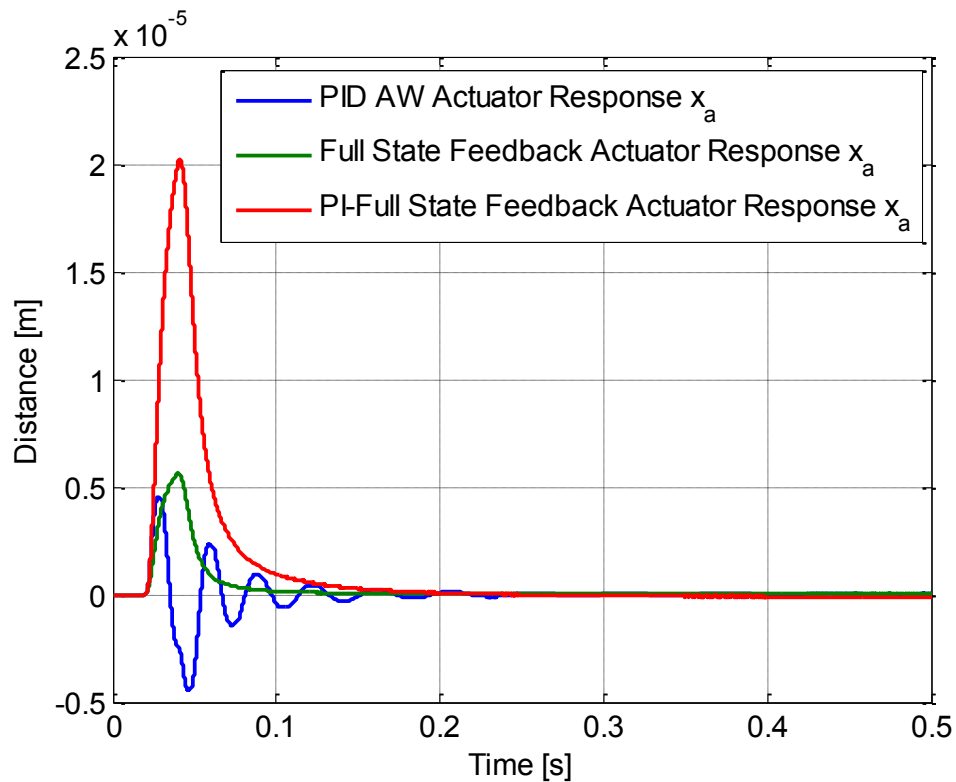


Fig. 4.22: Disturbance Rejection Responses Wide Pulse

From the figure, the PID controller has the largest amount of oscillations but has much smaller max amplitude than the PI-full state feedback. The PI-full state feedback, however, has the largest max amplitude. The full state feedback has the smallest max amplitude and also has the lowest settling time.

The two previous plots have been characterized by Table 4.11 to easily analyze the responses according to controller type.

Table 4.11: Controller Comparison for Disturbance Rejection

Controller Type	Max Abs. Value [m]		Actuator Settling Time [s]		Abs. Error	
	narrow	wide	narrow	wide	narrow	wide
PID with Anti-windup	7.033e-6	4.556e-6	0.263	0.293	19.0e-8	1.68e-7
Full State Feedback	4.165e-6	5.534e-6	0.152	0.154	9.32e-8	1.51e-7
PI-Full State Feedback	13.65e-6	20.20e-6	0.219	0.197	38.5e-8	6.18e-7

The max absolute value in the table refers to the largest absolute peak observed in the responses. The actuator settling time refers to the amount of time the actuator takes to return to the reference input in relation to the beginning of the simulation. Due to the fact that all the disturbances started at the same time and only the duration was varied allows easy comparison across each controller design. The absolute error is equal to the area under the actuator position curve where all points are assumed positive. This prevents any negative area from being added to the total area. This is also a way to take into account the oscillations in the responses. From the table, the best controller for removing the disturbance is the full state feedback controller. The full state feedback produces the smallest settling time and absolute error. The full state feedback also has the smallest max

value with the exception of the wide pulse disturbance, however, even then it is only slightly larger than the smallest. All of these characteristics mentioned reinforce the fact that the full state feedback is the best controller for rejecting the disturbances analyzed. Overall this makes the full state feedback the best design for the nonlinear nature of the model and also at rejecting disturbance inputs as shown above.

Chapter 5 Conclusions and Recommendations

5.1 Chapter Overview

This chapter discusses the end results of this thesis along with suggestions on areas of improvement for continuation of this work. The difficulties in modeling the system along with the model justification, controller designs, and implementation are also discussed.

5.2 Simulink Modeling

The main goal of this thesis was to produce a working model of an electrohydraulic actuator. There were many complications that arose during the creation of the Simulink model. The primary issue was accurately representing all the equations, explained in Chapter 2, and also determining values for certain variables that were not available. This made it difficult, especially, having to deal with both unknown values and debugging the Simulink model at the same time. The model itself came with its fair share of problems. As explained in Section 3.4, the hydraulic system modeled was also a nonlinear and stiff system. This made simulating the model difficult, especially, in light of the previous statements.

Parameter estimation was used to determine accurate values for the unknown values. The parameter estimation was conducted against the known step responses of the po8n transfer function. Additionally, the frequency response was compared for both the magnitude and phase of different parts of the system. This is discussed in Section 3.3.1. These justification methods and the comparison against literature values give validation to the model presented in this thesis. The way the system was designed also allows an individual to tune the parameters to

match a similar electrohydraulic actuator. As explained in the references this level of detail and method of modeling in Simulink is not usually accomplished due to the degree of difficulty. Based on the time response and frequency response justification, it was shown that each subsection of the model matched their respective parts. This also shows that the servovalve system can be described using nonlinear equations, rather than ignoring or approximating with a second or third order transfer function, effectively.

5.3 Controller Designs

Three different controller designs were designed and tested with the Simulink model. A PID controller with anti-windup, full state feedback with state estimator, and a hybrid PI full state feedback were tested. The ability to reject a pulse disturbance was also analyzed. All three models were tested; the best controller that both decreased the current area and increased the response time was the full state feedback controller. The full state feedback also performed the best at rejecting the disturbance. The idea to introduce a proportional and integral term (PI) to the controller to provide better tracking was the reason behind using the PI full state feedback controller. The current area was used as a way to gauge the efficiency of the controller. While both the PID and hybrid PI full state

feedback should be better for tracking and provide better response for disturbances, it was found that these models did not help increase the response time. The full state feedback model did increase the response time. The drawback to using full state feedback is the lack of tracking with inputs other than step inputs.

With the order of the transfer function and resulting state space representation, special scaled equations had to be developed for the full state feedback and state estimator gain. This was developed out a matter of necessity and to allow simulation of the state space control methods discussed a priori. This also has applications not just for this thesis but for other instances where there is a large number of poles, large magnitude poles, or both and pole placement needs to be accomplished. This can help solve the numerical issues that can arise by way of traditional pole placement algorithms

5.4 Recommendations

5.4.1 Simulation

As briefly mentioned, throughout this paper simulating the response was a major obstacle that had to be overcome with this research. There still are many issues with the solver and the way model was implemented. The solver had difficulty solving very fast systems, partially, due to the saturation limits and the conditional statements used, which can produce zero crossing detection problems. Future areas of study could consist of improving the solver and zero crossing detection and, additionally, finding more efficient ways of modeling the nonlinear system.

This caused issues with the type of controllers that could be tested with the nonlinear model. Additionally, there were limitations that prevented certain controller parameters from being analyzed. Ideally, the nonlinear model should have been able to produce responses that were not favorable, rather than receiving an error message. This also prevented an analysis of the digital controllers.

Finally due to the lack of experimental data, experimental analysis could be performed to provide further data for the Simulink model. By doing this the responses could be verified at larger step inputs and provide more justification for the model.

5.4.2 Controllers

As mentioned a more detailed analysis could be performed on the digital versions of the controllers mentioned above. With that being said, the nonlinear model would need to be improved upon to work better and be more reliable. Controller designs that could be tested include H_∞ controllers and other more robust controller designs, more adept at nonlinearities and modeling uncertainties. Additionally, different type of digital estimators could be analyzed for the input energy such as a Kalman estimator and a current estimator. Alternatively, an adaptive PID controller could be designed for use with a digital analysis. All of these controllers could be tested out on the nonlinear model, if the nonlinear model reliability was increased.

5.5 Conclusion

In conclusion, this thesis has demonstrated the modeling and justification of an electrohydraulic actuator. All aspects of the electrohydraulic actuator have been modeled in Simulink utilizing only standard Simulink blocks. This allows an individual to look at any value of interest during the simulation of the model. Additionally, all of the original equations developed were used in Simulink without linearizing. This allows an individual to notice the nonlinearities that are not available by linearized models or transfer functions for future areas of study.

Table 5.1 and Table 5.2 show the comparison of the different controller designs. From the tables it can be seen that the regular full state feedback design offered the best qualities out of all the controller designs.

Table 5.1: Current Comparison of Controller Designs

Controller Type	Current Area [A-s]		Max Current [A]	
	step 2.54e-4 [m]	step 2.54e-3 [m]	step 2.54e-4 [m]	step 2.54e-3 [m]
PID with Anti-windup	3.81e-6	3.51e-5	0.0089	0.04
Full State Feedback	1.49e-6	6.69e-6	9.922e-4	9.900e-3
PI-Full State Feedback	1.48e-6	1.00e-5	2.54e-4	2.5e-3
Plant	1.48e-6	1.00e-5	2.54e-4	2.5e-3

Table 5.2: Actuator Comparison of Controller Designs

Controller Type	Actuator Percent Overshoot [%]		Actuator Settling Time [s]	
	step 2.54e-4 [m]	step 2.54e-3 [m]	step 2.54e-4 [m]	step 2.54e-3 [m]
PID with Anti-windup	22.1695	87.5978	0.0459	0.0075
Full State Feedback	1.2503	0.2555	0.0085	0.0059
PI-Full State Feedback	0.1793	8.6441	0.0135	0.0172
Plant	0.1437	8.6149	0.0135	0.0172

The full state feedback controller offers the best settling time and also decreases the current area, as compared to, the plant or open loop design. The full state feedback controller does, however, have a larger max current than the open loop design; however it is less than the PID controller. Overall, the full state feedback offers the best response time by providing the smallest overshoot and also smallest settling time. The full state feedback also offers the smallest current area for any of the controller designs. The full state feedback, based on the disturbance analysis, also rejected the pulse disturbance the quickest of all the controllers mentioned. Finally, the full state feedback is still able to work appropriately even at larger step inputs where nonlinearity is more of an issue, which is evident by the robustness that the controller exhibits.

Appendix A Nonlinear and Linear Step Responses

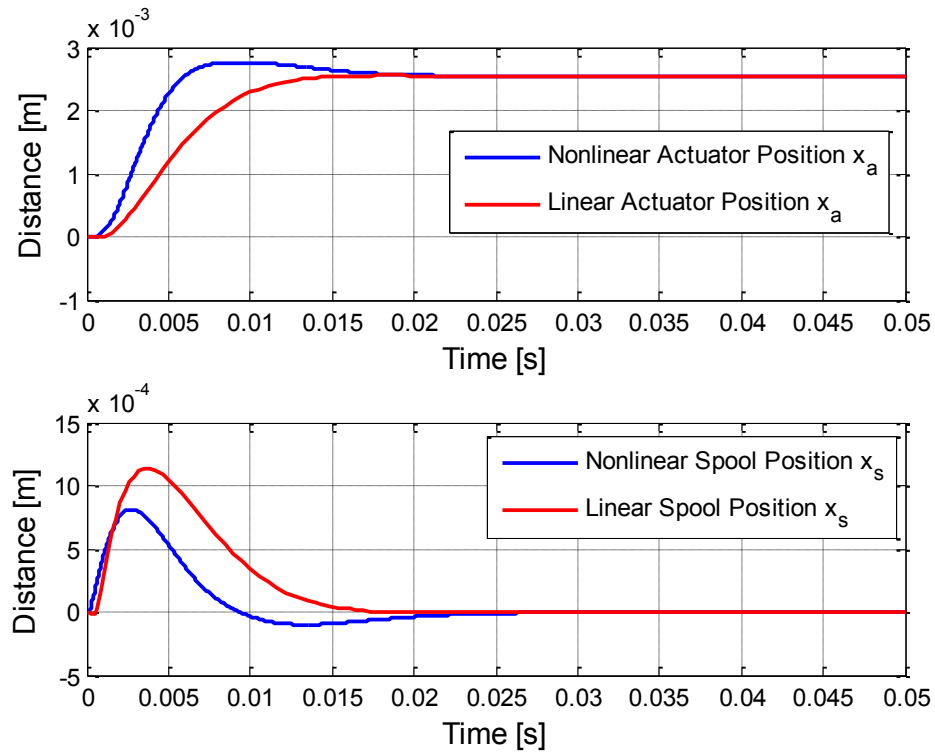


Fig. A.1: Step Input Response of $2.54e-3$ [m]

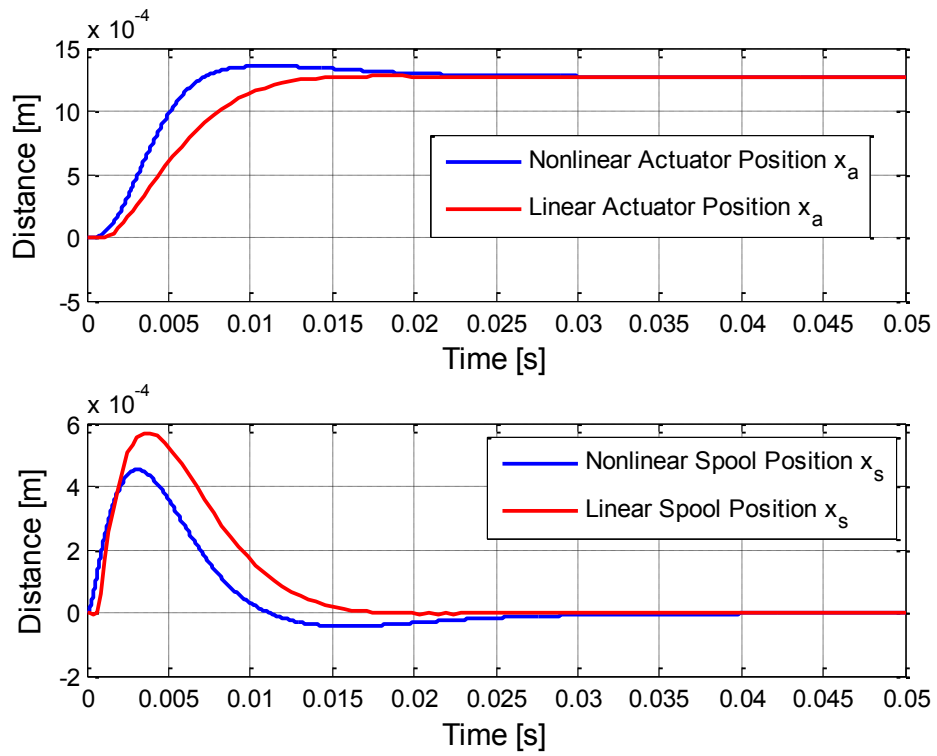


Fig. A.2: Step Input Response of $1.27\text{e-}3$ [m]

Appendix B Scaling factor and full state feedback

As mentioned before, full state feedback is a function utilized in the state space arrangement of a system to move the poles of a system to a better utilized position. Feedback gains are utilized to move poles to a more favorable location in the left half plane of a real imaginary plot.

Within the state space system

$$\begin{aligned}\dot{x} &= \mathbf{A}x + \mathbf{B}u \\ \mathbf{y} &= \mathbf{C}x + \mathbf{D}u\end{aligned}\tag{B.1}$$

where x is the state variable matrix, u is the input matrix, and \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} are the matrices used to define the system of interest.

For this explanation the \mathbf{D} term will be neglected from Eq. (B.1). With state feedback the input u is given by

$$\mathbf{u} = -\mathbf{k}\mathbf{x} = -[k_1 \ k_2 \ k_3 \ \dots \ k_n]\mathbf{x}\tag{B.2}$$

where k_i is the feedback gain for each state variable x required to move the poles of the system to a specific location.

The system must be controllable if the state feedback is going to be used. The controllability is dependent on if the controllability matrix has full row rank where

$$\mathcal{C} = [\mathbf{B} \ \mathbf{A}\mathbf{B} \ \mathbf{A}^2\mathbf{B} \ \mathbf{A}^3\mathbf{B} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}] \quad (\text{B.3})$$

is the controllability matrix \mathcal{C} , and n is the number of state variables. For full state feedback the observability matrix,

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} \quad (\text{B.4})$$

where n is length of the \mathbf{A} matrix, must have full column rank in order to utilize full state feedback. If this is not the case, then the states that are not observable can be estimated with a state estimator.

With both of these events fulfilled, the full state feedback can be defined in such a way to increase the response of a system (moving the poles further away from the origin), or moving the poles to a location to influence the characteristics of the response, such as the percent overshoot, rise time, damping, natural frequency, ... The feedback gains can be determined by using the MATLAB commands place or acker depending on the system.

During the linearization and creation of a transfer function to define a system there is a chance the transfer function will contain large values that will cause simulation errors. This can be remedied by using a scaling factor.

To begin assume, a transfer function exists which does not exhibit any numerical issues. For this example

$$H(s) = \frac{10s + 40}{s^2 + 200s + 10000} \quad (\text{B.5})$$

which has poles at -100, -100 and a zero at -4.

Equation (B.5) has a standard step input as shown below in Fig. B.1.

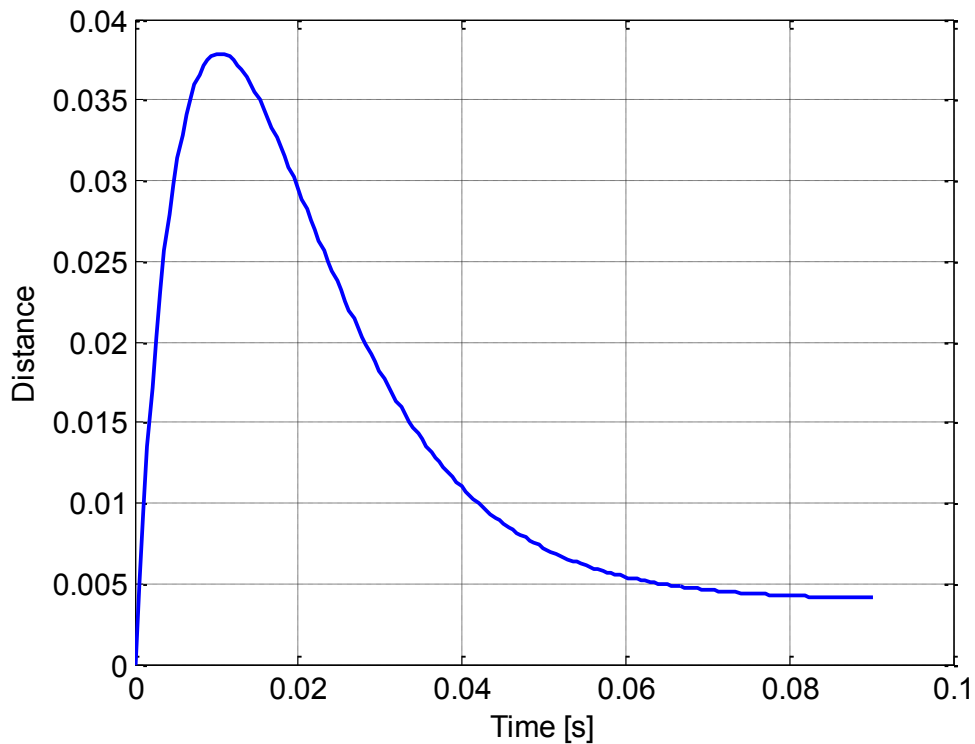


Fig. B.1: Step Response of H(s)

The bode plot of H(s) is shown below in Fig. B.2.

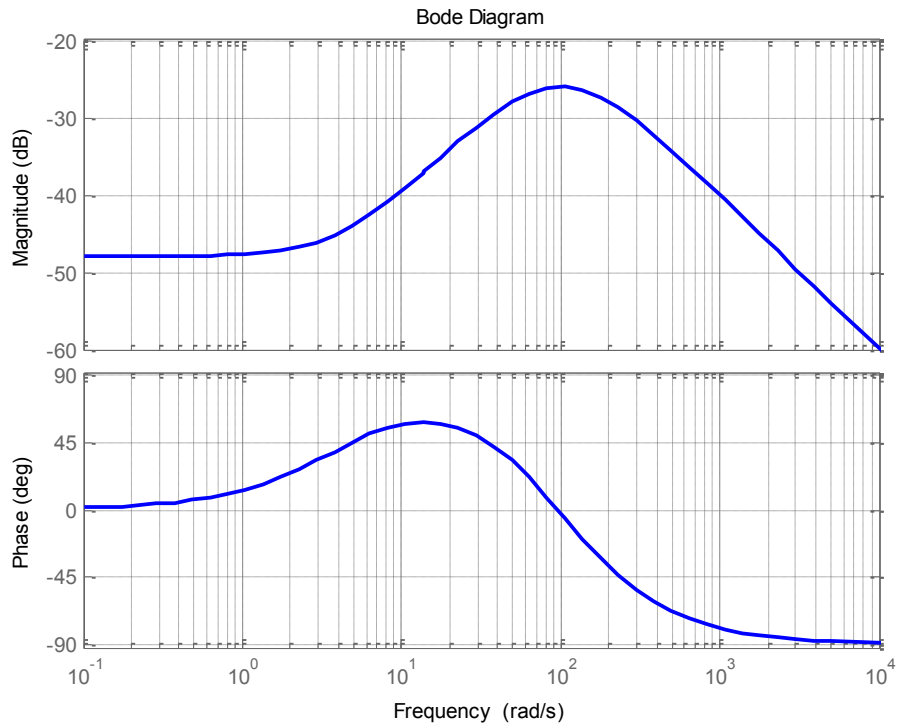


Fig. B.2: Bode Plot of H(s)

By introducing $s = 10s'$, where the scaling factor is 10, the scaled transfer function will become,

$$H(s') = \frac{s' + 0.4}{s'^2 + 20s' + 100} \quad (\text{B.6})$$

which has poles at -10, -10 and a zero at -0.4.

The step response of $H(s')$ is shown in Fig. B.3

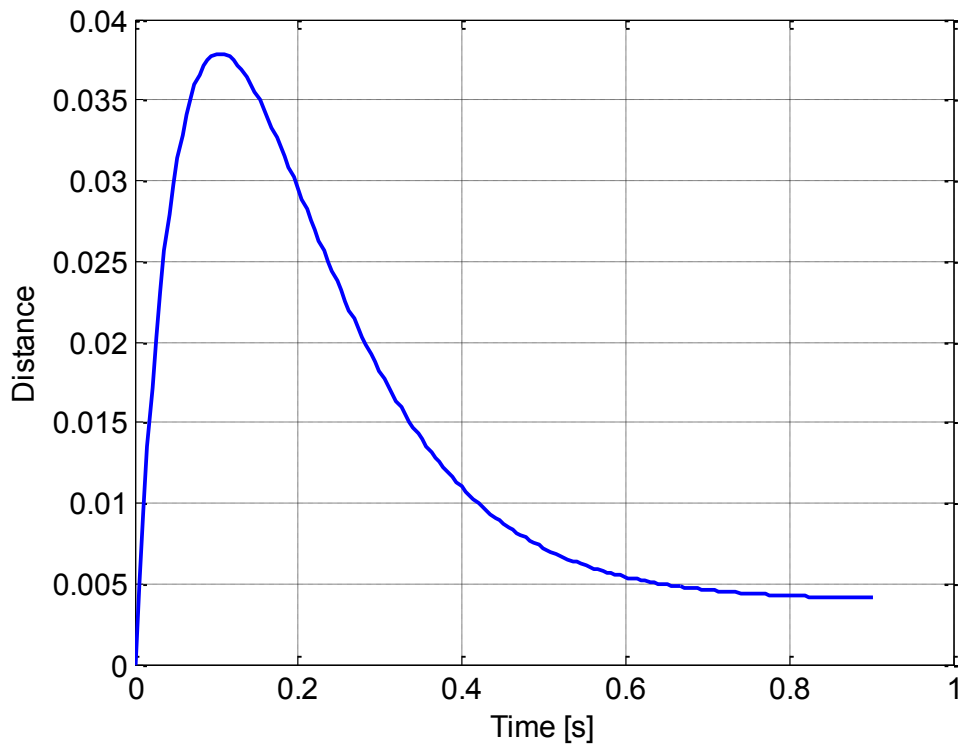


Fig. B.3: Step response of $H(s')$

The bode plot can be seen below in Fig. B.4.

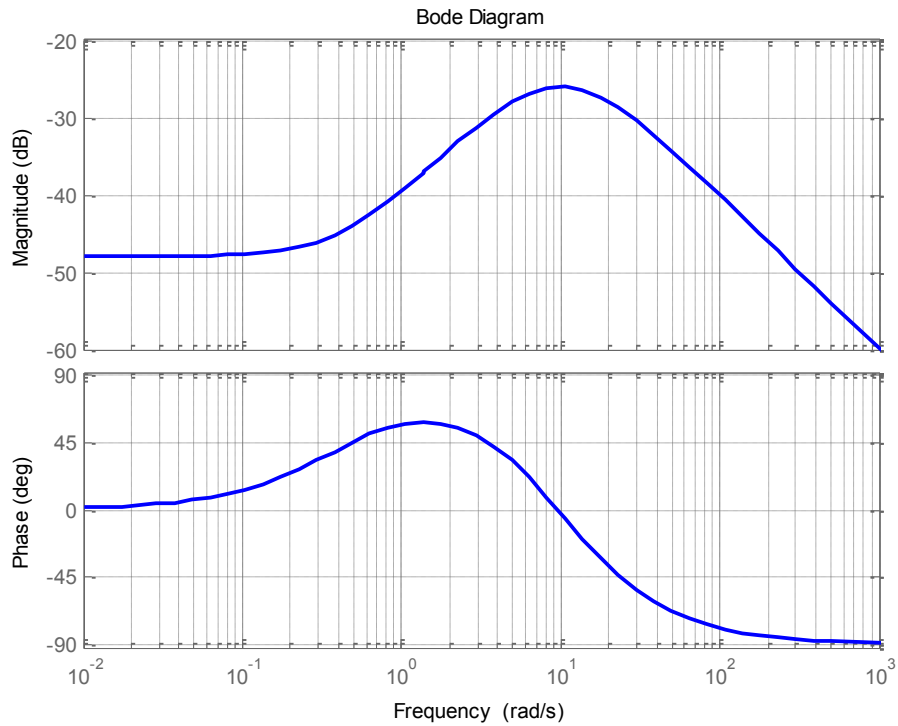


Fig. B.4: Bode Plot of $H(s')$

Table B.1, below, shows the comparison between the primary characteristics of the two transfer functions.

Table B.1: Comparison of Transfer Function with Scaling Factor

<u>Transfer Function</u>	<u>Peak Time</u>	<u>Poles/Zeros</u>	<u>Peak Magnitude</u>
$H(s)$	0.01 [sec]	P=-100, -100; Z=-4	100 [rad/sec]
$H(s')$	0.1 [sec]	P=-10, -10; Z=-0.4	10 [rad/sec]

From Table B.1, it can be seen that in addition to making the transfer function more numerically stable it also slows down the system. This is evident by the fact that the poles and zeros are scaled down by the scaling factor, which in turn slows down the system since the pole/zero combinations are closer to the origin by a

factor of the scaling factor used. By looking at the rise time and peak magnitude this reinforces that point.

Looking at the closed loop transfer function for the electrohydraulic actuator [4],

$$po8n(s) = \frac{num8}{den8} \quad (B.7)$$

where num8 is

$$num8 = -1.559e23s + 9.325e26 \quad (B.8)$$

and den8 is

$$\begin{aligned} den8 = & s^8 + 11090s^7 + 9.638e7s^6 + 5.464e11s^5 + 2.106e15s^4 \\ & + 5.972e18s^3 + 1.072e22s^2 + 5.166e24s \\ & + 9.325e26 \end{aligned} \quad (B.9)$$

This equation does have large terms for the polynomial, on the order of 10^{26} , that could potentially cause problems during simulation. This was fixed by using $s = 10,000s'$ with a scaling factor of 10,000. The scaled equation becomes,

$$po8n(s') = \frac{num8(s')}{den8(s')} \quad (B.10)$$

where num8 is

$$num8(s') = -155.9s' + 932.5 \quad (B.11)$$

and den8 is

$$\begin{aligned}
 den8(s') = s'^8 + 11.09s'^7 + 96.38s'^6 + 546.4s'^5 + 2106s'^4 \\
 + 5972s'^3 + 1.072e4s'^2 + 5166s' + 932.5
 \end{aligned}
 \tag{B.12}$$

The main issue with using Eq. (B.7), the non-scaled transfer function, is that errors will occur when converting to state space and trying to determine the controllability and observability of the system. By using Eq. (B.10), the controllability and observability matrices can be determined to have full rank. This shows that the system is fully stable and all the states are fully controllable and observable. Shown below is Fig. B.5, which shows the po8n(s) system is stable.

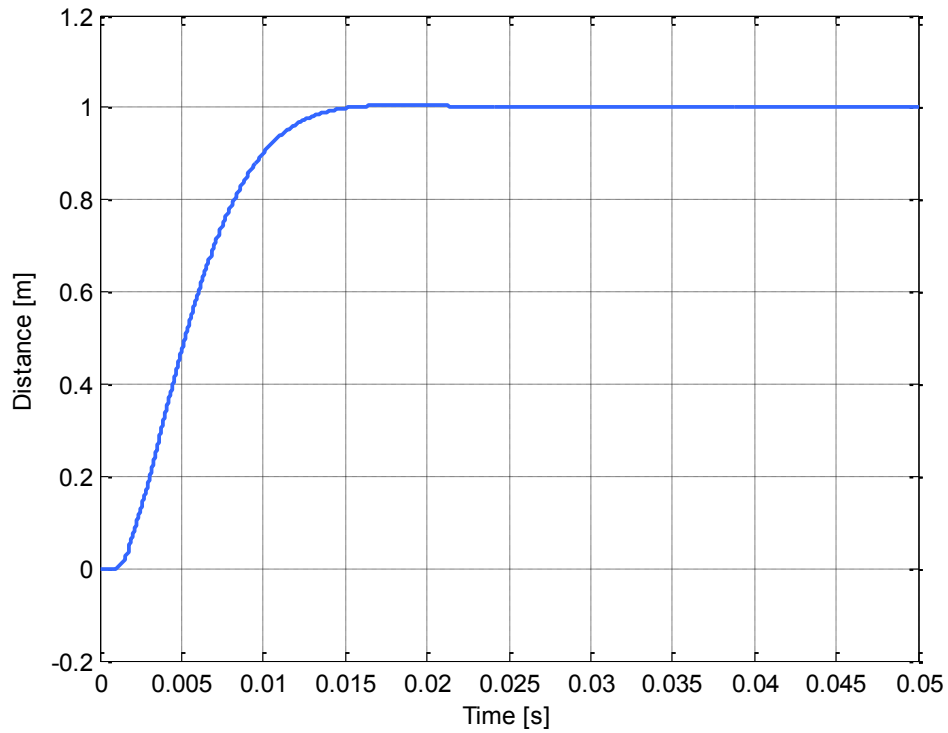


Fig. B.5: Step Response of $po8n(s)$ Transfer Function

The block diagram state space representation of full state feedback is shown in Fig. B.6 below.

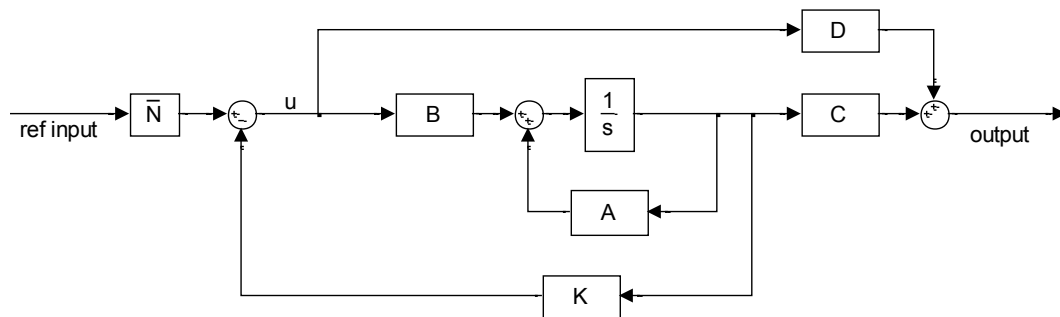


Fig. B.6: Block Diagram Representation of Full State Feedback

A, **B**, **C**, and **D** are the matrices that define the system, **K** is the full state feedback gain, and \bar{N} is the reference input gain. Due to the fact that the full state feedback is in effect moving the poles of the system and affecting the physical outcome of the system, the reference input gain needs to be included. This is primarily used to accurately track the supplied reference input.

The reference input is defined as [24],

$$\bar{N} = N_u + \mathbf{K}N_x \quad (\text{B.13})$$

where

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \quad (\text{B.14})$$

The reference input gain produces a response with no steady state error. Due to the size of the inverse matrix in Eq. (B.14), the Cayle-Hamilton Theorem [25] is used to break down the computation into more manageable sections.

The inverse matrix in Eq. (B.14) will from now on be referred to as **W**. The Cayley-Hamilton Theorem states that a square matrix of size $n \times n$, **W**, has the characteristic equation of the form

$$\lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_1\lambda + a_0 = 0 \quad (\text{B.15})$$

where a is coefficient of the polynomial function determined from the eigenvalues of \mathbf{W} . Equation (B.15) can then be rewritten in terms of the matrix \mathbf{W} as

$$\mathbf{W}^n + a_{n-1}\mathbf{W}^{n-1} + \dots + a_1\mathbf{W} + a_0\mathbf{I} = 0 \quad (\text{B.16})$$

rewriting Eq. (B.16)

$$\mathbf{W}^n + a_{n-1}\mathbf{W}^{n-1} + \dots + a_1\mathbf{W} = -a_0\mathbf{I} \quad (\text{B.17})$$

$$\mathbf{W}(\mathbf{W}^{n-1} + a_{n-1}\mathbf{W}^{n-2} + \dots + a_1\mathbf{I}) = -a_0\mathbf{I} \quad (\text{B.18})$$

$$\mathbf{W} \left(\frac{-1}{a_0} \right) (\mathbf{W}^{n-1} + a_{n-1}\mathbf{W}^{n-2} + \dots + a_1\mathbf{I}) = \mathbf{I} \quad (\text{B.19})$$

$$\mathbf{W}^{-1} = \left(\frac{-1}{a_0} \right) (\mathbf{W}^{n-1} + a_{n-1}\mathbf{W}^{n-2} + \dots + a_1\mathbf{I}) \quad (\text{B.20})$$

The inverse of the matrix can then be rewritten as Eq. (B.20). This is useful for working between the scaled state space model and the full scale state space model.

Utilizing the above derivation, Eq. (B.14) can then be rewritten as

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \mathbf{W}^{-1} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \quad (\text{B.21})$$

The standard form for determining the desired closed loop poles μ_i with full state feedback is defined as,

$$|sI - \mathbf{A} + \mathbf{BK}| = \prod (s - \mu_i) \quad (\text{B.22})$$

where \mathbf{K} is the full state feedback gain.

The state space representation of the $H(s)$, Eq. (B.5) is shown below as Eq. (B.23),

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -200 & -10000 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= [10 \quad 40]x \end{aligned} \quad (\text{B.23})$$

The scaled state space representation of $H(s')$, Eq. (B.6), is shown below as Eq. (B.24)

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -20 & -100 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= [1 \quad 0.4]x \end{aligned} \quad (\text{B.24})$$

The full state feedback gain scaling can be determined by using Eq. (B.25) as shown below

$$\mathbf{K}_{1,i} = \mathbf{K}_{s1,i} s f^i \quad (\text{B.25})$$

where i is the index of the feedback gain matrix starting with 1, \mathbf{K}_s is the scaled feedback gain calculated with acker or place MATLAB function, and sf is the scaling factor to slow down the system used. The scaled feedback gain is calculated in the scaled system knowing the actual, un-scaled and ideal, poles will

be a multiple of sf . This equation can be double checked by determining the eigenvalues of $[A-BK]$ to see if the values match the desired poles.

Table B.2 shows the means by which Eq. (B.25) is developed. From the pattern observed between different scaling factors of 10, 100, and 1000 it can be seen that the ratio follows Eq. (B.25), which can be used to calculate the full state feedback gain using a scaled system. The scaling ratio, K/K_s is determined by using the same poles, except scaled by the appropriate scaling factor and then calculating the ratio of the feedback gain. The scaling ratios were developed on transfer functions that were known to be free of numerical issues resulting from large coefficients.

Table B.2: Scaling Factor Full State Feedback

3rd order Transfer Function		2nd order Transfer Function	
Scaling Factor	K/K_s	Scaling Factor	K/K_s
s=10s'	[10 100 1000]	s=10s'	[10 100]
s=100s'	[100 10000 1e6]	s=100s'	[100 10000]
s=1000s'	[1000 1e6 1e9]	s=1000s'	[1000 1e6]

For instances where the states are not able to be sensed a state estimator can be developed. The state estimator is shown below in Fig. B.7.

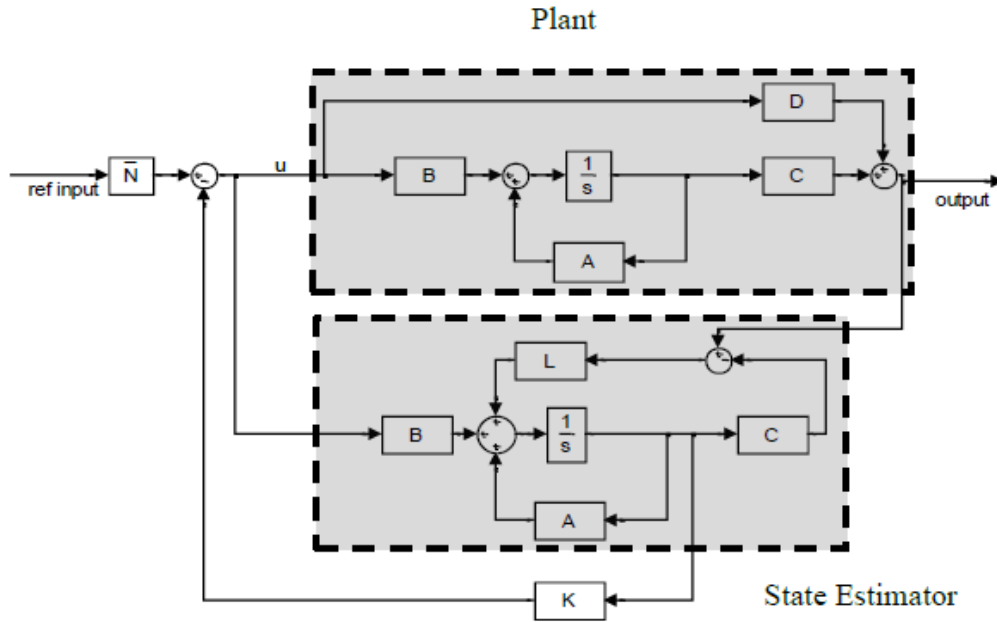


Fig. B.7: Full State Feedback and State Estimator

To determine the state estimator gain, L , the error between the estimated state variable \hat{x} and x is used. \hat{x} is defined as the state variable estimate of x . The state estimator works by estimating the state based on the difference between the actual plant output and the state estimator. The estimator gain is used to position the poles of the subsystem so the overall system is fast enough to be used with the actual plant.

Defining the state estimator as

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \quad (B.26)$$

$$\dot{\hat{x}} = (\mathbf{A} - \mathbf{LC})\hat{x} + \mathbf{B}u + \mathbf{L}y \quad (\text{B.27})$$

The error equation becomes,

$$e = x - \hat{x} \quad (\text{B.28})$$

taking the derivative with respect to time becomes,

$$\dot{e} = \dot{x} - \dot{\hat{x}} \quad (\text{B.29})$$

Inserting the appropriate values,

$$\dot{e} = [\mathbf{A}x + \mathbf{B}u] - [(\mathbf{A} - \mathbf{LC})\hat{x} + \mathbf{B}u + \mathbf{L}y] \quad (\text{B.30})$$

$$\dot{e} = (\mathbf{A} - \mathbf{LC})x - (\mathbf{A} - \mathbf{LC})\hat{x} \quad (\text{B.31})$$

After simplifying Eq. (B.31),

$$\dot{e} = (\mathbf{A} - \mathbf{LC})(x - \hat{x}) = (\mathbf{A} - \mathbf{LC})e \quad (\text{B.32})$$

Equation (B.32) governs the error in the estimation. In order for the error to go to zero quickly,

$$|s\mathbf{I} - (\mathbf{A} - \mathbf{LC})| = 0 \quad (\text{B.33})$$

or the eigenvalues of $\mathbf{A}-\mathbf{LC}$ need to be in the left-plane to be considered stable.

Typically, the poles of the state estimator are chosen to be three to five times faster than the closed loop poles of the plant resulting from the full state feedback.

This allows the states to be estimated faster than the plant itself.

Additionally due to the similarities between Eqs. (B.33) and (B.22) the MATLAB `place` or `acker` can be used with \mathbf{A}' and \mathbf{C}' in place of \mathbf{A} and \mathbf{B} , respectively, to calculate the state estimator gain \mathbf{L} .

Similarly the state estimator gain can be calculated as

$$\mathbf{L}_{i,1} = \mathbf{L}_{s,i,1} s f^{-(i-1)} \quad (\text{B.34})$$

where \mathbf{L} is the state estimator and \mathbf{L}_s is the scaled state estimator and i goes from 1 to the length of \mathbf{L}_s . \mathbf{L}_s is calculated from using the `place` or `acker` function in MATLAB with the transposed scaled \mathbf{A} and \mathbf{C} matrices. Again the ideal pole placements are defined in the scaled system knowing that the actual poles are a multiple of sf faster. Table B.3 shows the ratio that was used to create Eq. (B.34). This equation can be double checked by determining the eigenvalues of $[\mathbf{A}-\mathbf{L}\mathbf{C}]$ to see if the values match the desired estimator poles. The ratio was developed the same way as the full state feedback was, with the exception of using the state estimator analysis, instead of the full state feedback.

Table B.3: Scaling Factor Full State Estimator

3rd order Transfer Function		2nd order Transfer Function	
Scaling Factor	L/L_s	Scaling Factor	L/L_s
s=10s'	[1 0.1 0.01]'	s=10s'	[1 0.1]'
s=100s'	[1 0.01 0.0001]'	s=100s'	[1 0.01]'
s=1000s'	[1 0.001 1e-6]'	s=1000s'	[1 0.001]'

For the example above, the poles in $H(s)$ Eq. (B.5) are moved to -200, -200.

Using the `acker` function in MATLAB, the closed loop feedback gain becomes

[200 30000]. The step response with the feedback gain is shown below in Fig.

B.8

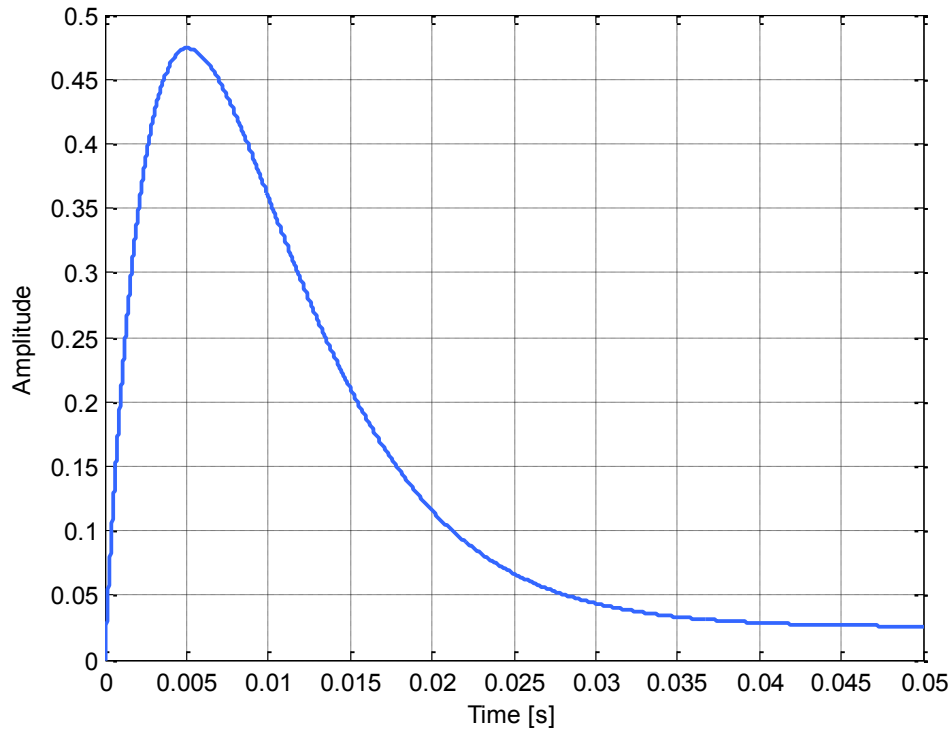


Fig. B.8: Step Response $H(s)$ with Full State Feedback Gain

Using the scaled $H(s')$ function Eq. (B.6), the poles are at -10, -10. Moving them to the scaled position of -20, -20, produces a full state feedback gain of [20 300].

The scaled step response with the full state feedback gain is shown below in Fig.

B.9.

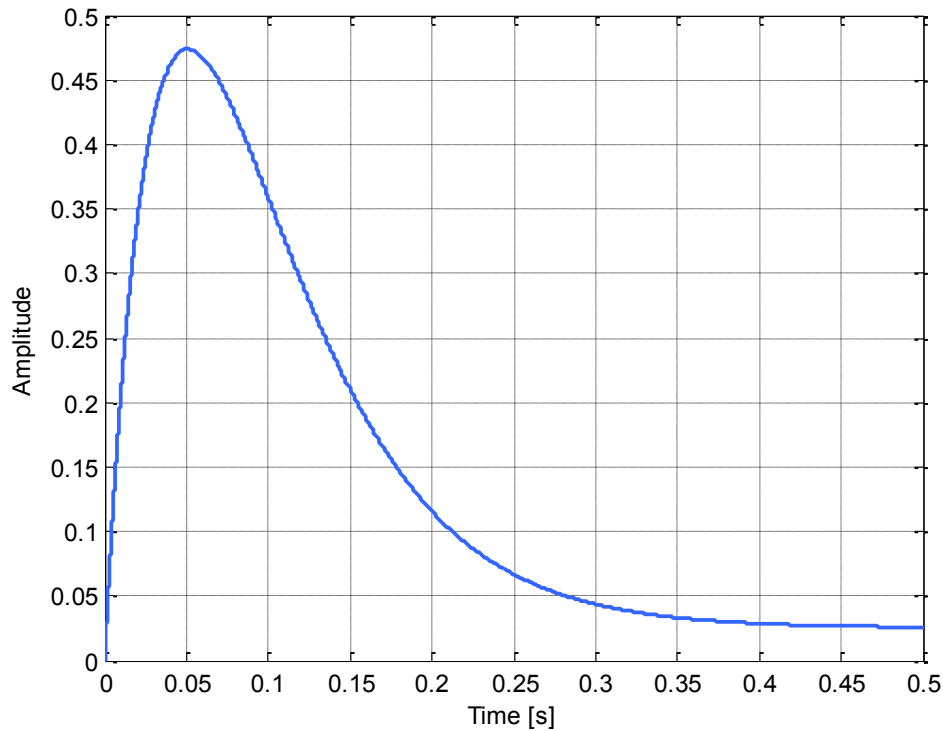


Fig. B.9: Step Response of $H(s')$ with Full State Feedback Gain

From Table B.4, the full state feedback gains are not scaled by the same scaling factor, in the example above 10 was used. This results in a scaling of 10 for the first gain and 100 for the second gain relating the scaled and un-scaled full state feedback gains, assuming the poles are scaled by the same amount.

Table B.4: Full State Feedback Gain Comparison

	Original Pole Location	Moved Pole Location	Full State Feedback Gain
$H(s)$	P=-100, -100;	P=-200, -200;	K=[200 30000]
$H(s')$	P=-10, -10;	P=-20, -20;	K=[20 300]

The full state feedback gain can easily be checked by validating Eq. (B.22).

REFERENCES

- [1] Levine W., ed., 2010, *The Control Handbook, Control Systems Advanced Methods*, CRC Press.
- [2] Jelali M., and Kroll A., 2003, *Hydraulic Servo-systems*, Springer, London.
- [3] Poley R., 2005, "DSP control of electro-hydraulic servo actuators," *Texas Instruments Appl. Rep.*, (January), pp. 1–26.
- [4] Kim D. H., and Tsao T.-C., 2000, "A Linearized Electrohydraulic Servovalve Model for Valve Dynamics Sensitivity Analysis and Control System Design," *J. Dyn. Syst. Meas. Control*, **122**(1), pp. 179–187.
- [5] Troxel N. A., and Yao B., 2011, "Hydraulic Cylinder Velocity Control With Energy Recovery: A Comparative Simulation Study," *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control, Volume 1*, ASME, pp. 169–176.
- [6] Doyle J., Smith R., and Enns D., 1987, "Control of plants with input saturation nonlinearities," *American Control Conference*, pp. 1034–1039.
- [7] Chen K., Huang M., and Fung R., 2011, "The comparisons of minimum-energy control of the mass-spring-damper system," *2011 9th World Congress on Intelligent Control and Automation, IEEE*, pp. 666–671.
- [8] Kim D. H., Moulton S., and Tsao T.-C., 1999, "A Modified Tustin Transformation for Improved Controller Performance with Applications to the Electrohydraulic Actuator," *Fluid power systems and technology*, pp. 125–130.
- [9] Chatzacos P., and Papadopoulos E., 2003, "On model-based control of hydraulic actuators," *Proceedings of RAAD*.
- [10] Maiti R., Saha R., and Watton J., 2002, "The static and dynamic characteristics of a pressure relief valve with a proportional solenoid-controlled pilot stage," *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.*, **216**(2), pp. 143–156.

- [11] Gordic D., Babic M., and Jovicic N., 2004, "Modelling of spool position feedback servovalves," *Int. J. Fluid Power*, **5**(1), pp. 37–50.
- [12] Fang J. H., Kong X. W., Zhu X., and Wei J. H., 2012, "The Modeling and Experimental Verification of a Servo-Proportional Valve," *Appl. Mech. Mater.*, **220-223**, pp. 1018–1022.
- [13] Åman R., 2011, "Methods and Models for Accelerating Dynamic Simulation of Fluid Power Circuits," Lappeenranta University of Technology.
- [14] Anderson R., and Li P., 2002, "Mathematical modeling of a two spool flow control servovalve using a pressure control pilot," *Journal of dynamic systems, measurement, and control*.
- [15] Alleyne A., 1996, "Nonlinear force control of an electro-hydraulic actuator," *Japan-USA Symposium on Flexible Automation*, pp. 193–200.
- [16] Peters D., 2002, "Electrohydraulic Valves... A Technical Look," Moog.
- [17] Ganji M., Behbahani S., and de Silva C. W., 2010, "Integrated modeling of an electro-hydraulic servo manipulator using linear graphs," *IEEE ICCA 2010*, IEEE, pp. 303–308.
- [18] Kim D. H., and Tsao T.-C., 1997, "Robust performance control of electrohydraulic actuators for camshaft machining," *ASME 1997 Fluid power systems and technology*, IEEE, p. 142.
- [19] Anderson W., 1988, *Controlling Electrohydraulic Systems*, CRC Press, New York.
- [20] Merrit H. E., 1967, *Hydraulic control systems*, John Wiley & Sons, New York.
- [21] Chapra S., and Canale R., 2006, *Numerical methods for engineers*, McGraw Hill.
- [22] MATLAB, 2012, version 7.14 (R2012a), The MathWorks Inc., Natick.
- [23] Levine W., ed., 2010, *The Control Handbook, Control Systems Fundamentals*, CRC Press.

- [24] Franklin G. F., Powell J. D., and Workman M., 1998, Digital Control of Dynamic Systems, Ellis-Kagle Press.
- [25] Bronson R., 1970, Matrix Methods an Introduction.