Multivariable Servomechanism Controller Design of Web Handling Systems

by

Weixuan Liu

A thesis submitted in conformity with the requirements for the Degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

©Copyright by Weixuan Liu, 2000
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-50385-2
To my family
Multivariable Servomechanism Controller Design
of Web Handling Systems
M.A.Sc., 2000
Weixuan Liu
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

In traditional web handling processes, web tension and speed are controlled assuming that the web system consists of a number of single input and single output systems. This assumption often results in large interactions occurring in the closed loop system between the control loops, and hence results in high quality control being difficult to achieve.

In this thesis, the control of the web handling processes is treated as a multivariable servomechanism problem. Three types of controller designs—the "cheap control servomechanism controller", the "high gain servomechanism controller" and the "tuning regulator" are studied and implemented on the University of Toronto web machine. The experimental results obtained show that these controllers provide excellent tension and speed response.
Acknowledgement

I am truly grateful to my supervisor, Professor Edward J. Davison for his patient guidance and financial support throughout my study and research in the System Control Group.

I am in debt to my family, who have done so much to make my dreams come true. Without their love and encouragement, I would achieve little.

Finally, I have to thank my colleagues in the System Control Group for their help and friendships. Special thanks go to Chuan Ma, Gaby Saad and Ken Pu, who have treated me as a family member and made my life enjoyable.
# Contents

1 Introduction 2
   1.1 Background Knowledge ........................................ 2
   1.2 The Rotoflex Web Handling Machine ......................... 3
   1.3 Outline of the Thesis ........................................ 4

2 Modeling For Web Tension Control 6
   2.1 Introduction ................................................ 6
   2.2 Modeling for the Web Mechanical System ................... 7
      2.2.1 The Model of a Free Web ............................... 7
      2.2.2 Models of Rollers ........................................ 10
      2.2.3 Models of Web Mechanical Systems ..................... 12

3 The Servomechanism Control Problem 25
   3.1 Robust Servomechanism Problems .............................. 25
   3.2 The Servo-compensator ...................................... 27
      3.2.1 The Controller Structure ................................ 27
      3.2.2 Stabilizing Controller Design .......................... 30

4 The Perfect Control Problem 31
   4.1 Introduction ................................................ 31
   4.2 The Perfect Control Problem ................................ 31
      4.2.1 Development of the Problem ............................ 31
      4.2.2 Design for Perfect Control ............................. 33
5 A High Gain Stabilizing controller

5.1 Introduction ................................................. 36

5.2 Mathematical Preliminaries ............................ 36
  5.2.1 Transmission Zeros .................................. 36
  5.2.2 Transmission Zero at Infinity [Mac89] ............ 37
  5.2.3 Minimum Phase System ............................... 37
  5.2.4 Hurwitz Polynomial ................................. 37

5.3 The multivariable root locus ............................. 37

5.4 Controller Synthesis ...................................... 39
  5.4.1 Approach 1–Decomposition .......................... 40
  5.4.2 Approach 2–Generalized Differential Interactor .... 42

5.5 Conclusion .................................................. 45

6 Tuning Regulator Controller ............................... 47

7 Controller Implementation ................................. 50

7.1 Introduction ............................................... 50

7.2 Models for Controller Implementations ................ 51
  7.2.1 A Reduced Order Model .............................. 51
  7.2.2 Another Model for the Web System ................. 55

7.3 Cheap controller design .................................. 60
  7.3.1 Experimental Results–Controller #1 ............... 61
  7.3.2 Discussion of the Experimental Results Obtained From Controller #1 ......... 66
  7.3.3 Experimental Results–Controller #2 ............... 70

7.4 High Gain Controller Design ............................. 73
  7.4.1 The Properties of the Web System ................. 73
  7.4.2 Design of The Saturation Controller ............... 74
  7.4.3 Combining The High Gain Control Design With The Saturation Controller Design ................................................. 77
  7.4.4 Simulation and Experimental Results ............... 77
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.5 Tuning Regulator Design</td>
<td>82</td>
</tr>
<tr>
<td>7.5.1 Preliminary Experiment</td>
<td>83</td>
</tr>
<tr>
<td>7.5.2 Results Obtained From Experiments</td>
<td>88</td>
</tr>
<tr>
<td>8 Conclusions</td>
<td>91</td>
</tr>
<tr>
<td>8.1 Main Contributions of This Thesis</td>
<td>91</td>
</tr>
<tr>
<td>8.2 Further Research</td>
<td>92</td>
</tr>
<tr>
<td>A Properties of the Saturation Controller</td>
<td>96</td>
</tr>
<tr>
<td>A.1 Structure of The Saturation Controller</td>
<td>96</td>
</tr>
<tr>
<td>A.2 The Response of the Saturation Controller</td>
<td>97</td>
</tr>
<tr>
<td>A.2.1 Case 1—Direct State Feedback</td>
<td>97</td>
</tr>
<tr>
<td>A.2.2 Using an observer</td>
<td>103</td>
</tr>
<tr>
<td>B Models of the University of Toronto Web Machine and Controllers</td>
<td>104</td>
</tr>
<tr>
<td>B.1 Models of the University of Toronto Machine</td>
<td>104</td>
</tr>
<tr>
<td>B.1.1 Model #1: A Reduced Order Low Frequency Model</td>
<td>105</td>
</tr>
<tr>
<td>B.1.2 Model #2: A Black Box 6th Order Model</td>
<td>105</td>
</tr>
<tr>
<td>B.2 Controllers</td>
<td>106</td>
</tr>
<tr>
<td>B.2.1 Cheap Control Servomechanism Controller</td>
<td>107</td>
</tr>
<tr>
<td>B.2.2 High Gain Servomechanism Controller</td>
<td>110</td>
</tr>
<tr>
<td>C Scripts For Controller Designs</td>
<td>113</td>
</tr>
<tr>
<td>C.1 Matlab Codes For Controller Design</td>
<td>113</td>
</tr>
<tr>
<td>C.1.1 The Script To Obtain the University of Toronto Web Machine State Space Model</td>
<td>113</td>
</tr>
<tr>
<td>C.1.2 The Script To Normalize The Linear System And Add Filters</td>
<td>114</td>
</tr>
<tr>
<td>C.1.3 Scripts for Cheap Control Servomechanism Controller Design</td>
<td>116</td>
</tr>
<tr>
<td>C.1.4 Scripts for High Gain Servomechanism Controller Design</td>
<td>120</td>
</tr>
<tr>
<td>C.1.5 The Script to Discretize the continuous time state space controller and to Export the Controller As A C Head File</td>
<td>124</td>
</tr>
<tr>
<td>C.2 C Codes For Controller Implementations</td>
<td>127</td>
</tr>
</tbody>
</table>
C.2.1 C Codes For The Implementation of the High Gain and Cheap Control Servomechanism Controllers . . . . . . . . . . . . . . . . 127
List of Tables

1  List of Notation ........................................... 1
7.1 List of Identified Parameters .............................. 52
7.2 Steady State Value Difference .............................. 83
## List of Figures

1.1 Overview of University of Toronto Machine ........................................... 4

2.1 The overall electrical/mechanical system of a web handling machine ........ 6
2.2 A Free Span Web .................................................................................... 7
2.3 A Single Tension Span ........................................................................... 8
2.4 A Roller in a Web System ....................................................................... 10
2.5 A Winder in a Web System ..................................................................... 11
2.6 A n + 1 span system ................................................................................ 12
2.7 Calculation of the inertia and radius of a winder .................................... 13
2.8 Reduced System Is a First Order RC circuit ......................................... 17
2.9 A High Frequency Model Module .......................................................... 19
2.10 High Frequency Model ........................................................................... 20
2.11 Block Diagram of the High Frequency Model ........................................ 21
2.12 The derivation of the model for the reduced order system ..................... 21

3.1 The general structure of a servomechanism controller ............................. 29

5.1 A Generic Multivariable Root Loci .......................................................... 38

7.1 The Illustrative Diagram of the Rotoflex Machine .................................... 50
7.2 The Reduced Low Frequency Model of the Rotoflex Machine ................ 51
7.3 The Response of the Open loop Real System vs. the Identified Model ......... 54
7.4 Comparison of The Response of the Black Box Model and The Real Machine–Rewinder Torque Step ................................................................. 57
7.5 Comparison of The Response of the Black Box Model and The Real Machine–Nip Torque Step ........................................ 58
7.6 Comparison of The Response of the Black Box Model and The Real Machine–Unwinder Torque Step ........................................ 59
7.7 The Web System Structure After Scaling ........................................ 60
7.8 The Structure of the Overall Controller ........................................ 62
7.9 Output Responses to Tension Fa Step Reference Input (controller #1) ........................................ 63
7.10 Output Responses to Tension Fb Step Reference Input (controller #1) ........................................ 64
7.11 Output Responses to velocity Step Reference Input (controller #1) ........................................ 65
7.12 Comparison of the closed loop responses of the experimental results and the theoretical results of the model #1 (controller #1) ........................................ 67
7.13 The Simulated Closed Loop Response of Tension Fb for Different Operating Points (controller #1 applied to model #1) ........................................ 69
7.14 The Closed Loop Experimental Response of Tension Fb for Different Operating Points (controller #1) ........................................ 69
7.15 The Closed Loop Simulated Response of Tension Fb for Different Web Thickness (controller #1 applied on model #1) ........................................ 70
7.16 Experimental Results of Response To Tension Fa Step Input (Controller #2) ........................................ 71
7.17 Experimental Results of Response To Tension Fb Step Input (Controller #2) ........................................ 72
7.18 Experimental Results of Response To Velocity Step Input (Controller #2) ........................................ 73
7.19 The Overall Structure of a High Gain Design – Saturation Controller ........................................ 75
7.20 Large Input Overshoot of the High Gain Controller ........................................ 76
7.21 Simulated Output of the High Gain Controller ........................................ 78
7.22 Simulated Output of the Cheap Control Perfect Controller ........................................ 78
7.23 The Input of the High Gain Controller ........................................ 79
7.24 The Input of the Cheap Control Perfect Controller ........................................ 79
7.25 Output Responses to Tension Fa Step Reference Input ........................................ 80
LIST OF FIGURES

7.26 Output Responses to Tension Fb Step Reference Input ............... 81
7.27 Output Responses to velocity Step Reference Input ................... 82
7.28 Open Loop Response to Torque Step Input \( \delta U = [0.6 \ 0 \ 0]^T \) ....... 84
7.29 Open Loop Response to Torque Step Input \( \delta U = [0 \ 0.6 \ 0]^T \) ........ 85
7.30 Open Loop Response to Torque Step Input \( \delta U = [0 \ 0 \ -0.3]^T \) ...... 86
7.31 Tuning Regulator Experiment: Responses to Fa step input .......... 88
7.32 Tuning Regulator Experiment: Responses to Fb step input .......... 89
7.33 Tuning Regulator Experiment: Responses to velocity step input ..... 90

A.1 The Overall Structure of A High Gain Design With Saturation Controller ........................................ 97
A.2 Simplified Structure of figureA.1 ........................................ 98
The following notation and expressions will be used throughout the thesis.

Table 1: List of Notation

- \( \mathbb{R} \): The set of real numbers.
- \( \mathbb{R}^+ \): The set of positive real numbers.
- \( \mathbb{C} \): The set of complex numbers.
- \( \mathbb{R}^n \): The set of the \( n \)-dimensional real vector space.
- \( \mathbb{C}^n \): The set of the \( n \)-dimensional complex vector space.
- \( \mathbb{R}^{m \times n} \): The set of the \( m \times n \) real matrices.
- \( \mathbb{C}^{m \times n} \): The set of the \( m \times n \) complex matrices.
- \( \mathbb{C}^- \): The set of complex numbers with strictly negative real parts.
- \( \mathbb{C}^+ \): The set of complex numbers lying on the closed right hand part of the complex plane.
- \( \text{sp}(M) \): The set of the eigenvalues of the square matrix \( M \).
- \( \mathbb{R}(s) \): The field of rational functions with real coefficients.
- \( \mathbb{R}[s] \): The commutative ring of polynomials with real coefficients.
- \( \mathbb{C}(s) \): The field of rational function with complex coefficients.
- \( \mathbb{C}[s] \): The commutative ring of polynomials with complex coefficients.
- \( \| \cdot \| \): The Euclidean norm of a matrix.
Chapter 1

Introduction

1.1 Background Knowledge

In industry, paper, plastic and other elastic thin materials are often used in the manufacturing of commercial products by using a continuous process. In this case, the paper or other material is typically unrolled from a large roll using a series of rollers and a rewinder, forming what is called a web.

To produce an end product from a raw web material, such as from a paper machine or a film extruder, two kinds of processes are involved: Web Converting and Web Handling. Web converting includes all those processes which are required to modify the physical properties of the web material such as Coating, Slitting, Metalizing, Drying and Embossing, etc. The web handling processes, on the other hand, consist of those processes which are associated with the transportation aspects of the web. The main purpose of the web handling process is to transport web with maximum throughput (speed) and with minimum damage[RoI98]. To achieve this, web tension control is crucial because of the following reasons:

1. Web tension affects the geometry of the web, such as the apparent length and width of the web.

2. High web tension prevents the loss of traction on the rollers; however too high web tension will cause a web break to occur.
3. Web tension control helps to reduce wrinkling. In particular, high process tension will help decrease the wrinkling caused by a misalignment of rollers; however, excessively high tension will cause more wrinkling to occur on very thin materials. Hence, appropriate web tension control is very important.

4. Web tension affects the wound-in tension and the shape of the final product roll, and hence the roll quality.

For these reasons, it is essential in web handling to control the web tension at a desired value as closely as possible. Normally, web tension should be set at 10-25% of the web's yield strength and should be kept within 10% of this value during the system's steady running state and 25% of this value at speed set-point changes [Roi98].

Almost all the components of a web machine influence the web tension. We will now study the properties of the web machine which was used in the experimental part of this thesis at the University of Toronto, Department of Electrical and Computer Engineering, and is manufactured by Rotoflex International Inc..

1.2 The Rotoflex Web Handling Machine

As a typical small size web handling machine, the University of Toronto web machine has all the necessary components associated with a web handling process.

Rollers are essential parts of a web handling machine. In any machine, there are two type of rollers: (i) externally torque driven rollers such as the unwinder, rewinder and the nip roller and (ii) the web driven rollers (idlers). These devices are also called "transport rollers" in industry because they are not intended to change the physical properties of the web. The traditional role of a "nipped roller" is to step the tension up or down between sections of processes, and hence create different tension zones for different processes. In designing a controller for a web system, the nipped roller torque input and the wound roller torque inputs (rewinder and unwinder) provide multiple inputs for multivariable tension/speed control. These torque inputs are regulated by PWM drives. The torque outputs from the PWM drives can be either positive or
negative, and hence can either act as "drives" or "brakes" in web control. Besides the nip and winders, there are also some web driven rollers (idlers), which provide additional inertia to the web system.

The tensions of the web system are measured by load cells (which provide better precision than the dancer system). To provide real-time monitoring of time varying information such as inertia of the unwinder and rewinder, there are two diameter sensors which measure the changing diameters of the unwinder and rewinder, in the U of T web machine.

1.3 Outline of the Thesis

The rest of the thesis deals with the control of web handling systems.

This thesis starts from the modeling of web handling systems; in particular, in chapter 2, a reduced order low frequency linear model and a high frequency model are developed for web system control design.

In chapters 3, the robust servomechanism is introduced, and the structure of the "servocompensator" is presented.
Chapters 4, 5 and 6 introduce the design of three stabilizing controllers. Two different types of perfect control design—a cheap control design and a high gain controller design are given in chapter 4 and 5 respectively, and in chapter 6, a "tuning regulator" controller, which does not require a mathematical model of the system is presented.

All three types of servomechanism controller design methodologies are implemented on the University of Toronto web handling machine, and the results are presented in chapter 7.

Chapter 8 concludes the work in this thesis and presents some suggestion for future research.
Chapter 2

Modeling For Web Tension Control

2.1 Introduction

To achieve satisfactory control of the web tension and velocity in a web machine, an approximate mathematical model describing the physical web system will be obtained. The complete electrical/mechanical components of the web handling machine influences the control of the web tension; however the web mechanic part of the system is the core part. It is also to be noted that the tension sensors introduce high frequency measurement noise to the whole web system, while the drive block places restrictions on the control signals (i.e. amplitude and rate thresholds of the output signals from the control system.). We will concentrate on the modeling of the most important part of the web system: the web mechanical system.

![Figure 2.1: The overall electrical/mechanical system of a web handling machine](image)

Figure 2.1: The overall electrical/mechanical system of a web handling machine
2.2 Modeling for the Web Mechanical System

The most influential components of a web system are the rollers and web. We will begin by developing mathematical models of these two prominent elements and then develop a model of the overall system.

2.2.1 The Model of a Free Web

![Figure 2.2: A Free Span Web](image)

A strip of web under longitudinal stretch will experience strains in all three directions MD, CD and ZD as shown in fig2.2:

\[
\begin{align*}
\varepsilon_x &= \frac{L_s - L_o}{L_o} = \frac{\Delta L_o}{L_o} \\
\varepsilon_w &= \frac{w_s - w_o}{w_o} = \frac{\Delta w_o}{w_o} \\
\varepsilon_h &= \frac{h_s - h_o}{h_o} = \frac{\Delta h_o}{h_o}
\end{align*}
\]

(2.1)

where the subscript \( s \) represents the state of being stretched and the subscript \( 0 \) represents the original unstretched state. The subscripts \( x, w, h \) represent the MD, CD and ZD directions respectively. In the following paragraphs, the subscription \( x \) for the MD direction will be omitted.

It follows, according to (2.1), that the mass per unit length of the stretched web \( m_s \), and the mass per unit length of the unstretched web \( m_o \) have the following
relationship:

\[
\rho_0 L w_0 h_0 = \rho_s (1 + \epsilon) L w_s h_s \\
\Rightarrow \rho_0 A_0 = (1 + \epsilon) \rho_s A_s \\
\Rightarrow \frac{\rho_0 A_0}{\rho_s A_s} = \frac{1}{1 + \epsilon}
\]

where \( \rho \) and \( A \) denote the density (mass per unit length) and the cross section area of the web span respectively. Here \( \epsilon \) denotes strain in the MD direction as described in (2.1).

![Figure 2.3: A Single Tension Span](image)

Now consider a one span web system. On applying the mass conservation law on the web span in fig 2.3, i.e, the rate of the mass increase in the web span equals the rate of mass entering the web span minus the rate of leaving this span, we obtain:

\[
\frac{dm}{dt} = \frac{d}{dx} \left[ \int_{x_1}^{x_2} \rho(x, t) A(x, t) dx \right] = \rho_2(t) A_2(t) v_2(t) - \rho_1(t) A_1(t) v_1(t)
\]  

(2.3)

Under the assumption that the strain in the web is uniformly distributed, the strain in the web span in figure 2.3 is given by \( \epsilon_1(x, t) = \epsilon_1(t) \), which implies that \( \rho(x, t) = \rho_1(t) \) and \( A(x, t) = A_1(t) \) must also be true.
This implies that:

\[ m = \int_{x_1}^{x_2} \rho(x,t)A(x,t)dx = L\rho_1(t)A_1(t) \quad (2.4) \]

On applying the result of (2.2) and (2.4) to (2.3), we then obtain the following dynamic equation for the web

\[ L \frac{d}{dt} \left( \frac{1}{1+\varepsilon_1(t)} \right) = \frac{v_2(t)}{1+\varepsilon_1(t)} - \frac{v_1(t)}{1+\varepsilon_1(t)} \quad (2.5) \]

Assuming the strain \( \varepsilon \) is very small \((\ll 1)\), then:

\[ \frac{1}{1+\varepsilon} \approx 1 - \varepsilon \quad (2.6) \]

On applying (2.6) to (2.5), we obtain the dynamic equation of the web for the small strain model

\[ \dot{\varepsilon}_1(t) = -\frac{1}{2}v_1\varepsilon_1(t) + \frac{1}{2}v_2\varepsilon_2(t) + \frac{1}{2}v_1(t) - \frac{1}{2}v_2(t) \quad (2.7) \]

However, our real interest in modeling is in the tension in the web span rather than the strain. In this case, from mechanics, it is known that tension and strain are approximately related by Hooke's law:

\[ T = AE\varepsilon \quad (2.8) \]

where \( T \) is the tension developed in the web, \( \varepsilon \) is the strain of the web from the unstretched state, \( A \) is the cross-section area of the web in the unstretched state, and \( E \), a constant, is the Young's modulus of the web.

Assume now that the cross-section area of the web, when at unstretched state, does not change along the web, and apply (2.8) to (2.7); we then obtain the dynamic
2.2.2 Models of Rollers

The roller is another very important component in a web handling machine. The winders and the web itself provide the necessary tools to produce the Tension, Nip and Torque (TNT) to produce a "good" roll, which is the ultimate goal of web handling processing.

\[
\dot{T}_1(t) = \frac{G}{L}(v_1(t) - v_2(t)) - \frac{1}{J}v_1(t)T_1(t) + \frac{1}{J}v_2(t)T_2(t)
\] (2.9)

where \(G \triangleq AE\).

A roller in a web system is driven by the web tension and corresponding external motor torque inputs. In a web machine, the overall system is designed for the web to interact with the roller in a state of either floating, sliding or tracking with a roller. Here we will only consider the case when the machine has been designed to work in the tracking mode.

In this case, the speed of a roller in a web system such as represented in figure 2.4 should satisfy the following dynamic equation:

\[
\frac{J}{r} \frac{du(t)}{dt} = T_1(t)r - T_2(t)r + U(t) - F
\] (2.10)
where $U$ is the external torque applied by the motor and $\mathcal{F}$ is the friction of the shaft.

\begin{equation}
J(t) \frac{d}{dt} \left( \frac{\omega(t)}{r(t)} \right) = T_1(t) r(t) - T_2(t) r(t) + U(t) - \mathcal{F} 
\end{equation}

(2.11)

On simplifying, the dynamic equation for $v(t)$ then becomes:

\begin{equation}
\frac{J(t)}{r(t)} \frac{dv(t)}{dt} = T_1(t) r(t) - T_2(t) r(t) + \frac{J(t)e}{2\pi r(t)} v^2(t) + U(t) - \mathcal{F}
\end{equation}

(2.12)

In the derivation of (2.12), the relationship:

\begin{equation}
\dot{r}(t) = \omega(t) \frac{e}{2\pi} = \frac{v(t)}{r(t)} \frac{e}{2\pi}
\end{equation}

(2.13)

where $\omega(t)$ is the angular speed and $e$ is the thickness of the web is used.

The friction torque $\mathcal{F}$ is generally a nonlinear function of the angular velocity. As observed in the identification study of the Rotoflex machine [Bor99], the friction force can be characterized by a quadratic function or a function of the form: $\mathcal{F} = (\omega - \alpha)^2 + d$, where $\alpha, e, c, d$, are constant coefficients and $\omega$ is the angular velocity of the shaft. However for practical use, the following linearized model for the friction force is often
enough:

\[ \mathcal{F} = \mathcal{F}_0 + b \omega = \mathcal{F}_0 + b \frac{v(t)}{T(t)} \]  

(2.14)

where \( \mathcal{F}_0 \) and \( b \) are constants.

### 2.2.3 Models of Web Mechanical Systems

![Image of a web system model](image)

Figure 2.6: A \( n + 1 \) span system

An overall model of the mechanical behavior of a web system, consisting of a unwinder, a rewinder and \( n \) non-winder rollers, can be directly obtained from the previous sub-component models obtained in sections 2.2.1 and 2.2.2 as following:

\[
\begin{align*}
\frac{dv_1(t)}{dt} &= -\frac{r_1(t)}{J_1(t)}T_1(t) + \frac{\epsilon}{2\pi r_1(t)}v_1^2(t) + \frac{r_1(t)}{J_1(t)}U_1(t) - \frac{r_1(t)}{J_1(t)}\mathcal{F}_1 \\
\frac{dv_i(t)}{dt} &= -\frac{r_i(t)}{J_i(t)}T_{i-1}(t) - \frac{r_i(t)}{J_i(t)}T_i(t) + \frac{r_i(t)}{J_i(t)}U_i(t) - \frac{r_i(t)}{J_i(t)}\mathcal{F}_i, \quad i = 2, \ldots, n + 1 \\
\frac{dv_{n+2}(t)}{dt} &= -\frac{r_{n+2}(t)}{J_{n+2}(t)}T_{n+1}(t) + \frac{\epsilon}{2\pi r_{n+2}(t)}v_{n+2}^2(t) + \frac{r_{n+2}(t)}{J_{n+2}(t)}U_{n+2}(t) - \frac{r_{n+2}(t)}{J_{n+2}(t)}\mathcal{F}_{n+2} \\
\dot{T}_j(t) &= \frac{\alpha}{L_j}(v_j(t) - v_{j+1}(t)) - \frac{1}{L_j}v_j(t)\dot{T}_j(t) + \frac{1}{L_j}v_{j+1}(t)\dot{T}_{j+1}(t), \quad j = 1, \ldots, n \\
\dot{T}_{n+1}(t) &= \frac{\alpha}{L_{n+1}}(v_{n+1}(t) - v_{n+2}(t)) - \frac{1}{L_{n+1}}v_{n+1}(t)\dot{T}_{n+1}(t) + \frac{1}{L_{n+1}}v_{n+2}(t)\dot{T}_\omega(t)
\end{align*}
\]

(2.15)

Here, if the first roller (roller number 1) is the rewinder, then the last roller (roller number \( n + 2 \)) is the unwinder, and they have varying radii and inertia, given by the
following equations:

\[
\begin{align*}
\dot{r}_1(t) &= \frac{v_1(t)}{2\pi r_1(t)} \\
\dot{r}_{n+2}(t) &= -\frac{v_{n+2}(t)}{2\pi r_{n+2}(t)} \\
J_1(t) &= J_0 + \frac{\pi \rho W}{2}(r_1^4 - r_{10}^4) \\
J_{n+2}(t) &= J_{(n+2)0} + \frac{\pi \rho W}{2}(r_{n+2}^4 - r_{(n+2)0}^4)
\end{align*}
\]  

(2.16)

where (see fig 2.7), the radius \( r \) and inertia \( J \) with a subscript 0 denotes the radius and inertia of the shaft, \( r_1 \) denotes the radius of the rewinder, \( J_1 \) denotes the inertia of the rewinder, \( r_{n+2} \) denotes the overall radius of the unwinder and \( J_{n+2} \) denotes the overall inertia of the unwinder. Here \( v_1 \) to \( v_{n+2} \) denote the linear speeds of the \( n+2 \) rollers, \( F_1 \) to \( F_{n+2} \) denote the friction torques on the rollers, \( U_1 \) to \( U_{n+2} \) denote the external motor torque applied to the rollers, \( T_1 \) to \( T_{n+1} \) denote the tensions in the \( n+1 \) web spans and \( T_w \) is the "wound in tension" in the unwinder, which is stored in the roll by the previous processes.

![](https://via.placeholder.com/150)

Figure 2.7: Calculation of the inertia and radius of a winder

This system is nonlinear. In order to apply linear multivariable control design methods to find a controller for this system, we must linearize (2.15) and (2.16). However, direct linearization for a single equilibrium point is not feasible for this system because the states \( r_1(t) \) and \( r_{n+1}(t) \) do not have steady state operating state. Hence, we will carry out a partial linearization, i.e. we will linearize the system on
the equilibrium space defined as:

\[ \{ \bar{T}_1, \ldots, \bar{T}_i, \ldots, \bar{T}_{n+1}, \bar{v}_i, \ldots, \bar{v}_j, \ldots, \bar{v}_{n+2} \} \times \{ r_1(t), r_{n+2}(t) \}, \quad \{ r_1(t), r_{n+2}(t) \} \in \mathbb{R}_+^2 \]

(2.17)

where \( \bar{T}_1, \ldots, \bar{T}_i, \ldots, \bar{T}_{n+1}, \bar{v}_i, \ldots, \bar{v}_j, \ldots, \bar{v}_{n+2} \) denote the steady state operating values.

In this case, the linearized model becomes:

\[
\begin{align*}
\frac{dv_i(t)}{dt} &= -\frac{r_i(t)}{J_i(t)} T_1(t) + \left( \frac{e}{\pi r_i(t)} - \frac{b_i}{J_i(t)} \right) v_1(t) + \frac{r_i(t)}{J_i(t)} U_1(t) \\
\frac{dv_j(t)}{dt} &= \frac{r_j}{J_j(t)} T_{i-1}(t) - \frac{r_j}{J_j(t)} T_i(t) + \left( -\frac{b_j}{J_j(t)} \right) v_i(t) + \frac{r_j}{J_j(t)} U_i(t), \quad i = 2, \ldots, n + 1 \\
\frac{dv_{n+2}(t)}{dt} &= \frac{r_{n+2}(t)}{J_{n+2}(t)} T_{n+1}(t) + \left( \frac{e}{\pi r_{n+2}(t)} - \frac{b_{n+2}}{J_{n+2}(t)} \right) v_{n+2}(t) + \frac{r_{n+2}(t)}{J_{n+2}(t)} U_{n+2}(t) \\
T_j(t) &= \frac{G - T_j(t)}{L_j} v_j(t) - \frac{G - T_{j+1}(t)}{L_j} v_{j+1}(t) - \frac{1}{L_j} \bar{v}_j(t) T_j(t) + \frac{1}{L_j} \bar{v}_{j+1}(t) T_{j+1}(t), \quad j = 1, \ldots, n \\
T_{n+1}(t) &= \frac{G - T_{n+1}(t)}{L_{n+1}} v_{n+1}(t) - \frac{G - T_{n+2}(t)}{L_{n+2}} v_{n+2}(t) - \frac{1}{L_{n+1}} \bar{v}_{n+1}(t) T_{n+1}(t) + \frac{1}{L_{n+2}} \bar{v}_{n+2}(t) T_{n+2}(t)
\end{align*}
\]

(2.18)

where the state variables \( v_i, i = 1, \ldots, n + 2 \) and \( T_j, j = 1, \ldots, n + 1 \) now denote the difference of the actual corresponding system state and the steady state operating values.

On examining the dynamic equation (2.16) of the radii, it is quite obvious that \( \dot{r}_i(t), i = 1, n + 2 \) is very small because the thickness \( e \) of the web is small (approximately \( 10^{-5} \text{m} \)). Hence the system (2.16) results in a slow varying system with respect to the radii \( r_i(t), r = 1, n + 2 \).

Another feature we can observe from the dynamic equations (2.15) is the multiple time scale structure. In particular, from (2.15), for \( j = 1, 2, \ldots, n + 1 \), we have:

\[
\dot{T}_j(t) = \frac{G}{L_j} (v_j(t) - v_{j+1}(t)) - \frac{1}{L_j} v_j(t) T_j(t) + \frac{1}{L_j} v_{j+1}(t) T_{j+1}(t)
\]

(2.19)

which can be rewritten as:
\[ \epsilon \dot{T}_j(t) = \frac{1}{L_j} (v_j(t) - v_{j+1}(t)) + \epsilon \left[ -\frac{1}{L_j} v_j(t) T_j(t) + \frac{1}{L_j} v_{j+1}(t) T_{j+1}(t) \right] \]

where \( \epsilon \equiv \frac{1}{l} \). Then since the web normally has a high modulus, \( G \) is very large, which implies that \( \epsilon \) is very small. Thus the web system has a two time scale structure, the very fast dynamics of the web tension components caused by the high value of the web modulus and the relatively very slow dynamics of the velocities components. This implies that further simplification can be obtained using singular perturbation methods.

The Reduced Slow Dynamic Model

Consider the singular perturbed system

\[ \begin{align*}
\dot{x} &= f(t, x, z, \epsilon) \\
\epsilon \ddot{z} &= g(t, x, z, \epsilon)
\end{align*} \tag{2.21} \]

where \( x \in \mathbb{R}^n \) and \( z \in \mathbb{R}^m \); then in the case when \( \epsilon \to 0 \), the reduced order system

\[ \dot{x} = f(t, x, h(x), 0) \tag{2.22} \]

can be found by solving \( g(t, x, h(x), 0) = 0 \) for \( h(x) \).

Consider now the web system (2.15):

\[ \begin{align*}
\epsilon \dot{T}_j(t) &= \frac{1}{L_j} (v_j(t) - v_{j+1}(t)) + \epsilon \left[ -\frac{1}{L_j} v_j(t) T_j(t) + \frac{1}{L_j} v_{j+1}(t) T_{j+1}(t) \right], \quad j = 1, \ldots, n \\
\epsilon \dot{T}_{n+1}(t) &= \frac{1}{L_{n+1}} (v_{n+1}(t) - v_{n+2}(t)) + \epsilon \left[ -\frac{1}{L_{n+1}} v_{n+1}(t) T_{n+1}(t) + \frac{1}{L_{n+1}} v_{n+2}(t) T_{n+2}(t) \right] 
\end{align*} \tag{2.23} \]

In this case, it can be easily observed that if \( \epsilon = 0 \), then the solution to (2.23) is given
by:

\[ v_1(t) = \cdots = v_i(t) = \cdots = v_{n+2}(t) = v, \quad i = 2, \cdots, n + 1 \quad (2.24) \]

This result reflects the general fact that when the stiffness of a web material is extremely high, the web material can be treated as rigid body, and hence all the web spans move with the same speed.

In this case, the dynamic equations for the velocities of the system are reduced to a single dynamic equation, i.e., \( v_i = v, \quad i = 1, 2, \cdots, n + 2 \) and the dynamic equations of the tensions are represented as only algebraic relationships.

A state space representation for the reduced system can be easily obtained in this case.

In (2.15), denote

\[
\begin{align*}
\beta_i & \triangleq \frac{r^2_i}{\bar{J}_i}, \quad i = 1, \cdots, n + 2 \\
\Phi_i & \triangleq \frac{r_i}{\bar{J}_i}, \quad i = 1, \cdots, n + 2 \\
\Delta_1 & \triangleq \left( \frac{\delta}{\pi r_1^2} - \frac{b_1}{\bar{J}_1} \right) \\
\Delta_{n+2} & \triangleq \left( \frac{\delta}{\pi r_{n+2}^2} - \frac{b_{n+2}}{\bar{J}_{n+2}} \right) \\
\Delta_i & \triangleq -\frac{b_i}{\bar{J}_i}, \quad i = 2, \cdots, n + 1
\end{align*}
\] (2.25)

then on letting \( v_i = v, \quad i = 1, 2, \cdots, n + 2 \), we obtain:

\[
\begin{align*}
\frac{dv(t)}{dt} & = -\beta_1 T_1(t) + \Delta_1 v(t) + \Phi_1 U_1(t) \\
\frac{dv(t)}{dt} & = \beta_i T_{i-1}(t) - \beta_i T_i(t) + \Delta_i v(t) + \Phi_i U_i(t), \quad i = 2, \cdots, n + 1 \\
\frac{dv(t)}{dt} & = \beta_{n+2} T_{n+1}(t) + \Delta_{n+2} v(t) + \Phi_{n+2} U_{n+2}(t)
\end{align*}
\] (2.26)

For every equation in (2.26), divide \( \frac{dv(t)}{dt} \) by \( \beta_i \) and add all of the resulting equations together; we hence obtain:
The output equation can be found by setting the right hand sides to be equal for all of the equations in (2.26). However we will end up with a very messy expression with no indication of the physical meaning. To overcome this problem, this equation will be derived later as a simplification of the high order model.

From (2.27), we can see that the $(2n+3)$th order system has been simplified to a first order RC circuit as given in fig 2.8: where

\[
\begin{align*}
\frac{dv(t)}{dt} &= \frac{\sum_{i=1}^{n+2} \frac{\partial}{\partial t}}{\sum_{i=1}^{n+2} \frac{1}{R_i}} u(t) + \left[ \frac{\phi_1}{\sum_{i=1}^{n+2} \frac{1}{R_i}} \cdots \frac{\phi_{n+2}}{\sum_{i=1}^{n+2} \frac{1}{R_i}} \right] \begin{bmatrix} U_1(t) \\ \vdots \\ U_{n+2}(t) \end{bmatrix} \\
&= \begin{bmatrix} U_1(t) \\ \vdots \\ U_{n+2}(t) \end{bmatrix}
\end{align*}
\] (2.27)

The output equation can be found by setting the right hand sides to be equal for all of the equations in (2.26). However we will end up with a very messy expression with no indication of the physical meaning. To overcome this problem, this equation will be derived later as a simplification of the high order model.

From (2.27), we can see that the $(2n+3)$th order system has been simplified to a first order RC circuit as given in fig 2.8: where

\[
\begin{align*}
C &\triangleq \sum_{i=1}^{n+2} \frac{1}{R_i} \\
Y &\triangleq -\sum_{i=1}^{n+2} \frac{\partial}{\partial t} \\
S_i &\triangleq \frac{\phi_i}{R_i} U_i(t), \quad i = 1, \cdots, n+2
\end{align*}
\] (2.28)

with $Y$ being the conductance, the reciprocal of the resistance.

The reduced system (2.27) has a very simple form. However, it does not capture the high frequency information in the system. To completely understand the stability properties of the system, we should include the dominant high frequency effects of the web system. To do this, we need to look at a more complicated model containing
high frequency information.

High Frequency Model of the Web System

To derive a high frequency model of the web system, consider the nonlinear dynamic equation for tensions in the web given by (2.20):

\[
\epsilon \dot{T}_j(t) = \frac{1}{L_j} (v_j(t) - v_{j+1}(t)) + \epsilon \left( -\frac{1}{L_j} v_j(t) T_j(t) + \frac{1}{L_j} v_{j+1}(t) T_{j+1}(t) \right), \, j = 1, \ldots, n
\]

\[
\epsilon \dot{T}_{n+1}(t) = \frac{1}{L_{n+1}} (v_{n+1}(t) - v_{n+2}(t)) + \epsilon \left( -\frac{1}{L_{n+1}} v_{n+1}(t) T_{n+1}(t) + \frac{1}{L_{n+1}} v_{n+2}(t) T_\omega(t) \right), \quad \frac{N_{n+1}(t)}{N_{n+1}(t)}
\]

(2.29)

where \( \epsilon = \frac{1}{\sigma} \).

Unlike the singular perturbation analysis approach, we now want to keep the detailed dynamic behavior of the tension terms. It is observed that because of the very small value of \( \epsilon \), the nonlinear components of the tension \( \epsilon N_j(t) \) and \( \epsilon N_{n+1}(t) \), \( j = 1, 2, \ldots, n + 1 \) of (2.29) is very small, i.e.

\[
\epsilon N_j(t) \ll \frac{1}{L_j} (v_j(t) - v_{j+1}(t))
\]

\[
\epsilon N_{n+1}(t) \ll \frac{1}{L_{n+1}} (v_{n+1}(t) - v_{n+2}(t))
\]

(2.30)

Hence as \( \epsilon \to 0 \), we can drop the nonlinear terms from the tension equations, and in this case, the tension equation (2.29) equations become linear equations:

\[
\dot{T}_j(t) = \frac{\epsilon}{L_j} (v_j(t) - v_{j+1}(t)), \quad j = 1, \ldots, n + 1
\]

(2.31)

This is just the dynamic equation for an ordinary spring. This implies that when nonlinear factors caused by changes of the density throughout web spans are negligible, a web span acts exactly as a spring.
On combining the linearized model derived before in (2.18) with (2.31), we now obtain a complete high frequency linear model:

\[
\begin{align*}
\frac{dv_1(t)}{dt} & = -\frac{r_1(t)}{J_1(t)} T_1(t) + \left(\frac{e}{\pi \tau(t)} - \frac{b_1}{J_1(t)} \right) v_1(t) + \frac{r_1(t)}{J_1(t)} U_1(t) \\
\frac{dv_i(t)}{dt} & = \frac{r_i(t)}{J_i(t)} T_{i-1}(t) - \frac{r_i(t)}{J_i(t)} T_i(t) + \left(\frac{e}{\pi \tau(t)} - \frac{b_i}{J_i(t)} \right) v_i(t) + \frac{r_i(t)}{J_i(t)} U_i(t), \quad i = 2, \ldots, n + 1 \\
\frac{dv_{n+2}(t)}{dt} & = \frac{r_{n+2}(t)}{J_{n+2}(t)} T_{n+1}(t) + \left(\frac{e}{\pi \tau(t)} - \frac{b_{n+2}}{J_{n+2}(t)} \right) v_{n+1}(t) + \frac{r_{n+2}(t)}{J_{n+2}(t)} U_{n+2}(t) \\
\dot{T}_j(t) & = \frac{C}{L_j} (v_j(t) - v_{j+1}(t)), \quad j = 1, 2, \ldots, n + 1
\end{align*}
\] 

(2.32)

Let us now associate \(\{v_i(t), T_i(t)\}, \quad i = 1, 2, \ldots, n + 1\) and \(\{v_{n+2}, T_w\}\) as corresponding to single module terms \(M_i\):

\[
M_i : \quad \frac{dv_i(t)}{dt} = \frac{r_i(t)}{J_i(t)} T_{i-1}(t) - \frac{r_i(t)}{J_i(t)} T_i(t) - f_i v_i(t) + \frac{r_i(t)}{J_i(t)} U_i(t) \\
\dot{T}_i(t) = \frac{C}{L_i} (v_i(t) - v_{i+1}(t)) \quad i = 1, 2, \ldots, n + 1
\]

(2.33)

where \(f_i = \frac{b_i}{J_i(t)} - \frac{e}{\pi \tau(t)}\), for \(i = 1, n + 2\) and \(f_i = \frac{b_i}{J_i(t)}\), for \(i = 2, \ldots, n + 1\).

We can describe these two equations by a capacitor-inductor-resistor circuit of figure 2.9.

![Figure 2.9: A High Frequency Model Module](image-url)
where:

\[
\begin{align*}
I_i &= \frac{L_i}{s} \\
C_i &= \frac{1}{s^2} \\
Y_i &= \frac{1}{s} f_i \\
S_i &= \frac{1}{s} U_i(t)
\end{align*}
\]  

(2.34)

From the diagram of the module of a single model (figure 2.9), we can see that web tension is an output from one module which becomes an input to the next module, which agrees with the well known fact that Web tension is transferred along the web spans. Every single module is a second order LCR ladder circuit.

The complete web system then is a series connection of these modules, starting from the roller 1 to roller \(n + 2\) as given in figure 2.10.

Alternately, it can be represented the block diagram 2.11.

Let us now return to the reduced system. It is obvious that the reduced slow system is only a simplification when the modulus \(G\) is set to \(\infty\) \((\epsilon \to 0)\), in which case the inductance \(L_i\) is 0. In this case, the resistors and capacitors form a simple parallel connection, and hence they can be lumped together forming the \(C\) and \(Y\) as given in figure 2.8.

We are now ready to derive the output equation for the reduced slow model.
Figure 2.11: Block Diagram of the High Frequency Model

Figure 2.12: The derivation of the model for the reduced order system
As shown in part C of figure 2.12, the output tension $T_i$ can be expressed as:

$$T_i = C_{i,all} \dot{v} + Y_{i,all} v - S_{i,all}$$  \hspace{1cm} (2.35)$$

where $C_{i,all}$, $Y_{i,all}$ and $S_{i,all}$ are the lumped capacitance, conductance and current sources from the module $i+1$ to module $n+2$ respectively, hence:

$$C_{i,all} = \sum_{k=i+1}^{n+2} C_k$$  
$$Y_{i,all} = \sum_{k=i+1}^{n+2} Y_k$$  
$$S_{i,all} = \sum_{k=i+1}^{n+2} S_k$$  \hspace{1cm} (2.36)$$

On carrying out a Laplace transform on (2.35), we obtain:

$$T_i = (sC_{i,all} + Y_{i,all}) v - S_{i,all}$$  \hspace{1cm} (2.37)$$

Perform the same procedure, we can also get that:

$$v = \frac{1}{(sC+Y)} S$$  \hspace{1cm} (2.38)$$

where $C$, $Y$ and $S$ are the overall lumped capacitance, conductance and current source from the module 1 to module $n+2$ respectively, given by:

$$C = \sum_{i=1}^{n+2} C_i$$  
$$Y = \sum_{i=1}^{n+2} Y_i$$  
$$S = \sum_{i=1}^{n+2} S_i$$  \hspace{1cm} (2.39)$$
On applying equation (2.38) to (2.37), we obtain:

\[
T_i = \frac{\dot{X}_{i-all} + Y_{i-all}}{\dot{S} + Y} S - S_{i-all} \\
= (Y_{i-all} - \frac{Y_{i-all}}{C}) \frac{1}{\dot{S} + Y} S + \frac{C_{i-all}}{C} S - S_{i-all} \\
= (Y_{i-all} - \frac{Y_{i-all}}{C}) v + \left( \frac{C_{i-all}}{C} \right) \left( S_1 + \cdots + \frac{C_{i-all}}{C} S_i \right) \\
+ \left( \frac{C_{i-all}}{C} - 1 \right) S_{i+1} + \cdots + \left( \frac{C_{i-all}}{C} - 1 \right) S_{n+2} \\
\text{n+2-i terms}
\]

(2.40)

On writing the output equation described by (2.40) for the reduced slow system in matrix form, and on applying the relationship \( S_j = \frac{1}{r_j} U_j(t) \), we obtain:

\[
\begin{bmatrix}
T_1 \\
\vdots \\
T_i \\
\vdots \\
T_{n+1}
\end{bmatrix}
= \begin{bmatrix}
C_1 \\
\vdots \\
C_i \\
\vdots \\
C_{n+1}
\end{bmatrix}
\begin{bmatrix}
D_{11} & \cdots & D_{1j} & \cdots & D_{1(n+2)} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
D_{i(n+1)1} & \cdots & D_{i(n+1)j} & \cdots & D_{i(n+1)(n+2)}
\end{bmatrix}
\begin{bmatrix}
U_1 \\
\vdots \\
U_i \\
\vdots \\
U_{n+1}
\end{bmatrix}
\]

(2.41)

where

\[
C_i = Y_{i-all} - \frac{Y_{i-all}}{C} \quad i = 1, \cdots, n + 1 \\
D_{ij} = \frac{C_{i-all}}{C} \frac{1}{r_j} \quad i = 1, \cdots, n + 1, \text{ and } j = 1, \cdots, i \\
D_{ij} = (\frac{C_{i-all}}{C} - 1) \frac{1}{r_j} \quad i = 1, \cdots, n + 1, \text{ and } j = i + 1, \cdots, n + 2
\]

(2.42)

As a summary, on combining the output equation (2.41) and the dynamic equation (2.27), we then obtain the complete state space model of the low frequency model of
the web system:

\[
\dot{v} = \frac{\sum_{i=1}^{n+2} \left( \frac{\partial F_i}{\partial x} \right) + \frac{\partial T}{\partial x} + \frac{\partial D_i}{\partial x}}{\sum_{i=1}^{n+2} F_i} v \\
+ \left[ \frac{1}{\sum_{i=1}^{n+2} F_i} \cdots \frac{1}{\sum_{i=1}^{n+2} F_i} \cdots \frac{1}{\sum_{i=1}^{n+2} F_i} \right] \begin{bmatrix} U_1 & \cdots & U_j & \cdots & U_{n+2} \end{bmatrix}^T
\]

\[
\begin{bmatrix} v \\ T \end{bmatrix} = \begin{bmatrix} 1 \\ C_T \end{bmatrix} u + \begin{bmatrix} 0 \\ D_T \end{bmatrix} \begin{bmatrix} U_1 & \cdots & U_j & \cdots & U_{n+2} \end{bmatrix}^T
\]

(2.43)

where \( T, C_T \) and \( D_T \) are given in 2.41.

The modeling method proposed here provides physical insight into the system, and provides a simple systematic numerical algorithm to determine the approximate space representation of a web system when the number of web spans is large (\( \geq 3 \)).
Chapter 3

The Servomechanism Control Problem

3.1 Robust Servomechanism Problems

Let a linear time invariant system be described by the state-space model:

\[
\begin{align*}
\dot{x} &= Ax + Bu + B_0w \\
y &= Cx + Du + F_0w \\
y_m &= C_m x + D_m u + F_m w \\
e &= y - y_{ref}
\end{align*}
\]

(3.1)

where \( x \in \mathbb{R}^n \) are the states, \( u \in \mathbb{R}^m \) are the inputs, \( w \in \mathbb{R}^n \) are the unmeasurable disturbances, \( y \in \mathbb{R}^r \) are the outputs to be regulated, \( y_m \in \mathbb{R}^{m} \) are the measurable outputs, \( y_{ref} \in \mathbb{R}^r \) are the reference inputs, and \( e \in \mathbb{R}^r \) are the error signals. It is assumed that \( A, B, C, D, C_m, D_m \) are known and \( E, F \) may or may not be known.

Consider the class of reference/disturbance signals which are described by the
following differential equations:

\[
\begin{align*}
\dot{\xi}_r &= A_r \xi_r, \quad y_{\text{ref}} = C_r \xi_r, \quad \xi_r \in \mathbb{R}^{n_r}; \\
\dot{\xi}_d &= A_d \xi_d, \quad \omega = C_d \xi_d, \quad \xi_d \in \mathbb{R}^{n_d};
\end{align*}
\]  

(3.2)

where the pairs \((C_r, A_r)\) and \((C_d, A_d)\) are observable, and where \(\text{rank}(C_r) = r\) and \(\text{rank}(C_d) = \text{rank} \begin{pmatrix} E \\ F \end{pmatrix} = \Omega\). For nontriviality, we consider only the cases where \(\text{sp}(A_r) \in \mathbb{C}^+\), and \(\text{sp}(A_d) \in \mathbb{C}^+\).

Let \(\Lambda_r(s)\) and \(\Lambda_d(s)\) denote the minimal polynomials of \(A_r\) and \(A_d\) respectively. Define \(\lambda_i, i = 1, 2, \ldots, p\) to be the zeros (multiplicities included) of the least common multiple polynomial \(\Lambda(s)\) of \(\Lambda_r(s)\) and \(\Lambda_d(s)\). These zeros are also called the design frequencies of the robust servomechanism problem.

Remark 3.1 It is to be noted that common types of signals such as constant, ramp and sinusoid signals all satisfy (3.2).

It is now desired to find a robust controller which will achieve the following control objectives [Dav76b]:

1. The closed-loop system is stable.

2. Asymptotic regulation occurs (i.e., \(e(t) \to 0\) as \(t \to \infty\)) for all reference inputs \(y_{\text{ref}}\) and disturbances \(\omega\) described by (3.2), and for all plant and controller initial states.

3. Asymptotic regulation occurs for any perturbations in the plant parameters \((C, A, B, D, C_m, D_m)\) of (3.1), which do not de-stabilize the resultant perturbed system.

Such a control problem is called a robust servomechanism control problem.

Any dynamic linear controller which solves the robust servomechanism problem
is assumed to have the following structure:

\[
\begin{align*}
\dot{x}_c &= A_c x_c + B_c y_m + \hat{B}_c y_{\text{ref}} \\
u &= C_c x_c + D_c y_m + \hat{D}_c y_{\text{ref}}
\end{align*}
\] (3.3)

where \( x_c \in \mathbb{R}^{n_c} \) is the controller state, and \( u \in \mathbb{R}^m \) is the output of the controller to the input of system (3.1).

The following existence conditions are obtained to solve this problem:

**Lemma 3.1** [Dav76b] The necessary and sufficient conditions that there exists a solution to the robust servomechanism problem for (3.1) and (3.2) are that the following conditions should all hold:

1. \((A, B)\) is stabilizable.
2. \((C, A)\) is detectable.
3. \(m \geq r\).
4. The transmission zeros of \((C, A, B, D)\) do not coincide with \(\lambda_i, i = 1, 2, \ldots, q\).
5. \(y\) is contained in \(y_m\), i.e., the outputs to be regulated are measurable.

### 3.2 The Servo-compensator

#### 3.2.1 The Controller Structure

Given the design frequencies \(\lambda_i, i = 1, 2, \cdots, p\), define a matrix \(\phi \in \mathbb{R}^{p \times p}\)

\[
\phi \triangleq \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-\delta_1 & -\delta_2 & -\delta_3 & \cdots & -\delta_p
\end{bmatrix}
\] (3.4)
where $\delta_1, \delta_2, \ldots, \delta_p$ are the coefficients of the following polynomial:

$$\lambda^p + \delta_p \lambda^{p-1} + \delta_{p-1} \lambda^{p-2} + \cdots + \delta_2 \lambda + \delta_1 \triangleq \prod_{i=1}^{p} (\lambda - \lambda_i) \tag{3.5}$$

The servo-compensator for (3.1) then has the following structure:

$$\begin{align*}
\dot{\xi} &= \phi^* \xi + \beta^* e \\
e &= y - y_{ref}
\end{align*} \tag{3.6}$$

where in the above equation

$$\begin{align*}
\phi^* &\triangleq \text{diag}(\phi, \phi, \cdots, \phi) \\
\beta^* &\triangleq \text{diag}(\gamma, \gamma, \cdots, \gamma)
\end{align*} \tag{3.7}$$

where

$$\gamma = (0, 0, 0, \cdots, 1)' \quad \gamma \in \mathbb{R}^p \tag{3.8}$$

Assuming that a solution to the servomechanism problem for (3.1) exists, a controller which solves the servomechanism problem for (3.1) is then given by:

$$u = K_0 \xi + K_1 \dot{\xi} \tag{3.9}$$

The state $\dot{x}$ of (3.9) is the output of a stabilizing controller which has the following general structure

$$\begin{align*}
\dot{\eta} &= A_s \eta + B_s u_s \\
\dot{x} &= C_s \eta + D_s u_s
\end{align*} \tag{3.10}$$

where $u_s \triangleq [y_{ref}^T \ y_m^T \ \xi^T \ u^T]^T$

The servo-compensator serves to achieve the control objectives of asymptotic
tracking and robust control. The role of the stabilizing controller is to stabilize the resultant overall augmented system, obtained when the servo-compensator (3.6) is applied to (3.1) to produce:

$$
\begin{align*}
\begin{pmatrix}
\dot{x} \\
\dot{\xi}
\end{pmatrix}
&= 
\begin{pmatrix}
A & 0 \\
-\beta^* C & \phi^*
\end{pmatrix}
\begin{pmatrix}
x \\
\xi
\end{pmatrix}
+ 
\begin{pmatrix}
B \\
-\beta^* D
\end{pmatrix} u
+ 
\begin{pmatrix}
E \\
-\beta^* F
\end{pmatrix} \omega
+ 
\begin{pmatrix}
0 \\
\beta^*
\end{pmatrix} y_{ref} \\
\end{align*}
\begin{align*}
\begin{pmatrix}
y_m \\
\xi
\end{pmatrix}
&= 
\begin{pmatrix}
C_m & 0 \\
0 & I
\end{pmatrix}
\begin{pmatrix}
x \\
\xi
\end{pmatrix}
+ 
\begin{pmatrix}
D_m \\
0
\end{pmatrix} u
+ 
\begin{pmatrix}
F_m \\
0
\end{pmatrix} \omega
\end{align*}
(3.11)
$$

where, if the conditions of lemma 3.1 hold, the triple

$$
\left[ (C_{aug}, A_{aug}, B_{aug}) \right]
= 
\left[ 
\begin{pmatrix}
C & 0 \\
0 & I
\end{pmatrix}, 
\begin{pmatrix}
A & 0 \\
-\beta^* \phi^*
\end{pmatrix}, 
\begin{pmatrix}
B \\
-\beta^* D
\end{pmatrix}
\right]
(3.12)
$$

is stabilizable and detectable, and has the same fixed modes as the plant \((C, A, B)\).

Figure 3.1: The general structure of a servomechanism controller
3.2.2 Stabilizing Controller Design

Different methods can be used to design the stabilizing controller (3.10) for the servomechanism controller. In this thesis, three methods will be used: a cheap control approach, a high gain stabilizing controller approach and a tuning regulator controller approach.
Chapter 4

The Perfect Control Problem

4.1 Introduction

From the past chapter, it is clear that given a class of modeled tracking/disturbance signals (design signals) and the plant model satisfying Lemma 3.1, there exists a solution to the servomechanism problem so that asymptotic reference tracking/disturbance rejection occurs for any arbitrary plant perturbations which do not produce instability for the overall closed-loop system. The controller must contain a servocompensator and a stabilizing controller, where the design of the servocompensator is unique and the design of stabilizing controller is not. It is now desired to solve the following type of problem: given a class of unmodeled signals which lie outside of the design signals, find a stabilizing controller so that arbitrary good approximate error regulation occurs and arbitrarily good transient response occurs for all the bounded initial conditions of the plant and controller. The above problem is called the Robust Servomechanism Problem with Perfect Control (RSPPC)[DS87].

4.2 The Perfect Control Problem

4.2.1 Development of the Problem

A description of the RSPPC will be defined here.
CHAPTER 4. THE PERFECT CONTROL PROBLEM

Given a plant described by (3.1), consider the class of unmodeled reference/disturbance signals which are linear combinations of signals of the following type:

\[ y_{ref} = t^{k_r}e^{\lambda_r t}\bar{y}_{ref} \]
\[ \omega = t^{k_\omega}e^{\lambda_\omega t}\bar{\omega} \]

where \( k_r, k_\omega \) are nonnegative integers, \( \lambda_r \in \mathbb{C}^+ \) and \( \lambda_\omega \in \mathbb{C}^+ \) occur in conjugate pairs, and \( \bar{y}_{ref} \in R^n, \bar{\omega} \in R^n \).

Instead of considering a single feedback controller (3.9) and a single stabilizing controller (3.10), consider now a family of controllers parameterized by a single parameter \( \epsilon \), i.e.

\[ u = K_{\xi}^\epsilon \dot{x} + K_{\xi}^\epsilon \xi \]  

where \( \xi \) is the output the servo-compensator (3.6) and \( \dot{x} \) is the output of a stabilizing controller:

\[ \dot{\eta} = A_s^\epsilon \eta + B_s^\epsilon u_s \]
\[ \dot{x} = C_s^\epsilon \eta + D_s^\epsilon u_s \]

where \( u_s = [y_{ref}^T \ y_m^T \ \xi^T \ u^T]^T \).

Assume now that the conditions of Lemma 3.1 are satisfied and that for each fixed \( \epsilon > 0 \), a servomechanism controller described by (4.2) to (4.3) has been found to solve the robust servomechanism problem for the design frequencies described by (3.1).

Assume now that the unmodeled reference/disturbance signals (4.1) are applied to the resultant closed loop system; it is desired now that for any fixed \( \lambda \in \mathbb{C}^+ \) in the signals (4.1) the closed-loop system should possess the following properties [DS87]:

1. Achieving Arbitrarily Good Approximate Error Regulation (AGAER), i.e. \( \forall \delta > 0, \forall \Delta > 0 \) and all initial conditions \( x(0), \xi(0), \eta(0), \xi_\omega(0), \xi_m(0) \) located on their respective unit spheres, there exists a \( \epsilon > 0 \) and \( T > 0 \) such that the resultant steady-state error, which is denoted by \( e_{ss}(t) \), has the properties that \( ||e_{ss}(t)|| < \delta, \forall t \in [T, T + \Delta] \).
2. Achieving Perfect Control,

Let the output of the resultant closed loop system be denoted by:

\[ y(s) = K^s_r(s)y_{ref}(s) + K^s_\omega(s)\omega(s) + H^s_2(s)x(0) + H^s_\xi(s)\xi(0) + H^s_\eta(s)\eta(0) \]  

(4.4)

where \( K^s_r(s), K^s_\omega(s), H^s_2(s), H^s_\xi(s), H^s_\eta(s) \) are the corresponding transfer functions obtained after applying the servomechanism controller (4.2) and (4.3) to the plant (3.1). It is then desired that, in the point-wise convergence sense:

(a) \( \lim_{\epsilon \to 0} K^s_r(s) = I \), i.e., perfect decoupled tracking occurs.

(b) \( \lim_{\epsilon \to 0} K^s_\omega(s) = 0 \), i.e., perfect disturbance rejection occurs.

(c) perfect initial condition response occurs, i.e.

i. \( \lim_{\epsilon \to 0} H^s_2(s) = 0 \)

ii. \( \lim_{\epsilon \to 0} H^s_\xi(s) = 0 \)

iii. \( \lim_{\epsilon \to 0} H^s_\eta(s) = 0 \)

(d) No unbounded peaking occurs as \( \epsilon \to 0 \), i.e.,

\[ \sup_{\epsilon > 0}\sup_{t \in [0, T]} e^{f}(t)e^{f}(t) < \infty, \forall t < \infty \]

When \( \lim_{\epsilon \to 0} H^s_\xi(s) = 0 \) does not hold, the problem is called a \textit{robust servomechanism problem with perfect control}, otherwise it is called a \textit{robust servomechanism problem with complete perfect control}.

4.2.2 Design for Perfect Control

It is shown in [DS87] that there exists no solution to the \textit{robust servomechanism problem with complete perfect control}. However, there exists a solution to the \textit{robust servomechanism problem with perfect control} provided the following conditions all hold:

\textbf{Lemma 4.1} [DS87] \textit{Given the plant (3.1), there exists a solution to the RSP with perfect control if and only if the following conditions all hold:}
1. \((C_m, A)\) is detectable.

2. \((C, A, B, D)\) is minimal phase and \(m \geq r\).

3. \(y\) is contained in \(y_m\).

Perfect Controller Design Using Cheap Control

Assume now that the above conditions hold for a given plant. Consider the system in which the plant (3.1) is augmented with the servocompensator (3.6):

\[
\begin{align*}
\begin{pmatrix}
\dot{x} \\
\dot{\xi}
\end{pmatrix} &=
\begin{pmatrix}
A & 0 \\
-\beta^*C & \phi^*
\end{pmatrix}
\begin{pmatrix}
x \\
\xi
\end{pmatrix} +
\begin{pmatrix}
B \\
-\beta^*D
\end{pmatrix} u +
\begin{pmatrix}
E \\
-\beta^*F
\end{pmatrix} \omega +
\begin{pmatrix}
0 \\
\beta^*
\end{pmatrix} y_{ref}
\end{align*}
\]

\((4.5)\)

Define \(D \in R^{r \times r_p}\) to be

\[
D \triangleq \text{block diag} \left(\alpha, \alpha, \ldots, \alpha\right)
\]

\((4.6)\)

where \(\alpha \triangleq (1, 0, \ldots, 0)\), and

Define \(z \in R^r\) to be

\[
z = (0 \, D)
\begin{pmatrix}
x \\
\xi
\end{pmatrix}
\]

\((4.7)\)

Then we form the system \((\hat{C}, \hat{A}, \hat{B})\), where:

\[
\begin{align*}
\hat{A} & \triangleq 
\begin{pmatrix}
A & 0 \\
-\beta^*C & \phi^*
\end{pmatrix} \\
\hat{B} & \triangleq 
\begin{pmatrix}
B \\
-\beta^*D
\end{pmatrix} \\
\hat{C} & \triangleq (0 \, D)
\end{align*}
\]

\((4.8)\)
A perfect controller is now found by finding a feedback controller

$$u = K_1^e x + K_2^e \xi$$

(4.9)

to minimize the following performance index

$$J_e = \int_0^\infty (x' z + e u' u) dt$$

(4.10)

where $e > 0$ is a scalar.

It is well known that a solution to the above control problem is given by:

$$K^e \triangleq (K_1^e, K_2^e) = -\frac{1}{e} \hat{B} P_e$$

(4.11)

where $P_e$ is the unique positive semidefinite solution of the Algebraic Riccati equation

$$\hat{A}' P_e + P_e \hat{A} + C'C = \frac{1}{e} P_e B B' P_e$$

(4.12)

The perfect controller is then implemented by using an observer, i.e.,

$$u = K_1^e \hat{x} + K_2^e \xi$$

(4.13)

where:

$$\dot{\hat{x}} = (A - k_o C_m) \hat{x} + k_o(y_m - D_m u) + Bu$$

(4.14)

where $K_o$ is the observer gain found so that $A - K_o C_m$ is stable.

Other Perfect Controllers

The Design of the perfect controller is not unique. In the next chapter, a new type of high gain controller design [ZD94a] for minimum phase plants, which requires only information regarding the system’s infinite transmission zeros (IZ) will be introduced.
Chapter 5

A High Gain Stabilizing controller

5.1 Introduction

High gain controllers have been widely used as stabilizing controllers because of their ability to compensate for nonlinearity and uncertainties in the system. Also, because the amplitude of the sensitivity function can be significantly reduced, disturbance rejection of the closed loop system can be greatly improved.

5.2 Mathematical Preliminaries

5.2.1 Transmission Zeros

Let \( G(s) \) be a rational transfer function matrix; then it can be transformed to the Smith-McMillan form.

\[
M(s) = \text{diag} \left[ \frac{\epsilon_1(s)}{\psi_1(s)}, \frac{\epsilon_2(s)}{\psi_2(s)}, \ldots, \frac{\epsilon_r(s)}{\psi_r(s)}, 0, \ldots, 0 \right]
\]  

(5.1)

where \( \epsilon_i(s), \psi_i(s) \) are polynomials with the property that for all \( i = 1, 2, \ldots, r \), \( \epsilon_{i-1}(s) \) divides \( \epsilon_i(s) \) and \( \psi_{i-1}(s) \) divides \( \psi_i(s) \).
The transmission zeros are defined as the zeros of polynomial

\[ Z(s) = \epsilon_1(s)\epsilon_2(s) \cdots \epsilon_r(s) \]  

(5.2)

If the transmission zero has multiplicity \( \delta \), it is said to be a transmission zero of order \( \delta \). In the following chapters, a transmission zero will be simply called a zero.

5.2.2 Transmission Zero at Infinity \([\text{Mac89}]\)

Let \( H(\lambda) \triangleq G(\frac{1}{\lambda}) \); then the infinite transmission zeros of \( G(s) \) of order \( \sigma \) at \( s = \infty \) are defined to be the zeros of \( H(s) \) of order \( \sigma \) at \( s = 0 \). In the following chapters, an infinite transmission zero will be denoted by \( \text{IZ} \).

5.2.3 Minimum Phase System

A system is called minimum phase if none of its transmission zeros lies in the closed right hand plant \( \mathcal{C}^+ \).

5.2.4 Hurwitz Polynomial

A \( n \)th order real coefficient polynomial on \( \lambda \)

\[ f(\lambda) = a_0 \lambda^n + a_1 \lambda^{n-1} + \cdots + a_{n-1} \lambda + a_n \]  

(5.3)

is called a Hurwitz polynomial if all of its zeros have negative real parts.

5.3 The multivariable root locus

In the SISO case, the eigenvalues of a closed loop system are very easy to analyze for high gain feedback. However, for a multivariable system, due to the multi-input and multi-output geometrical structure, the behavior of the closed loop system eigenvalues, as the feedback gain increases, is much more complicated than for the SISO
To study the behavior of multivariable systems under high gain, it is convenient to study the multi-variable root locus of the system.

It is well known that for a SISO strictly proper system with $n$ poles and $m$ zeros, the root loci begin at the open-loop poles, and $m$ branches of the root loci terminate at the finite zeros of the system, with the other $n-m$ branches going to infinity along asymptotes with angles of:

$$\alpha_i = \frac{2(i+1)}{n-m} \pi, \ i = 0, 1, \cdots, n-m-1$$  \hspace{1cm} (5.4)

which intersect at the point

$$h = \frac{(\text{sum of open loop poles})-(\text{sum of open loop zeros})}{n-m}$$  \hspace{1cm} (5.5)

However when one studies the multivariable case, the root locus plot is not this straightforward. The following is a plot of the root locus of a strictly proper square multivariable system, whose complete description of this system is described in [KS76].

![Figure 5.1: A Generic Multivariable Root Loci](image)

In the above plot, we observe that a multivariable system can have IZs of multiple
orders. The root loci approach zeros of different orders with different rates in a Butterworth pattern.

For systems with infinite zeros of order larger than 1, there may be families of root loci approaching such infinite zeros. For example, if a system has \( v \) \( i \)th order zeros, then for every \( i \)th order IZ, say, \( j \)th \( i \)th order zero, the root loci approach this infinite zero (IZ) along asymptotes satisfying the following equation defining a Butterworth Pattern

\[
s = [−λk]^{\frac{1}{i}}
\]

where \( λ \) is some complex constant.

In this case, there are \( i \) branches of root loci. They approach infinity with a rate of \( |(λk)^{\frac{1}{i}}| \), and have angles with the positive axis of

\[
α = \frac{1}{i}[\text{Arg}(−λ) + 2tπ], \quad t = 0, 1, \cdots, i - 1
\]

The center (called a pivot in [KE79]) of the above rays of radiation does not necessarily lie on the real axis and may come in complex conjugate pairs[KE79], which is very different from the SISO case.

It is clear that whenever Butterworth Patterns of order higher than 2 occurs in a system, this system is unstable at high gain feedback. Hence the main design objective in high gain controller design is to eliminate the occurrence of high order Butterworth Patterns. We want the closed loop system root loci to approach the finite zeros of the minimal phase system or/and go off to infinity in the open left hand complex plane, i.e., the closed loop system can have only IZ of order 1.

### 5.4 Controller Synthesis

The controller will be designed for square system, i.e., the number of inputs \( m \) of a system is equal to the number of the outputs \( r \). Recognizing the fact that a non-square invertible and minimal phase system can be squared down to a square minimal
5.4.1 Approach 1–Decomposition

The structure of the IZs of a given system is the most important information in high gain controller design. This structure can be constructed by the spectrum decomposition or the singular value decomposition.

Decomposition of the Transfer Function Matrix

Theorem 5.1 [ZD94b] Given an invertible system \((C, A, B)\), there exist two non-singular matrices \(P, Q \in \mathbb{R}^{n \times m}\) such that:

\[
PC = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_h \end{bmatrix}, \quad BQ = [B_1 \ B_2 \cdots \ B_h]
\]

where \(C_i \in \mathbb{R}^{n \times n}, B_i \in \mathbb{R}^{n \times n}\), \(\gamma_i = \text{rank}(C_i), \ i = 1, 2, \ldots, h\) satisfy the following conditions:

\[
C_i M_i^{j-1} B_i = 0, \ i = 1, 2, \ldots, h, \ j = 1, 2, \ldots, \sigma_i - 1;
\]

\[
\Sigma_i = C_i M_i^{\sigma_i - 1} B_i \text{ is nonsingular};
\]

\[
C_i M_i^{j-1} B_k = 0, \ i = 1, 2, \ldots, h, \ j = 1, 2, \ldots, \sigma_i, \ k > i;
\]

\[
C_i M_k^{j-1} B_k = 0, \ i = 2, 3, \ldots, h, \ j = 1, 2, \ldots, \sigma_k, \ k < i;
\]

\[
M_{i+1} = M_i - B_i K_i, \ M_1 = A;
\]

\[
K_i = \Sigma_i^{-1} C_i M_i^{\sigma_i};
\]

\[
0 < \sigma_1 < \sigma_2 < \cdots < \sigma_h
\]

where \(\sigma_i, \ i = 1, 2, \ldots, h\) are the orders of the IZs of this system.

Then from [KS76], the decomposed systems \((C_i, M_i, B_i)\) has only IZs with orders higher than \(\sigma_i, \ i = 1, 2, \cdots, h\).
CHAPTER 5. A HIGH GAIN STABILIZING CONTROLLER

The details of using this method to find the IZ structure can be found in [KS76] and [KE79]. Another method to determine the IZ structure can be found from Kailath's work in [Kai80], where a bilinear transformation is used.

Controller Structure

With these decomposition results, we can design a controller on the following system:

\[
\begin{align*}
\dot{z}(t) &= Az(t) + BQv(t) \\
w(t) &= PCz(t)
\end{align*}
\]

Denote

\[
H^* = PC(sI - A)^{-1}BQ
\]

and

\[
H = C(sI - A)^{-1}B
\]

Consider a controller structure of the following:

\[
u(s) = G_1 C(s, \epsilon) G_2 y(s)
\]

where \(G_1(s), G_2(s)\) are constant matrices.

Let \(G_1 = QG\) and \(G_2 = P\), where \(G\) is an arbitrary matrix which will be chosen later in the controller design.

Note the fact that the characteristic polynomial \(P(s)\) of the closed loop system on applying controller (5.13) to the original system (5.12) is given by:

\[
P(s) = \det(I_m - G_1 C(s, \epsilon) G_2 H(s)) = \det(I_m - GC(s, \epsilon) H^*(s))
\]
CHAPTER 5. A HIGH GAIN STABILIZING CONTROLLER

to applying a controller

\[ v(s) = GC(s, \varepsilon)w(s) \]  

(5.15)

to the decomposed system (5.10).

Choose now:

\[ C(s, \varepsilon) = \frac{1}{\varepsilon^0} \text{diag}(C_i(s, \varepsilon)I_{r_i}) \quad i = 1, 2, \ldots, h \]  

(5.16)

\( C_i(s, \varepsilon) \) will be designed to compensate for the open loop IZ while maintaining a stable proper controller and without introducing new high order IZ as \( \varepsilon \to 0 \). Define the following structure for \( C_i(s, \varepsilon) \)

\[ C_i(s, \varepsilon) \triangleq \frac{\delta_i(s)}{\prod_{j=1}^{r_i}(s^{p_j}+1)}, \quad i = 1, 2, \ldots, h \]  

(5.17)

where \( \delta_i \) is chosen to be Hurwitz polynomial of order \( \sigma_i - 1 \) and \( r_i \geq \delta_i - 1 \).

This high gain controller has the following property:

**Theorem 5.2 [ZD94b]** Consider a minimum phase square system (5.12) and the class of monic Hurwitz polynomial \( \delta_i(s) \) of degree \( \sigma_i - 1 \), \( i = 1, 2, \ldots, h \), and consider the high-gain compensator (5.19) with (5.16) and (5.17), where \( r_i \geq \sigma_i - 1 \). Choose \( p_0 = 1, p_j = 2, j = 1, 2, \ldots, h \), \( G_1 = QG, G_2 = P \), and let \( G = \text{diag}(G_1, G_2, \ldots, G_h) \) such that \( \text{sp}(\Sigma_i G_i) \subset C^{-} \); then as \( \varepsilon \to 0 \), the closed loop system is asymptotically stable, and its eigenvalues are given by (multiplicities excluded):

\[ \Lambda_i + O(\varepsilon), \quad \frac{\Lambda_i G_i}{\varepsilon} + O(1), \quad \frac{1}{\varepsilon^{p_j}} + O(1), \quad i = 1, 2, \ldots, h \]  

(5.18)

where \( \Lambda_0 \) denotes the finite transmission zeros of the open loop system \((C, A, B)\), and \( \Lambda_i \) denotes the zeros of \( \delta_i(s) \).

5.4.2 Approach 2–Generalized Differential Interactor
Dynamic Feedback Transformation

Instead of decomposing the system and tweaking the system not to have high order IZ, a simple substitution is to find a dynamic invariant feedback to make the system have only 1st order IZ. To do this, let us introduce the concept of a generalized differential interactor [ZD94a].

**Definition 5.1** A $m \times m$ polynomial matrix $\tilde{X}_H(s)$ is called the (right) generalized differential interactor of $H(s)$ if it has the following structure

$$\tilde{X}_H(s) \triangleq \tilde{\Delta}(s)\tilde{\Gamma}(s) \quad (5.19)$$

where $\tilde{\Delta}(s) \triangleq \text{diag}(\delta_1(s), \delta_2(s), \ldots, \delta_m(s))$, with $\delta_i(s)$ a monic real-coefficient Hurwitz polynomial of degree $\delta_i - 1$, $i = 1, 2, \ldots, m$, and where $\tilde{\Gamma}(s)$ is an upper triangular polynomial matrix with an integer 1 on the diagonal, such that

$$\lim_{s \to \infty} H(s)\tilde{X}_H(s) = 0$$

$$\lim_{s \to \infty} sH(s)\tilde{X}_H(s) = \tilde{K}_H \quad (5.20)$$

where $\tilde{K}_H$ is a non-singular constant matrix.

A detailed way of determining the interactor $\tilde{X}_H(s)$ has been given in [ZD94a].

The definition 5.1 implies that the following holds:

**Theorem 5.3** Given a non-singular $m \times m$ strictly proper transfer function matrix $H(s)$, then there always exists a generalized differential interactor $\tilde{X}_H(s)$ such that

$$H(s)\tilde{X}_H(s) = C(sI_n - A)^{-1}\tilde{B} \quad (5.21)$$

where $\tilde{B}$ is a $n \times m$ constant and $C\tilde{B}$ is nonsingular.

The fact that $C\tilde{B}$ is full rank states that this system can have only $m$ 1st order IZ and no other higher order IZ. Denote the overall order of the system $(C, A, \tilde{B})$ to
be $\bar{n}$; then it has $\bar{n} - m$ finite zeros, and they are given by

$$\Lambda_0, \Lambda_1, \Lambda_2, \cdots, \Lambda_m$$  \hspace{1cm} (5.22)

where $\Lambda_0$ denotes the finite zeros of the system $(C, A, B)$ and $\Lambda_1, \Lambda_2, \cdots, \Lambda_m$ are the set of the zeros of the Hurwitz polynomial $\delta_1(s), \delta_2(s), \cdots, \delta_m(s)$ respectively.

The above property of the system $\tilde{\Sigma} \triangleq (C, A, \tilde{B})$ makes the high gain controller design for $\tilde{\Sigma}$ very easy. A static controller

$$\tilde{u} = \frac{1}{\varepsilon}Gy(s)$$  \hspace{1cm} (5.23)

where $\tilde{u}$ is the input to $\tilde{\Sigma}$ and $G$ is chosen such that $\lambda(C\tilde{B}G) \in C^-$, will suffice.

This controller has the following property:

Lemma 5.1 [ZD94a] Given a minimum phase square system $(C, A, B)$, consider the transformed system $(C, A, \tilde{B})$ defined in theorem 5.3; then after applying the controller (5.23), where $\lambda(C\tilde{B}G) \in C^-$, the closed loop system

$$\dot{z} = Ax + \tilde{B}\tilde{u}$$  
$$y = Cz$$  
$$\tilde{u} = \frac{1}{\varepsilon}Gy$$  \hspace{1cm} (5.24)

has the property that as $\varepsilon \to 0$, it is asymptotically stable, and its eigenvalues are given by

$$\Lambda_i + O(\varepsilon), \ i = 1, 2, \cdots, m; \ \frac{\lambda(C\tilde{B}G)}{\varepsilon} + O(1)$$  \hspace{1cm} (5.25)

where $O(\varepsilon)$ denotes the order of the small scaler $\varepsilon$.

The result (5.25) results from singular perturbation theory [Kok86]. Using the same theory, Zhang and Davison in [ZD94a] have proposed a dynamic high gain
feedback controller of the form:

$$\bar{u} = \frac{1}{\epsilon} \text{diag} \left( \frac{1}{(r_i^2 + 1)^{r_i}} \right) G_y(s) \quad i = 1, 2, \ldots, m$$ \hspace{1cm} (5.26)

where $r_1, r_2, \ldots, r_m$ are positive integers. This controller results in an asymptotically stable closed loop system with a time scale of 3.

**Lemma 5.2** Given a minimum phase square system, consider the system $(C, A, \hat{B})$ as defined in (5.21), and apply the controller (5.26), where $\lambda(C\hat{B}G) \in C^{-}$; then as $\epsilon \to 0$, the closed loop system is asymptotically stable and the closed loop eigenvalues are given by

$$\lambda_i + O(\epsilon), \quad i = 0, 1, \ldots, m;$$

$$\frac{\lambda(C\hat{B}G)}{\epsilon} + O(1);$$

$$\frac{\lambda(A_{ni})}{\epsilon^2} + O(1), \quad i = 1, 2, \ldots, m$$ \hspace{1cm} (5.27)

where the $R^{r_i \times r_i}$ matrix $A_{ni}$ is defined as the

$$A_{ni} \triangleq \begin{bmatrix}
-1 & 1 & 0 & \cdots & 0 & 0 \\
0 & -1 & 1 & \cdots & 0 & 0 \\
0 & 0 & -1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -1 & 1 \\
0 & 0 & 0 & \cdots & 0 & -1
\end{bmatrix}$$ \hspace{1cm} (5.28)

### 5.5 Conclusion

The previous controllers described in chapters 4 and 5 have the property that they require either: (a) a "satisfactory" complete model describing the plant, or (b) a knowledge of some aspects of the plant, e.g. that the plant be minimum phase and a knowledge of the Markov parameters of the plant. It is often difficult in industrial control problems to determine such \textit{a priori} knowledge, and the question arises: can one still design a controller to solve the RSP when this knowledge is not available? In
[Dav76b], Davison proposed a type of multivariable tuning regulator which can solve the RSP provided certain mild conditions hold.
Chapter 6

Tuning Regulator Controller

As discussed before, to solve the robust servomechanism problem using the servocompensator, two design steps should be carried out: the servocompensator design and the stabilizing controller design. The servocompensator design is based only on the signals to be tracked/rejected, and hence is independent of the plant information. Plant information is only needed for stabilizing the overall system. Let us examine what information is required to accomplish this.

Given a plant:

\[
\begin{align*}
\dot{x} &= Ax + Bu + Ew \\
y &= Cx + Du + Fw \\
e &= y - y_{ref}
\end{align*}
\]

where \( y_m \equiv y \), then from Lemma 3.1, the conditions for a solution to exist to solve the robust servomechanism problem are:

1. \((A, B)\) is stabilizable.

2. \((C, A)\) is detectable.

3. \(m \geq r\).

4. The transmission zeros of \((C, A, B, D)\) do not coincide with \(\lambda_i, \ i = 1, 2, \ldots, p\).
where $\lambda_1, \lambda_2, \ldots, \lambda_\bar{p}$ are the roots for the characteristic polynomial for the reference/disturbance signals:

$$\lambda^\bar{p} + \delta_p \lambda^{\bar{p}-1} + \delta_{\bar{p}-1} \lambda^{\bar{p}-2} + \cdots + \delta_2 \lambda + \delta_1 \triangleq \prod_{i=1}^{\bar{p}} (\lambda - \lambda_i) \quad (6.2)$$

On making the assumption that the plant is open loop stable, we can see that the first two conditions are automatically satisfied. The third and forth conditions are equivalent to the rank condition of the system matrix:

$$\text{rank} \begin{pmatrix} \lambda_i I - A & B \\ -C & D \end{pmatrix} = n + \min(m, r), \quad i = 1, 2, \ldots, \bar{p} \quad (6.3)$$

which is equivalent to the condition:

$$\text{rank}(G_{\lambda_i}) \triangleq \text{rank}(D - C(\lambda I - A)^{-1}B) = \min(m, r), \quad \forall i = 1, 2, \ldots, \bar{p} \quad (6.4)$$

where $G_{\lambda_i}$ is the steady state gain of the plant at the frequency $\lambda_i$; this condition can be experimentally determined [Dav76a].

Thus the existence of a solution to the servomechanism problem can be checked experimentally by determining the steady state gain matrices (6.4). Assuming that such a solution exists, it is then pointed out in [Dav76a] that a controller which solves the servomechanism problem is given by:

$$u = K_2 \eta \quad (6.5)$$

where $\eta$ is the output of the servo-compensator associated with $\lambda_1, \lambda_2, \ldots, \lambda_\bar{p}$, and where $K_2$ can be determined experimentally in conjunction with some one dimensional "online tuning"; moreover, when the control objective is to track/reject only constant signals, there is a very simple way to determine $K_2$.

**Lemma 6.1** [Dav76b] Given the stable system (6.1), where the number of inputs $m$ is at least equal to the number of outputs $r$, let the servocompensator $\dot{\eta} = y - \ldots$
$y_{ref}$ be applied; then if $\text{rank}(D - CA^{-1}B) = r$, there exists an $\epsilon^* > 0$, so that on applying $u = \epsilon K \eta$, where $K = (D - CA^{-1}B)^\dagger \equiv (D - CA^{-1}B)'\{(D - CA^{-1}B)(D - CA^{-1}B)\}'^{-1}$, the resultant closed loop system is stable $\forall \epsilon \in (0, \epsilon^*)$, and provides asymptotic tracking/rejection for constant signals.
Chapter 7

Controller Implementation

7.1 Introduction

In this chapter, three controller designs: cheap controller design, high gain controller design and the tuning regulator controller design will be implemented on the University of Toronto web machine.

The University of Toronto web machine is a two span machine as illustrated in fig 7.1.

![Diagram of the Rotoflex Machine](image)

Figure 7.1: The Illustrative Diagram of the Rotoflex Machine

The major components of this system consist of an unwinder, a winder, a nip and the web connecting them. In between the rewinder and the nip, and between
the nip and the unwinder, there are a number of rollers as shown in figure 1.1. In system identification and controller design, these idlers are ignored, which introduces uncertainty to the system. This uncertainty is assumed to be insignificant.

7.2 Models for Controller Implementations

7.2.1 A Reduced Order Model

The web material used in the experiments carried out in the thesis is paper, which has a very high elasticity modulus, and hence can be considered as a stiff material. The web model is assumed to be the reduced order low frequency model.

It can be verified from the modeling of chapter 2, that the web system has the structure as shown in figure 7.2.

![Diagram of Rotoflex Machine Model](Figure 7.2: The Reduced Low Frequency Model of the Rotoflex Machine)
where

\[
\begin{align*}
S_r &= \frac{U_r}{r_r} \\
S_n &= \frac{U_n}{r_n} \\
S_u &= \frac{U_u}{r_u} \\
C &= C_r + C_n + C_u = \frac{1}{r_r^2} + \frac{1}{r_n^2} + \frac{1}{r_u^2} \\
Y &= Y_r + Y_n + Y_u \approx \frac{b_r}{r_r^2} + \frac{b_n}{r_n^2} + \frac{b_u}{r_u^2} \\
C_\Delta &= \sum_{i=1}^{n} \frac{1}{r_i^2} \\
Y_\Delta &= \sum_{i=1}^{n} \frac{b_i}{r_i^2}
\end{align*}
\] (7.1)

where the subscriptions \( r \), \( n \) and \( u \) denote the rewinder, nip and unwinder respectively and the subscripts \( 1, \cdots, n \) represent the \( n \) idlers. Here \( C \) and \( Y \) denote the lumped effect of the rewinder, nip and winder, and \( C_\Delta \) and \( Y_\Delta \) denote the lumped effect of all idlers on the system. Hence we can see that the idlers add a perturbation term to the system parameters for the reduced order model.

Using the parameters in Table 7.1 obtained from the identification results in [Bor99], we are able to obtain a reduced order model (7.2) by ignoring the idlers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>meaning</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_{oa} )</td>
<td>Radius of the bare unwinder</td>
<td>0.0415m</td>
</tr>
<tr>
<td>( r_b )</td>
<td>Radius of transport roller (Nip)</td>
<td>0.0415m</td>
</tr>
<tr>
<td>( r_{ob} )</td>
<td>Radius of bare rewinder</td>
<td>0.0415m</td>
</tr>
<tr>
<td>( J_{oa} )</td>
<td>Inertia of bare unwinder</td>
<td>0.0175kgm²</td>
</tr>
<tr>
<td>( J_b )</td>
<td>Inertia of transport roller</td>
<td>0.0322kgm²</td>
</tr>
<tr>
<td>( J_{oc} )</td>
<td>Inertia of bare rewinder</td>
<td>0.0234kgm²</td>
</tr>
<tr>
<td>( b_a )</td>
<td>Damping friction coefficient of unwinder</td>
<td>( 1/254Nms )</td>
</tr>
<tr>
<td>( b_b )</td>
<td>Damping friction coefficient of transport roller</td>
<td>( 1/165Nms )</td>
</tr>
<tr>
<td>( b_c )</td>
<td>Damping friction coefficient of rewinder</td>
<td>( 1/258Nms )</td>
</tr>
<tr>
<td>( e )</td>
<td>Web thickness</td>
<td>( 0.00015/(2\times\pi) \text{ m} )</td>
</tr>
</tbody>
</table>
and fixing the radius \( r_a \) at 0.0833m.

\[
\begin{align*}
\dot{v} &= -0.16796v + [0.47961 \ 0.84649 \ 0.41162] [m_a \ m_b \ m_c]^T \\
\begin{bmatrix}
T_a \\
v \\
T_b \\
\end{bmatrix} &=
\begin{bmatrix}
0.033741 & 11.465 & -3.8615 & -1.8777 \\
1 & 0 & 0 & 0 \\
-0.34031 & 2.4895 & 4.3938 & -9.5806 \\
\end{bmatrix}
\begin{bmatrix}
m_a \\
m_b \\
m_c \\
\end{bmatrix}
\end{align*}
\] (7.2)

Since the real system has filters of the structure:

\[
F(s) = \frac{1}{0.1s+1}
\] (7.3)

connected to each of the tension outputs, such an addition of filters was applied to system (7.2) to obtain the following \( n = 3 \) model of the system:

\[
\begin{align*}
\dot{x}_1 &= [10 \ 0 \ 0.13496] [x_1] + [45.859 \ -15.446 \ -7.5109] [m_a] \\
\dot{x}_2 &= [0 \ -10 \ -1.3613] [x_2] + [9.9579 \ 17.575 \ -38.323] [m_b] \\
\dot{x}_3 &= [0 \ 0 \ -0.16796] [x_3] + [0.47961 \ 0.84649 \ 0.41162] [m_c] \\
\end{align*}
\] (7.4)

The response of the system (7.4) is compared to the response of the real plant by carrying out a set of open loop rewinder torque step input experiments at the operating point \( r_a = 0.0833m \). The plot 7.3 shows the difference between the responses of the system (7.4) and the response of the actual system for the case of a step input in the rewinder torque.

We observe from figure 7.3 that the ignored idlers have little effect on the tension responses. A step input on rewinder torque causes almost the same amplitudes of tension increase and tension responses to occur for both the model (7.4) and the
Figure 7.3: The Response of the Open loop Real System vs. the Identified Model

real system. However, on the other hand, the speed responses of the identified model and the real system differ significantly.

A step input in rewinder torque causes approximately 5 times greater change in speed response for the model (7.4) than for the real system. This is because the real system has more friction than the model (7.4). From figure 7.2, we know that that under a step increase $\delta S_1$ on the rewinder torque input, the amplitude of the resulting speed change $\delta v$ at steady state is determined by the overall friction of the system:

$$\delta v = \frac{\delta S_1}{\frac{1}{\sum_{i=1}^{n} \frac{b_i}{T_i}} + \frac{1}{T} + \frac{1}{\Delta} \approx \frac{\delta S_1}{\frac{1}{\sum_{i=1}^{n} \frac{b_i}{T_i}} + \frac{1}{T}}$$

(7.5)

With no significant difference between the friction coefficients of the external driven rollers (rewinder, nip and unwinder) and the idlers, the idlers play a significant role in (7.5) because of their large number, which results in a speed response amplitude of the model (7.4) significantly larger than the real response.

The speed response of the model (7.4) is also significantly slower than the real system. The time constant of the speed response of the real system is determined by
the overall inertia \((C + C_\Delta)\) and the overall damping \((Y + Y_\Delta)\):

\[
\tau = \frac{C + C_\Delta}{Y + Y_\Delta}
\]  \((7.6)\)

The unmodeled idlers will result in an increase in both the overall inertia and the overall damping. To determine which one has the dominant influence, we will compare the time constant of the model (7.4) with the real plant. By fitting a first order dynamic equation to the response of the real system we obtain a time constant of \(\frac{1}{0.8227}\) for the response of the real machine. This is about 5 times faster than the time constant \(\frac{1}{0.1680}\) of the model (7.4), which is determined by:

\[
\tau = \frac{C}{Y} \quad (7.7)
\]

From previous observation, we know that the steady state change of the model 7.4 is about 5 times larger than the real system, which means that \(Y + Y_\Delta\) is about 5 times larger than \(Y\). This implies that the \(C + C_\Delta\) is almost equal to \(C\). Hence, the dominant influence of the idlers is the damping \((Y_\Delta\) or the idler frictions) which they introduce to the system, rather than the inertia. This conclusion agrees with our assumption that the idlers inertia is insignificant.

Overall speaking, model (7.4) has a very good match in tension responses with the real plant. For speed response, the main difference of the real system and model (7.4) is that the real system has more damping.

The final identified system used for controller design in this thesis is obtained by fixing \(r_a = 0.0415m\) (smallest radius) and \(r_c = 0.15\) (largest radius). The \(n = 3\) system obtained in this case is called the identified system #1, which is described in Appendix B.1.

### 7.2.2 Another Model for the Web System

From the open loop step experiments, we can also apply a multivariable state space
identification method as described in [Dav99] to obtain a LTI model, by treating the web system as a black box. In this case, a 6th order system is identified which gives a better speed response match than model #1. This model is called identified model #2. (Complete information of the 6th order model can be found in Appendix B.1.)

Figure 7.4 to 7.6 show the step responses of the identified 6th order system and the real system, which have excellent agreement as compared to the case of model #1.
Figure 7.4: Comparison of the Response of the Black Box Model and the Real Machine–Rewinder Torque Step
Figure 7.5: Comparison of the Response of the Black Box Model and the Real System for Tension, Torque, and Speed.
Figure 7.6: Comparison of the Response of the Black Box Model and the Real Machine—Unwinder Torque Step
In the following sections, controllers will be designed based on both models and the results obtained will be compared.

In this section, three design methodologies: cheap control design, high-gain control design and tuning regulator design will be carried out.

### 7.3 Cheap controller design

**Choosing the Right Scaling**

It can be observed that both model #1 and #2 of the web system form a controllable and observable system. However, we note that the outputs of these models can have very different scale ranges, i.e., the tensions can vary from 0N to over 100N, whereas the machine speed varies only within ±5m/s for normal operation. To avoid numerical problems in the controller design, we therefore need to normalize the outputs.

The following transformation is made:

\[
y_{\text{new}} = \begin{bmatrix}
\frac{1}{130} & 0 & 0 \\
0 & \frac{1}{5} & 0 \\
0 & 0 & \frac{1}{130}
\end{bmatrix}
\begin{bmatrix}
T_a \\
v \\
T_b
\end{bmatrix}
\]  

(7.8)

This normalizes the magnitude of the output to lie in the interval [0 1] (see figure 7.7).

![Figure 7.7: The Web System Structure After Scaling](image-url)
Design of the Servocompensator

The reference signals to the system are assumed to be constant corresponding to the tension and velocity set points which are to be tracked. There are several main sources of disturbance to the web system:

1. Constant type offset signals. These signals include sources such as the static friction torque.

2. Periodic type disturbance signals. These signals include disturbances associated with the mechanical vibration in the web system such as the periodical disturbance arising from unbalanced rollers.

3. Stochastic type disturbance signals. These signals include signals such as the wound-in tension $T_w(t)$ from the unwinder, which according to (2.15), has very little influence on the system dynamics.

Since the frequency information of the periodic type disturbance and the spectrum information of the stochastic type disturbance are unknown, the assumption of assuming constant disturbances will only be made.

Thus, the servocompensator can be simply chosen to be:

$$\dot{\xi} = y - y_{ref}$$  \hspace{1cm} (7.9)

where $y$ is the plant output, $y_{ref}$ is the reference signal and $\xi$ is the servocompensator state.

The final closed loop system has the structure as given in figure 7.8.

where $T = \text{diag}(\frac{1}{135}, \frac{1}{5}, \frac{1}{135})$ and $K$ is found by solving the cheap control problem described in chapter 4.

7.3.1 Experimental Results—Controller #1

A controller is first designed based on the model #1, and the resulting controller is called controller #1. The following plots given in figure 7.9 and figure 7.11 give the
closed loop response for the above controller when the cheap control gain $c$ is chosen to be $1e-4$ and the observer gain $eo$ is chosen to be $1e-4$. The complete model of this controller can be found in Appendix B.1.
Figure 7.9: Output Responses to Tension Fa Step Reference Input (controller #1)
Figure 7.10: Output Responses to Tension Fb Step Reference Input (controller #1)
Figure 7.11: Output Responses to velocity Step Reference Input (controller #1)
7.3.2 Discussion of the Experimental Results Obtained From Controller #1

(a) Transient Response and Interactions

A very important criteria for controller design is the transient response time. As we can see from figure 7.9 to 7.11, the tension response has a settling time of approximately 0.3s and a speed response of less than 1s, which is much faster than the SISO PID controllers commercially used in web system control. Another criteria for controller performance is the interaction between outputs, i.e., it is desired that tension set point change in one span should not cause variation in the response of the tension of the other span or the machine speed. Also the speed set point change should cause as little variation in the tension responses. As can be seen in figures 7.9, 7.10 and 7.11, the largest interaction occurs when the speed set point is changed. However, in this case the variation of tension is approximately 1N (corresponding to less than 5% of the tension set points), which is negligible by the industrial criteria of 25% of the tension set point values.

(b) Steady State Variations

The maximum variation of tension at steady state is approximately 1 to 2N (corresponding to 2% - 5% of the tension reference values), which is much lower than the required industrial criteria of 10% of the set point value.

(c) Comparison With Theoretic Results

It is interesting to compare the experimental results of the real machine with the results obtained from the simulation of the identified model #1 to determine the effect of unmodeled uncertainties on the overall closed loop system.

It is observed from figure 7.12 that the experimental results and the results obtained from the simulation on the model #1 are very similar. They have almost the same response speeds. It is interesting to note that the large uncertainty of system dynamics caused by the idlers as discussed in section 7.2.1 has very little effect on the
Figure 7.12: Comparison of the closed loop responses of the experimental results and the theoretical results of the model #1 (controller #1)

closed loop system. This is because that the large uncertainties in system dynamic matrix $A$ are very small compared to the feedback terms.

(d) Further Discussion—Control of Nonlinear System

To use linear control design for nonlinear systems, the general practice is to treat the nonlinear system as a family of LTI systems at several operating points, which is indexed by some state variables and possibly some exogenous parameters, and then design a family of linear controllers for each of the fixed operating points, and thence use gain-scheduling. In the web system, the dominate nonlinear aspects of the web
machine can be approximated by a family of LTI systems:

\[
\begin{bmatrix}
\dot{v} \\
T
\end{bmatrix} = A(r_a) \begin{bmatrix}
v \\
T
\end{bmatrix} + B(r_a)U(t)
\] (7.10)

which is parameterized by the radius \( r_a \). A new LTI approximation of the web system (corresponding to a new value of \( r_a \)) and a new controller is then designed when the desired performance of the LTI controller can not be maintained for the closed loop nonlinear system. Figure 7.13 and 7.14 give the response of the nonlinear web system (7.10) at different operating points, using the previous designed LTI cheap control servocompensator controller #1. It is interesting to note that for all of the operating ranges, the LTI controller produces similar responses. Thus, the gain scheduling procedure described above is not necessary.
Figure 7.13: The Simulated Closed Loop Response of Tension $F_b$ for Different Operating Points (controller #1 applied to model #1)

Figure 7.14: The Closed Loop Experimental Response of Tension $F_b$ for Different Operating Points (controller #1)
From the modeling discussion of chapter 2, the scheduling variable \( r_a \) has the dynamic equation:

\[
\dot{r}_a(t) = \frac{\varepsilon v_a(t)}{2\pi r_a(t)}
\]  

(7.11)

and so the variation speed of \( r_a \) is determined by the web speed and the web thickness \( \varepsilon \). As a further check, it is shown in figure 7.15, that under the same single LTI controller \( \#1 \), the closed loop nonlinear system remains stable and does not significantly deteriorate in response for web thickness up to 5mm for the same web material.

![Figure 7.15: The Closed Loop Simulated Response of Tension Fb for Different Web Thickness (controller \#1 applied on model \#1)](image)

7.3.3 Experimental Results—Controller \#2

A cheap servomechanism controller was also designed based on the 6th order model \#2 with \( \varepsilon = 10^{-4} \) and \( \epsilon_a = 10^{-4} \). This controller is called controller \#2. Figure 7.16 to 7.18 show the experimental results obtained when this controller is applied to the web machine. A complete description of this controller is given in Appendix B.1.
Figure 7.16: Experimental Results of Response To Tension Fa Step Input (Controller #2)
Figure 7.17: Experimental Results of Response To Tension Fb Step Input (Controller #2)
We can observe in this case that the controller #2 based on the $n = 6$ model #2 provides excellent tracking for the closed loop system. Interestingly, however, the response is not better than the controller #1 designed based on the $n = 3$ model #1.

### 7.4 High Gain Controller Design

#### 7.4.1 The Properties of the Web System

The high gain controller is designed based on the reduced order low frequency model, which is a minimum phase system. Hence the high gain controller design methodology introduced in chapter 5 can be applied.
As a further observation, we observe that the web system to be controlled (model #1) has a full rank $CB$ matrix, which means that it has no finite transmission zeros and has only 3 first order infinite transmission zeros, with no higher order infinite transmission zeros.

### 7.4.2 Design of The Saturation Controller

A large obstacle for high gain controller implementation is the input saturation problem, i.e., as the loop gain becomes larger, the plant inputs become larger than which the real physical plant can provide. Given now the high gain controller design introduced in Chapter 5, assume that the high gain controller is chosen to be the form

$$\hat{u} = \frac{1}{s}G(s)e(s)$$

as introduced in (5.23) of section 5.4.2. This implies that when the web system to be controlled is started from rest ($y(0) = 0$), the control input amplitude will become unbounded as $\epsilon \to 0$.

The input saturation problem is a common issue for all controller designs. However it is of particular importance in the high gain control design because unlike other design methodologies such as the cheap control design, decreasing the loop gain may result in an unstable closed loop system for the high gain design. Clamping the control input signal also may not solve the problem because this will not guarantee overall stability or asymptotic tracking of the system.

To resolve this problem, we will design a cascaded saturation controller and high gain controller structure as indicated in figure 7.19.

The saturation controller is chosen to be the following [DJ91]:

$$\begin{align*}
\dot{x}_1 &= y - y_{ref} \\
z &= \epsilon_t(-\bar{D} - \bar{C}\bar{A}^{-1}\bar{B})^{-1}x_1 + \bar{C}\bar{A}^{-1}\hat{x}
\end{align*}$$

where $\hat{x}$ is the observed state of the composite system consisting of the high gain controller and the plant, and $\bar{A}, \bar{B}, \bar{C}$ and $\bar{D}$ are the system matrices of the same system.
Figure 7.19: The Overall Structure of a High Gain Design – Saturation Controller

(see figure 7.19). $z$ is the saturation controller output, and $\epsilon_l > 0$ is the saturation controller gain. It can be noted that in this saturation controller design, a servocompensator is included, and hence closed loop asymptotic tracking is guaranteed.

Properties of the Saturation Controller

To illustrate the properties of the saturation controller, it is necessary to examine the plant input under a high gain controller to see what elements should be constrained in the plant input.
Figure 7.20: Large Input Overshoot of the High Gain Controller
Figure 7.20 shows a representative plant input response of the closed loop system composed of the web system (model #1) and a high gain controller under a step tension reference set point change. It is observed from figure 7.20 that at the reference change instant, the error of the output and the reference signal becomes very large, which causes the large overshoot shown in figure 7.20.

The saturation controller (7.13) has the property that by on-line tuning the saturation controller gain $\epsilon$, we can reduce the overshoot of the plant input to allowable values. The smaller the gain $\epsilon$, the smaller the overshoot. However, reducing $\epsilon$ will also slow down the plant output response.

The detailed analysis of the properties of the saturation controller is presented in appendix A.

7.4.3 Combining The High Gain Control Design With The Saturation Controller Design

The design is carried out in two stages. First we design the high gain controller to stabilize the web system and achieve a "best" response, and then we apply a saturation controller to constrain the input amplitude.

The two stage saturation and high gain design methodology provides the freedom to achieve desirable control properties and satisfy the control input constraint. The following section gives some representative simulation and experimental results obtained using the above high gain-saturation controller design on the Rotoflex machine.

7.4.4 Simulation and Experimental Results

Comparison of Simulation Results

The high gain controller is implemented using the decomposition method introduced in section 5.4.1 (the complete design scripts can be found in appendix C.1.4). The high gain controller gain is chosen to be $4e - 1$ and the saturation controller gain is
chosen to be 30 in the final design (see appendix B.1 for the complete model for this controller).

The resulting controller displays very good control properties.

Simulation results show that the high gain controller has little interaction between the output channels (almost achieving decoupled control), under the same control magnitude input constraint as used in the cheap control design. As shown in figure 7.21, we see that a 0.3m/s speed set point change approximately causes only 0.05% of variation in tensions which is some 10 times better than the cheap control case (see figure 7.22).

![Figure 7.21: Simulated Output of the High Gain Controller](image1)

![Figure 7.22: Simulated Output of the Cheap Control Perfect Controller](image2)
Figures 7.23 and 7.24 give the corresponding control inputs for the high gain-saturation controller, and the cheap controller design. It is interesting to note that, with almost the same control effort (the input amplitude), the high gain-saturation controller produces much better control results than the cheap gain control design.

![Figure 7.23: The Input of the High Gain Controller](image1)

![Figure 7.24: The Input of the Cheap Control Perfect Controller](image2)

**Experimental Results**

Figures 7.25 to 7.27 give the experimental closed loop responses obtained on the web machine with the above high gain-saturation controller applied. It is observed that the high gain-saturation controller provides a response speed similar to the cheap control perfect controller.
It is also seen from figures 7.25 to 7.27, that the closed loop system under high gain-saturation control shows no interaction when a reference set point changes, which is an improvement compared to the cheap control experimental results of figures 7.9 to 7.11.

Figure 7.25: Output Responses to Tension Fa Step Reference Input
Figure 7.26: Output Responses to Tension Fb Step Reference Input
7.5 Tuning Regulator Design

From the previous controller implementation discussion, we recognize that a "good identified model" is very important in controller design. However, this identification process is a very tedious procedure, and a reliable model is always very difficult to obtain. Hence a different methodology which bypasses the plant model identification procedure of controller design is appealing. If the plant is assumed to be LTI and open loop stable, then it is shown in [Dav76a] that a tuning regulator design approach can be used which requires no mathematical model of the system.

In web control, the main control objective is to track constant tension and constant
velocity set points, and reject any constant unmeasurable disturbance in the system. From the results in chapter 6, the solvability verification condition and controller design for this problem can be easily done by carrying out a set of open loop step response experiments.

7.5.1 Preliminary Experiment

On using the approach of [Dav76a], it is assumed that the plant to be controlled can be described by a LTI model. In the case when this assumption can not be satisfied, then the approach of [Dav76a] can be extended to nonlinear plants by applying gain-scheduling techniques, e.g., see [SD86].

In the case of the web system, the analysis carried out in the modeling section of chapter 2 indicates that the web system is in fact a nonlinear system. However this analysis also indicates that the web system can be approximately described by a LTI model, and so this assumption will also be made here (as was done using the previous controller design approaches). The analysis of chapter 2 also indicates that the web system can be described by a model which is open loop stable, as has been demonstrated by experiment, and so the assumptions required by the tuning regulator approach hold.

A set of open loop step torque input experiments was then carried out on the University of Toronto web machine for the case when \( r_a = 0.0732m \) and \( r_c = 0.0853m \), and figures 7.28 to 7.30 give the response of the outputs \( T_a, v \) and \( T_b \).

The corresponding steady state values obtained under these step inputs are listed in table 7.2.

Table 7.2: Steady State Value Difference

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Steady State Value Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \delta F_a^{ss} )</td>
</tr>
<tr>
<td>Torque Step Input ( \delta U_1 = [0.6, 0, 0]^T )</td>
<td>7.7637</td>
</tr>
<tr>
<td>Torque Step Input ( \delta U_2 = [0, 0.6, 0]^T )</td>
<td>-1.5849</td>
</tr>
<tr>
<td>Torque Step Input ( \delta U_3 = [0, 0, -0.3]^T )</td>
<td>1.3173</td>
</tr>
</tbody>
</table>
Figure 7.28: Open Loop Response to Torque Step Input $\delta U = [0.6 \ 0 \ 0]^T$
Figure 7.29: Open Loop Response to Torque Step Input $\delta U = [0 \ 0.6 \ 0]^T$
According to the results of chapter 6, when the control objective is tracking/rejecting constant signals, the information obtained in table 7.2 is only required in order to solve the servomechanism problem for the web system using the tuning regulator controller approach.

Determination of the Steady State Gain Matrix \( G \)

**Definition 7.1** *Given a LTI system:*

\[
\dot{x} = Ax + Bu + E\omega \\
y = Cx + Du + F\omega
\]

(7.14)
the steady-steady gain matrix of the system (7.14) for constant reference input signals is given by:

\[
G = -CA^{-1}B + D
\]  

(7.15)

It then follows from chapter 6 that there exists a servomechanism controller for the web system for constant reference/disturbance signals if and only if \( \text{rank}(G) = \text{dim}(y) \) the number of inputs of the web system is at least equal to the number of outputs and the gain matrix \( G \) has a rank not smaller than the number of outputs.

For the web system used in this experiment, the gain matrix \( G \) can be determined from table 7.2:

\[
G = -CA^{-1}B + D = \\
\begin{bmatrix}
7.7637 & -1.5849 & 1.3173 \\
0.3667 & 0.3778 & -0.1453 \\
0.7962 & 0.7796 & 5.2389 \\
12.9394 & -2.6415 & -4.3912 \\
0.6111 & 0.6296 & 0.4842 \\
1.3271 & 1.2993 & -17.4631 \\
\end{bmatrix} \cdot [\delta U_1 \ \delta U_2 \ \delta U_3]^{-1}
\]

(7.16)

and there exists a solution to the servomechanism problem since:

\[
\text{rank}(G) = 3 = \text{number of outputs}
\]  

(7.17)

The tuning Regulator is given by:

\[
u = \epsilon_p G^{-1} \begin{bmatrix} T_{\text{ref}} - T_a \\ v_{\text{ref}} - v \\ T_{\text{bref}} - T_b \end{bmatrix} + \epsilon_r G^{-1} \begin{bmatrix} \int_{t_0}^{t} (T_{\text{ref}} - T_a) dt \\ \int_{t_0}^{t} (v_{\text{ref}} - v) dt \\ \int_{t_0}^{t} (T_{\text{bref}} - T_b) dt \end{bmatrix}
\]

(7.18)

where \( T_{\text{ref}}, v, \) and \( T_{\text{bref}} \) are reference signals for tension \( a, \) velocity and tension \( b \) respectively, and \( T_a, v, \) and \( T_b \) denote the actual measured tension \( a, \) velocity and
tension. $\epsilon_p > 0$ and $\epsilon_I > 0$ are scalars found by "on-line" tuning to achieve the best control effect.

7.5.2 Results Obtained From Experiments

After applying the controller 7.18 to the real plant, the resultant closed loop step responses obtained are given in 7.31 to 7.33.

![Graphs showing responses to step inputs](image)

Figure 7.31: Tuning Regulator Experiment: Responses to $F_a$ step input
Figure 7.32: Tuning Regulator Experiment: Responses to Fb step input
Behavior of the Tuning Regulator

From figure 7.31 to 7.33, it can be observed that the tension responses display satisfactory behavior with respect to both transient state and steady state, with the time constant less than 1 second. However, we also observe that the speed input response is somewhat oscillatory. This could be improved by adding a rate feedback term in the controller 7.18.
Chapter 8

Conclusions

8.1 Main Contributions of This Thesis

The web handling system has been treated as a multivariable servomechanism control problem in this thesis, and three different types of servomechanism controllers have been designed and implemented on the University of Toronto web machine.

1. A new modeling method has been presented. Using this method, a high frequency model and a reduced order low frequency model have been developed.

2. A 6th order LTI model of the web system was initially obtained by applying "black box" identification techniques, which appeared to have a better fit to the input/output behavior of the web system compared to the nominal 3rd order LTI benchmark model used in the web system.

3. Using both the $n = 6$ and $n = 3$ models of the web system, a "cheap control servomechanism controller" was designed and implemented on the University of Toronto web handling machine. The resulting closed loop system displayed excellent tension and speed response, and the output response was very fast with very little interaction occurring between output channels.
4. A new type of controller—a "high gain-saturation servomechanism controller", was introduced and implemented on the same web system. This controller design produced excellent tension and speed response, with zero interaction occurring between the output channels.

5. Finally, under the assumption that a model of the web system is not available, on carrying out a set of steady state experiments, a “tuning regulator servomechanism controller” was designed and implemented on the University of Toronto web machine. This controller also achieved excellent response.

8.2 Further Research

A desirable goal of servomechanism controller design is to design a controller that achieves the required control objectives with as little a priori knowledge of the underlying plant as possible. The tuning regulator is an example of a very good design methodology which achieves this goal. However, given that the web system parameters may change over time due to aging, mechanical wear, etc., it may be desirable to have a controller which "adapts" to such changes. Hence some type of adaptive or self-tuning modification of this controller is needed in future implementation of the tuning regulator design methodology to industrial web systems.
Bibliography


Appendix A

Properties of the Saturation Controller

In the high gain controller design, due to nature of the high loop gain, when the error is large (i.e. at the instant of reference set point change), the input to the plant may become too large for the physical plant to provide. In this Appendix, a modified type of saturation controller will be introduced. In this saturation controller design, given a plant and a high gain controller, by applying on-line tuning of the saturation controller gain, it can always be ensured that the maximum value of the plant input can be reduced to acceptable levels.

A.1 Structure of The Saturation Controller

The sole purpose of the saturation controller is to limit the plant input amplitudes. Hence we will first design our high gain controller (or other type of controller), and then apply the saturation controller to the resulting closed loop system consisting of the high gain controller and the plant. This type of controller structure is illustrated in figure A.1.
The saturation controller is chosen to be the following [DJ91]:

\[
\begin{align*}
\dot{x}_l &= y - y_{ref} \\
z &= \epsilon_l (- (\bar{D} - \bar{C} \bar{A}^{-1} \bar{B})^{-1} x_l + (\bar{D} - \bar{C} \bar{A}^{-1} \bar{B})^{-1} \bar{C} \bar{A}^{-1} \dot{x})
\end{align*}
\] (A.1)

where \( \dot{x} \) is the can be the exact state feedback or the estimated state of the closed loop system consisting of the high gain controller and the plant. We denote this composite system to be \((\bar{C}, \bar{A}, \bar{B}, \bar{D})\). Here \( z \) is the saturation controller output, and \( \epsilon_l > 0 \) is the saturation controller gain. It can be noted that in this saturation controller design, a servocompensator is included; hence closed loop asymptotic tracking is guaranteed.

### A.2 The Response of the Saturation Controller

#### A.2.1 Case 1−Direct State Feedback

On carrying out a simple coordinate transformation:

\[
\begin{align*}
\bar{x} &= x \\
\bar{x}_l &= x_l - \bar{C} \bar{A}^{-1} x
\end{align*}
\] (A.2)

we can solve for \( x(t) \) in figure A.1, assuming that the reference input \( y_{ref} \) and the disturbance \( \omega \) are constant.
This response has the Laplace transform of:

\[
z(s) = \frac{e^{-\epsilon t}}{s + \epsilon} \left( \tilde{C} \tilde{A}^{-1} \tilde{B} - \tilde{D} \right)^{-1} \left[ \tilde{F} - \tilde{C} \tilde{A}^{-1} \tilde{E}, -I \right] \left[ \begin{array}{c} \omega \\ \omega_{\text{ref}} \end{array} \right]
\]

which means that figure A.1 can be transformed to a simpler structure in figure A.2

\[
T_f(s) = \text{diag}\left(\frac{\epsilon_1}{s + \epsilon}, \cdots, \frac{\epsilon_r}{s + \epsilon}\right)
\]

where \( r \) is the number of outputs of the plant.

Denote the transfer function matrices of the high gain controller and the plant to be \( G_h(s) \) and \( G_P(s) \), then the inputs \( u(s) \) to the plant can be expressed as:

\[
u(s) = \frac{G_h(s)(I + G_P(s)G_h(s))^{-1}T_f(s)z_{ss}}{G}
\]

which can be expressed as:

\[
u(s) = \left( G_0 + sG_s(s) \right) T_f(s) \frac{z_{ss}}{s}
\]
where $G_0$ is a constant matrix and $G_s(s)$ is a proper rational matrix of $s$.

The decomposition of (A.6) exists and is unique since we can write $G$ as:

$$
G = \begin{bmatrix}
g_{11}(s) & \cdots & g_{1r}(s) \\
\vdots & & \vdots \\
g_{m1}(s) & \cdots & g_{mr}(s)
\end{bmatrix}
$$

where, due to the fact that both the high gain controller and the plant transfer function matrices are proper, every element $g_{ij}(s)$, $i = 1, \cdots, m; j = 1, \cdots, r$ of $G$ is proper. Hence we can decompose $g_{ij}(s)$ into a constant part and a part which is a multiple of $s$:

$$
g_{ij}(s) = \frac{n_{ij}(s)}{d_{ij}(s)}
= \frac{n_{ij}^0 + s n_{ij}^1(s)}{d_{ij}^0 + s d_{ij}^1(s)}
= g_{ij}^0 + s g_{ij}^1(s)
$$

where $n_{ij}^0$, $d_{ij}^0$, and $g_{ij}^0$ are constants, $n_{ij}^1(s)$ and $d_{ij}^1(s)$ are polynomial function of $s$, and $g_{ij}^1(s)$ is a proper rational function of $s$.

Then the result (A.6) arises directly from (A.9).

Thus we can decompose $u(s)$ as $u(s) = u_0(s) + u_s(s)$:

where:

$$
u_0(s) = G_0 T_f(s) \frac{z_{ss}}{s}
= \frac{1}{s+q_s} G_0 z_{ss}
$$

(A.10)

and

$$
u_s(s) = G_s(s) T_f(s) \frac{z_{ss}}{s}
= \frac{1}{s+q_s} G_s(s) z_{ss}
$$

(A.11)
APPENDIX A. PROPERTIES OF THE SATURATION CONTROLLER

The Relationship Between \( u_0(t) \) And The Saturation Control Gain \( \varepsilon_t \)

The corresponding time domain response of \( u_0(s) \) is:

\[
u_0(t) = (1 - e^{-\varepsilon_t})G_0z_{ss}
\]  \( \text{(A.12)} \)

We can easily conclude from A.12 that the response of \( u_0(t) \) has the following properties:

1. The steady state response of \( u_0(t) \) is \( G_0z_{ss} \).

2. There is no overshoot in the response of \( u_0(t) \).

3. The rate of the change of \( u_0(t) \):

\[
\dot{u}_0(t) = \varepsilon_t e^{-\varepsilon_t}G_0z_{ss},
\]  \( \text{(A.13)} \)

at any instant \( t \) can be made arbitrarily small by simply decreasing \( \varepsilon_t \), i.e., the input can be made as slow as possible.

We can see from the above properties, that the input \( u_0(t) \) provides the necessary input for asymptotic tracking. Hence we can call it the tracking input.

The Relationship Between \( u_s(t) \) And The Saturation Control Gain \( \varepsilon_t \)

On expanding \( u_s(s) \), we obtain:

\[
u_s(s) = \frac{s}{s + \varepsilon_t}G_s(s)z_{ss} = \frac{s}{s + \varepsilon_t} \begin{bmatrix} g_{t1}^s(s) & \cdots & g_{t1}^s(s) \\ \vdots & \vdots & \vdots \\ g_{m1}^s(s) & \cdots & g_{mr}^s(s) \end{bmatrix} \begin{bmatrix} z_{ss}^1 \\ \vdots \\ z_{ss}^r \end{bmatrix}
\]  \( \text{(A.14)} \)

\[
= \begin{bmatrix} \frac{s}{s + \varepsilon_t}g_{t1}^s(s)z_{ss}^1 + \cdots + \frac{s}{s + \varepsilon_t}g_{t1}^s(s)z_{ss}^r \\ \vdots \\ \frac{s}{s + \varepsilon_t}g_{m1}^s(s)z_{ss}^1 + \cdots + \frac{s}{s + \varepsilon_t}g_{m1}^s(s)z_{ss}^r \\ \vdots \\ \frac{s}{s + \varepsilon_t}g_{mr}^s(s)z_{ss}^1 + \cdots + \frac{s}{s + \varepsilon_t}g_{mr}^s(s)z_{ss}^r \end{bmatrix}
\]

\[
= \begin{bmatrix} u_1^s(s) & \cdots & u_n^s(s) \end{bmatrix}^T
\]
We can now solve for $u_i(t)$ by carrying out an inverse Laplace transform of $u_i^j(s)$, $i = 1, \cdots, m$. We will do this by carrying out the inverse Laplace transform of each single component of $u_i^j(s)$:

$$
\mathcal{L}^{-1}(u_i^j(s)) = \mathcal{L}^{-1}(\frac{g_{ij}^j(s)}{s + \epsilon_l})z_i^j(t)
$$

(A.15)

where $i = 1, \cdots, m$ and $j = 1, \cdots, r$.

From equation (A.9), we know that set of the poles of $G_s(s)$ lies in the set of the poles of $G(s)$; hence we can express $g_{ij}(s)$ as:

$$
g_{ij}^j(s) = \sum_{k=1}^{f_i}(\sum_{l=1}^{f_a} \frac{C_{kl}}{(s + \lambda_k)^l}) + \sum_{a=1}^{f_a}(\sum_{b=1}^{f_a} \frac{C_{ab}}{(s + \sigma_a + j\omega_k)^b})
$$

(A.16)

where $C_{kl}$, $k = 1, \cdots, \delta_1$, $l = 1, \cdots, \delta_k$ and $C_{ab}$, $a = 1, \cdots, \delta_2$, $k = 1, \cdots, \delta_a$ are constant coefficients. Here $-\lambda_k$ are the real stable poles of $G(s)$, and $-\sigma_a \pm j\omega_k$ are the complex conjugate pairs of stable poles of the closed loop composite system consisting of the plant and the high gain controller. $\delta_k$ is the multiplicity of the $k$th real pole and $\delta_a$ is the multiplicity of the complex conjugate poles.

$\mathcal{L}^{-1}(u_i^j(s))$ can be obtained by solving for each

$$
\mathcal{L}^{-1}\left(\frac{1}{(s + \lambda_k)^l s + \epsilon_l}\right)
$$

(A.17)

and

$$
\mathcal{L}^{-1}\left(\frac{1}{(s + \sigma_a + j\omega_k)^b s + \epsilon_l}\right)
$$

(A.18)

For simplicity, we will just discuss the case of (A.17). Similar results can be obtained for the case of (A.18).
Now from complex analysis, we know that:

\[
\frac{1}{(s+\lambda_k)^l} \frac{\epsilon_l}{s+\epsilon_l} = \epsilon_l \left( \frac{C_0}{s+\epsilon_l} + \sum_{i=1}^{l} \frac{C_i}{(s+\lambda_k)^i} \right)
\]

where:

\[
\begin{align*}
C_0 &= \frac{(-1)^l}{(-\lambda_k+\epsilon_l)^l} \\
C_i &= \frac{(-1)^{l-i}}{(-\lambda_k+\epsilon_l)^{l+1-i}}, \ i = 1, \ldots, l
\end{align*}
\]

And so from the result of (A.20), we obtain:

\[
u_R^l(t) = \mathcal{L}^{-1} \left( \frac{1}{(s+\lambda_k)^l} \frac{\epsilon_l}{s+\epsilon_l} \right) = \epsilon_l \left( \frac{C_0 e^{-\epsilon_l t}}{A(t)} + \sum_{i=1}^{l} \frac{C_i \epsilon_l^{i-1} e^{-\lambda_k t}}{(i-1)! B(t)} \right)
\]

On examining the structure of \(u_R^l(t)\) given in A.21 and A.20, the following properties of \(u_a(t)\) now directly follows:

1. The steady state response of \(u_a(t)\) is 0.

2. The maximum amplitude of every element \(u_A^i(t)\), \(i = 1, \ldots, r\) of \(u_a(t)\) can be made arbitrarily small by simply reducing \(\epsilon_l\).

Conclusions

By decomposing the plant input into two parts: the input \(u_0(t)\) and the input \(u_a(t)\), it is easy to understand how the saturation controller works, and how the gain \(\epsilon_l\) influences the maximum amplitude of the plant input. Reducing the gain \(\epsilon_l\) causes the input \(u_a(t)\) to approach 0, and hence the plant input approaches the tracking input \(u_0(t)\), which has no input overshoot at all. However, reducing \(\epsilon_l\) will also reduce the speed of the input \(u_0(t)\), and hence will also slow down the plant input \(u(t)\) and the system output response.
A.2.2 Using an observer

According to [DJ91], when an estimate of the state $\hat{x}$ in (A.1) is obtained from an observer:

$$\dot{\hat{x}} = (\bar{A} - K_o \bar{C})\hat{x} + K_o(y - \bar{D}u) + \bar{B}u$$  \hspace{1cm} (A.22)

the corresponding response of $z(s)$ is given by:

$$z(s) = \frac{\epsilon_l}{s + \epsilon_l} (\bar{C} \bar{A}^{-1} \bar{B} - \bar{D})^{-1} [\bar{F} - \bar{C} \bar{A}^{-1} \bar{E}, -I] \begin{bmatrix} \omega(s) \\ y_{ref}(s) \end{bmatrix}$$

$$- \frac{\epsilon_l s}{s + \epsilon_l} \frac{(\bar{C} \bar{A}^{-1} \bar{B} - \bar{D})^{-1} \bar{C} \bar{A}^{-1} (s - \bar{A} + K_o \bar{C})^{-1} \bar{E} \omega(s)}{P_1(s)}$$

$$+ \frac{P_1(s)}{P_2(s)}$$  \hspace{1cm} (A.23)

We observe from (A.23) that the response of $z(s)$ consists of two parts $P_1(s)$ and $P_2(s)$. $P_1(s)$ is exactly the same as for the direct state feedback case, and $P_2(s)$ is an extra term which results when the observed state is used.

In this case, when the reference and the disturbance signals are constant, the same conclusions made in section A.2.1 still hold.
Appendix B

Models of the University of Toronto Web Machine and Controllers

B.1 Models of the University of Toronto Machine

Models are expressed in the following state space representation:

$$\begin{align*}
\dot{x} &= A_r x + B_r U \\
y &= C_r x
\end{align*}$$

(B.1)

where $u = [m_a \ m_b \ m_c]$ and $y = [F_a \ v \ F_c]$, with:

- $m_a$ the torque output to the rewinder drive (signal M28 in MTSC32 manual)
- $m_b$ the torque output to the nip drive (signal M29 in MTSC32 manual)
- $m_c$ the torque output to the unwinder drive (signal M30 in MTSC32 manual)
- $F_a$ the tension of the rewinding zone
- $v$ the velocity of the nip
- $F_b$ the tension of the unwinding zone
All models include filters of the form:

\[ F(s) = \frac{1}{s^{0.1} + 1} \]  \hspace{1cm} (B.2)

connecting to each tension output.

### B.1.1 Model #1: A Reduced Order Low Frequency Model

\[ \mathbf{A}_r = \begin{bmatrix} -10 & 0 & -2.4742 \\ 0 & -10 & -4.2565 \\ 0 & 0 & -0.16413 \end{bmatrix} \]

\[ \mathbf{B}_r = \begin{bmatrix} 69.54 & -26.846 & -7.4274 \\ 19.955 & 19.955 & -21.146 \\ 0.6624 & 0.6624 & 0.18326 \end{bmatrix} \]

\[ \mathbf{C}_r = \begin{bmatrix} 0.019231 & 0 & 0 \\ 0 & 0 & 0.2 \\ 0 & 0.019231 & 0 \end{bmatrix} \]

### B.1.2 Model #2: A Black Box 6th Order Model

\[ \mathbf{A}_r = \begin{bmatrix} -11.3151 & 0.0937 & -0.4417 & -0.0187 & 0.0056 & -0.0000 \\ 1.5026 & -3.1116 & -0.5832 & -0.0245 & 0.0056 & -0.0003 \\ 0.3527 & 0.0347 & -6.2573 & -0.0056 & 0.0030 & -0.0017 \\ 0.8439 & 0.0715 & -0.3368 & -2.1966 & 0.0010 & -0.0007 \\ -1.2398 & -0.1067 & 0.5027 & 0.0202 & -0.8294 & 0.0016 \\ 0.0585 & -0.0007 & 0.0029 & -0.0011 & -0.0015 & -3.1000 \end{bmatrix} \]
\[ Br = \begin{pmatrix} 7.6328 & -0.2568 & -89.5580 \\ 6.4710 & -40.1816 & -76.5029 \\ -86.8171 & -0.3430 & 4.9728 \\ -10.4544 & -77.1662 & 16.2484 \\ 59.9650 & 50.8595 & 21.9618 \\ 86.9371 & -11.4346 & -24.5354 \end{pmatrix} \]

\[ Cr = \begin{pmatrix} -0.0186 & 0.1447 & -0.6668 & 0.0030 & -0.0000 & -0.0409 \\ -0.0013 & -0.0005 & 0.0011 & 0.0002 & 0.0104 & 0.0000 \\ 1.7052 & -0.0364 & 0.1543 & -0.0312 & -0.0008 & 0.0491 \end{pmatrix} \]

### B.2 Controllers

Controllers are given by the following model:

\[
\begin{align*}
\dot{x}_c &= A_c x_c + B_c U_c \\
U &= C_c x_c 
\end{align*}
\]  

(B.3)

where

- \( U_{c}[1] \) the reference signal \( F_{a\text{ref}} \) of tension \( F_a \) (tension of the rewinding zone)
- \( U_{c}[2] \) the reference signal \( v_{\text{ref}} \) of nip velocity \( v \)
- \( U_{c}[3] \) the reference signal \( F_{b\text{ref}} \) of tension \( F_b \) (tension of the unwinding zone)
- \( U_{c}[4] \) \( F_{a\text{ref}} - F_a \)
- \( U_{c}[5] \) \( v_{\text{ref}} - v \)
- \( U_{c}[6] \) \( F_{b\text{ref}} - F_b \)

and \( U \) is the torque input vector to the drives as described in section B.1.
B.2.1 Cheap Control Servomechanism Controller

Controller #1: Controller Design Based On The Reduced Order Low Frequency Model

For $e = 1e - 4$ and $eo = 1e - 4$, the controller is given by:

$$\begin{align*}
Ac &= \\
1.0e+003 &
\begin{bmatrix}
-0.1528 & -0.0183 & -0.0969 & -7.3808 & -1.0343 & -0.7553 \\
-0.0185 & -0.0701 & -0.1675 & -0.8426 & -1.4912 & -3.0825 \\
-0.0005 & -0.0009 & -0.0216 & -0.0214 & -0.0899 & -0.0238 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{align*}$$

$$\begin{align*}
Bc &= \\
&
\begin{bmatrix}
7.0440 & 98.0424 & 22.2063 & -7.0440 & -98.0424 & -22.2063 \\
0.2095 & 15.5180 & 0.3626 & -0.2095 & -15.5180 & -0.3626 \\
0 & 0 & 0 & -0.0077 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.2000 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.0077
\end{bmatrix}
\end{align*}$$

$$\begin{align*}
Cc &= \\
&
\begin{bmatrix}
-0.1081 & -0.0184 & -2.8733 & -85.5875 & -48.5367 & -17.8599 \\
0.0727 & -0.0602 & -4.8672 & 51.1881 & -74.5665 & -42.6570 \\
0.0086 & 0.1437 & -4.2224 & 7.3868 & -45.6512 & 88.6646
\end{bmatrix}
\end{align*}$$
Controller #2: Controller Design Based On The 6th Order Black Box Model

\[ Ac = \]

\[ 1.0 \times 10^3 \times \]

<table>
<thead>
<tr>
<th>Columns 1 through 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.1247  0.0010  -0.0039  0.0022  0.0021  -0.0030  -1.8727</td>
</tr>
<tr>
<td>-0.0979  -0.0044  0.0038  0.0020  0.0106  -0.0021  -3.3632</td>
</tr>
<tr>
<td>0.0034   0.0009  -0.0468  0.0002  0.0064  -0.0025  7.7515</td>
</tr>
<tr>
<td>0.0314   -0.0032  0.0109  -0.0026  0.0178  0.0015  -2.2714</td>
</tr>
<tr>
<td>0.0259   -0.0046  0.0228  -0.0008  -0.0191  0.0020  -2.7788</td>
</tr>
<tr>
<td>-0.0282  -0.0097  0.0434  0.0003  -0.0037  -0.0010  -8.5418</td>
</tr>
<tr>
<td>0        0        0        0        0        0        0</td>
</tr>
<tr>
<td>0        0        0        0        0        0        0</td>
</tr>
<tr>
<td>0        0        0        0        0        0        0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Columns 8 through 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.9403  -8.5743</td>
</tr>
<tr>
<td>5.0565  -6.1816</td>
</tr>
<tr>
<td>3.9115  0.4843</td>
</tr>
<tr>
<td>6.6299  3.7635</td>
</tr>
<tr>
<td>-7.6487 0.6514</td>
</tr>
<tr>
<td>-2.4199 -2.0221</td>
</tr>
<tr>
<td>0        0</td>
</tr>
<tr>
<td>0        0</td>
</tr>
<tr>
<td>0        0</td>
</tr>
</tbody>
</table>
\[ B_c = \]
\[ 1.0 \times 10^3 \times \]
\[
\begin{array}{cccccccc}
0.0083 & -0.1091 & 0.0622 & -0.0083 & 0.1091 & -0.0622 \\
0.0151 & -0.7003 & 0.0548 & -0.0151 & 0.7003 & -0.0548 \\
-0.0557 & -0.4175 & -0.0027 & 0.0557 & 0.4175 & 0.0027 \\
0.0068 & -1.2509 & -0.0169 & -0.0068 & 1.2509 & 0.0169 \\
0.0310 & 1.2829 & -0.0141 & -0.0310 & -1.2829 & 0.0141 \\
0.0622 & 0.2459 & 0.0166 & -0.0622 & -0.2459 & -0.0166 \\
0 & 0 & 0 & -0.0000 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.0002 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.0000
\end{array}
\]

\[ C_c = \]

**Columns 1 through 7**

\[
\begin{array}{cccccccc}
0.0043 & -0.0143 & 0.0494 & -0.0009 & -0.0244 & 0.0046 & -88.7028 \\
-0.0236 & 0.0148 & -0.0416 & 0.0010 & -0.0620 & -0.0054 & 44.2376 \\
0.0834 & 0.0000 & -0.0004 & -0.0033 & -0.0123 & 0.0037 & 13.2234
\end{array}
\]

**Columns 8 through 9**

\[-46.1716 & 0.0230 \\
-85.0014 & -28.5972 \\
-25.3563 & 95.8238\]
B.2.2 High Gain Servomechanism Controller

This controller is a cascade connection of a high gain controller and saturation controller. When the high gain controller gain is $4e - 1$ and saturation controller gain is 30, the overall controller is given by:

$$Ac =$$

Columns 1 through 6

\[
\begin{array}{ccccccc}
-6.2500e+000 & 0 & 0 & -1.5706e+002 & 9.8369e-015 & -3.9783e+000 \\
0 & -6.2500e+000 & 0 & 1.5706e+002 & -1.5706e+002 & -7.3479e+000 \\
0 & 0 & -6.2500e+000 & 5.8198e-014 & 5.6768e+002 & -1.0689e+000 \\
0 & 0 & 0 & -2.7041e+001 & -4.1601e-004 & -8.0490e-000 \\
0 & 0 & 0 & -4.1601e-004 & -2.7042e+001 & -1.3847e-000 \\
0 & 0 & 0 & 7.4150e-002 & 1.2756e-001 & -2.5319e+000 \\
0 & 0 & 0 & -4.3291e+002 & 2.7588e+000 & -1.1117e+000 \\
0 & 0 & 0 & 4.3747e+002 & -4.2942e+002 & -2.0525e+000 \\
0 & 0 & 0 & 4.2924e+000 & 1.5779e+003 & -2.9811e+000 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}
\]

Columns 7 through 12

\[
\begin{array}{cccccccc}
-3.0000e+001 & -9.4555e-016 & -4.3628e-016 & -1.9632e+003 & -1.0842e+003 & 1.6196e-01 \\
-3.5527e-015 & -3.0000e+001 & 8.8818e-016 & 1.9632e+003 & -1.9751e+003 & -1.9632e+000 \\
3.5527e-015 & 0 & -3.0000e+001 & 5.6134e-013 & -2.6966e+003 & 7.0960e+000 \\
8.6132e-001 & -3.3251e-001 & -9.1995e-002 & 0 & 0 \\
2.4717e-001 & 2.4717e-001 & -2.6191e-001 & 0 & 0 \\
1.3127e-001 & 1.3127e-001 & 3.6318e-002 & 0 & 0 \\
\end{array}
\]
### APPENDIX B. MODELS OF THE UNIVERSITY OF TORONTO WEB MACHINE AND $C$

\[
\begin{align*}
B_c &= \\
-3.5527e-015 & -3.6250e+001 & 8.8818e-016 & 1.9632e+003 & -1.9751e+003 & -1.9632e+003 \\
3.5527e-015 & 0 & -3.6250e+001 & 5.6134e+013 & -2.6966e+003 & 7.0960e+000 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{align*}
\]

\[
C_c =
\begin{align*}
\text{Columns 1 through 6} \\
9.9088e-001 & 0 & 0 & 0 & 0 & 0 \\
0 & 9.9088e-001 & 0 & 0 & 0 & 0 \\
0 & 0 & 9.9088e-001 & 0 & 0 & 0 \\
\text{Columns 7 through 12} \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{align*}
\]
APPENDIX B. MODELS OF THE UNIVERSITY OF TORONTO WEB MACHINE AND C

\[
\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
Appendix C

Scripts For Controller Designs

C.1 Matlab Codes For Controller Design

C.1.1 The Script To Obtain the University of Toronto Web Machine State Space Model

This function is used to obtain the reduced order low frequency model of the University of Toronto web machine model. It uses the modeling method described in chapter 2.

Function: \[ \text{[linSys]} = \text{get\_lin\_system}(\text{info}); \]

% This function is used to get a linear system.
% Input: info (Plant information vector)
% info(1) to info(3): inertia of the three rollers (from rewinder to unwinder),
% info(4) to info(6): radius of the three rollers (from rewinder to unwinder),
% info(7) to info(9): friction coefficients of the three rollers (from rewinder to unwinder).
% Output: lin_sys (the linear reduced order state space model obtained)

113
function [lin_sys] = get_lin_system(info)

Ja = info(1); J0b = info(2); Jc = info(3);
ra = info(4); rb = info(5); rc = info(6);
bA = info(7); bB = info(8); bC = info(9);

Ca = Ja/(ra^2); Cb = J0b/(rb^2); Cc = Jc/(rc^2); Ya = bA/(ra^2);
Yb = bB/(rb^2); Yc = bC/(rc^2);

C = Ca + Cb + Cc; Y = Ya + Yb + Yc;

A = -Y/C; B_1 = (1/ra)/C; B_2 = (1/rb)/C; B_3 = (1/rc)/C;

C_1 = Cb + Cc; C_2 = Cc; Y_1 = Yb + Yc; Y_2 = Yc;
C_1 = Y_1 - (Y*C_1)/C; C_2 = Y_2 - (Y*C_2)/C;

D_11 = (C_1/C)*(1/ra); D_12 = ((C_1/C) - 1)*(1/rb); D_13 = ((C_1/C) - 1)*(1/rc);
D_21 = (C_2/C)*(1/ra); D_22 = (C_2/C)*(1/rb); D_23 = (C_2/C - 1)*(1/rc);

%%%%%% The system %%%
my_A = A; my_B = [B_1 B_2 B_3]; my_C = [C_1; C_2];
my_D = [D_11 D_12 D_13; 0 0 0; D_21 D_22 D_23];

lin_sys = ss(my_A, my_B, my_C, my_D);

C.1.2 The Script To Normalize The Linear System And Add Filters

Function: my_sys = system_relative(Base_max)

% Input: Base_max is the vector of the maximum value of the outputs
% Output: my_sys is the normalized system with filters connected to
% the tension measurements.

function my_sys = system_relative(Base_max)

inv_B = ones(1,length(Base_max))./Base_max;

inv_mat_B = diag(inv_B);

% web properties
rho = 1100 ; W = 0.168 ; e =0.00015/(2*pi) ; prW2 = pi*rho*W/2 ;

% bare shaft inertias
J0a = 0.01745 ; J0b = 0.03223 ; J0c = 0.023363;

r0 = 0.0415;

Rmin = 0.0415 ; Rmax = 0.15 ;
ra = Rmin ;
rc = Rmax;
rb = r0 ;

Jb = J0b ;

% Shaft inertias (range) with web on
Ja = J0a + prW2*(ra.^4 - r0^4) ;
Jc = J0c + prW2*(rc.^4 - r0^4) ;

% Damping factors for the shafts, denoted ma,mb,mc
ma = 1/254.4878 ; mb = 1/165.0867 ; mc=1/258.3219 ;
Ia = Ja; Ib = Jb; Ic=Jc;

info = [Ja;Jb;Jc;ra;rb;rc;ma;mb;mc];

[my_sys_int] = get_lin_system(info);

% This part will calculate the system after filtering

filter_tf=tf([1 0;0 1 0;0 0 1],[[0.1 1] 1 1 1;1 1 1 1 [0.1 1]]);

filter_ss = ss(filter_tf);

my_sys_norel = series(my_sys_int,filter_ss);

my_sys_norel.c = inv_mat_B*my_sys_norel.c; my_sys_norel.d = inv_mat_B*my_sys_norel.d;

my_sys=ss(my_sys_norel.a,my_sys_norel.b,my_sys_norel.c,my_sys_norel.d);

C.1.3 Scripts for Cheap Control Servomechanism Controller Design

Cheap Control Design

(Modified from E.J. Davison's routine "servob.m")

Function: [k,ko]=servobd(a,b,c,d,e,eo)

function [k,ko]=servobd(a,b,c,d,e,eo)

% This function is used to find the cheap controller gain and the obs
% SERVOBD
% [k,ko]=servob(a,b,c,d,e,eo)
% u=k2*neta +k1*(observed state), where k=[k1 k2]

[n,m]=size(b); [r,n]=size(c);

aa=[a zeros(n,r); c zeros(r,r)]; bb=[b;d]; cc=[zeros(r,n) eye(r)];

qq=cc'*cc; rr=e*eye(m);
kk=-lqr(aa,bb,qq,rr);

K1=kk(:,1:n); K2=kk(:,n+1:n+r); aac=aa+bb*kk;

% Observer Design
ko = observe(eo,a,b,c); % observer gain

k=kk;

Obtaining the State Space Representation of the Cheap Control Servomechanism Controller

Function: [cont_ss] = servo_de(system,e,eo)

% This function obtains the state space model of the controller from
% the given plant state space model using the function "servobd".
% The input to the obtained controller is [yref error]', the output of
% the controller is the input u to the plant, i.e., the controller has
% the following state space representation:
% dot(Xc) = cont_ss.a*Xc + cont_ss.b*[yref error]' 
% u = cont_ss.c*Xc + cont_ss.d*[yref error]'

function [cont_ss] = servo_de(system,e,eo);
APPENDIX C. SCRIPTS FOR CONTROLLER DESIGNS

AO = system.a; BO = system.b; CO = system.c; DO = system.d;

[nb, mb] = size(BO);
[rc, nc] = size(CO);

n = nb;            % number of states of the given system
r = rc;            % number of outputs of the given system
m = mb;            % number of inputs of the given system

% Now use the servob routine to compute the controller

[K, Ko] = servobd(AO, BO, CO, DO, e, eo);

% Now construct the controller state space representation
% The controller is of the following structure
% \( u = k_1 x + k_2 \dot{y} \)
% Where \( x \) is the observed state from a full order linear state observer:
% \[
% \dot{\hat{x}}(x) = A \cdot (\hat{x}(x)) + B \cdot u + K_0 \cdot (y - \hat{y}(x))
% \]
% \[
% \hat{y} = C \cdot \hat{x} + D \cdot u
% \]
% and \( \dot{\dot{y}} = y = yref - error \)

K1 = K(:,1:n); K2 = K(:,n+1:n+r);

% For the controller state space representation, the input to the
% controller is the output from the plant, and the output of the
% controller is the input to the plant

Acont = [AO - Ko*CO + (BO-Ko*DO)*K1 (BO-Ko*DO)*K2; zeros(r,n) zeros(r,r)];
Bcont = [Ko - Ko; zeros(r,r) - eye(r,r)];
Ccont = [K1 K2];
Dcont = zeros(m,r*2);
The Overall Design Script For The Cheap Control Servomechanism Controller Design

% First normalize the obtained linear system and serially connect each tension
% outputs to a sensor as the real system.
Fa_max = 130; Fb_max = 130; Vb_max = 5;
Base_max = [Fa_max Vb_max Fb_max];

plant = system_relative(Base_max);

% here get the controller
[cont_ss] = servo_de(plant);

% Now the actual designed controller will take the
% relative reference and error as the inputs, so it
% is necessary to modify the B, D matrices, here assume
% the inputs to the controller are references and errors.

Base_c = [Base_max Base_max];

inv_Base_c = ones(1,length(Base_c))./Base_c;
m_inv_Bc = diag(inv_Base_c);

cont_ss.B = cont.ss.B * m_inv_Bc;
cont.ss.D = cont.ss.D * m_inv_Bc;

% Now export this controller to C
C.1.4 Scripts for High Gain Servomechanism Controller Design

Specify the high gain controller gain "epsi_h", the weighting scaler for the observer design "eo" and the saturation controller gain "given_theta". The high gain controller gain "epsi_h" should start from a very small value as to ensure stability. "eo" should be chosen from a large value as 1e4, this result in a slow response speed, then it should be tuned to smaller values, satisfactory response speed is achieved. The saturation is chosen from a very large value, the resultant closed loop system may be slow, the value is the to achieve desired response.

epsi_h = 4e-1; eo = 1e-3; given_theta = 30;

Fa_max = 200; Fb_max = 200; Vb_max = 5;
Base_max = [Fa_max Vb_max Fb_max];

% first get the plant system, this system export the relative output, \( y_{\%} = \frac{y}{y_{\text{max}}} \) and the state are relative states \( x_{\%} = \frac{x}{x_{\text{max}}} \);

% This function is similar to "system_relative.m", the difference is that it also normalize the plant state also to be in \([0 1]\).
plant = system_state_relative(Base_max);

[r,n] = size(c); [n,m] = size(b);

dr = 0;
APPENDIX C. SCRIPTS FOR CONTROLLER DESIGNS

% High gain controller design

[Xi,M,bta,L,G,H,K,p]=shgp(a,b,c,epsi_h,dr);

% The state space representation of the high gain controller
% The high gain controller is expressed in the following form:
% \[ \dot{X}_h = A_{c,H}X_h + B_{c,H}(y-v) \]
% \[ u = C_{c,H}X_h + D_{c,H}(y-v) \]

A_{c,H} = inv(Xi)*M;
B_{c,H} = inv(Xi)*(1/epsi_h^bta)*L*G;
C_{c,H} = (epsi_h^bta)*H;
D_{c,H} = K*G;

hg_ss = ss(A_{c,H}, B_{c,H}, C_{c,H}, D_{c,H});

% The composite system of the plant and the high gain controller.
% Because hg_ss.d is 0, we obtain a simpler expression as following:

comp_a = [plant.a plant.b*hg_ss.c;hg_ss.b*plant.c hg_ss.a];
comp_b = [zeros(length(plant.a),r);-hg_ss.b];
comp_c = [plant.c zeros(length(hg_ss.a))];
comp_d = zeros(r);

comp_ss = ss(comp_a,comp_b,comp_c,comp_d);

% Now design the input limiting controller
% first need to scale the outputs to make |yref'|yref|<=1
% for this case, the outputs are already relative outputs,
% which means the system has already been scaled

tou = inv(comp_ss.d-comp_ss.c*inv(comp_ss.a)*comp_ss.b);
theta = given_theta;

% observer design

ko = observe(eo,compss.a,comp_ss.b,comp_ss.c); % observer gain

% now get the saturation controller for the composite system of the form:
%      dot(eta) = y-yref
%          u   = theta*tou*comp_ss.c*inv(comp_ss.a)*hat(x)-theta*tau*eta
% where hat(x) is the observed state from the following full order observe
%      dot(hat(x)) = comp_ss.a*hat(x)+comp_ss.b*u+ko(y-comp_ss.c*hat(x)-comp_ss.d
% This controller can then be expressed by:
%      dot(X1) = all_Cont_ss.a*X1 + all_Cont_ss.b*[yref error]',
%      v      = all_Cont_ss.c*X1 + all_Cont_ss.d*[yref error]',

all_Ccont = [theta*tou*comp_ss.c*inv(comp_ss.a) (-theta)*tou];
all_Dcont = zeros(m,2*r);

n_xhat = length(comp_ss.a);

all_Acont = [(comp_ss.a-ko*comp_ss.c) zeros(n_xhat,r);
             zeros(r,n_xhat) zeros(r)] +[(comp_ss.b-ko*comp_ss.d)*all_Ccont;zeros(r,n_xhat+r)];
all_Bcont = [ko -ko;zeros(r) -eye(r)];

all_Cont_ss = ss(all_Acont,all_Bcont,all_Ccont,all_Dcont);
Now it is needed to transform the controller to the form that can be applied to the input of the real plant, rather than the input of the saturation controller. This is done by augmenting the controller:

\[ \dot{X_l} = a_{l,Cont,ss}.a*X_l + a_{l,Cont,ss}.b*[yref\ error]' \]
\[ v = a_{l,Cont,ss}.c*X_l + a_{l,Cont,ss}.d*[yref\ error]' \]

and the high gain controller:

\[ \dot{X_h} = A_{c,H}X_h + B_{c,H}(y-v) \]
\[ u = C_{c,H}X_h \]

to get a new system:

\[ \dot{X_c} = overall_{Cont,ss}.a*X_c + overall_{Cont,ss}.b*[yref\ error]' \]
\[ u = overall_{Cont,ss}.c*X_c + overall_{Cont,ss}.d*[yref\ error]' \]

\[
\begin{align*}
n_{hg} &= \text{length}(hg_{ss}.a); \\
n_{all\ Cont} &= \text{length}(all\ Cont_{ss}.a); \\

overall_A &= [hg_{ss}.a \ (-hg_{ss}.b)*all\ Cont_{ss}.c; \\
&\quad \text{zeros}(n_{all\ Cont},n_{hg})\ all\ Cont_{ss}.a]; \\
overall_B &= [-hg_{ss}.b*all\ Cont_{ss}.d+[hg_{ss}.b\ (-hg_{ss}.b)];all\ Cont_{ss}.b]; \\
overall_C &= [hg_{ss}.c\ \text{zeros}(r,n_{all\ Cont})]; \\
overall_D &= \text{zeros}(m,2*r); \\

\end{align*}
\]

Because the controller is designed to control the relative outputs, scaling on e, yref should be made, i.e., transformation should be made to B and Dcont.

\[
\begin{align*}
u_{range} &= [\text{Base_max}\ Base_max]; \\
u_{scale} &= \text{ones}(1,\text{length}(u_{range}))/u_{range}; \\
u_{m\_scale} &= \text{diag}(u_{scale}); \\
\end{align*}
\]
overall_B = overall_B*u_m_scale;
overall_D = overall_D*u_m_scale;

overall_Cont_ss = ss(overall_A,overall_B,overall_C,overall_D);

% Now export to C

Output_Controller(overall_Cont_ss,'..\program\high_gain_cont.h');

C.1.5 The Script to Discretize the continuous time state space controller and to Export the Controller As A C Head File

Function: [ ] = Output_Controller(Cont,file)

% Input: Cont: State Space Representation of the Controller
% file: Name of the output C header file for the controller

function [] = Output_Controller(Cont,file)

Acont = Cont.a;
Bcont = Cont.b;
Ccont = Cont.c;
Dcont = Cont.d;

% The continous model is discretized by a bilinear transformation. This will % project the open left hand side coordinate plane into the unit circle, hen % the discretized model will not change the stability property of the contin % time model.

[A,B,C,D] = bilinear(Acont,Bcont,Ccont,Dcont,400) ;
% Now get the dimension informations of the system

[n,m] = size(B); [r,nr] = size(C);

out = fopen(file,'wb');

% Here the name "servocompensator" is the name of the control function to be
% called in the C program.
fprintf(out,'void servocompensator(double *Cont_inputs,double
*Plant_inputs)\n'); fprintf(out,'\n');

fprintf(out,'float xd[%i];\n',length(A));
fprintf(out,'int i;\n'); fprintf(out,'int j;\n');

fprintf(out,'static float x[%i]={'\n',n});
for i = 1:n-1
    fprintf(out,'0,');
end fprintf(out,'0};\n');

fprintf(out,'if(State!=6)\n');
for i = 1:n
    fprintf(out,'x[%i]=0;\n',i-1) ;
end
for i = 1:m
    fprintf(out,'Cont_inputs[%i]=0;\n',i-1) ;
end
for i' = 1:r
    fprintf(out,'Plant_inputs[%i]=0;\n',i-1) ;
end


for i = 1:r
    fprintf(out,'Plant_inputs[%i]=',i-1);
    for j = 1:n
        fprintf(out,'(%f)*x[%i]+',C(i,j),j-1);
    end
    fprintf(out,'0')
    for j = 1:m
        fprintf(out,'+(%f)*Cont_inputs[%i]',D(i,j),j-1);
    end
    fprintf(out,'
');end

for i = 1:n
    fprintf(out,'xd[%i]=',i-1);
    for j = 1:n
        fprintf(out,'(%f)*x[%i]+',A(i,j),j-1);
    end
    fprintf(out,'0')
    for j = 1:m
        fprintf(out,'+(%f)*Cont_inputs[%i]',B(i,j),j-1);
    end
    fprintf(out,'
');end


fprintf(out, '
');
fprintf(out, 'for(i=0;i<%i;i++)
');
fprintf(out, 'x[i] = xd[i];
');

fclose(out);

C.2 C Codes For Controller Implementations

C.2.1 C Codes For The Implementation of the High Gain and Cheap Control Servomechanism Controllers

//UofT.c -- 1 October 1997, vrp Web Technology, FA

//To be implemented at University of Toronto

//The following functions must be included:

// void UofT_Init(void); --called by the MTSC32 during init phase
// void UofT_Interrupt(void); --called by the MTSC32 at every sampling inst

/****** NECESSARY HEADER INCLUDES ******/

#include <sysdef.h>
#include <ec32.h>
#include <kernel32.h>
#include <bin32.h>
#include <ana32_11.h>
#include <enc32_29.h>
#include <math.h>

#include "servocompensator.h" // this name should be changed to

// appropriate head file of the
// controller to be implemented

#define State signals[35]

/***********************************************/

/************ UOFT AVAILABLE DEFINES **********/

/* Operation Switches */
#define sw_7_7 switches[7] & 0x40
#define sw_7_8 switches[7] & 0x80
#define sw_7_9 switches[7] & 0x100
#define sw_7_10 switches[7] & 0x200
#define sw_7_11 switches[7] & 0x400
#define sw_7_12 switches[7] & 0x800
#define sw_7_13 switches[7] & 0x1000
#define sw_7_14 switches[7] & 0x2000
#define sw_7_15 switches[7] & 0x4000
#define sw_7_16 switches[7] & 0x8000
#define sw_7_17 switches[7] & 0x10000
#define sw_7_18 switches[7] & 0x20000
#define sw_7_19 switches[7] & 0x40000
#define sw_7_20 switches[7] & 0x80000
#define sw_7_21 switches[7] & 0x100000
#define sw_7_22 switches[7] & 0x200000
#define sw_7_23 switches[7] & 0x400000
#define sw_7_24 switches[7] & 0x800000
#define sw_7_25 switches[7] & 0x1000000
#define sw_7_26 switches[7] & 0x2000000
#define sw_7_27 switches[7] & 0x4000000
#define sw_7_28 switches[7] & 0x8000000
APPENDIX C. SCRIPTS FOR CONTROLLER DESIGNS

```c
#define sw_7_29 switches[7] & 0x10000000
#define sw_7_30 switches[7] & 0x20000000
#define sw_7_31 switches[7] & 0x40000000
#define sw_7_32 switches[7] & 0x80000000

/* Monitor Signals */

// define the plant references to be monitored
#define mFa_ref signals_ex[23]
#define mVb_ref signals_ex[24]
#define mFb_ref signals_ex[25]

// define error signal to be monitored
#define Fa_err signals_ex[26]
#define Vb_err signals_ex[27]
#define Fb_err signals_ex[28]

// define the modified tensions to be monitored
#define mFa signals_ex[30]
#define mFb signals_ex[31]

// define the plant output values
#define Fa_modi (signals[13]+0.31795)/1.4457
// linearity between the measurement and the real tension. In future work,
// these linearity values should be modified from the corresponding set up
// values.
#define Vb_act signals[2]*(3.1415926/30)*0.0415
#define Fb_modi (signals[14]-1.7522)/1.3314 //linearity between measu/rea

// define the reference values
/* Reference values */

// define the control signals

#define ma desired_values[54] // Torque to shaft A
#define mb desired_values[55] // Torque to shaft B
#define mc desired_values[56] // Torque to shaft C

/* Setup Values */ #define offset

***************************************************************************/

/****************************************** INIT FUNCTION **************/

void UofT_Init(void) {
    version_string[0]=0;
    strcpy (version_string,"ServoCompensator\0");
    // This string should be changed to name of the controller to be
    // implemented.

    return;
}  
/******************************************

/****************************************** INTERRUPT FUNCTION **************/

void UofT_Interrupt(void) {
float Cont_inputs[6];
float Plant_inputs[3];

Cont_inputs[0] = Ref_Fa;
Cont_inputs[1] = Ref_Vb;
Cont_inputs[2] = Ref_Fb;
Cont_inputs[3] = Ref_Fa - Fa Modi;
Cont_inputs[4] = Ref_Vb - Vb_act;
Cont_inputs[5] = Ref_Fb - Fb Modi;

servocompensator(Cont_inputs, Plant_inputs);
// This function should be changed to the controller function name in
// the appropriate head file of the controller to be implemented.

/* have to adjust the outputs here */
output_ma = Plant_inputs[0];
output_mb = Plant_inputs[1];
output_mc = Plant_inputs[2];

if (State==4)
{
    Plant_inputs[0] = Ref_Fa*signals[54];
    // refa * ra, start up behavior, rollers are static
    Plant_inputs[2] = -Ref_Fb*signals[114];
    // refb * rc, start up behavior, rollers are static
}

ma = (Plant_inputs[0] + 0.055743)/0.84455 ;
mb = (Plant_inputs[1] - 0.042655)/0.54887 ;
mc = (Plant_inputs[2] - 0.064914)/1.174 ;
// These coefficient values are used to correct for the linearity of the
// input signals. They should be modified by the corresponding set up
// values in the future work.

mFa_ref = Cont_inputs[0];
mVb_ref = Cont_inputs[1];
mFb_ref = Cont_inputs[2];

Fa_err = Cont_inputs[3];
Vb_err = Cont_inputs[4];
Fb_err = Cont_inputs[5];

mFa = Fa_modi;

mFb = Fb_modi;

} }/**********************************************************************/