Linear Robust Control in Indirect Deformable Object Manipulation

LINEAR ROBUST CONTROL IN INDIRECT DEFORMABLE OBJECT MANIPULATION

BY

STEVEN KINIO, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2013) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Linear Robust Control in Indirect Deformable Object Manipulation
AUTHOR:	Steven Kinio B.Eng., (Electrical and Biomedical Engineering) McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Alexandru Patriciu
NUMBER OF PAGES:	xvi, 107

This thesis is dedicated to my amazing parents. Thanks for all the sacrifices you made raising me (looking at you mom), and for being supportive in everything that I do. You guys are awesome!

Abstract

Robotic platforms have several characteristics such as speed and precision that make them enticing for use in medical procedures. Companies such as Intuitive Medical and Titan Medical have taken advantage of these features to introduce surgical robots for minimally invasive procedures. Such robots aim to reduce procedure and patient recovery times. Current technology requires platforms to be master-slave manipulators controlled by a surgeon, effectively converting the robot into an expensive surgical tool. Research into the interaction between robotic platforms and deformable objects such as human tissue is necessary in the development of autonomous and semi-autonomous surgical systems.

This thesis investigates a class of robust linear controllers based on a worst case performance measure known as the H_∞ norm, for the purpose of performing so called Indirect Deformable Object Manipulation (IDOM). This task allows positional regulation of regions of interest in a deformable object without directly interacting with them, enabling tasks such as stabilization of tumors during biopsies or automatic suturing. A complete approach to generating linear H_∞ based controllers is presented, from derivation of a plant model to the actual synthesis of the controller.

The introduction of model uncertainty requires μ synthesis techniques, which extend H_∞ designs to produce highly robust controller solutions. In addition to H_∞

and μ synthesis designs, the thesis presents an approach to design an optimal PID controller with gains that minimize the H_∞ norm of a weighted plant. The three control approaches are simulated performing set point regulation in MATLAB^{TM}'s *simulink*. Simulations included disturbance inputs and noises to test stability and robustness of the approaches. H_∞ controllers had the best robust performance of the controllers simulated, although all controllers simulated were stable. The H_∞ and PID controllers were validated in an experimental setting, with experiments performed on two different deformable synthetic materials. It was found that H_∞ techniques were highly robust and provided good tracking performance for a material that behaved in a relatively elastic manner, but failed to track well when applied to a highly non-linear rubber compound. PID based control was outperformed by H_∞ control in experiments performed on the elastic material, but proved to be superior when faced with the nonlinear material. These experimental findings are discussed and a linear H_∞ control design approach is proposed.

Acknowledgements

I'd like to acknowledge Dr. Alexandru Patriciu for being an amazing supervisor and friend these past few years, as well as a great teacher in my undergraduate years. I'd also like to thank Dr. Sirouspour and Dr. Davidson for their advice and help throughout my Masters, and for putting up with my pestering. Finally, I'd like to thank Cheryl Gies for always being friendly and helpful.

Notation and abbreviations

Abbreviations:

SISO	Single Input Single Output
MIMO	Multiple Input Multiple Output
IDOM	Indirect Deformable Object Manipulation
MP	Manipulation Point
CP	Control Point
DO	Deformable Object
PID	Proportional Integrative Derivative
FEM	Finite Elements Model
BIBO	Bounded Input Bounded Output
LSMP	Lumped Spring Mass Damper
SVD	Singular Value Decomposition
DGKF	Doyle, Glover, Khargonekar and Francis, A famous solution to the H_∞ Control Problem
ARE	Algebraic Ricatti Equation
LQR	Linear Quadratic Regulation
LQE	Linear Quadratic Estimation
BaB	Branch and Bound Search
SQP	Sequential Quadratic Programming

State-

Space:

A	State Matrix
B	Input Matrix
C	Output Matrix
D	Feedthrough Matrix
x	State Vector

Deformable-

Object:

K	Stiffness Matrix
Ξ	Damping Matrix
M	Mass Matrix

Modified-

Plant:

$(.)_r$	Reduced Plant Matrix
$(.)_e$	Error Weight Matrix
$(.)_u$	Control Weight Matrix
$(.)_n$	Noise Weight Matrix
$\mathcal{K}_r$	Krylov subspace of order r
Δ	Uncertain System
$W_{(.)}$	Weighting System

Controllers:

I	Identity Matrix
I_{ij}	The element in the i^{th} row and j^{th} column of Matrix I
0	Zero Matrix
G	Block Dynamical System
P	Block Plant
K	Controller Dynamical System
w	Exogenous Input Signal
u	Control Input Signal
F_d	Disturbance Force Input Signal
w_{noise}	Normalized White noise Input signal
$\Phi(.)$	Barrier Function For Unconstrained Optimization
$\bar{\sigma}(G)$	Largest singular value of system G
γ	Upper Bound of ∞ norm of system
μ	Structured Singular Value, bound for uncertain systems
K_p	Proportional Gain Term of PID Controller
K_d	Derivative Gain Term of PID Controller
K_i	Integrative Gain Term of PID Controller

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Manipulation of deformable objects is a key field of research in the development of autonomous and semi-autonomous medical robotic systems. Discoveries in this field benefit not only medicine, but a large variety of industrial applications as well, such as food processing and textile manufacturing.

This thesis explores several robust linear control design approaches for the task of Indirect Deformable Object Manipulation (IDOM). This task, also referred to in the literature as Indirect Simultaneous Positioning (ISP), was first introduced by Wada *et al.* (1998). IDOM involves using N manipulators grasping a deformable object at N locations, henceforth referred to as manipulation points (MP), to indirectly regulate the position of M control points (CP). Figure 1.1 illustrates this for a task involving two robotic actuators moving two control points.

For the applications investigated in this thesis, a region of the deformable object is assumed to be connected to ground. In a real application setting, this ground can

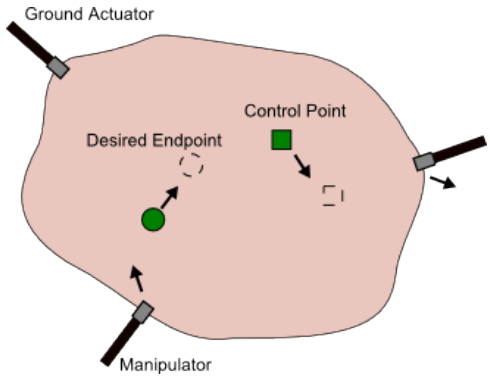

Figure 1.1: Indirect Set Point Regulation

also take the form of an additional actuator attached to the object. The complex behaviour of deformable objects pose many challenges to designers wishing to craft optimal controllers for IDOM tasks. In biomedical applications, a large number of variables influence the material parameters of human tissue, such as a subject's age, sex, and fitness levels. These factors, coupled with the overall nonlinear nature of human tissue, mean that conventional linear modelling techniques may fail to provide accurate predictions of deformation behaviour. These challenges require that controller designs take parameter uncertainty into consideration, in order to provide stability margins that are acceptable for medical applications.

IDOM has a variety of applications that make it an important research field. In textile manufacturing, fabric must be guided underneath sewing needles along certain paths to produce the appropriate stitch patterns. The region of the fabric that must go underneath the needle cannot be directly handled, necessitating an indirect control approach. This type of task was discussed by Wada *et al.* (2001). In terms of biomedical applications, there are several procedures which can utilize IDOM techniques, including suturing tasks, tissue approximation, and tissue stabilization in tasks such as needle biopsies. Suturing has been explored by Nageotte *et al.* (2005),

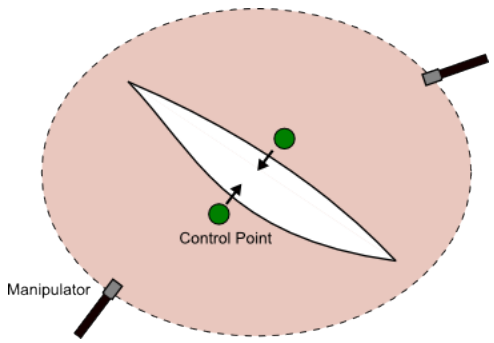

Figure 1.2: Indirect Deformation Control in Suturing Task

Khabbaz and Patriciu (2011), and Kang and Wen (2000), all of whom described various approaches to use a robot to autonomously sew shut sutures.

Figure 1.2 shows how indirect manipulation could be used to hold incisions together to simplify the suturing task. Tissue stabilization is another application field for IDOM. Stabilization, coupled with imaging technologies such as MRI, can help radiologists increase the biopsies accuracy. This task has been investigated by Smolen and Patriciu (2009), but for different control architectures than those investigated in this thesis. Another potential biomedical application for indirect control is related to the task of tissue retraction. Retraction, illustrated in Figure 1.3 involves moving flaps of tissue away from the surgical workspace. Robotic assisted retraction is discussed in Poulose *et al.* (1999) and Patil and Alterovitz (2010).

1.2 Prior Work

Papers addressing topics related to deformable objects (DO) can be divided into two broad categories, those that focus primarily on modelling the behaviour of such objects, and those that focus on control tasks involving deformable objects.

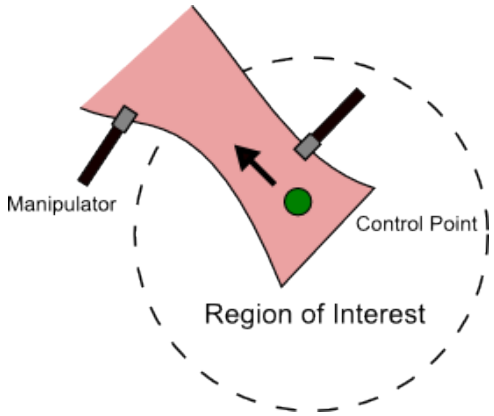

Figure 1.3: Indirect Deformation Control in Tissue Retraction Task

1.2.1 Modelling Review

Initial papers regarding modelling of DOs were focused primarily on deformable object modelling for computer graphics applications. An early paper in this field was published by Terzopoulos *et al.* (1987). The paper presented a detailed derivation of the process the authors used to model flexible objects, beginning with an object's dynamics and energies, and resulting in a mass-spring-damper formulation. McInerney and Terzopoulos (1996) provided a survey of the variety of techniques used to model DOs, ranging from energy-minimizing formulations to probabilistic models. Another survey, detailing modelling approaches such as continuum mechanics, was presented in Gibson and Mirtich (1997). More recent papers relevant to IDOM include Bro-Nielsen (1998), who discussed using finite elements modelling techniques to emulate the behaviour of human tissue for use in surgical simulations. Their plant model was also comprised of mass, damping and stiffness matrices, with damping being defined as a gain scaled version of the mass matrix. Henrich *et al.* (1999) presented a paper discussing different classes of deformation that can be encountered in robotic tasks, ranging from elastic to plastic deformation behaviour. The paper

included a discussion on the effect of motion on the deformation behaviour of non-rigid objects. Tokumoto *et al.* (1999) described a method to model the deformation behaviour of viscoelastic objects. They proposed a mesh composed multiple nodes connected with so-called four element models, which consists of a spring and damper in parallel connected to a spring and damper in series.

1.2.2 Control Approaches for Deformable Object Manipulation

We begin our review of control of deformable objects with Wada *et al.* (1998), who first described the IDOM task in literature. This paper modelled the deformable object with a mesh of springs, and used an iterative control law to regulate the desired position of control points. An observer was used to estimate the position of unmeasured points in the mesh. Experiments performed on acrylic-wool knitted fabrics validated the authors' approach. Wada *et al.* (2001) expanded on their 1998 paper, using PID control to perform simultaneous set point regulation, with applications in textile manufacturing. Simulations were performed to prove the validity of the approach. In a similar vein, Wakamatsu *et al.* (2006) proposed a planning method to knot and unknot deformable objects for industrial and medical applications, and Koustoumpardis and Aspragathos (2007) designed a neural network controller to regulate forces applied to fabrics during sewing tasks. Koustoumpardis et al. contrasted their neural net with a PID controller, concluding that although PID control had satisfactory performance, the neural network was more robust.

Robustness considerations have been the focus of several papers involving control of deformable objects. Hirai and Wada (2000) proposed a robust control approach

to IDOM, introducing three uncertain terms into their formulation to account for unmodelled behaviour of their control points, manipulation points, and uncontrolled points. Their controller was generated based on a linearized form of the uncertain plant. The practicality of the approach was demonstrated through experimentation, in which the controller was used to simultaneously regulate three control points on an acrylic-wool fabric, mirroring Wada *et al.* (1998). Torabi *et al.* (2009) expanded on the control law presented by Hirai and Wada to stabilize tissue during prostate needle insertion procedures. The authors introduced an integrating term to the controller to combat steady state error. The approach was tested in simulations with a 2D spring mass environment modelling the prostate and surrounding tissues. The controller performed well, reducing error from 2cm in uncontrolled systems to 1mm. Fanson and Patriciu (2010) presented an output regulation based approach to IDOM, capable of compensating for both uncertainties and nonlinearities in the deformable object model. The approach was validated through simulation and experimentation of a task involving one CP and one MP.

Several papers have discussed IDOM control for medical applications, including Smolen and Patriciu (2009), who proposed a Jacobian transform based control law, based on a meshless model of a deformable object such as human tissue. Mallapragada (2009) proposed a PI controller in combination with a planner for real time tumor manipulation in breast biopsies. Reference signals were fed to the PI controller by the planner based on target location and needle orientation. Their controller was used to ensure biopsy targets stayed in line with the biopsy needle during insertion procedures.

Interested readers can find a comprehensive overview of deformable object manipulation in Henrich and Worn (2000) textbook on the subject. A detailed discussion of stability considerations associated with IDOM control can be found in Shibata and Hirai (2006), who extended the spring meshes discussed by Wada et al. and added mass and damping considerations to their model of the deformable object. Their paper used tuned PID gains to control a 1-D linear IDOM task. The authors concluded that stable gains for IDOM tasks are both upper and lower bounded. Due to the generalized nature of the stability analysis performed, their work can be extended to higher dimension problems.

1.2.3 Investigated Control Laws

As discussed in Gibson and Mirtich (1997), Hirai and Wada (2000), Torabi *et al.* (2009) and Fanson and Patriciu (2010), uncertainties and nonlinearities inherent in the process of modelling DOs require controllers capable of compensating for model mismatch. For linear control approaches, this means that adaptive or robust control is necessary. This thesis focuses on a robust control approach. The two main controller architectures investigated in this thesis are the H_∞ controller and the PID controller.

H_∞ control is an extension of H_2 control methods, which is itself an extension of Linear Quadratic Control (LQR). A detailed discussion of relevant aspects of H_∞ control is covered in Chapter 4, for now it suffices to say that it is based on minimizing worst case gains on disturbance inputs. Early papers on the topic, such as Francis (1987), focus on a transfer function based solution. Modern state space approaches to the problem were first presented by Doyle *et al.* (1988). The synthesis approach

to creating these controllers has since been expanded upon, with a solution for general plants presented by Zhou (1998) and linear matrix inequality based solutions presented by Gahinet and Apkarian (1994).

PID control is widely used in industry, see Yamamoto and Hashimoto (1991), and has appeared in various IDOM papers, see Wada *et al.* (2001), Koustoumpardis and Aspragathos (2007), Mallapragada (2009); these factors meant it was a good candidate to contrast to the performance of H_∞ control. Parameters for the PID controller are obtained through tuning, such as that outlined in Ziegler and Nichols (1993), and through an optimization procedure. Optimal PID control is not a new subject, but most literature focuses on optimal SISO control instead of the MIMO problems discussed in this thesis. An important paper in this region includes Astrom *et al.* (1998), who presented a two phase transfer function based optimization approach based on initially optimizing integral gains for the control task before moving into a full PID design. Their approach takes into account robustness considerations by including the maximum sensitivity to disturbances M_s into their objective function. Through plots of the robustness criteria, the authors show that PID optimization is an inherently non-convex problem. Another important paper relevant to topics investigated in this thesis is Panagopoulos *et al.* (2002), who proposed a methodology in which to apply H_∞ loop shaping principles to the design of PID controllers. They focus primarily on the SISO problem case, and their objective function is based primarily on the ∞-norm of the robustness measure M_s. They utilize weighting functions to tune the balance between robustness and performance of their subsequently designed controllers.

Other papers on selection of PID parameters include Blanchini *et al.* (2004), who suggested a geometric approach to choosing PID parameters satisfying specified H_∞

bounds, and Ho (2003), who presented an H_∞ based PID optimization approach. Blanchini's approach requires fixing one DOF in a 3 element PID design (ie fixing K_p, K_i or K_d), and generating plots of 2D plots of regions satisfying a combination of stability and upper H_∞ bound γ restrictions. Ho's approach details primarily a SISO case. Bounds for the optimization variables are calculated and plotted by fixing K_p, and generating 2D regions similar to those described by Blanchini et al. This allows Ho to simplify the optimization into a convex formulation, and drastically reduced his computational cost.

1.3 Contributions

This thesis investigates the effectiveness of H_∞ controller synthesis approaches when applied to the task of indirect deformable object manipulation. A derivation of a deformable object model is presented, and modifications to the plant such as uncertainties and weighting functions are discussed. The extension of the design approach to use μ synthesis methods is briefly covered. To contrast the performance of the H_∞ controller, a method to synthesize a MIMO PID controller with an optimal closed loop ∞-norm is derived. The controllers are simulated for various plant models, and experiments are performed on the H_∞ controller and both tuned and optimal PID controllers to determine their behaviour in a practical settings. This research has resulted in the following papers

- A Comparative Study of H_∞ and PID Control for Indirect Deformable Object Manipulation, ROBIO 2012, Steven Kinio and Alexandru Patriciu

- An Investigation of Linear Control Approaches for Indirect Deformable Object Manipulation (To Be Submitted), Steven Kinio, Alexandru Patriciu

1.4 Thesis Layout

This thesis is organized as follows: Chapter 2 outlines the deformable object modelling. Chapter 3 details how designers can modify the basic plant model derived in Chapter 2 to reduce computational complexity, incorporate design objectives, and account for the effects of uncertainty. Chapter 4 provides an overview of H_∞ control, and discusses the formulation approach used to place the plant in proper form to synthesize controllers. The Chapter concludes by extending the H_∞ plant to solve μ synthesis control problems. Chapter 5 discusses an approach that enables designers to produce optimal gain parameters for a stable MIMO PID controller. Chapter 6 provides the reader with performance of the controllers discussed earlier in the thesis, with simulations and experimental results being presented. Chapter 7 draws conclusions from the results presented in Chapter 6, and discusses future avenues of research.

Chapter 2

Modelling the Deformable Object

The controller design approaches investigated in this thesis require mathematical representations, or models, of the systems that the robots will interact with. For the IDOM task, the model provides the effect of forces applied at MPs on the position of CPs. Wada *et al.* (1998) extended the basic concept of a spring equation to include multiple interconnected nodes attached via springs, resulting in

$$Kx = f \tag{2.1}$$

where K is referred to as a stiffness matrix, x is a vector of nodal displacements, and f is a vector of forces applied to the nodes in the model. In this thesis, K was generated Finite Elements techniques.

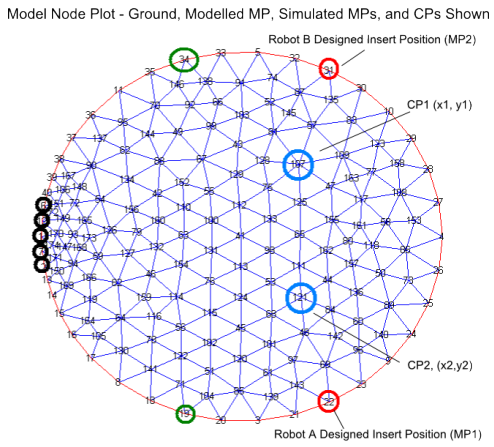

Figure 2.1: Finite Elements Model of Circular Deformable Object

2.1 Finite Elements Modelling

Finite elements is a broad and complex topic, and for a detailed description of FEM theory, we refer the reader to Hughes (2000) and Bhavikatti (2007). In this section, we will briefly describe how to apply FEM methods to generate a nominal plant for an IDOM task. The modelling process for a 2D object involves defining boundaries of an object, and then meshing the bounded region to generate a large number of nodes. See Figure 2.1 for a representation of the node mesh generated for a 2D circular deformable object. The nodes within the mesh are interconnected, and their interactions can be written in equation form.

If the conventional FEM approach is used to obtain these equations, see Hughes (2000), they can be written in the form

$$\begin{aligned} K_{11}x_1 + K_{12}x_2 + \ldots + K_{1n}x_n &= F_1 \\ &\vdots \\ K_{n1}x_1 + K_{n2}x_2 + \ldots + K_{nn}x_n &= F_n \end{aligned} \tag{2.2}$$

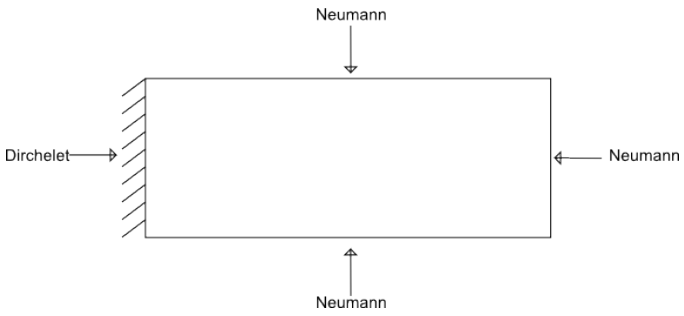

Figure 2.2: Examples of Object Boundaries

which when written in matrix form matches equation 2.1. Software packages such as MATLAB^{TM}'s *pdetool* and FLEXPDE^{TM} greatly simplify the use of FEM. To build a finite elements model using such software, designers specify boundary conditions, geometric dimensions, and material properties such as Young's modulus and the Poisson ratio. As stated above, boundary conditions play an important role in the synthesis of an FEM model. There are two primary boundary conditions used to define a deformable object model, Dirchelet boundaries and Neumann boundaries. Figure 2.2 illustrates how these boundaries are defined for a simple rectangular deformable object that is grounded on one of its minor edges.

2.1.1 Dirchelet Boundary

Dirchelet boundaries, also referred to as first-type boundary conditions, fix the solution of PDEs along the boundary to a defined value. In the case of modelling DOs, this means fixing deformations of certain nodes. These boundaries can be used to enable nodes on the to boundary to behave as if they were grounded, if the allowable displacement is set to 0. Grounding nodes of the model allows for a unique solution for deformation configurations. A Dirchelet boundary acting as a ground takes the

form

$$x_i = 0 \tag{2.3}$$

where x_i is the displacement associated with a node i. Once a grounded node is set, the system of equations seen in 2.2 must be modified to result in a full rank stiffness matrix. Rows and columns associated with the bounded node are removed Polycarpou (2006), creating a new nonsingular square matrix.

2.1.2 Neumann Boundary

Neumann boundaries enable the edges of a model to deform. This is facilitated by imposing a derivative value at the boundary. In MATLAB$^{\text{TM}}$'s *pdetool* environment, Neumann boundaries are defined using the mixed boundary formulation

$$\tau n + Qx = g \tag{2.4}$$

Where x is displacement of nodes in our model, g is the desired force density, Q is spring scaling matrix, and n is a normal to the boundary, and τ is the stress. Converting equation 2.4 into Neumann form that enables edge deformation is achieved by setting $g = 0$, forcing the node to displace in order to maintain an equilibrium force of 0 N along the boundary, and setting $Q = 0$.

2.1.3 Refining the Model

DO models containing only stiffness terms lack responses to velocity and acceleration data. A common approach to improve the behaviour of such models is to add damping and inertial considerations into the plant formulation, as seen in Bro-Nielsen (1998)

and Terzopoulos *et al.* (1987).

A model incorporating these elements is discretized to

$$M\ddot{x} + Kx + \Xi\dot{x} = F \tag{2.5}$$

Where M is a mass matrix that generates inertial forces, K is a stiffness matrix behaving like a mesh of springs, Ξ is a matrix of damping terms, and x is vector of nodal displacements.

MATLAB$^{\text{TM}}$'s *pdetool* was used to generate the matrix K from 2.1. This synthesis approach left the Mass and Damping matrices undefined, and thus manual definition of these matrices was required. Two main approaches for defining Ξ were considered. The first is to build a matrix with the same structure as K. In this approach, the damping is defined as

$$\Xi = \alpha K \tag{2.6}$$

where α is an appropriately selected scaling term. This formulation is equivalent to attaching dampers in parallel with all the spring connections within the DO model.

For flat deformable objects, damping can be approximated as viscous friction between the object and the work surface. In this case, we define Ξ as

$$\Xi = \beta I \tag{2.7}$$

where β is a suitable damping coefficient. A variation of this block diagonal form of damping can be seen in Bro-Nielsen (1998), where the authors model damping as a scaled version of the Mass matrix. This formulation is advantageous in applications where damping is not uniform through the DO.

The Mass matrix is always defined in diagonal form. Each diagonal element of the mass matrix is associated with the inertial force in that dimension of a particular node in the model. For a 2D system with uniformly distributed mass, the diagonal terms of the matrix M can be defined as

$$m_{ii} = M_t/N \tag{2.8}$$

where m_{ii} is the diagonal mass element, M_t is the total mass of the object, and N is the number of nodes in the FEM element node. Introducing varying diagonal terms into M results in a non-uniform mass distribution.

Designers should also be aware that the linear matrices produced by FEM may not accurately reflect the behaviour of nonlinear systems. This can be partially addressed by introducing uncertainties into the overall plant model, and this will be discussed in later sections of this thesis. Due to the large number of states generated with FEM techniques, it is computationally expensive to introduce uncertainty into the resultant plant models. We will briefly explore an alternate approach, known as Lumped Spring-Mass-Damper, that has significantly fewer states.

2.2 Lumped Spring-Mass-Damper

A lumped spring mass damper model, or LSMP as we will refer to them as, attempts to describe the behaviour of the deformable object with a minimal number of states. For the particular problem considered in this thesis, seen in Figure 2.3, the two manipulation points and two control points would be the only states of the plant. The immediate appeal of this approach is that the low number of states required to

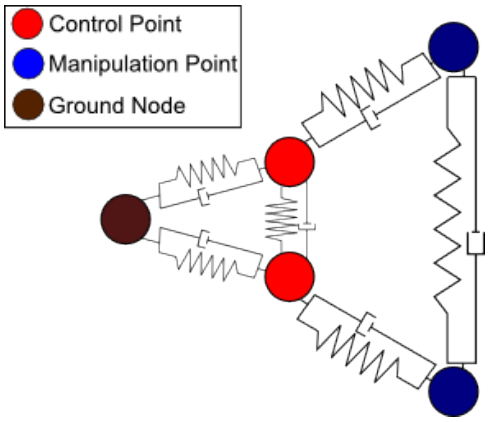

Figure 2.3: Examples of a Lumped Spring Mass Damper System

describe the plant results in a relatively low computational costs. This also makes LSMP a good candidate for systems where the designer wishes to include uncertainty, since uncertain terms may be tailored for each individual element in the model. Conversely, the greatest strength of this approach is also its greatest weakness, since the behaviour of the material between the input nodes and the output controls nodes must be lumped into single parameters. This generates two main problems, the first being difficulty in finding suitable lumped parameters, and secondly, a lack of precision in modelling the behaviour of the system once lumped parameters are selected. Depending on how the LSMP is initialized, it may be impossible to define "good" lumped parameters.

The lack of precision can be overcome when designers are seeking to build an uncertain plant, but for applications without uncertainty, this characteristic can prove to be a great hinderance.

2.3 First Order Plant Model

The H_∞ controllers investigated in this thesis are based on state space control techniques. A standard state space system is written in the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2.9}$$

Equation 2.5 contains an acceleration term $\ddot{x}$ which can be dealt with in two ways. The easiest way is to assume that the mass of each individual region of the deformable object is so low that the engineer can assume there is a negligible inertial force associated with the mass. This assumption works well when the DO is very thin, and can also help for low mass systems that would otherwise become poorly conditioned if mass was considered. The massless plant becomes

$$\Xi\dot{x} + Kx = Bu$$

rearranging, we get the standard form

$$\dot{x} = -\Xi^{-1}Kx + \Xi^{-1}Bu \tag{2.10}$$

For a standard Mass Spring Damper system, we must define a parameterized state $\hat{x}$ where

$$\hat{x} = \begin{bmatrix} x^T & \dot{x}^T \end{bmatrix}^T \tag{2.11}$$

Arranging 2.5 with regards to $\hat{x}$ results in

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}\Xi \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} u$$

This form reveals why equation 2.10 is relevant. Note that

$$M^{-1} = \frac{1}{m} I \tag{2.12}$$

Low values of m result in

$$m \to 0, M^{-1} \to \infty$$

This results in a poorly conditioned system, making system 2.10 the better choice.

The final major design choice in the definition of the plant is to select how forces from the controller will be applied to the deformable object. We will explore two approaches. The first, and simplest approach, is to determine individual nodes that are closest to the intended insertion/grasping point of contact of the actuators. Let us illustrate this concept with Figure 2.4.

In this figure, forces are applied in the region near node 3, thus the selector matrix takes the form

$$\hat{x} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

For forces in more than one dimension, ie x y z, states are usually represented in blocks corresponding to each dimension. Thus, for a system with 12 nodes, modelled

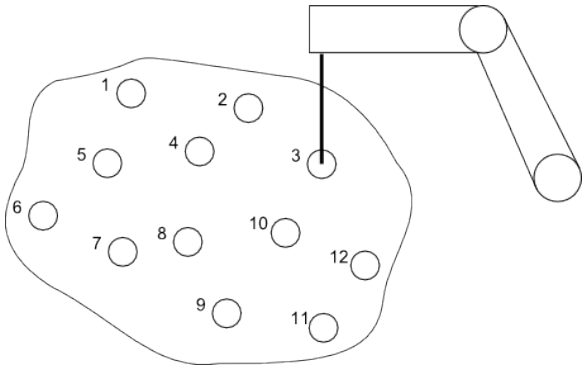

Figure 2.4: Deformable Object with FEM Plant

in 3D, the first 12 states would be associated with x forces, the next 12 would be y forces, and the final 12 would be z forces. Selector matrices for such a system use offsets to apply component forces to each of these states. Therefore, for the system seen in Figure 2.4, B would have 3 columns one dealing with each component force from the robot.

Although simple, this approach has several drawbacks: Forces must be approximated to be applied to a point mass, and must be treated as if they were exactly on the node, which may not reflect reality.

An alternative approach that more closely models a real application of forces involves distributing force inputs through a "force field" into the plant. This selector matrix applies a function based force, with magnitude of that force based on distances of nodes from the insertion point. Referring to Figure 2.4, our force application profile might look like Figure 2.5, where the actual force distribution would appear like a spike as seen in Figure 2.6.

This method is advantageous in the sense that if a force is applied between two nodes in the model, force can be distributed between them to try and better model the physical interactions.

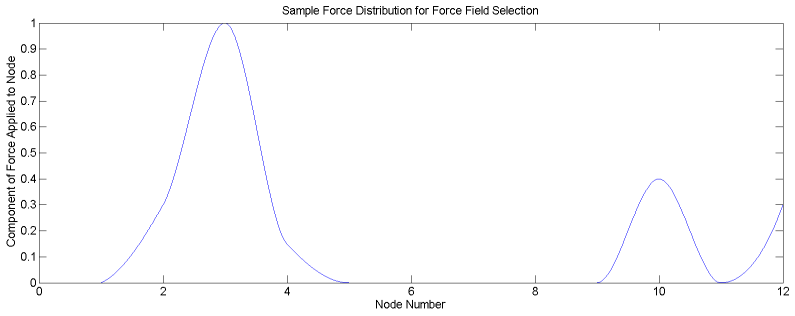

Figure 2.5: Sample Force Field Force Distribution

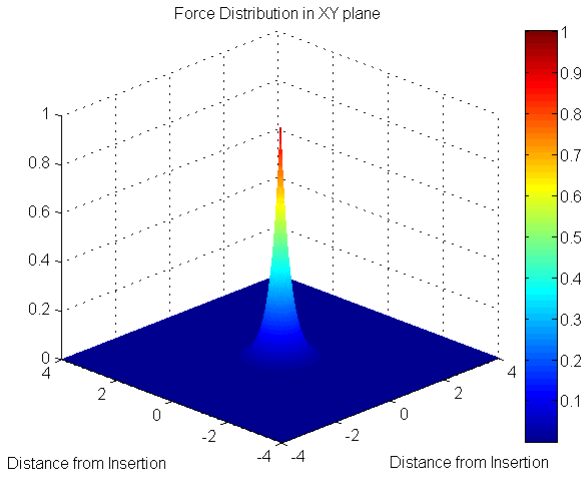

Figure 2.6: Force Field Distributed in XY Plane

Chapter 3

Plant Modification for Controller Synthesis

The nominal plant derived in the previous chapter only describes the expected behaviour of the material they are planning to manipulate. By modifying the nominal plant in the ways discussed in this chapter, engineers can take into account performance measures and computational issues that arise when designing optimal controllers. This chapter will discuss model reduction techniques which can simplify high order plants into more compact representations and loop shaping/ plant weighting techniques to provide solutions methods with design goal information. It will also include a brief discussion into the manner in which uncertain terms can be introduced to plant model.

We begin by discussing model reduction

3.1 Model Reduction

If the deformable object plant contains a large number of states, model reduction techniques can be used to reduce this number to a more manageable value. A. C. Antoulas and Gugercin (2001) provides a comprehensive overview of model reduction algorithms. This thesis focused primarily on two main forms of model reduction, these being a Schur based model truncation approach, detailed by Safonov and Chiang (1989), and a structure preserving reduction using Krylov subspace projections, presented by Li and Bai (2006). We will briefly discuss the general approach of these two techniques, and discuss their pros and cons.

3.1.1 Model Truncation

Schur based model truncation, as described by Safanov and Chiang, reduces the order of a system through use of Schur decomposition. Schur decomposition of a matrix E is

$$E = VUV^{-1} \tag{3.1}$$

where V is unitary, meaning $V^{-1} = V^*$, and U is an upper triangular matrix that shares the same eigenvalues as E. Safanov and Chiang's technique decomposes the product of the reachability Grammian P and observability Grammian Q of our system

$$G = C(sI - A)^{-1}B + D \tag{3.2}$$

The Grammians are obtained by solving the two Lyaponuv equations

$$\begin{aligned} PA^T + AP + BB^T = 0 \\ QA + A^TQ + C^TC = 0 \end{aligned} \tag{3.3}$$

When the Schur decomposition of the matrix product PQ has been completed, two matrices V_A and V_D are designed such that U's diagonal terms are in ascending and descending order respectively. Designers discard the smallest k columns of V_A and V_D, using the resultant matrices as bases for the eigenspace of their newly reduced system. For a detailed description of the method, readers are referred to Safonov and Chiang (1989).

This method performs no balancing on the reduced model to eliminate the ill-conditioned solutions associated with the balancing process, Safonov and Chiang (1989). Although this generalizes the solution, and allows it to address a larger number of input systems, the lack of balancing means designers must be careful on how many states they wish to crop from their nominal plant formulation. Model truncation enables designers to specify the number of states that they wish to maintain in their reduced system, which can be advantageous for applications with limited computational resources. Figure 3.1 compares the bode response of a nominal MIMO plant with 600 states first input to first output transfer function with the same transfer function for several reduced plants.

The shape of the responses are extremely close to one another in DC and low frequency bands, but the higher frequencies begin to have gain disparities in both slope and magnitude. The phase response of the systems remains similar, although the phase is shifted between the original system model and the final reduced plant. This

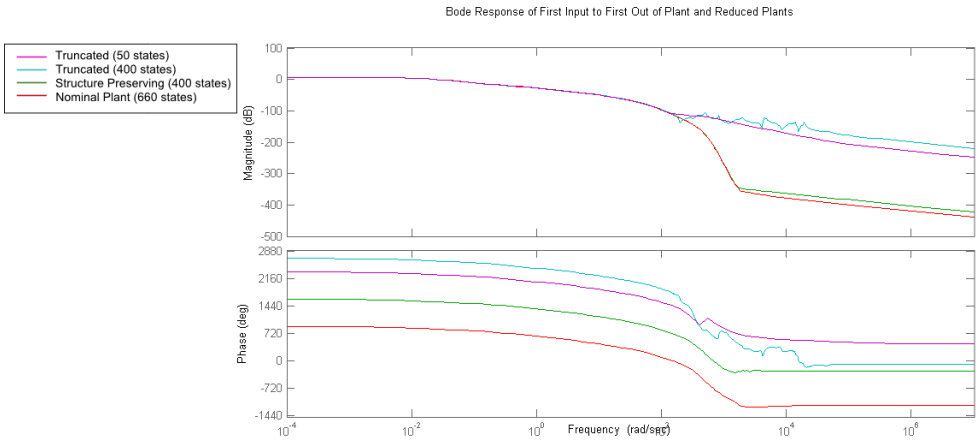

Figure 3.1: Model Reduction Performance

performance makes sense, since the truncation process eliminates the smallest/fastest poles of the system, resulting in a loss of modelling accuracy in high frequency bands. Attempts to reduce the number of states truncated from the original plant do not necessarily provide better behaviour in the truncated system, as seen in Figure 3.1, where the 400 state truncated plant still has divergent behaviour from that of the nominal system.

An additional disadvantage of truncation is that the method eliminates any structure that might have existed in the original block matrix definition of the system. This prevents designers from applying modifications to specific components of the plant, such as adding uncertainty to certain parameters in the reduced plant.

3.1.2 Structure Preserving Model Reduction

The structure preserving reduction used in this thesis was based off of methods described by Li and Bai (2006). Structure preserving reduction techniques are used when designers want to retain the original block structure of their nominal plant

after a reduction has been applied. Thus, for a Mass-Spring-Damper plant such as those derived in Chapter 2, we would reduce

$$A = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}\Xi \end{bmatrix} \tag{3.4}$$

to the system

$$A = \begin{bmatrix} 0 & I_r \\ -M_r^{-1}K_r & -M_r^{-1}\Xi_r \end{bmatrix} \tag{3.5}$$

Li and Bai use Krylov subspace projections in order to reduce a plant while still retaining its original structure, where a Krylov subspace of order r is defined as

$$\mathcal{K}_r(A, b) = \text{span}(b, Ab, A^2b, A^3b, \ldots A^{r-1}b) \tag{3.6}$$

For the plant described by 3.2, the reduced plant elements are calculated as

$$\begin{aligned} C_r &= XC \\ A_r &= YAX \\ B_r &= YBX \\ D_r &= YD \end{aligned} \tag{3.7}$$

X and Y are calculated by the following algorithm proposed by Li and Bai's Li and Bai (2006), where orth(Z) solves for the orthonormal basis of span(Z).

Once the iterations are complete, designers define

$$\tilde{X} = (Q_1, Q_2, \ldots Q_k) \tag{3.8}$$

Algorithm 1 Calculate X,Y

$\hat{A} = A$
$\hat{Q} = \hat{A}^{-1}D$
$Q_1 = \text{orth}(\hat{Q})$
for $j = 1$ to $k-1$ **do**
  $\hat{Q} = \hat{A}^{-1}BQ_j$
  for $j = 1$ to $k-1$ **do**
    $\hat{Q} = \hat{Q} - Q_i(Q_i^*\hat{Q})$
  end for
  $Q_{j+1} = \text{orth}(\hat{Q})$
end for

and then partition it into two block matrices

$$\tilde{X} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} \tag{3.9}$$

where $\tilde{X}_1$ contains the first N_1 rows of $\tilde{X}$, and X_2 contains the remaining rows of $\tilde{X}$. The value N_1 for our purposes is defined as the number of nodes in our FEM formulation of the plant, multiplied by the dimensionality of the control problem.

The values of X and Y are defined as

$$X = Y = \begin{bmatrix} \text{orth}(\tilde{X}_1) & 0 \\ 0 & \text{orth}(\tilde{X}_2) \end{bmatrix} \tag{3.10}$$

3.2 Modelling Plant Uncertainty

A model of a plant is always an approximation of the real behaviour of that system. Variation between selected parameters and the actual material can have large consequences on the stability and tracking accuracy of the final closed loop system.

Adding uncertainty into the plant enables designers to present an array of models to a controller synthesis technique, Zhou (1998). Weighting plays an important role when defining uncertain plants, since uncertainty is defined initially as a gain bounded all-pass parameter, with equal bounded gain behaviour across the entire frequency spectrum.

Uncertainty can be added to elements in a plant in two ways, either additively, or in a multiplicative manner. For a response G, we define additive uncertainty as

$$G_{unc} = G + W_1 \Delta W_2 \tag{3.11}$$

where W_1 and W_2 shape the uncertainty response, and Δ represents our uncertainty. Δ is usually given a basic norm bound, such as $\|\Delta\|_\infty \leq 1$. We define multiplicative uncertainty as

$$G_{unc} = G(I + W_1 \Delta W_2) \tag{3.12}$$

Multiplicative uncertainty is advantageous to additive uncertainty since it inherits the structure of the system it is applied to. Regardless of how the uncertainty is defined, there are three main points where uncertainty can be embedded into the plant.

Input uncertainty, as seen in Figure 3.2 occurs when the actuator used to apply force commands to the plant behaves in an uncertain manner. In the experiments performed for this thesis, the robot actuators interacted with the deformable object by grasping metal rods, which were then inserted into holders attached to the object itself. This medium of contact was unmodelled, and thus interactions between the metal rod, the holders, the robot, and the object all introduce uncertain behaviour

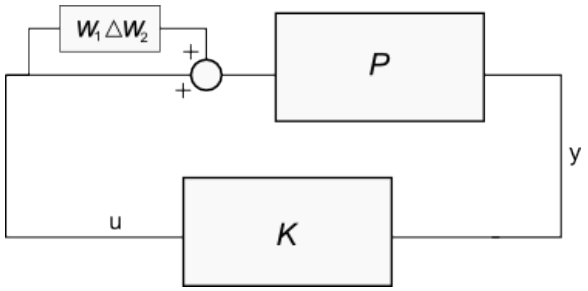

Figure 3.2: Input Uncertainty

on the forces commanded by the controller and those applied to the system itself. Our resultant uncertain plant P_{unc} is defined

$$P_{unc} = (I + W_1 \Delta W_2) P \tag{3.13}$$

Output uncertainty is applied on the measured output of a plant, and can result from mismatch in modelled positions of tracked states in the model versus the application task, and in the case of camera based tracking, skew associated with viewing angle of the task. We write this uncertain plant as

$$P_{unc} = P(I + W_1 \Delta W_2) \tag{3.14}$$

The final main class of uncertainties used by designers is internal plant uncertainty. This uncertainty can be applied to certain parameters within the plant, such as the damping in a spring-mass-damper plant. This type of uncertainty can be converted into one of the converted into one of the previous two uncertainties described above. For a FEM system

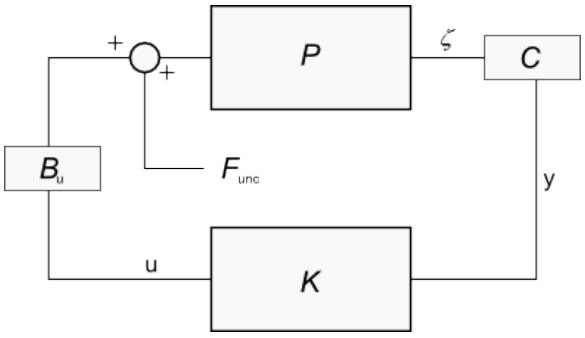

Figure 3.3: Internal Uncertainty Formulated In Force Input

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -M^{-1}K + \Delta_K & -M^{-1}\Xi + \Delta_\Xi \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} u \tag{3.15}$$

where Δ_K and Δ_Ξ are internal uncertainty matrices. We can separate our uncertain terms from our base plant,

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}\Xi \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} u + \begin{bmatrix} 0 & 0 \\ \Delta_K & \Delta_\Xi \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \tag{3.16}$$

these now behave as a force input F_{unc}, and we can rewrite our plant as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}\Xi \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} u + \begin{bmatrix} 0 \\ B_{unc} \end{bmatrix} F_{unc} \tag{3.17}$$

Resulting in a plant seen if Figure 3.3, with a plant P, controller system K, sensor C, control force selector B_u and uncertain "force" selector B_{unc}.

We can rearrange this block, pulling our uncertain force through the plant to

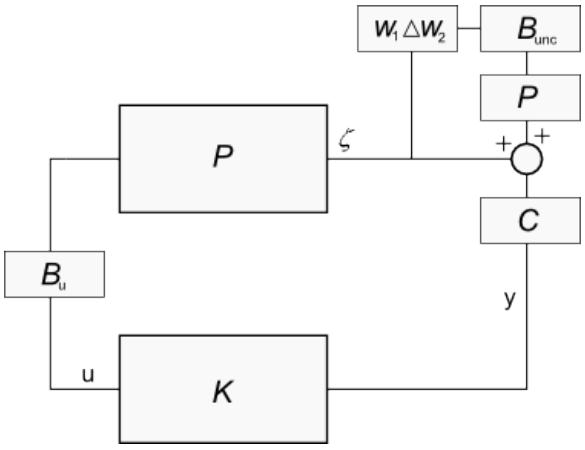

Figure 3.4: Modifying Internal Uncertainty

produce an additive state vector. This results in the closed loop seen in Figure 3.4.

3.3 Loop Shaping

Loop shaping is the process in which designers add weighting functions to their modelled system, in order to specify constraints in the design or desired performance specifications. There are a large variety of potential weights available to designers, as seen in Zhou (1998), Hu *et al.* (2000),Lundstrm *et al.* (1991) and many others.

This thesis focused on using three weights. A performance weight on the error output, emphasizing DC and low frequency tracking behaviour at the expensive of higher frequency regulation; a penalty weight on the control signal, again penalizing high frequency commands; a weight to shape a noise disturbance on the states fed into the controller. These weights are referred to respectively as W_e, W_u, and W_n. An equilibrium tracking plant incorporating these weight functions can be seen in Figure 3.5.

All these performance weights are defined as dynamic functions, and are usually,

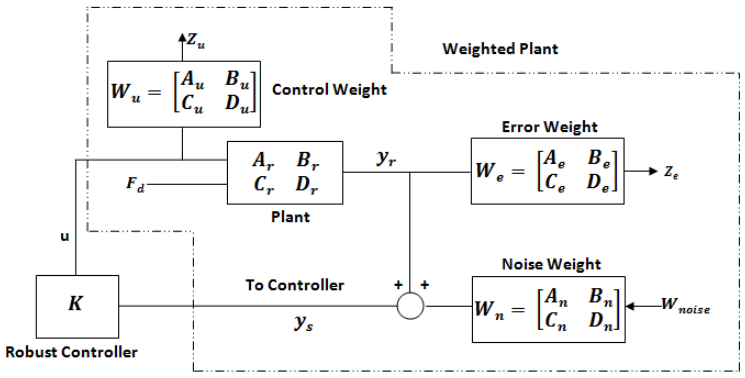

Figure 3.5: Weighted Plant and Controller

for the sake of ease, defined in the Laplace domain, and then converted into state space form. The first weight we will discuss is our performance weight W_e. W_e puts a frequency weight on the error signal, allowing designers to focus control on frequency bands that they want the best performance and disturbance rejection in. For the particular problems dealt with in this thesis, set point regulation was the primary task of interest, and thus, DC and low frequency error regulation were of primary concern to the designer. This suggests that our application's W_e should manner similar to that of a low pass filter. Indeed, the base form of W_e is defined as

$$W_e = \frac{s+1}{s+\epsilon} \tag{3.18}$$

The term ϵ pushes the pole of the weight off of the $j\omega$ axis. If designers wish to have a larger difference between their "passband" performance region and their "stopband", they can just base their weight off a higher order LPF derivation.

The weight W_u is used to model limitations of the actuators used by the controller. This enables designers to limit command forces generated by the controller, as well as limit the rate of changes in the commands. For this work W_u was selected according

to the methodology suggested by Zhou (1998), taking the form

$$W_u = \lambda \frac{s + W_{bc}/M_u}{\varepsilon s + W_{bc}} \tag{3.19}$$

Where λ is a tuned gain parameter, which in conjunction with the gain on W_u helps the designer select between performance and controller robustness. Where W_{bc} is the actuator/robot bandwidth, M_u is the desired maximum desired force, and ϵ is a constant to shift the pole away from the $j\omega$ axis. For our system, M_u was selected to be 5 N, and W_{bc} was selected to be 2.5.

W_n, models the noise of the control point measurement system. In our experimental setup the control points positions are provided by a stereoscopic camera. W_n was assumed to act as a weight on a normalized Poisson noise distribution. To determine the scaling factor to be used, a set of 10000 samples was generated from the camera, and standard deviation was calculated. W_n was then selected as

$$W_n = \frac{s/(10/2\pi) + 1}{s/(50/2\pi) + 1} \tag{3.20}$$

This function was selected as a low pass filter with a cutoff frequency equal with the sampling rate of our sensor camera. For higher dimensional systems, noise in each dimension can be accounted for by defining $\widetilde{W_n}$ as

$$\widetilde{W_n} = diag(n_i W_n)$$

Where $n_1 \dots n_m$ are the standard deviation noise components of each of the dimensional directions respectively.

3.4 Position Control Versus Force Control

The discussion thus far has focused on force control. One disadvantage of force control is the assumption that forces sensed by the robot correspond to forces applied to the deformable object. This may be untrue, especially if the force sensors attached to the robot actuator do not directly interact with the deformable object. Also, most existing surgical robots are position controlled. We can show that that assuming position interaction leads to the same plant form as the force interaction.

For a simple DO model composed of springs and dampers, the standard force control formulation is

$$\dot{x} = -\Xi^{-1}Kx - \Xi^{-1}Bf \tag{3.21}$$

Where Ξ is the damping matrix, K is the stiffness matrix, B is a force selector, and f is a vector of command forces from our controller. This can be partitioned into the form

$$\begin{bmatrix} \dot{x_{nc}} \\ \dot{x_c} \end{bmatrix} = \Xi^{-1} \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} x_{nc} \\ x_c \end{bmatrix} + \Xi^{-1} \begin{bmatrix} 0 \\ f \end{bmatrix} \tag{3.22}$$

Where x_{nc} are deformations of nodes where force is not directly applied, and x_c are the manipulated nodes. For position control, we must rearrange 3.22 to treat the manipulation states as our control input, and set our input force vector $f = 0$. We assume that the robot has sufficient force to ignore dynamics of the controlled nodes. The resultant system is of the form

$$\dot{x}_{nc} = \tilde{\Xi}_A^{-1}K_{11}x_{nc} + \tilde{\Xi}_B^{-1}K_{12}x_c \tag{3.23}$$

Where $\tilde{\Xi}_A$ and $\tilde{\Xi}_B$ are appropriately repartitioned damping parameters. This is

now in standard state space form, with the state matrix $A = \tilde{\Xi}_A^{-1} K_{11}$, and the control input matrix $B = \tilde{\Xi}_B^{-1} K_{12}$. Our new control signal is now x_c.

Chapter 4

H_∞ Control

4.1 The ∞-Norm

The infinity norm of a system $G(s)$ is defined as Zhou (1998)

$$\|G\|_\infty = \sup_\omega \bar{\sigma}\{G(j\omega)\} \tag{4.1}$$

That is to say, the infinity norm of a system is the largest singular value of that system for any frequency response. The definition in equation 4.1 must be discretized to become computationally feasible, since ω can take an infinite number of values. Designers wishing to approximate their system's ∞-norm must select a sampled frequency vector ω_d, ensuring they have sufficient resolution in their operation bandwidth to fully cover their design constraints. With ω_d, the problem becomes formulated as

$$\|G\|_\infty = \sup_{\omega_d \in R^{1xN}} [\bar{\sigma}\{G(j\omega_{d_1})\}, \ldots, \bar{\sigma}\{G(j\omega_{d_N})\}] \tag{4.2}$$

where $\bar{\sigma}\{G(j\omega_{d_i})\}$ can be calculated by performing an SVD (singular value decomposition) on $G(j\omega_{d_i})$

$$U\Sigma V^T = G(j\omega_{d_i})$$
$$\Sigma = U^T G(j\omega_{d_i})V$$

To obtain a vectorized form of Σ such that the sup operator is applicable, we multiply this formulation by a 1s vector of the form $\mathbf{1} = [1, 1, \ldots, 1]^T \in R^{1xN}$, where $\Sigma \in R^{NxN}$. Thus, the ∞-norm is defined as

$$\|G\|_\infty = \sup_{\omega_d \in R^{1xN}} (\sup \mathbf{1}U^T G(j\omega_{d_i})V, \ldots, \sup \mathbf{1}U^T G(j\omega_{d_N})V) \tag{4.3}$$

The formulation presented in 4.3 is easy to implement in mathematical software such as MATLAB$^{\mathrm{TM}}$. One drawback of this approach invovles the difficulty in selecting an appropriately sampled version of ω. There exists an alternative method based on the state space form of a transfer function which eliminates the need to select a vector ω_d. It can be shown that for any matrix M

$$\bar{\sigma}(M) < \gamma$$

if

$$\gamma^2 I - M^* M > 0 \tag{4.4}$$

This indirect relationship, Zhou (1998), can be coupled with a search algorithm such as a bisection search to find a tight upper bound on the value of $\bar{\sigma}(M)$, thus allowing it to be approximated as γ. For a system $G(s)$, relation 4.5 is rewritten in

the form Zhou (1998)

$$\Phi(s) = \gamma^2 I - G\tilde{}(s)G(s) > 0 \tag{4.5}$$

where $G\tilde{}(s) = G^T(-s)$, and $s \in \mathbb{R}$

To prove the resultant matrix is positive definite, it suffices to show that $\Phi^{-1}(s)$ contains no poles on the imaginary axis. Zhou (1998) derive the state space form of $\Phi^{-1}(s)$, which has an $\tilde{A}$ matrix of the form

$$\tilde{A} := \begin{bmatrix} A + BR^{-1}D^*C & BR^{-1}B^* \\ -C^*(I + DR^{-1}D^*)C & -(A + BR^{-1}D^*C)^* \end{bmatrix} \tag{4.6}$$

where $R = \gamma^2 I - D^*D$

This matrix is relabelled H and is referred to as the Hamiltonian matrix of our system $G(s)$. Taking an eigendecompostion of H allows a binary search to be constructed to find the tight bound γ. For detailed discussion of this binary search, the reader is referred to Zhou (1998).

To understand the significance of γ bound to the ∞-norm of our system one must consider the Small Gain Theorem

4.2 Small Gain Theorem

Small gain theorem provides a simple way to examine the robustness of a closed loop system, for a detailed discussion, the reader is referred to Slotine and LI (1991).

Given a system composed of a controlled closed loop M and uncertainty Δ, shown in Figure 4.1, we wish to determine the susceptibility of the closed loop to disturbance

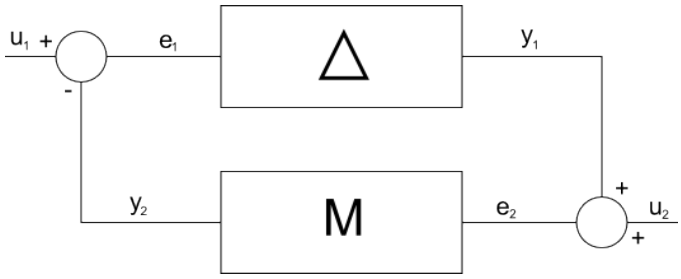

Figure 4.1: Closed Loop System for Small Gain Example

inputs. We define

$$\begin{aligned} e_1 &= u_1 - Me_2 \\ e_2 &= u_2 + \Delta e_1 \end{aligned} \tag{4.7}$$

This system closely resembles a deformable object manipulation task with disturbance inputs such as sensor noise and unmodelled dynamics. Investigating the magnitude of the error signals, the BIBO conditions for M and Δ are

$$\begin{aligned} \|y_1\| &\leq \gamma_1\|e_1\| + \beta_1 \\ \|y_2\| &\leq \gamma_2\|e_2\| + \beta_2 \end{aligned} \tag{4.8}$$

Given these conditions, we can now write

$$\begin{aligned} \|e_1\| &\leq \|u_1\| + (\gamma_2\|e_2\| + \beta_1) \\ \|e_1\| &\leq \|u_1\| + \gamma_2(\|u_2\| + \gamma_1\|e_1\| + \beta_2) + \beta_1 \\ \|e_1\| &\leq \|u_1\| + \gamma_2\|u_2\| + \gamma_1\gamma_2\|e_1\| + \beta_1 + \gamma_2\beta_2 \\ \|e_1\| &\leq \tfrac{1}{1-\gamma_1\gamma_2}(\|u_1\| + \gamma_2\|u_2\| + \beta_1 + \gamma_2\beta_2) \end{aligned} \tag{4.9}$$

As long as $\gamma_1\gamma_2 < 1$, the error resultant from disturbance signals is upper bounded, and thus the system is stable. When $\gamma_1\gamma_2 > 1$, equation 4.9 no longer holds, and it becomes difficult to prove a perturbed system remains stable. In regards to H_∞ control, recall that the ∞-norm is defined as the largest singular value of the system,

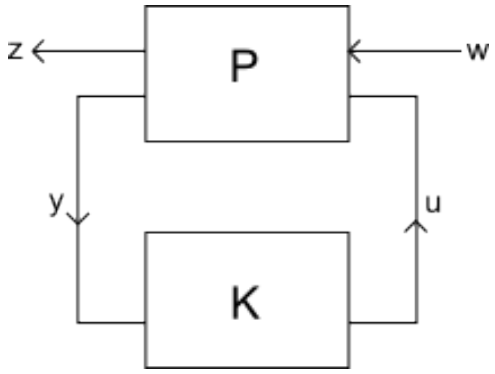

Figure 4.2: Closed Loop System for H_∞ Control

which corresponds to the largest gain of the system over all frequencies. Minimizing the γ_2 bound of our controlled closed loop M enables larger gains in the uncertain system Δ. Thus, reducing a controlled system's ∞-norm enables it to tolerate larger uncertainties.

4.3 H_∞ Controller

Now let us begin discussing H_∞ controller design techniques. The main goal of an H_∞ design process is to produce a dynamical controller system K that minimizes the transfer function $T_{z\omega}$'s ∞-norm. The closed loop for this class of controllers is usually represented in the form seen in Figure 4.2.

The plant P takes in an exogenous input ω, which contains reference signals, disturbance inputs, and noise sources, and the control input u. In turn, it outputs a measurement signal y to the controller and an error signal z. As seen earlier in this Chapter, it is difficult to directly calculate the ∞-norm, and thus most controller design techniques are described as sub-optimal, and focus on generating controllers that produce closed loops with an ∞-norm lower that γ.

There are several approaches to produce H_∞ controllers. Important ones are the Youla Parameterization, Francis (1987), and the so called DGKF two Ricatti approach. The Youla Parametrization is primarily a transfer function based approach, and a detailed description of its workings can be found in Francis (1987). DGKF was used to generate the H_∞ controllers presented in this thesis, and an in depth description of its derivation can be found in Doyle *et al.* (1989) and Zhou (1998). We will discuss the components of DGKF relevant to our approach.

4.3.1 DGKF

DGKF encompasses two H_∞ solutions methods, one for a simplified plant, and the other more general method to solve problems that cannot be put in the simplified form. This divide is based off a difference in the weighted plants that are fed into the synthesis process, and is entirely dependant on the way our error output z is defined.

For a simplified system, we define

$$z = C_1 x + D_{12} u$$

whereas for our general system, z is defined as

$$z = C_1 x + D_{11} u + D_{12} \omega$$

Several assumptions are made that must be satisfied to enable synthesis of stable controllers. For a simplified system, these include

1. (A,B_1) is controllable, (C_1,A) is observable

2. (A,B_2) is stabilizable, (C_2,A) is detectable
3. $D_{12}^* \begin{bmatrix} C_1 & D_{12} \end{bmatrix} = \begin{bmatrix} 0 & I \end{bmatrix}$
4. $\begin{bmatrix} B_1 & D_{21} \end{bmatrix} D_{21}^* = \begin{bmatrix} 0 \\ I \end{bmatrix}$

For the generalized synthesis problem, these assumptions are loosened to require

1. (A,B_2) is stabilizable, (C_2,A) is detectable
2. $D_{12} = \begin{bmatrix} 0 \\ I \end{bmatrix}$
3. $D_{21} = \begin{bmatrix} 0 & I \end{bmatrix}$

The final two assumptions for the general case can be obtained through reparameterization of a problem, and thus are less binding than the first assumption.

Both the simplified and general solutions are based off of creating controllers by solving a pair of Algebraic Ricatti Equations (ARE). An ARE is an algebraic version of a quadratic problem. AREs are of the form

$$A^*X + XA + XRX + Q = 0$$

This form is associated with a Hamiltonian matrix of the form

$$\begin{bmatrix} A & R \\ -Q & -A^* \end{bmatrix} \tag{4.10}$$

Astute readers will note that 4.6 matches the form seen 4.10, thus meaning that the ∞-norm γ bound can be written as a quadratic problem. The DGKF synthesis

of a controller uses two Ricatti equations, written in their Hamiltonian form and labeled $H\infty$ and J_∞. These matrices are defined with slight differences between the simplified and general solutions approaches, and interested readers are encouraged to read Zhou (1998) for details. To provide the reader with some context behind the synthesis process, we will briefly look at H_∞ and J_∞ as defined for the simplified problem.

$$H_\infty := \begin{bmatrix} A & \gamma^2 B_1 B_1^* - B_2 B_2^* \\ -C_1^* C_1 & -A^* \end{bmatrix} \tag{4.11}$$

$$J_\infty := \begin{bmatrix} A^* & \gamma^2 C_1^* C_1 - C_2^* C_2 \\ -B_1 B_1^* & -A \end{bmatrix} \tag{4.12}$$

There is a readily apparent duality between the matrices 4.11 and 4.12. Our dual terms include

$$A \leftrightarrow A^*$$

$$C_1 \leftrightarrow B_1^*$$

$$C_2 \leftrightarrow B_2^*$$

To understand the origins of these matrices, and the reason for their dual appearance, we will briefly look at LQR and LQE optimization problems. In a Linear Quadratic Regular (LQR) optimization, the objective function can be formulated in the Hamiltonian form How (2007)

$$H_{LQR} = \frac{1}{2}(x^T R_{xx} x + u^T R_{uu} u) + p^T (Ax + B_u u) \tag{4.13}$$

which can be rewritten in the form

$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A & -B_u R_{uu}^{-1} B_u^T \\ -C_z^T R_{zz} C_z & -A^T \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} \tag{4.14}$$

There is a striking resemblance between 4.14 and 4.11. If we substitute $R_{zz} = I$, $B_u = \begin{bmatrix} B_1 & B_2 \end{bmatrix}$, $C_z = C_1$, and

$$R_{uu}^{-1} = \begin{bmatrix} y^{-2} & 0 \\ 0 & -1 \end{bmatrix}$$

we can equate the two matrices entirely. Thus, it is evident that the Hamiltonian H_∞ plays a role in regulation component of our controller system K. Linear Quadratic Estimation (LQE) has dualities with an LQR problem How (2007), the relevant ones for our discussion of H_∞ control being

$$A \leftrightarrow A^T$$
$$B \leftrightarrow C_y^T$$
$$C_z \leftrightarrow B_w^T$$
$$R_{zz} \leftrightarrow R_{ww}$$
$$R_{uu} \leftrightarrow R_{vv}$$

This duality bears resemblance to that seen between H_∞ and J_∞, and so we can conclude that J_∞ is used to provide state estimation capabilities to the gain system K.

4.3.2 H_∞ Controller Framework

The weighted plant for our H_∞ implementation is defined by the following derivation. We utilize three of the weighting functions discussed in Chapter 3, these being: W_u to measure performance of our controller signal and penalize control signals in frequency bands we are uninterested in; W_e to provide tracking feedback to our cost function, and W_n, to allow the inclusion of sensor noise information into our controller formulation.

The reduced form of the plants derived in Chapter 2 can be written as

$$\begin{aligned} \dot{x}_r &= A_r x_r + B_r \begin{bmatrix} u \\ F_d \end{bmatrix} \\ y &= C_r x_r \end{aligned} \tag{4.15}$$

We define our sensor noise with the dynamical system

$$\begin{aligned} \dot{x}_n &= A_n x_n + B_n w_{noise} \\ y_{noise} &= C_n x_n + D_n \omega_{noise} \end{aligned} \tag{4.16}$$

We feed our error signal to the controller

$$y_s = r - (C_r x_r + C_n x_n + D_n \omega_{noise}) \tag{4.17}$$

Where r is our tracking signal. We regulate our tracking error

$$\begin{aligned} \dot{x}_e &= A_e x_e + B_e C_r x_r \\ z_e &= r - C_e x_r + D_e C_r x_r \end{aligned} \tag{4.18}$$

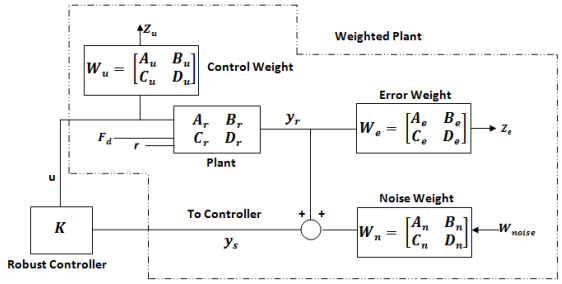

Figure 4.3: Weighted Plant and Controller

We include states in our plant to provide the controller penalty cost

$$\begin{aligned} \dot{x}_u &= A_u x_u + B_u u \\ z_u &= C_u x_u + D_u u \end{aligned} \tag{4.19}$$

Let us now define a new combined state vector ζ as

$$\zeta = \begin{bmatrix} {x_r}^T & {x_e}^T & {x_n}^T & {x_u}^T \end{bmatrix}^T \tag{4.20}$$

and our exogenous input vector

$$\omega = \begin{bmatrix} r^T & {F_d}^T & {\omega_{noise}}^T \end{bmatrix}^T \tag{4.21}$$

This results in the weighted plant seen in Figure 4.3

The state space representation of the new combined system is

$$\dot{\zeta} = \begin{bmatrix} A_r & 0 & 0 & 0 \\ -B_e C_r & A_e & 0 & 0 \\ 0 & 0 & A_n & 0 \\ 0 & 0 & 0 & A_u \end{bmatrix} \zeta + \begin{bmatrix} 0 & B_r^{F_d} & 0 & B_r^u \\ B_e & 0 & 0 & 0 \\ 0 & 0 & B_n & 0 \\ 0 & 0 & 0 & B_u \end{bmatrix} \begin{bmatrix} r \\ F_d \\ w_{noise} \\ u \end{bmatrix} \tag{4.22}$$

Where we define

$$B_r = \begin{bmatrix} B_r^u & B_r^{F_d} \end{bmatrix}$$

The output becomes

$$\begin{bmatrix} z_e \\ z_u \\ y_s \end{bmatrix} = \begin{bmatrix} -D_e C_r & C_e & 0 & 0 \\ 0 & 0 & 0 & C_u \\ -C_r & 0 & -C_n & 0 \end{bmatrix} \zeta + \begin{bmatrix} D_e & 0 & 0 & 0 \\ 0 & 0 & 0 & D_u \\ I & 0 & -D_n & 0 \end{bmatrix} \begin{bmatrix} r \\ F_d \\ w_{noise} \\ u \end{bmatrix} \tag{4.23}$$

We must now determine if our weighted plant meets the requirements of the DGKF approach. First, let us examine our D matrix to determine whether our problem enables us to use the simplified solution approach, or if we must use the general synthesis approach.

Our D block is split into four components, D_{11}, D_{12}, D_{21} and D_{22}. We define

$$D_{11} = \begin{bmatrix} D_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$D_{12} = \begin{bmatrix} 0 \\ D_u \end{bmatrix}$$

$$D_{21} = \begin{bmatrix} I & 0 & -D_n \end{bmatrix}$$

and

$$D_{22} = \begin{bmatrix} 0 \end{bmatrix}$$

Since D_{11} is nonzero, this means our problem falls into the general H_∞ approach outlined by Zhou et al Zhou (1998). All that remains before we can move to producing controllers is to show that (A,B_1) is stabilizable, and (C_2,A) is detectable.

We define B_1 as

$$B_1 = \begin{bmatrix} 0 & B_r^{F_d} & 0 \\ B_e & 0 & 0 \\ 0 & 0 & B_n \\ 0 & 0 & 0 \end{bmatrix}$$

To prove stabilizability, we must show that rank $\begin{bmatrix} \lambda I - A & B_1 \end{bmatrix} = n$ for all the postive eigenvalues of A. If we calculate the poles of our matrix A, we find that all eigenvalues of the plant are already 0, and thus the system is indeed stabilizable. Likewise, detectability of the pair (C_2,A) requires that all unstable poles of A are observable, however, since the plant is already stable, we can conclude that the pair is indeed detectable.

4.4 Extension to μ Synthesis

Modelling deformable objects, especially in biomedical applications, can involve a fair deal of uncertainty in the parameters selected. This uncertainty can potentially lead to unstable closed loops, which poses a serious problem, particularly for medical procedures, where the safety of a patient is of paramount importance. A direct successor of the H_∞ controller design methods discussed in this Chapter is μ synthesis. The μ synthesis control approach optimizes the controller based on the structured singular value μ of the closed loop system. The structured singular value of a system is a measure of the smallest perturbation that can be applied to that closed loop before it destabilizes Moser (1993). Given a system matrix M,

$$M = C^{-1}(sI - A)B + D \tag{4.24}$$

the structured singular value of a system with a particular uncertainty Δ is defined as Moser (1993)

$$\mu(M) = \frac{1}{\min\{\bar{\sigma}(\Delta) : \det(I + M\Delta) = 0\}} \tag{4.25}$$

It can be shown that $\mu(M)$ is bounded by Zhou (1998)

$$\max_U \rho(UM) \leq \mu(M) \leq \inf_D \bar{\sigma}(DMD^{-1}) \tag{4.26}$$

It can be shown that μ closely follows its upper bound value. The matrix D is selected to be stable, minimum phase, and have the property $D\Delta = \Delta D$. The upper bound of our structured singular value μ now looks like a scaled version of the closed loop ∞-norms that have been discussed in this Chapter. This provides the basis of

the most common μ synthesis approach, known as D-K synthesis.

D-K synthesis iteratively alternates between optimizing D terms, and optimizing a controller gain K. The iteration continues until the solution of D converges. At this point in the synthesis, all that remains is to solve one final H_∞ problem, and the optimal controller is obtained. A detailed description of the process can be found in Zhou (1998) and Moser (1993).

For our particular application, all three types of uncertainty described in Chapter 3 were valid to implement. We opted to select uncertainty in our spring and damping parameters to model the variability inherent in working with tissue. Stiffness and damping were selected since they are more difficult to estimate in an accurate manner than mass without experimentation. This form of uncertainty produced a plant of the form

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -M^{-1}K + \Delta & -M^{-1}\Xi + \Delta \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} u \tag{4.27}$$

This uncertainty was rearranged into the form seen in Figure 3.4. We use a state estimator initialized with the original states of the system to feed the state into the uncertainty, producing the closed loop system seen in Figure 4.4

The resultant block containing Δ was parameterized as the uncertain system $\tilde{\Delta}$, as seen in Figure 4.5. This transform enables us to drastically reduce the number of inputs and outputs from our uncertainty block, reducing the overall computational complexity of any synthesis technique substantially. For example, in this thesis, plants were contained upwards of 300 states associated with any particular parameter, but

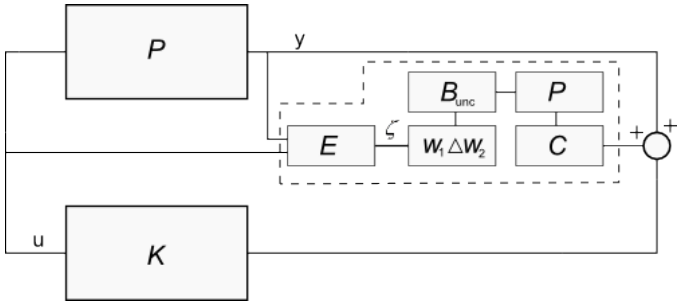

Figure 4.4: Uncertainty in μ Synthesis Approach

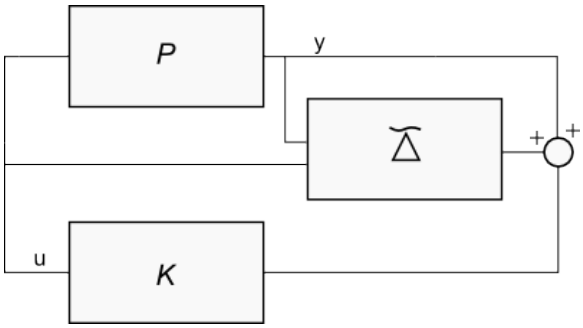

Figure 4.5: Uncertainty in μ Synthesis Approach After Parameterization

only had 4 outputs and 4 control inputs. Thus, using this transform, the computational complexity of the uncertainty is reduced by close to two orders of magnitude. Performance of μ synthesis controllers synthesized with this approach is demonstrated in Chapter 6.

Chapter 5

Optimal PID Control

The popularity of PID control in industry, see Yamamoto and Hashimoto (1991), coupled with its use in papers focused on DOs, see Wada *et al.* (2001) and Koustoumpardis and Aspragathos (2007) provided motivation to compare its performance to that of H_∞ controllers. In order to achieve a fair comparison, this Chapter describes the process in which optimal PID gains are calculated such that the bound of $\|T_{zw}\|_\infty$ is minimized.

5.1 Definition of the Closed Loop System

Recall from our discussion on the formulation of the H_∞ controller that our weighted plant is composed of the following subsystems:

1. A reduced model of the deformable object, described by the system

$$\begin{aligned} \dot{x}_r &= A_r x_r + B_r \begin{bmatrix} u \\ F_d \end{bmatrix} \\ y &= C_r x_r \end{aligned} \tag{5.1}$$

2. An output error weight, described by

$$\begin{aligned} \dot{x}_e &= A_e x_e + B_e(r - C_r x_r - C_n x_n - D_n w_{noise}) \\ z_e &= C_e x_r + D_e(r - C_r x_r - C_n x_n - D_n w_{noise}) \end{aligned} \tag{5.2}$$

3. A sensor noise shaping system, described by

$$\begin{aligned} \dot{x}_n &= A_n x_n + B_n w_{noise} \\ y_s &= C_r x_r + C_n x_n + D_n w_{noise} \end{aligned} \tag{5.3}$$

4. A control penalty weight function, described by

$$\begin{aligned} \dot{x}_u &= A_u x_u + B_u u \\ z_u &= C_u x_u + D_u u \end{aligned} \tag{5.4}$$

We define our error signal e as

$$e = r - C_r x_r - C_n x_n - D_n w_{noise} \tag{5.5}$$

and its derivative $\dot{e}$ as

$$\dot{e} = \dot{r} - C_r(A_r x_r + B_r^u u + B_r^{F_d} F_d) - C_n(A_n x_n + B_n w_{noise}) - D_n \dot{w}_{noise} \tag{5.6}$$

The control signal is

$$u = K_p e + K_d \dot{e} + K_i \int e \tag{5.7}$$

To handle the integral term in our state space system, we parameterize

$$\int e = q \tag{5.8}$$

Then the state equation becomes

$$\dot{q} = r - Crx_r - C_n x_n - D_n w_{noise} \tag{5.9}$$

The output vector z (similarly with H_∞ design) is

$$z = \begin{bmatrix} {z_e}^T & {z_u}^T \end{bmatrix}^T \tag{5.10}$$

We assume our noise is an unknown bandlimited function, and we parameterize a new noise variable

$$\tilde{w}_{noise} = \dot{w}_{noise} + w_{noise} \tag{5.11}$$

The exogenous input is defined as

$$w = \begin{bmatrix} r^T & \dot{r}^T & F_d^T & \tilde{w}_{noise}^T \end{bmatrix}^T \tag{5.12}$$

Let us now define a new combined state vector ζ as

$$\zeta = \begin{bmatrix} {x_r}^T & {x_e}^T & {x_n}^T & {x_u}^T & q^T \end{bmatrix}^T \tag{5.13}$$

We now derive the closed loop form our our weighted system,

For clarity's sake, we parameterize the following equation by the symbol Υ

$$\Upsilon = (I + K_d C_r B_r^u)^{-1} \tag{5.14}$$

The state space representation of the new combined system - G_{CL} - is

$$\dot{\zeta} = \begin{bmatrix} A_r - B_r^u\Upsilon(K_pC_r + K_dC_rA_r) & 0 & -B_r^u\Upsilon(K_pC_n + K_dC_nA_n) & 0 & B_r^u\Upsilon K_i \\ -B_eC_r & A_e & -B_eC_n & 0 & 0 \\ 0 & 0 & A_n & 0 & 0 \\ -B_u\Upsilon(K_pC_r + K_dC_rA_r) & 0 & -B_u\Upsilon(K_pC_n + K_dC_nA_n) & A_u & B_u\Upsilon K_i \\ -C_r & 0 & -A_n & 0 & 0 \end{bmatrix} \zeta$$

$$+ \begin{bmatrix} B_r^u\Upsilon K_p & B_r^u\Upsilon K_d & (-B_r^u\Upsilon K_dC_r + I)B_r^{F_d} & -B_r^u\Upsilon K_pD_n \\ B_e & 0 & 0 & -B_eD_n \\ 0 & 0 & 0 & B_n \\ B_u\Upsilon K_p & B_u\Upsilon K_d & (-B_u\Upsilon K_dC_r)B_r^{F_d} & -B_u\Upsilon K_pD_n \\ I & 0 & 0 & -D_n \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \\ F_d \\ \tilde{w}_{noise} \end{bmatrix}$$

Where we define

$$B_r = \begin{bmatrix} B_u^r & B_{F_d}^r \end{bmatrix}$$

The output becomes

$$\begin{bmatrix} z_e \\ z_u \end{bmatrix} = \begin{bmatrix} -D_e C_r & C_e & -D_e C_n & 0 & 0 \\ -D_u \Upsilon (K_p C_r + K_d C_r A_r) & 0 & -D_u \Upsilon (K_p C_n + K_d C_n A_n) & C_u & D_u \Upsilon K_i \end{bmatrix} \zeta$$

$$+ \begin{bmatrix} D_e & 0 & 0 & D_e D_n \\ D_u \Upsilon K_p & D_u \Upsilon K_d & -D_u \Upsilon K_d C_r B_{F_d}^r & -D_u \Upsilon K_p D_n \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \\ F_d \\ \tilde{w}_{noise} \end{bmatrix}$$

Then, the state space form of our transfer function T_{zw} is

$$\begin{aligned} \dot{x_{CL}} &= A_{CL}(\kappa) x_{CL} + B_{CL}(\kappa) w \\ z &= C_{CL}(\kappa) x_{CL} + D_{CL}(\kappa) w \end{aligned} \tag{5.15}$$

With the state matrices $A_{CL} \ldots D_{CL}$ being functions of the controller gains $K_{p_i}, K_{d_i}, K_{i_i}$, which are collectively grouped as the optimization variable κ.

5.2 Definition of the Optimization Problem

As in the H_∞ controller design case, we want to generate a controller that minimizes the ∞ system norm of our closed loop transfer matrix. The ∞ norm is defined as Zhou (1998)

$$\|G\|_\infty := \sup_w \overline{\sigma}\{G(jw)\} \tag{5.16}$$

This norm is difficult to calculate directly, but can be approximated by calculating

an upper bound γ. Thus, we can now write our optimization in the general form

$$\min_{\kappa} \gamma \tag{5.17}$$

A discussion of methods in which to calculate the ∞ norm was presented in Chapter 4, and thus 5.17 is easily implementable.

5.2.1 Stability Constraint

Minimizing γ will produce PID controllers that take into consideration performance and robustness conditions, however we still need to guarantee stability of the approach before we can use the parameters in actual control tasks. To allow simple computation of the stability constraint, we use pole based stability, which is based on calculating the eigenvalues of $A_{CL}(\kappa)$, the A matrix of our closed loop system. However, we can simplify this calculation, since we do not have to take into consideration our weights W_u and W_e. Thus, our reduced $\bar{A}_{CL}(\kappa)$ can be written

$$\bar{A}_{CL}(\kappa) = \begin{bmatrix} A_r - B_r^u \Upsilon (K_p C_r + K_d C_r A_r) & -B_r^u \Upsilon (K_p C_n + K_d C_n A_n) & B_r^u \Upsilon K_i \\ 0 & A_n & 0 \\ C_r & A_n & 0 \end{bmatrix} \tag{5.18}$$

The stability constraint can now be formulated as

$$real(eig(\bar{A}_{CL}(\kappa)) + \epsilon \leq 0 \tag{5.19}$$

Where ϵ is a positive real number used to push poles a specified distance from

the jw axis. Note that stability can also be included in the objective function of the optimization as a barrier function, inducing large scale penalties the closer the poles are to the positive real axis. The stability constraint generates a barrier function of the form Boyd (2004)

$$\Phi(\kappa) = -\log\left(real(eig(\bar{A}_{CL}(\kappa))) - \epsilon\right) \tag{5.20}$$

Additional Considerations

Since we are primarily focused on MIMO systems, our PID controller contains more than the three gains found in a SISO formulation. The gain values K_p, K_d and K_i are no longer simple scalars, they are now full gain matrices, with at least N to as many as N^2 individual parameters each, where N is the number of control points multiplied by the dimensionality of the target problem. The gain matrices can take a variety of forms, ranging from simple block diagonals where all gains along the diagonal are equal, to more complex gain matrices containing off diagonal components with varying gains. To simplify matters, designers should try to utilize a simple 3 parameter block diagonal designs initially, and then move to more complex solutions configurations if this does not provide a stabilizable closed loop.

If a simple block diagonal approach fails to provide a stable solution, it usually suffices to couple the dimensional component gains of each control signal. Thus, gains should be coupled for all x-dimensional error, y-dimensional error, and potentially z-dimensional error. For a 2D problem, the resultant coupled gain matrix would take the form

$$\widetilde{K_x} = \begin{bmatrix} K_{x_1} & 0 & K_{x_2} & 0 \\ 0 & K_{x_1} & 0 & K_{x_2} \\ K_{x_2} & 0 & K_{x_1} & 0 \\ 0 & K_{x_2} & 0 & K_{x_1} \end{bmatrix} \tag{5.21}$$

5.2.2 Optimization Setup

An important first step in our optimization is to to define the bounds of the feasible set of controller gains. For the controllers we are interested in, the defining attribute of the feasible set is that gain parameters produce a stable closed loop. Published results exist that provide methodologies to determine the bounded set of parameters that generate a stable closed loop, see Tan *et al.* (2006), Soylemez *et al.* (2003), however, these focus on SISO applications. To minimize computational complexity of this task for our problem, we do a search with coarse gain parameters. To give the designer an easy way to visualize the feasible set, we recommend defining stability bounds by iterating through three parameters at a time, as seen in Figure 5.1.

Figure 5.1 is calculated by taking the poles of the matrix 5.18, for a range of gain parameters of K_p and K_i, with K_d being fixed. For this particular plant, K_p and K_d were defined as block diagonal gains, and the K_i gain matrix was defined as a two parameter matrix of the form 5.21. To include considerations of K_d on the stability of the closed loop and its effect on the feasible set, multiple stability plots were generated for different fixed values of K_d, and a final feasible region was defined by intersecting these disparate plots. This intersecting of plots is shown in Figure 5.2.

The nonconvex behaviour of the stability bounds leaves the designer with some

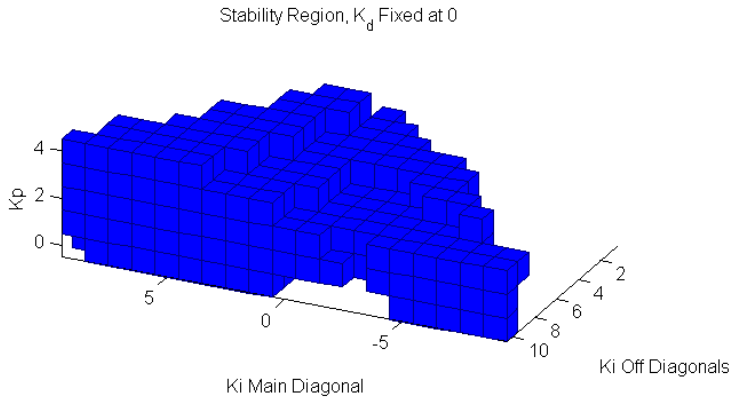

Figure 5.1: Sample Coarse Stability Plot for Feasible Set Definition

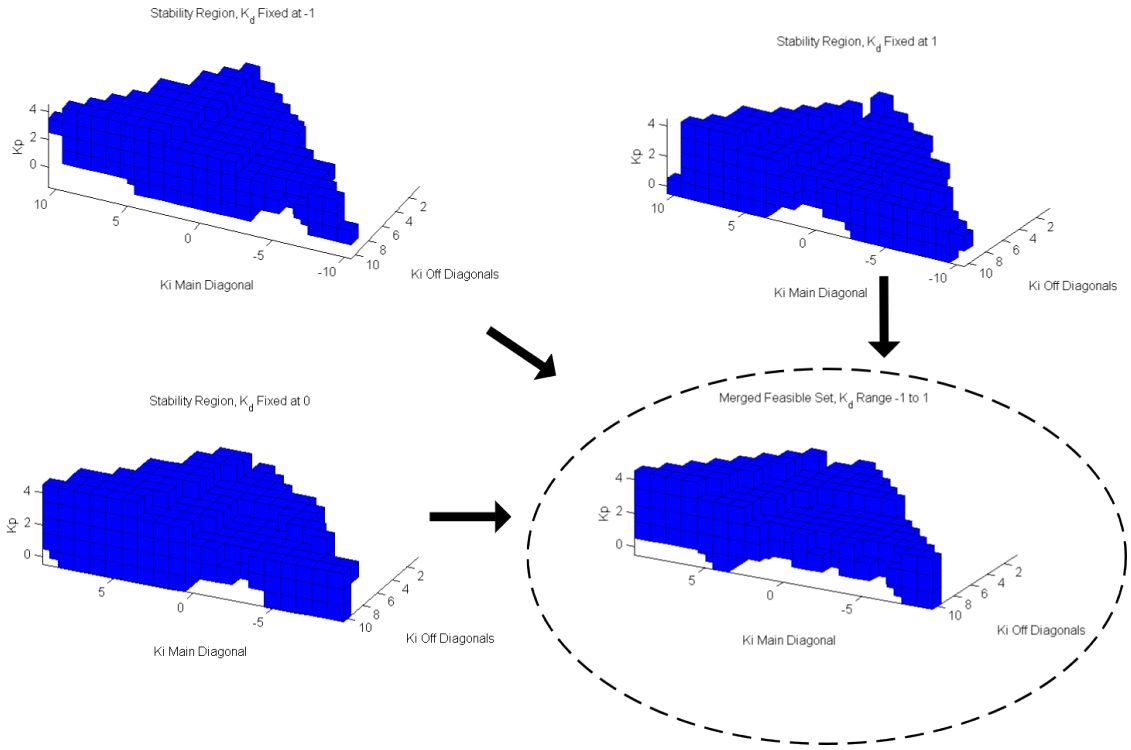

Figure 5.2: Visualizing the Feasible Set for More than Three Gains

difficult choices. The bounds cannot be specified easily, since each individual gain may be valid only in combination with other gain parameters. One method in handling these bounds is to define nonlinear bounding functions, and build these functions based on the stable set obtained during the coarse looping process. The disadvantage of this method lies in the added layer of complexity introduced by generating proper bounding functions. Another way of handling this behaviour is to use barrier functions discussed earlier in the Chapter. This method does not necessarily invalidate the usefulness of generating the a plot of the feasible set in the first place, since the designer can now select random points within the set to act as an initial point for optimization routines. A final approach is to try and break apart the nonconvex region into a large number of convex subregions. Figure 5.3 shows how this might work for a two parameter nonconvex feasible set. This process for 3D shapes is discussed by Bischoff and Kobbelt (2002), who propose a method to convert the volumes of polygons into sets of overlapping ellipsoids. If the designer does not care about overlap of solvable regions, one way to go about decomposing the problem is to identify the boundaries of the feasible set, and then randomly sample "seed" points from points in this set. The seed points can then be used to grow ellipsoids, or spheroids, until they hit another boundary of the set. In this way, the interior of the feasible set will be filled with convex regions.

The nonconvex nature of our PID's feasible set agrees with the findings of Astrom *et al.* (1998),who showed that robustness measures for a PID controller are related in a nonconvex manner to the values of the three PID gains. To investigate whether this behaviour extends to a MIMO H_∞ case, we calculate the ∞-norm of a closed loop in the coarse manner that we generated the feasible set seen in Figure 5.1. Using

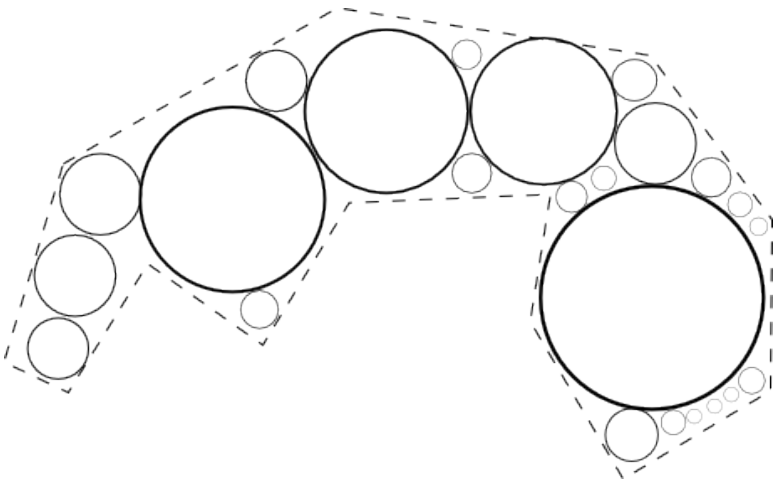

Figure 5.3: Subdividing a Nonconvex Feasible Set into Convex Subsets

bisection search to obtain the ∞-norm and looping through multiple controllers, we generated the map of ∞-norms seen in Figure 5.4

This figure is difficult to interpret, with the vast bulk of the feasible set appearing to be the same colour, and thus have the same apparent ∞-norm. To try and better understand the relation between our controller gains and our performance measure, let us smooth out our "outlier" data, to observe trends within the blue region of our plot. Capping our maximal ∞-norm at 20 results in Figure 5.5

We can now start seeing trends in our norm, with the norm decreasing as K_p decreases. It is evident that in regions along the bounds of stability, the ∞-norm becomes very large. Focusing on a particular fixed K_p surface provides even better resolution, as can be seen in Figure 5.6

This figure shows two minimal norms, separated by a ridge, conclusively proving that our problem is nonconvex. This attribute means that designers will need to select a number of starting points for their optimization, and select the best solution amongst the different solutions as their final controller design.

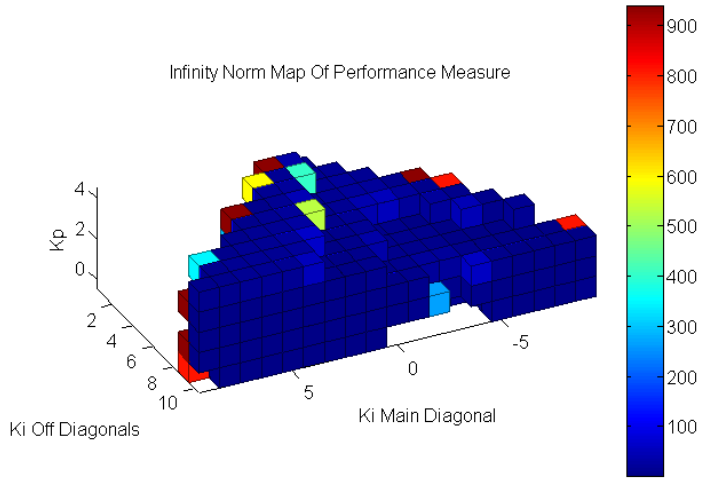

Figure 5.4: Coarse Plot of H_∞ Norms of Closed Loop

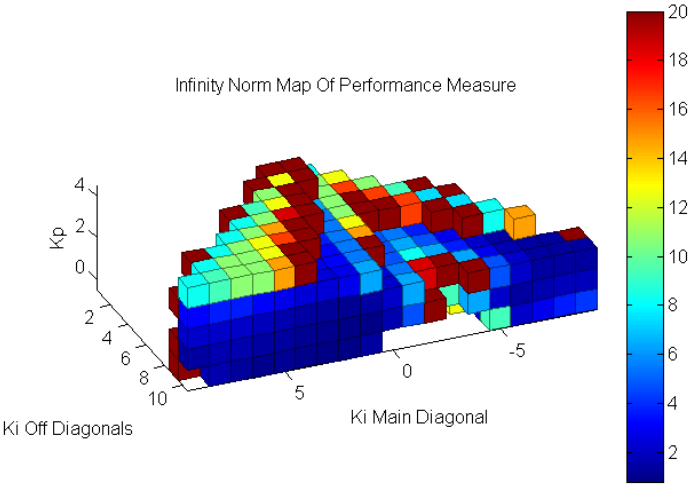

Figure 5.5: Coarse Plot of H_∞ Norms of Closed Loop With Saturation of 20

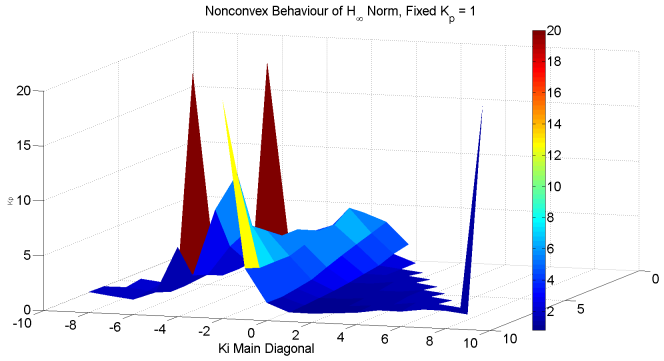

Figure 5.6: Surface Plot of H_∞ Norms of Closed Loop With Saturation of 20, K_p Fixed at 1

5.2.3 Optimization Method

Our full optimization problem can be written in the form

$$\begin{aligned} \underset{\kappa}{\text{minimize}} \quad & \gamma + \Phi(\kappa) \\ \text{subject to} \quad & real(eig(\bar{A}_{CL}(\kappa)) + \epsilon \leq 0 \end{aligned}$$

Where κ is a vector the our controller gains, and $\Phi(\kappa)$ is our stability barrier function. The nonconvex nature of this problem makes it difficult to find a globally optimal solution. We selected an approach known as Sequential Quadratic Programming (SQP), due to the ease in which it could be implemented in MATLAB$^{\text{TM}}$'s $fmincon$ function.

Readers interested in alternate approaches that were investigated may refer to the appendix.

Sequential Quadratic Programming

Sequential Quadratic Programming, or SQP for short, is a popular optimization approach for dealing with nonlinear programs with nonlinear constraints. SQP works

by breaking up a nonlinear problem into a sequence of quadratic subproblems, with each iteration attempting to more accurately approximate the actual behaviour of the objective than the preceding step, Boggs and Tolle (1995). A key advantage of SQP is that it does not require feasible points while it searches for the optimal solution. This is advantageous when nonlinear constraints become overly complex. According to Boggs and Tolle (1995),the internal quadratic subproblems solved in each step will only determine a local minimal solution. To determine a global optimal solution, a merit function ϕ is defined. This function must reduce its value for each stage of the optimization that brings the solution closer to the global minima. In order to achieve this, ϕ is usually based on an unconstrained version of the problem's objective function Boggs and Tolle (1995). A properly selected ϕ will give the designer a reasonably good chance of finding a global solution to their problem. SQP, like Pattern Search which is discussed in the Appendix, is implemented in MATLAB, and thus is a strong candidate to generate an optimal PID controller.

Chapter 6

Simulations and Experiments

This chapter covers the simulation and experimentation aspects of the work done for this thesis.

6.1 Simulations

All controllers investigated in this thesis were evaluated through simulations in MATLAB^{TM}'s Simulink environment before experimental testing. Figure 6.1 shows the model file used to evaluate the stability, performance, and robustness characteristics of the different controllers. To simulate sensor noise, a biased white noise is added to the output of the plant before being fed to controllers. The bias was based on values observed during experimentation. The plant itself has a disturbance force applied to the control nodes of the plant model to simulate unmodelled dynamics. The disturbance was simulated as a bandlimited white noise signal, with low sampling rate (10 Hz). All the simulation results presented in the following subsections are fed the same noise signal and disturbances in order to fairly compare their performance.

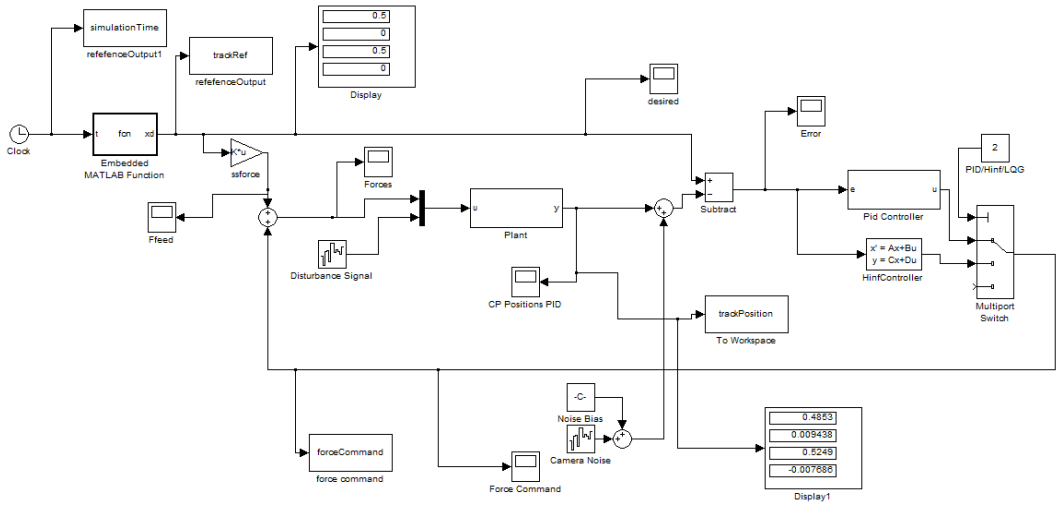

Figure 6.1: MATLAB^{TM} Simulink File Used To Simulate Controllers

6.1.1 H_∞ Simulations

Over the course of the research carried out for this thesis, a variety of controllers were generated and simulated, with the main variables between simulations being the scaling term λ on the control cost weight W_u, and the inclusion of various values of feed forward force. Figures 6.2 through 6.6 display the effect that λ plays on the tracking performance of the controller. Dashed lines represent the desired set point, and solid lines represent the actual positions of control points. The simulated task involved a step stretch command for both control points in the x-plane, followed by y-dimension commands pulling apart the two control points. One key limiting factor on the scaling seen in these plots was the timestep required to solve the controllers for smaller values of λ. Low values of λ reduce the smoothness of the closed loop system, and thus reduce the timestep necessary to solve controller in a stable manner. This may prevent controllers with small λ values from being used in a practical setting,

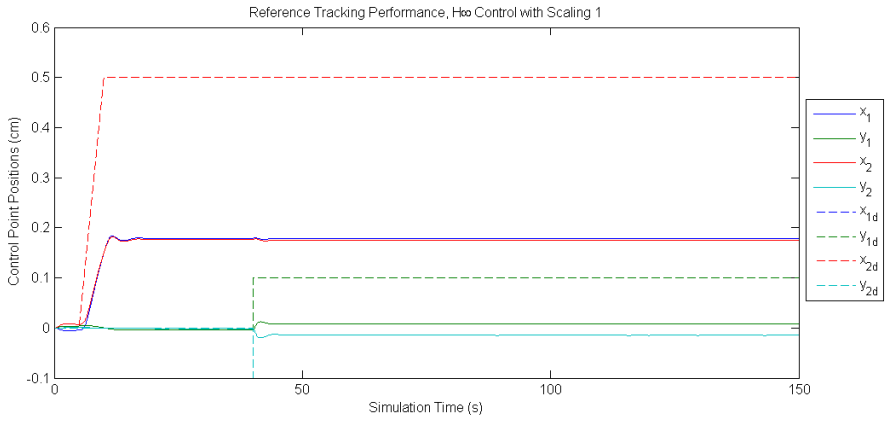

Figure 6.2: H_∞ Set Point Tracking Performance, $\lambda = 1$ Scaling of W_u

depending on the lower bound of timesteps available to the application platform. This limitation is unfortunate, since low stiffness plants often require small λ values to have decent performance, as can be seen in the difference between Figure 6.2 and that of Figure 6.6. The reason for the impact of λ is due to the relationship between the two subcomponents of the output vector z, z_e and z_u. Without any gain scaling, the ∞-norm of the transfer matrices for these two output vectors are almost two orders of magnitude apart. Scaling W_u also reduces the γ value of the final solution, providing added incentive to tune this parameter during the weighted plant's design, however, modifying W_u with low values of λ has a robustness tradeoff. As λ decreases, the susceptibility of the controller to disturbances and sources of noise increases, eventually leading to unstable systems in an experimental setting. It has been empirically determined that λ scaling up to one order of magnitude provides reasonable increases in performance, without unacceptable loss of robustness.

When a feed forward force F_{ff} is added to the system, the performance is drastically improved. A controller with a scaling factor of $\lambda = 0.1$, seen without F_{ff} in Figure 6.4, has superior performance in Figure 6.7 when combined with F_{ff}.

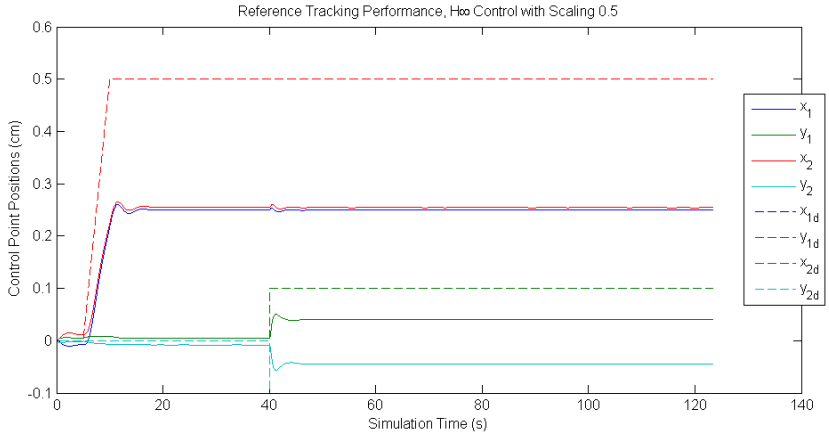

Figure 6.3: H_∞ Set Point Tracking Performance, $\lambda = 0.5$ Scaling of W_u

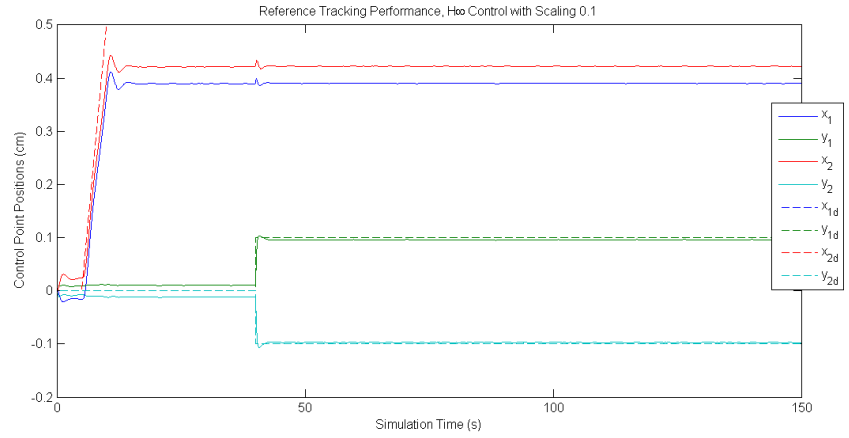

Figure 6.4: H_∞ Set Point Tracking Performance, $\lambda = 0.1$ Scaling of W_u

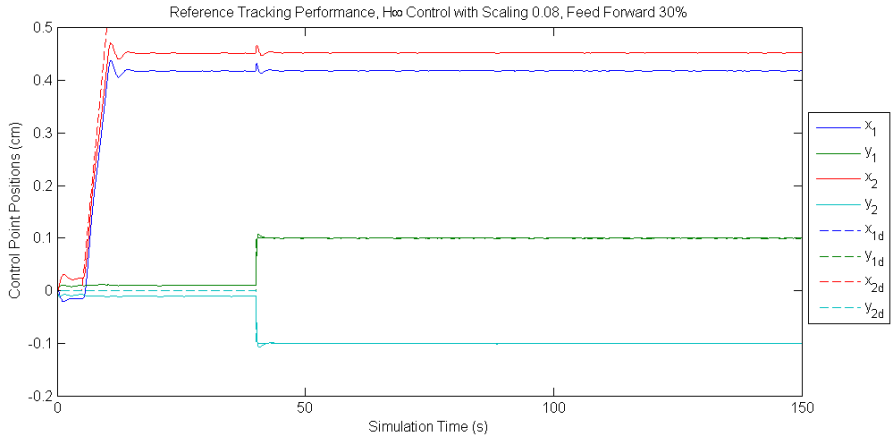

Figure 6.5: H_∞ Set Point Tracking Performance, $\lambda = 0.08$ Scaling of W_u, $F_{ff} = 30\%$

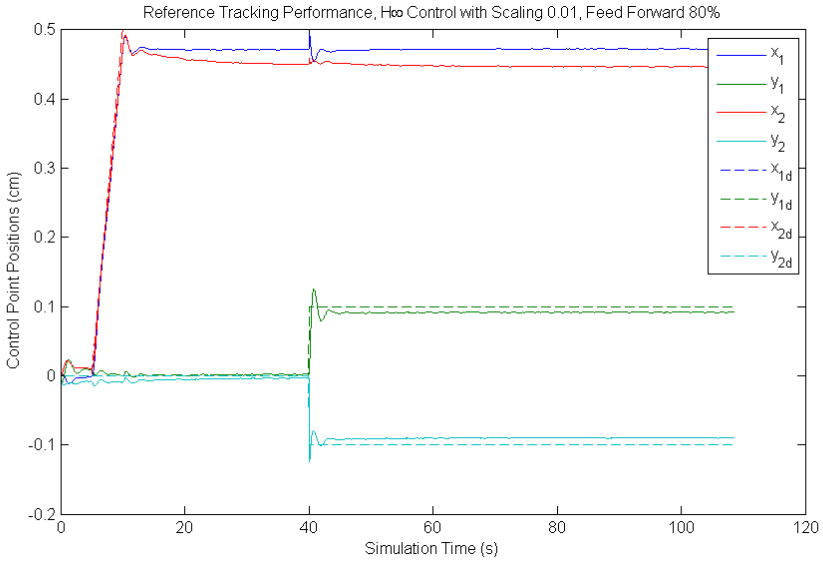

Figure 6.6: H_∞ Set Point Tracking Performance, $\lambda = 0.01$ Scaling of W_u, $F_{ff} = 80\%$

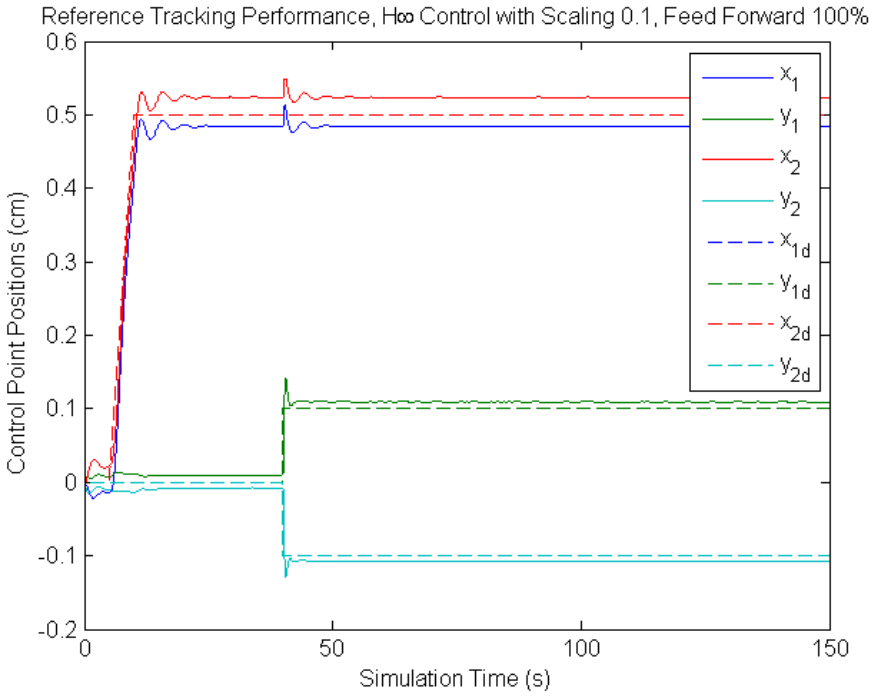

Figure 6.7: H_∞ Set Point Tracking Performance, $\lambda = 0.1$ Scaling of W_u, $F_{ff} = 100\%$

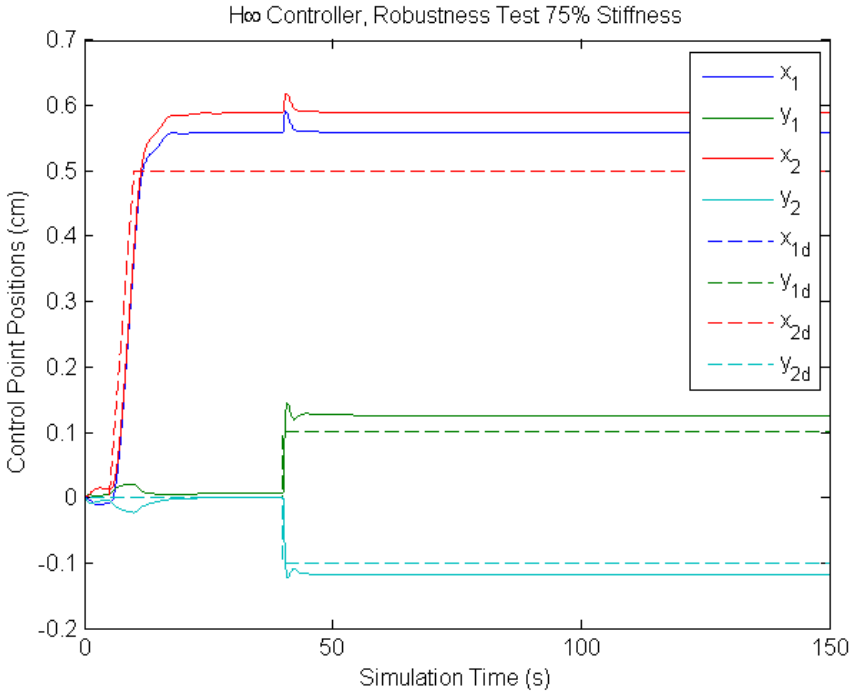

Figure 6.8: H_∞ Robust Performance, $\lambda = 1$ Scaling of W_u, 75% Stiffness

Robustness simulations were also performed on the H_∞ controllers, with simulations being performed for plants with 90% and 75% of the modelled stiffness. All robustness tests included a F_{ff} term which was calculated for the original plant. Controllers with scalings of both $\lambda = 1$ and $\lambda = 0.1$ were simulated to test the effect of scaling on the stability of the controllers. Figures 6.8 and 6.9 show robustness of unscaled W_u based controllers, and Figures 6.10 and 6.11 show performance of controllers when scaling W_u with a λ of 0.1.

The overshoot seen in Figure 6.8 is largely due to the Feed Forward Force. Due to W_u being unscaled, the designed controller's maximum applicable force is also low, resulting in minimal compensation of F_{ff}. In contrast, inclusion of minimal amounts of scaling such as in Figure 6.10, result in robust tracking performance that is far

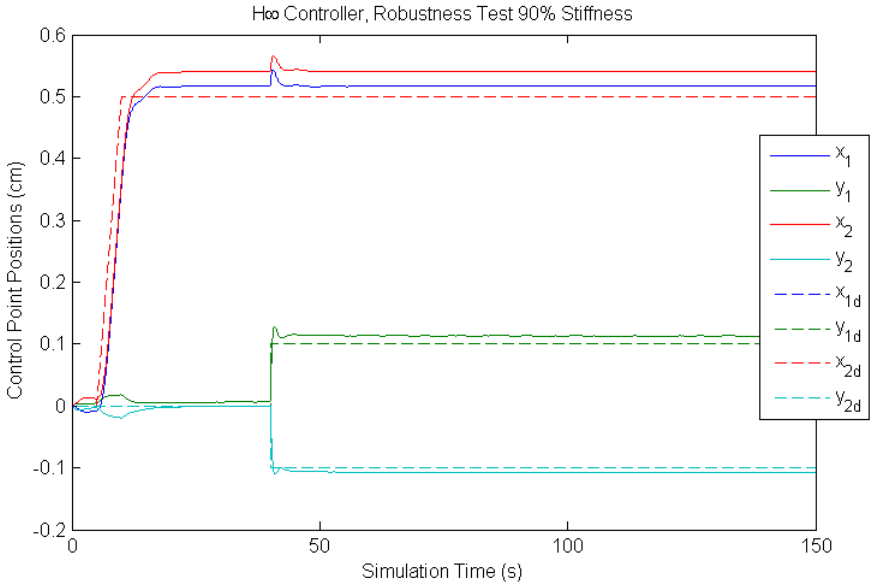

Figure 6.9: H_∞ Robust Performance, $\lambda = 1$ Scaling of W_u, 90% Stiffness

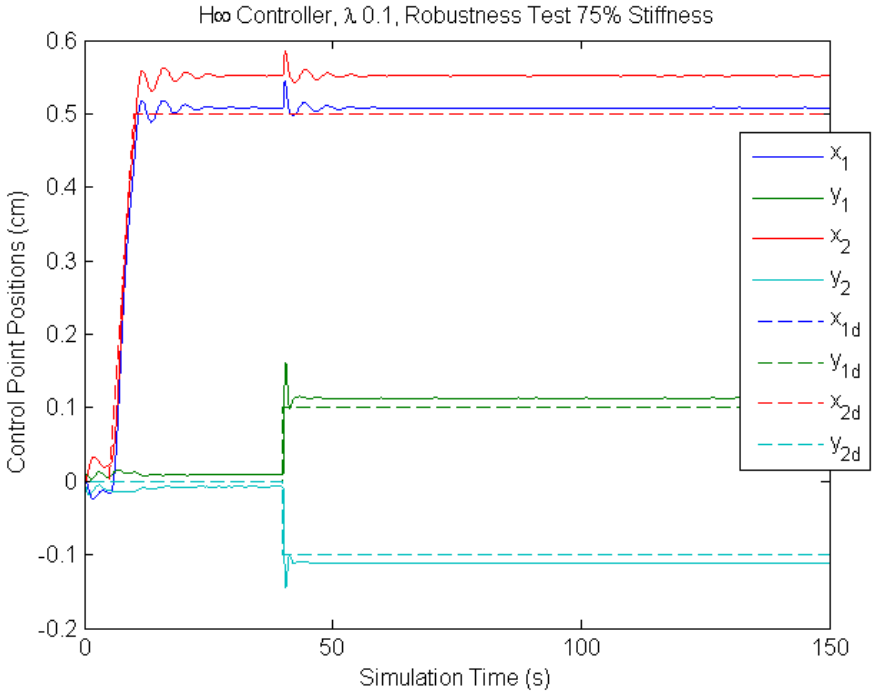

Figure 6.10: H_∞ Robust Performance, $\lambda = 0.1$ Scaling of W_u, 75% Stiffness

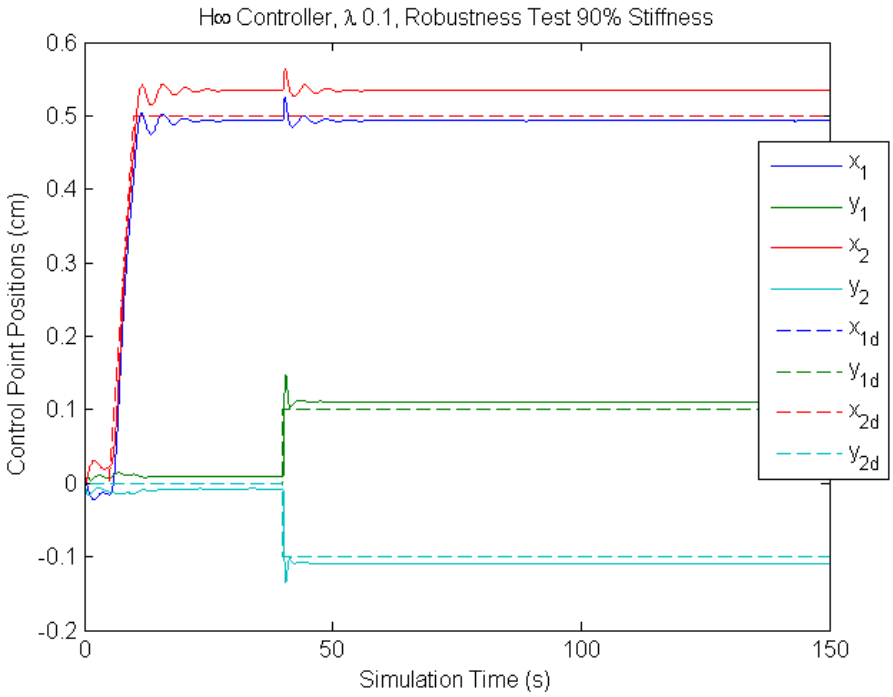

Figure 6.11: H_∞ Robust Performance, $\lambda = 0.1$ Scaling of W_u, 90% Stiffness

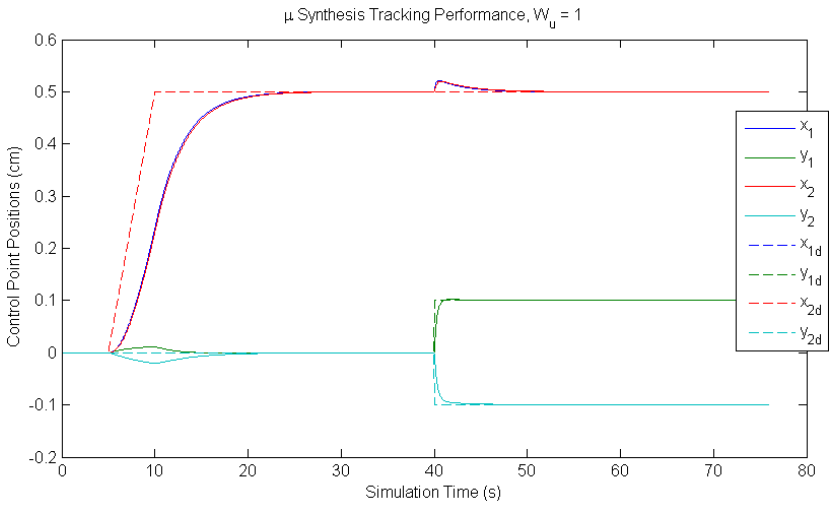

Figure 6.12: μ Synthesis Set Point Tracking, $\lambda = 0.1$ Scaling of W_u

closer to the ideal tracking case.

6.1.2 μ Synthesis Simulations

The weighted plant used to generate the H_∞ controllers was also used to build μ synthesis based controllers. To save time, μ synthesis was only performed on weighted plants that generated well performing H_∞ controllers. The controllers were generated using D-K synthesis through MATLAB^{TM}'s robust control toolbox. Uncertainty in the plant was defined as an observer type uncertainty discussed in Chapter 4. A μ synthesis controller with a μ of 1.6 was obtained with this approach. Figure 6.12 shows the set point tracking for a μ synthesis controller based on the $\lambda = 0.1$ scaled weighted plant used to create the H_∞ controller seen in Figures 6.4, 6.7, 6.10, and 6.11,

Note that the tracking performance for the μ based controller is superior to the H_∞ controller counterpart. Robustness simulations were performed, resulting in Figures

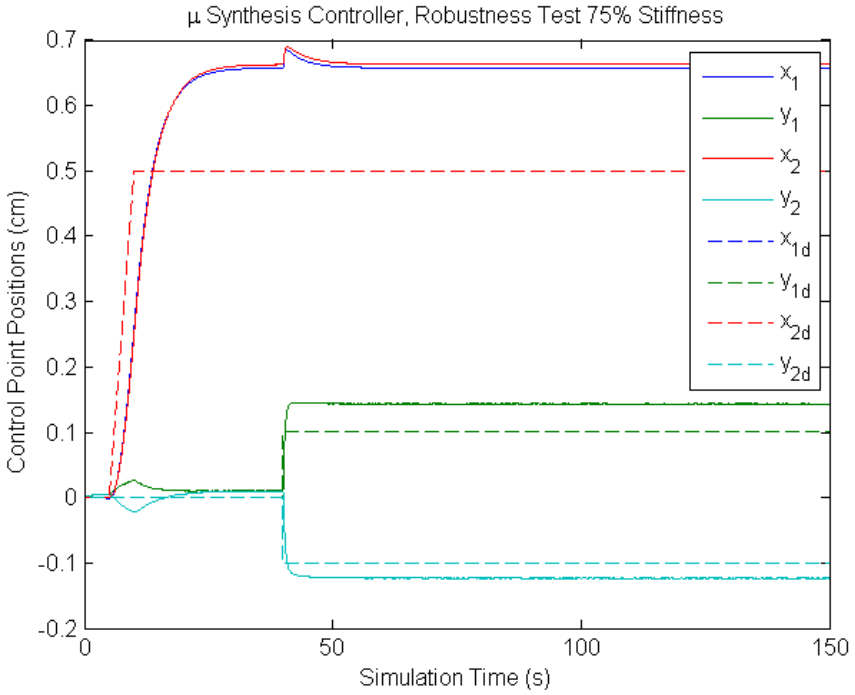

Figure 6.13: μ Synthesis Robust Performance, $\lambda = 0.1$ Scaling of W_u, 75% Stiffness

6.13 and 6.14. It is interesting to note that the H_∞ controller actually compensated the change in the plant's structure better than the μ synthesis controller. An explanation for this behaviour may be due to way in which the uncertainty in the plant was defined. Due to the need to reduce computational complexity, the uncertainty is defined in an abstract manner, which may limit the effectiveness of D-K synthesis.

6.1.3 Optimal PID Simulation

The following plots show the performance of a PI controller with a minimal ∞ norm bound $\gamma = 0.905$. This controller was synthesized using the techniques presented in Chapter 5. Note that due to the nonconvex nature of the optimization process, this controller may not be the most optimal configuration of controller gains available.

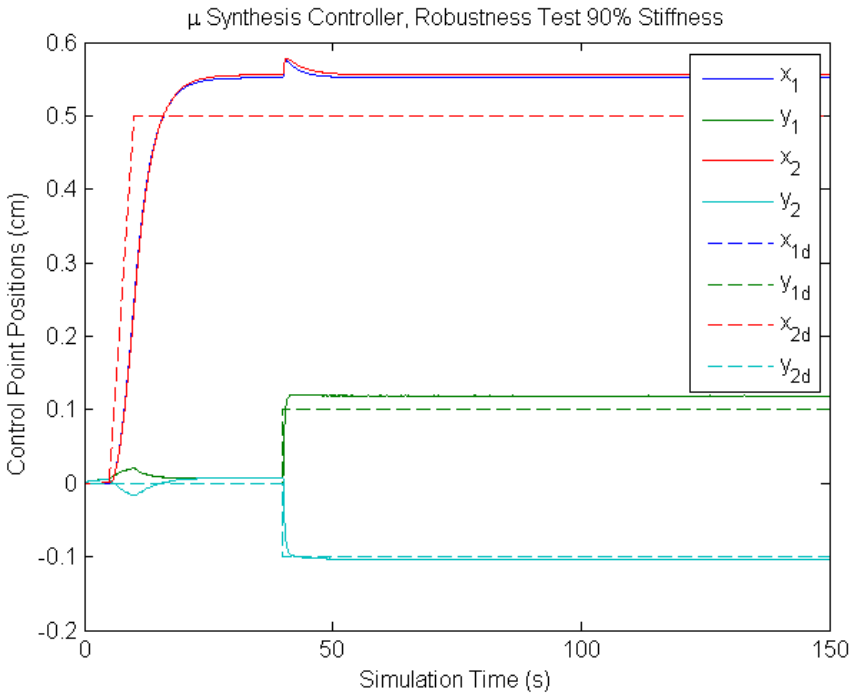

Figure 6.14: μ Synthesis Robust Performance, $\lambda = 0.1$ Scaling of W_u, 90% Stiffness

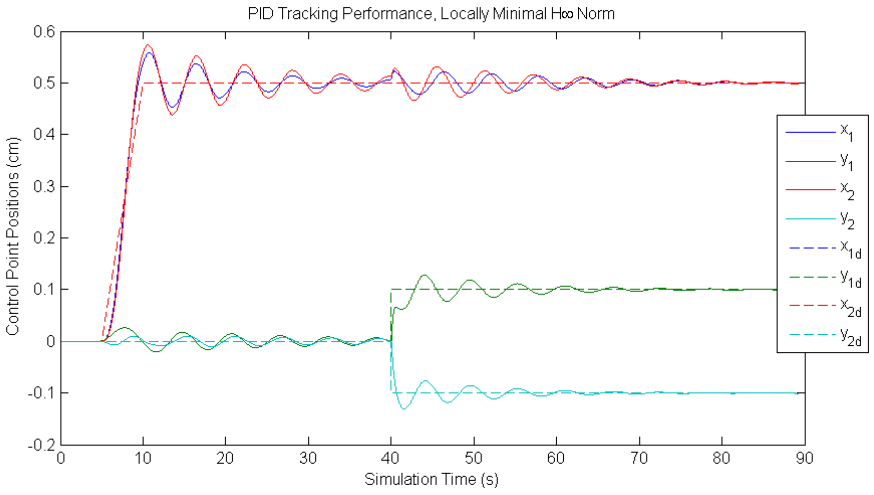

Figure 6.15: Optimal PID Set Point Tracking, $\lambda = 1$ Scaling of W_u

Gain parameters were obtained by using MATLAB^{TM}'s $fmincon$ function to solve an SQP approximation of the objective function. Figure 6.15 shows the set point tracking performance of the controller. Note that the controller is underdamped, since the objective function of the optimization exclusively focused on stability and the minimization of the closed loop ∞-norm. Inclusion of K_d may help reduce the oscillations of the underdamped system, but this will also have an impact on the ∞-norm.

Figure 6.16 and 6.17 show the simulations of the robustness tests performed on the PI controller. Note that when the plants stiffness is 75% of its nominal value, the controller has extreme oscillatory behaviour. Although this controller was designed to minimize an ∞-norm, it doesn't have the same robustness as seen in the previous two controller designs, thus making it the weakest candidate for applications of the three simulated control architectures.

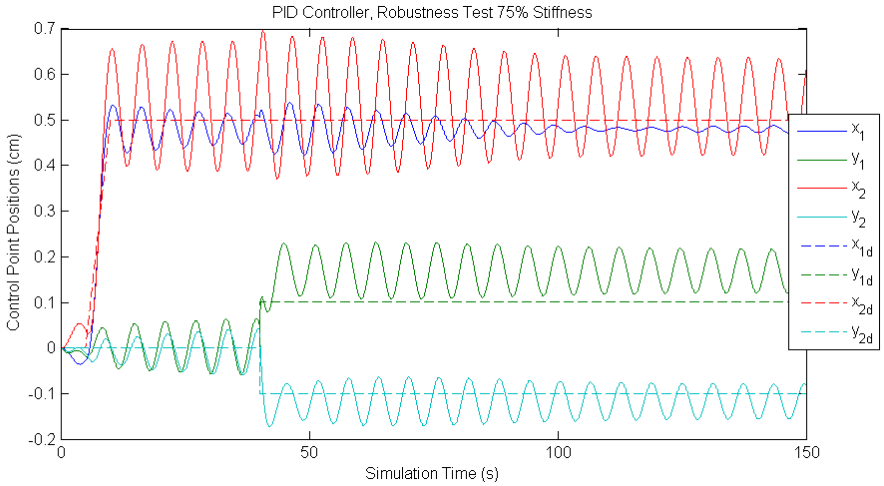

Figure 6.16: Optimal PID Robust Performance, $\lambda = 1$ Scaling of W_u, 90% Stiffness

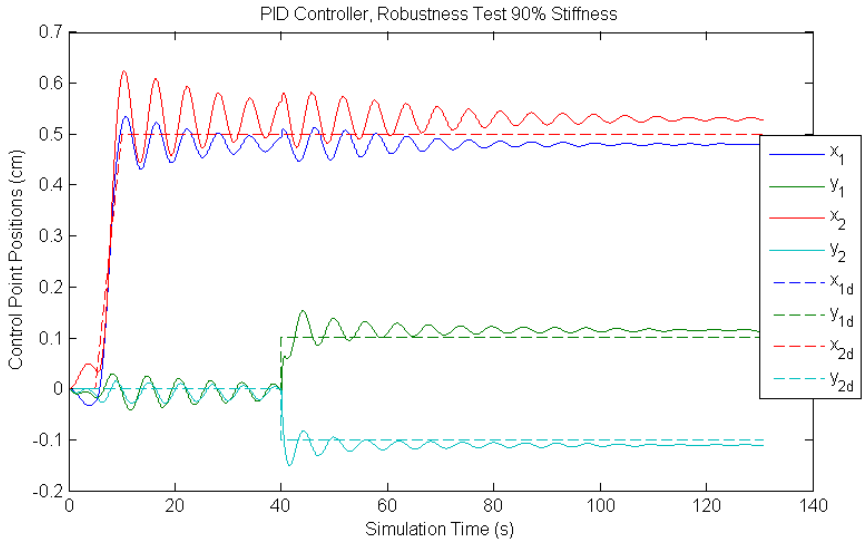

Figure 6.17: Optimal PID Robust Performance, $\lambda = 1$ Scaling of W_u, 90% Stiffness

6.2 Experimental Setup

After simulations were completed, experimentation was performed to complete evaluation of the control approach. Two phases of experiments were carried out on two different materials. Experiments were performed using two CRS Catalyst 5 robots fitted with force sensors, and controlled by an open-source controller by Quanser (2012). The deformable object used during the first phase of testing was a 0.15875 cm thick silicone rubber sheet, cut into a 7.5 cm radius circle. Polyether foam was placed below the sheet to prevent folding when actuation was performed in the negative x (towards ground) direction. The second phase of experiments was carried out on a nonlinear rubber compound sheet, 7.5 cm in radius and 2 cm thick. The material used in the second set of experiments was sticky, and thus metal bearings were used to reduce the effect of friction on the experimental results. Approximately 5 cm of the circumference of each sheet was fastened to the workspace with clamps to form a ground. Positions of the control points were determined through the use of a stereoscopic Bumblee2 camera from Point Grey Research Point Grey Research (2011) tracking the location of LED markers. These positions were tracked through use of a C++ executable. Command forces generated by the deformable object controller were fed into a transposed Jacobian based force controller, details of which are described by Sciavicco and Siciliano (2000). Figure 6.18 shows a block schematic of the experimental system. Figure 6.19 shows the experimental setup for the first group of experiments, and Figure 6.20 shows the nonlinear material used in the second phase.

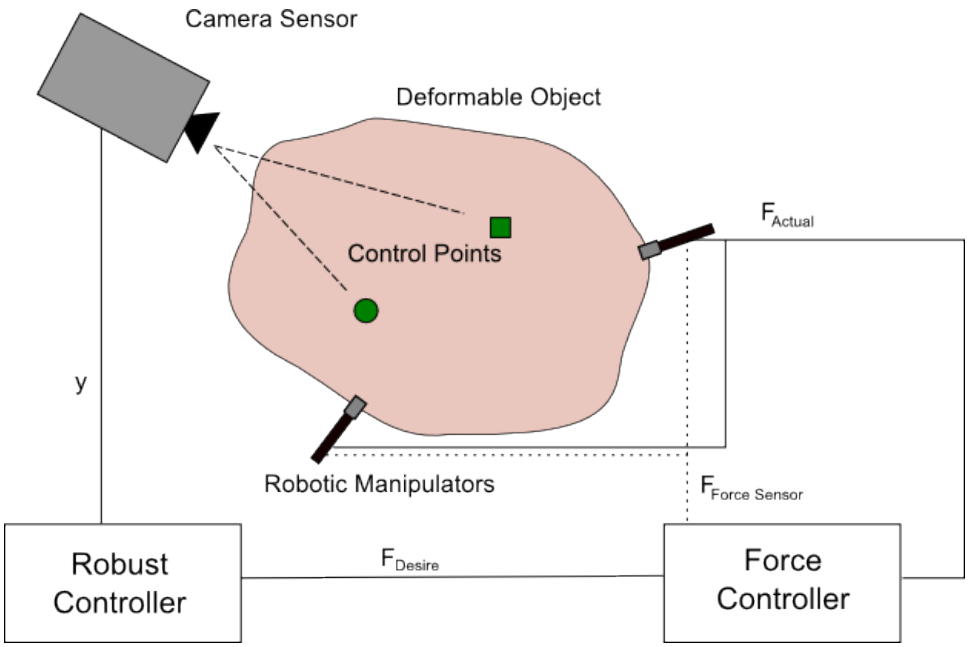

Figure 6.18: Block Schematic Of Controller Implementation

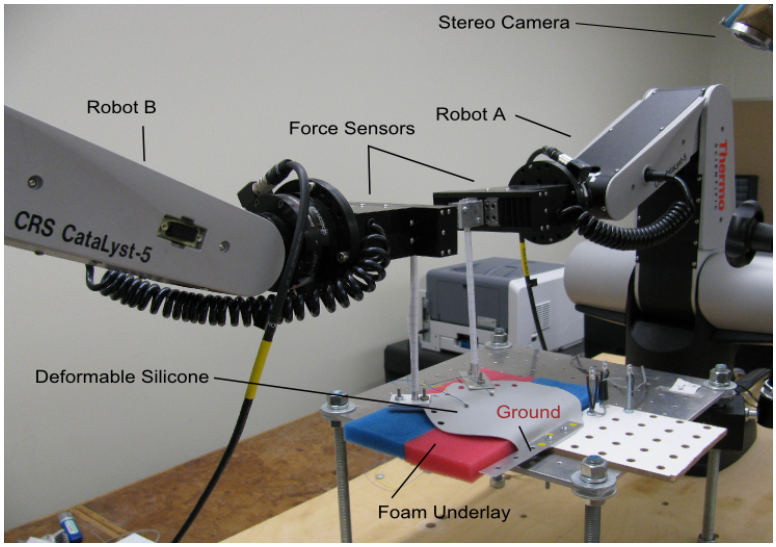

Figure 6.19: Experimental Setup with Silicone Rubber Sheet

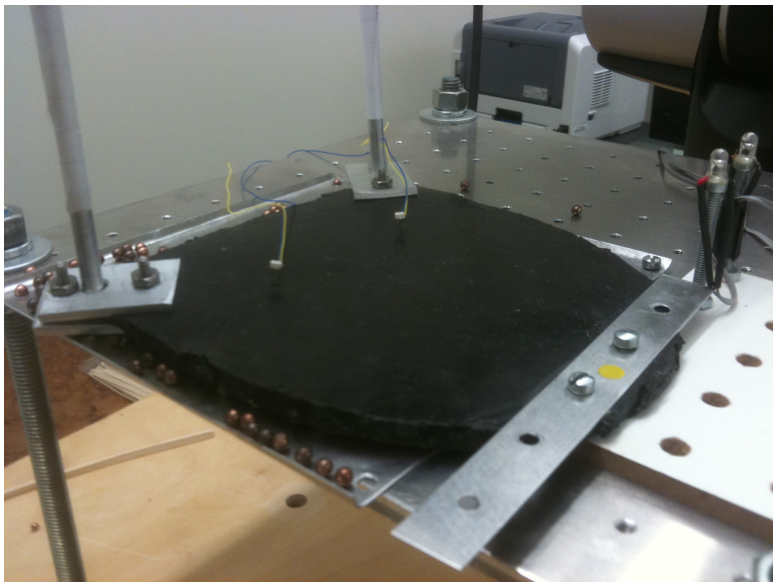

Figure 6.20: Experimental Material Used in Second Phase of Experiments

6.3 Experimental Results

Experiments were performed on three control approaches within the two experimental phases. The evaluated controllers included variously weighted H_∞ controllers, a tuned PID controller, and an optimal PID controller that was synthesized using techniques discussed in Chapter 5. The tuned PID controller was initially tuned based on simulations performed using a FEM plant that was used during synthesis of the H_∞ and optimal PID controllers. Due to deterioration of the material used in the intial phase of experiments, the optimal PID controller was only tested with the second nonlinear rubber compound.

In the first set of experiments, the tuned PID controller and H_∞ controller were subjected to a reference signal that tested stretching performance in both the x and y axes. The reference profile can be seen in Figure 6.21

Performance of the H_∞ controlled system, presented in Figure 6.22, was excellent,

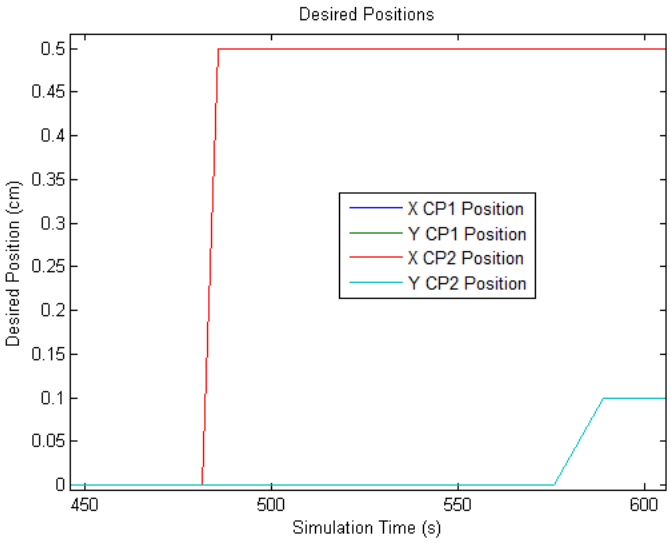

Figure 6.21: Phase One Experimental Reference Signal Profile

with errors staying within the range of 0.5 mm, except for overshoots that are clearly visible when commands are changed. Large overshoots seen in all experimental plots are the result of a deficiency in the software used to record experiments. Whenever a command was issued during an experiment, the software would fail to record data. This results in the sharp spikes in error, followed by ramps due to extrapolation by MATLAB^{TM} of the missing data. This deficiency was corrected during the second phase of experiments through use of a set point script.

It is important to note that stretch commands were tracked in a superior manner to compressions. This behaviour was due to the fact that the silicone rubber sheet folded during compression, and as a result the force sensors on the robots had reduced sensitivity to force changes. Figure 6.23 shows the performance of the H_∞ controller in response to the first ramp command seen in Figure 6.21. Slack between the actuator and object resulted in the delay seen in the figure, however, even with this slack, the control point error still began converging to zero within 10 seconds of

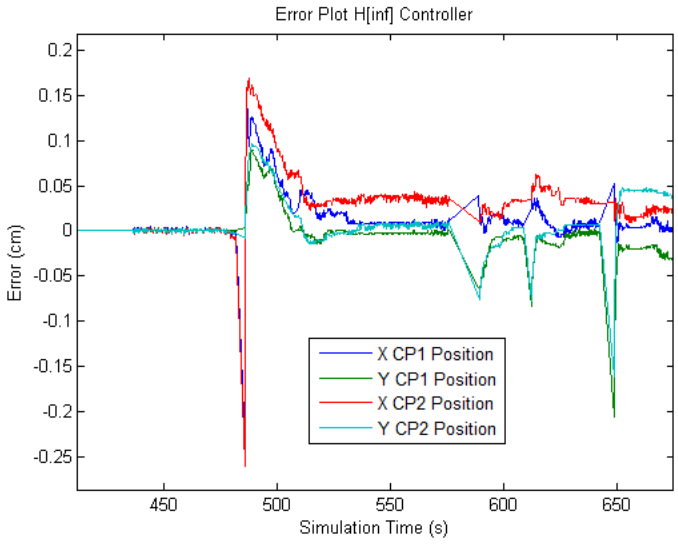

Figure 6.22: Phase One H_∞ Set Point Performance, $W_u = 1$

the start of visible actuation.

The initial tuned PID controller based off the plant model was marginally stable, with extreme oscillations occurring during the experimental procedure, seen in Figure 6.24

An second set of gains for the tuned PID controller was then acquired through experimental tuning. This resulted in a stable controller, whose performance is seen in Figure 6.25. Note the tracking here is similar to that of the H_∞ controller, supporting the findings of Wada *et al.* (1998) and Mallapragada (2009).

The optimal PID controller was tested during the second phase of experiments. This material had several characteristics which caused large difficulties for the model based approaches investigated throughout the thesis. One major issue was determination of the object's material parameters. In all experiments, the feed forward force that was applied was based off of a FEM model, whose stiffness was defined based off

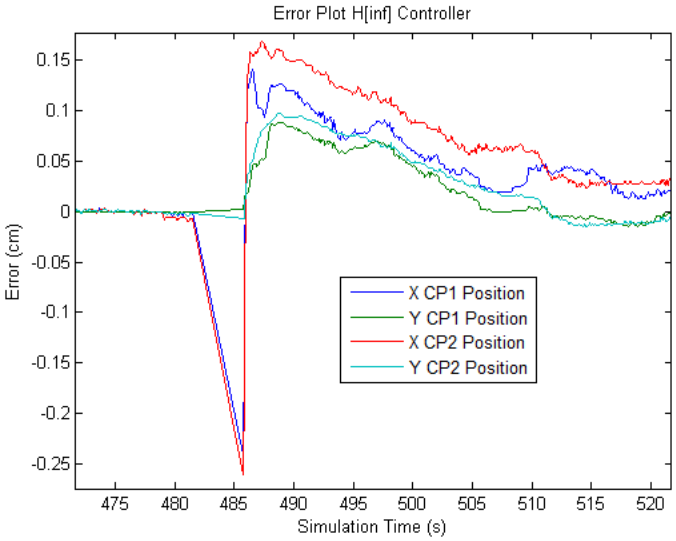

Figure 6.23: Phase One H_∞ Ramp Response Performance, $W_u = 1$

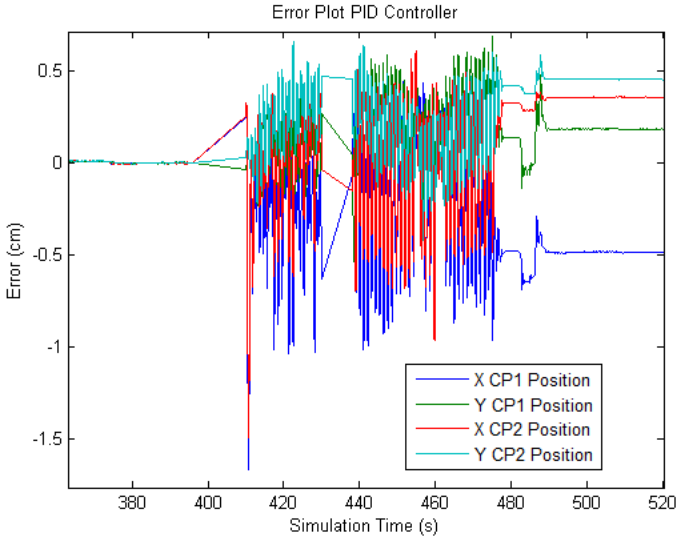

Figure 6.24: Phase One Model Tuned PID Performance

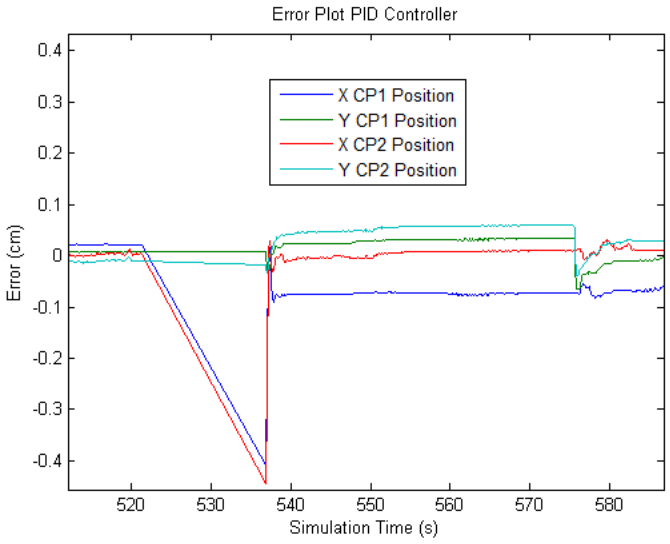

Figure 6.25: Phase One Experimentally Tuned PID Performance

of a Young's modulus and poisson ratio obtained from material data sheets. During experimentation, it was found that the FEM model was far stiffer than the material appeared to be during experimentation. This resulted in a much smaller region of stability for the optimal PID controller than that seen in Figure 5.1. This mismatch also greatly impacted the effectiveness of H_∞ controllers. In addition to this stiffness mismatch, the material demonstrated creep, which negatively impacted the ability of controllers to perform the IDOM task. Due to time constraints, a more realistic tissue phantom could not be acquired. Thus, the second phase of experiments illustrate how these controllers behave when confronted with nonlinear plants.

The set point reference signal for the optimal PID controller can be seen in Figure 6.26. This was the last experiment performed with manual modification of the reference signal. Further experiments performed in phase two used the harmonized reference seen in Figure 6.27

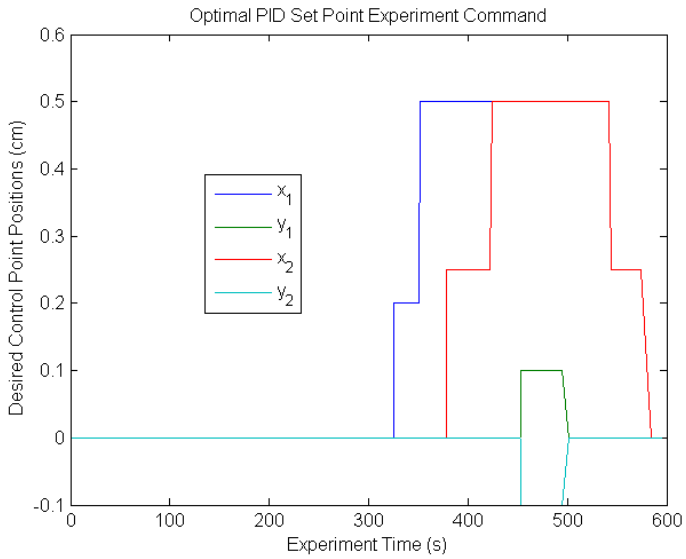

Figure 6.26: Optimal PID Reference Signal Profile

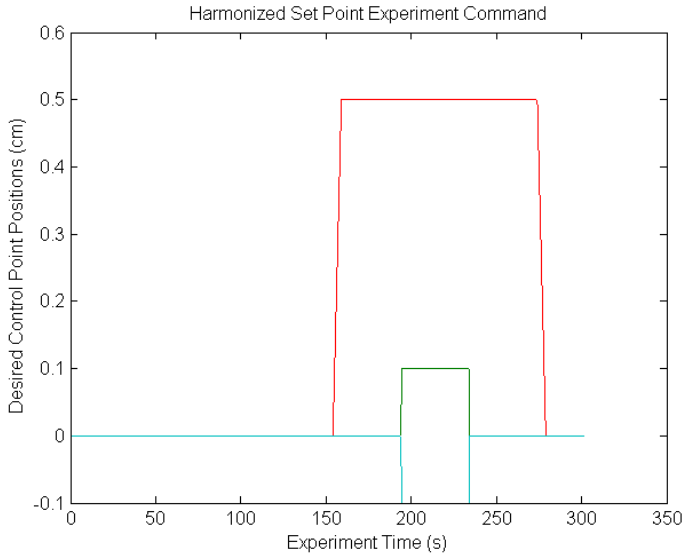

Figure 6.27: Harmonized Reference Signal Profile

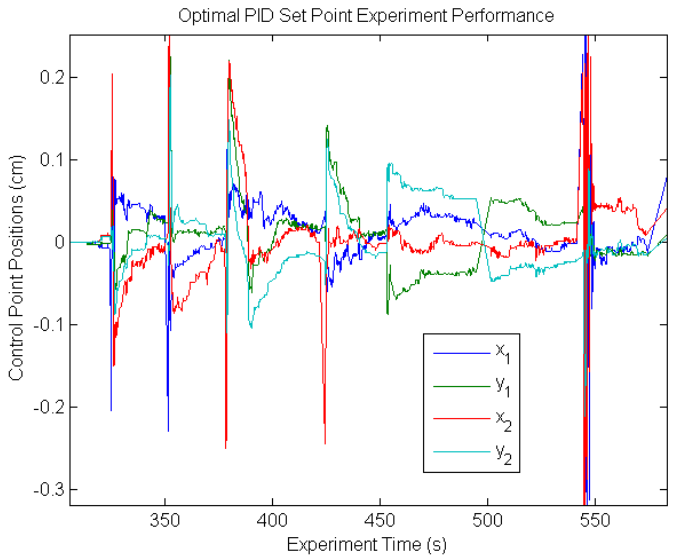

Figure 6.28: Phase Two Optimal PID Set Point Performance

Since PID control is not reliant on any model, the optimal PID controller performed well during set point experiments. The tracking profile can be seen in Figure 6.28. Note that once more the large spikes in error are due to steps in the reference signal. Error quickly decays below 1 mm, and within 25 seconds of a step command the error is close to 0.5 mm.

We now contrast this tracking performance with that of two H_∞ controllers, synthesized with $W_u = 1$ and $W_u = 0.5$, seen in Figure 6.29 and 6.30 respectively. Note the nonlinear behaviour of the plant has reduced the error regulation of the H_∞'s significantly from that seen in Figure 6.22. Average tracking error for these controllers is 1 mm, with a worst case error of 2.5 mm, which is unacceptable for experiments with small reference signals. Mirroring the simulations, the controller with $W_u = 0.5$ had better tracking performance than that of $W_u = 1$.

If we compare the controllers in terms of command forces, as seen in Figures

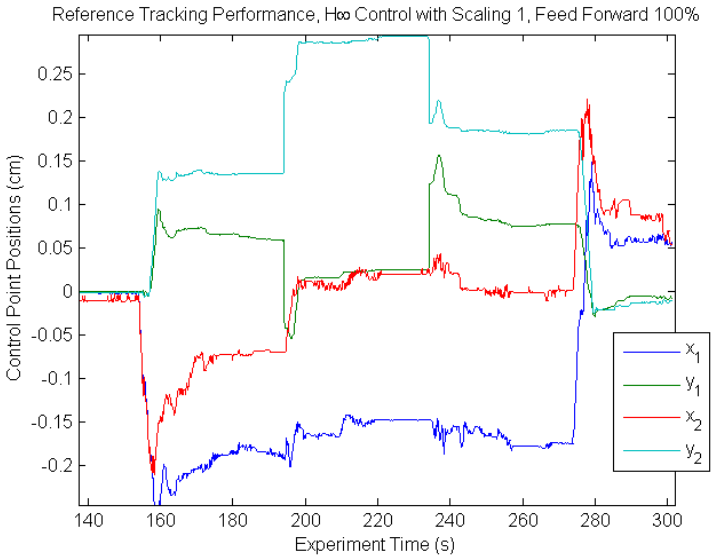

Figure 6.29: Phase Two H_∞ Set Point Performance, $W_u = 1$

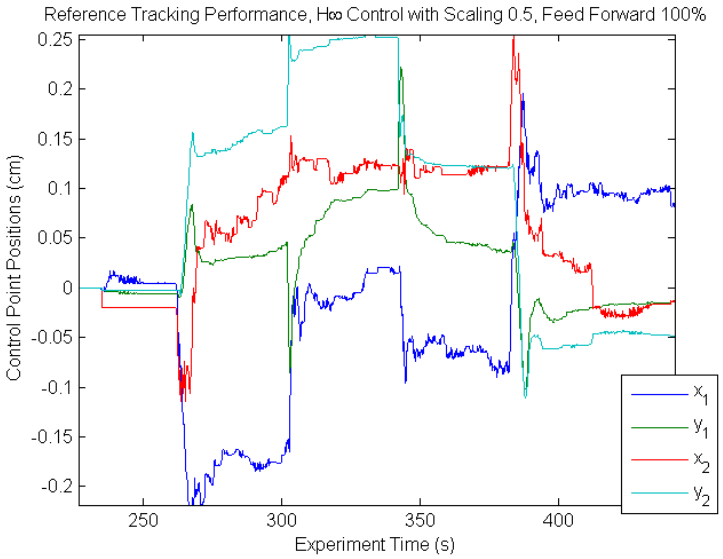

Figure 6.30: Phase Two H_∞ Set Point Performance, $W_u = 0.5$

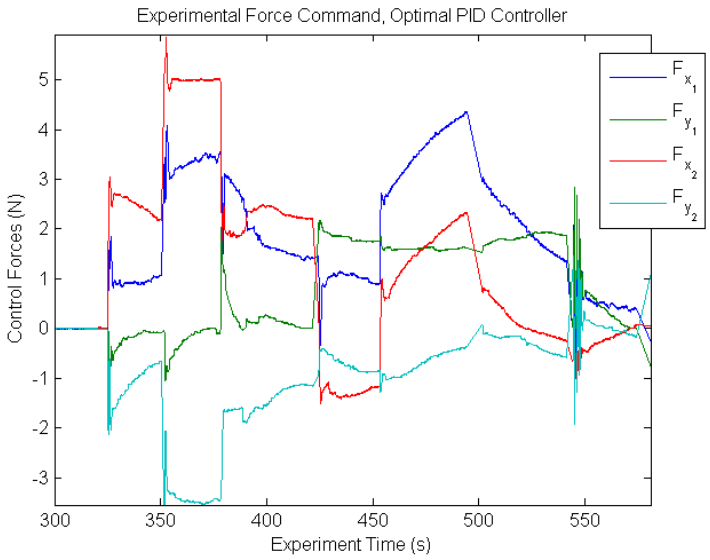

Figure 6.31: Phase Two Optimal PID Force Command

6.31,6.32 and 6.33, we can see the major cause of this disparity in performance. The H_∞ controller has far smaller command forces than the Optimal PID controller. The maximal allowable control force that an H_∞ can apply is dictated by the scaling of W_u, with smaller scalings allowing larger forces. Thus, H_∞ controllers with smaller W_u weights may have performed closer to the PID controller, if they had been discretizable.

Unfortunately, due to time constraints, μ synthesis controllers were not experimentally validated. It is likely their performance would have been similar to that observed for the H_∞ controllers.

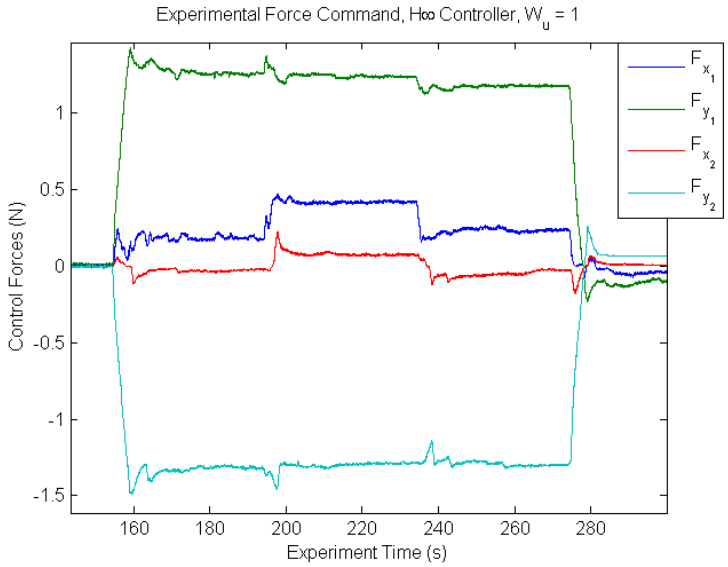

Figure 6.32: Phase Two H_∞ Force Command, $W_u = 1$

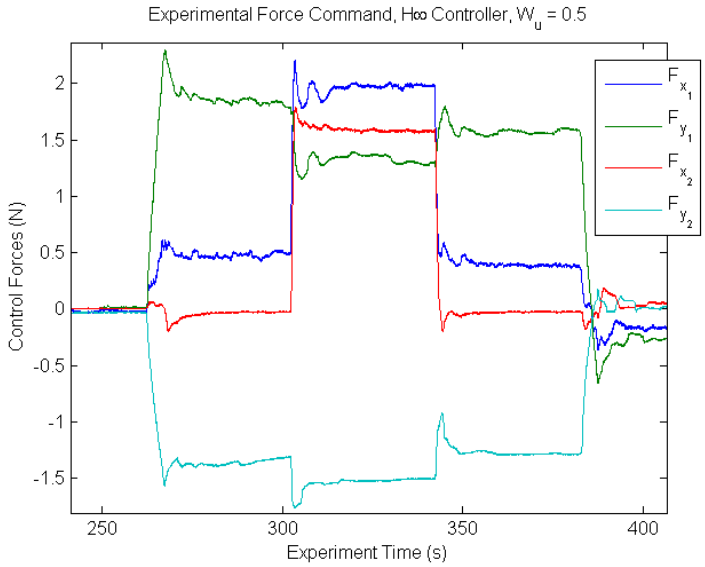

Figure 6.33: Phase Two H_∞ Force Command, $W_u = 0.5$

Chapter 7

Discussion and Conclusion

7.1 Discussion

The simulation and experiments provided some insight into the feasibility of using H_∞ based controller synthesis for IDOM tasks. For plants that behave in a linear manner, such as the material used during the Phase One experiments, H_∞ control appears to be a good candidate. The tracking performance of H_∞ for such plants was relatively fast and resulted in tracking error near 0.5 mm. The introduction of the polyether foam, used to reduce folding of the thin silicone rubber material, allowed validation of the robustness of the H_∞ design, due to its effect on the stiffness of the experimental plant. When contrasted to traditional tuning based PID approaches, the tracking performance was similar, but the key advantage of the H_∞ approach was the ability to generate a stable controller directly from the FEM plant model.

In phase two, the optimal PID based on H_∞ techniques proved to be superior to the model dependant H_∞ control designs. Although all controllers were stable, the optimal PID controller was able to fully compensate for the mismatch between

the modelled plant and that observed during experimentation. One key point to note in the H_∞ controller's defence, is the effect of scaling the W_u term with λ. In simulation, it was determined that good tracking for H_∞ controllers occurred around $\lambda = 0.05$. Unfortunately, weighted plants with low λ scalings resulted in increasingly nonsmooth controller behaviour. This proved problematic, since stable implementation of the resultant controllers required successively smaller timesteps to avoid unstable behaviour.

The robotic platform used for experiments required controllers to be implemented with MATLAB$^{\text{TM}}$'s *ode*1, with a fixed lower bound on the timestep of 1 ms. This constraint limited scaling to a lower bound of $\lambda = 0.5$. The experimental performance of H_∞ controllers seen in Figures 6.29 and 6.30 was similar to their simulated behaviour, indicating that better scaled H_∞ may have corrected the observed poor performance characteristics. To enable controllers with better scaling to be used in an experimental setting, it may be viable to use a truncation based reduction approach on the H_∞ controller itself. The removal of fast poles through truncation may reduce the timestep necessary to use the controller.

These findings allow a robust controller synthesis approach for IDOM tasks to be proposed. Designers should initially attempt to build an H_∞ controller for their plant, modifying λ to acquire satisfactory tracking performance in simulations. If the resultant controller becomes unstable at the minimum timestep allowed by the application platform, and reduction techniques fail to alleviate the problem, the designer may then switch over to the optimal PID synthesis approach, which uses the same weighted plant that was designed for the H_∞ problem. By following this approach, the designer can take advantage of the superior performance of H_∞ methods for plants

in which both H_∞ and PID work, and can also take advantage of the optimal PID's ability to regulate an IDOM task for nonlinear plants where the model dependant H_∞ methods falter.

The μ synthesis based controllers designed for the IDOM task investigated in this thesis performed in a very similar manner to H_∞ controlled systems. The manner in which the uncertainty for the weighted plant was defined was an ad-hock measure used primarily to allow computation of the controller in a reasonable timeframe, and resulted in a relatively high optimal μ value of 1.6. This high μ helps explain why the H_∞ designs had superior robust performance to μ based controllers during simulations.

It is evident that μ based approaches are far more computationally expensive to design that equivalent H_∞ controllers, and systems containing large numbers of states exacerbate this problem. This makes them an ill-fit with FEM models, unless custom synthesis code is written, or large amounts of computational power is available. In future studies, LSMP based μ synthesis controllers should be compared to the FEM based H_∞ controllers investigated over the course of this thesis. The integral role of uncertainty in μ synthesis allows μ based controllers to overcome the shortcomings of LSMP, and the low numbers of states associated with the approach would allow detailed customization of uncertain parameters.

7.2 Conclusion

Worst case based robust control techniques such as H_∞ control provide promising performance characteristics for deformable objects, within reasonable bounds. For deformable objects without excess nonlinear behaviour, H_∞ provides stable, robust,

and reasonably fast tracking in set point regulation tasks. Tradeoffs in performance for increased robustness were an inherent feature of the controller designs investigated throughout this thesis. Construction of the weighted plant model for purposes of H_∞ controller synthesis provides the basis for alternate controller design approaches such as H_∞ based optimal PID and μ synthesis. The flexibility to produce multiple controllers with minimal effort makes H_∞ controller synthesis an ideal candidate for IDOM tasks where experimental tuning is infeasible. By utilizing the multi-control design approach, the model dependant limitations of H_∞ techniques can be minimized, while still retaining stability and decent performance characteristics.

Future work in producing stable, high performance IDOM controllers may wish to focus on nonlinear approaches, which may not be constrained by the same tradeoffs inherent in linear H_∞ based methods. Further avenues of potential research include building a plant specifically designed for μ synthesis, or a weighted plant including an integrator for purposes of building H_∞ controllers.

Appendix A

Alternate Global Optimization Approaches

This section of the appendix details two alternate global optimization methods that were considered, but ultimately discarded, for the purposes of synthesising optimal PID controller gains. They are

- Branch and Bound Searches
- Pattern Search Algorithms

Pattern Search Algorithms

Pattern Search provides another method to solve the type of nonconvex problem that PID optimization presents. The approach works for both unconstrained and linearly constrained optimizations and makes use of the barrier functions described earlier in this chapter Audet and J. E. Dennis (2003). The basic premise of Pattern Search (PS) involves building meshes around a current incumbent solution point.

Searches are then performed within these meshes for points that give smaller objective function values. PS is split into multiple iterations, each composed of two main steps, $SEARCH$ and $POLL$. As its name implies, $SEARCH$ performs the afore mentioned scan of a mesh of points surrounding the current incumbent solution. If $SEARCH$ succeeds, the mesh is coarsened to search a larger area of the solution space. If $SEARCH$ fails to produce a point with a lower objective value, $POLL$ is called. $POLL$ scans neighbouring regions to seek a lower objective value solution. If both $SEARCH$ and $POLL$ fail to replace the current incumbent point, it is labeled a *localmeshoptimizer* Audet and J. E. Dennis (2003), and the mesh is refined to search more tightly around the current optimum. In this way, PS converges onto an optimal solution, although, depending on the initial mesh selected for early iterates, this optimum may not be the global optimal solution. Pattern Search is implemented in MATLAB, making it a good candidate to solve the PID optimization. Readers who want a more in depth discussion of this optimization technique are referred to Audet and J. E. Dennis (2003) and Audet and Dennis Jr (2004).

Brand and Bound Searches

Branch and bound searches are based on the idea of generating a tight bound on a solution set, similar in the way that the D-K solution to μ synthesis finds a final structured singular value for its controller design. Given an objective function $F(X)$, we define an upper and lower bound function such that Boyd (2008)

$$\Phi_{lb}(X) \leq \min F(X) \leq \Phi_{ub}(X) \tag{A.1}$$

The bounding functions Φ_{lb} and Φ_{lb} are designed by the engineer performing the

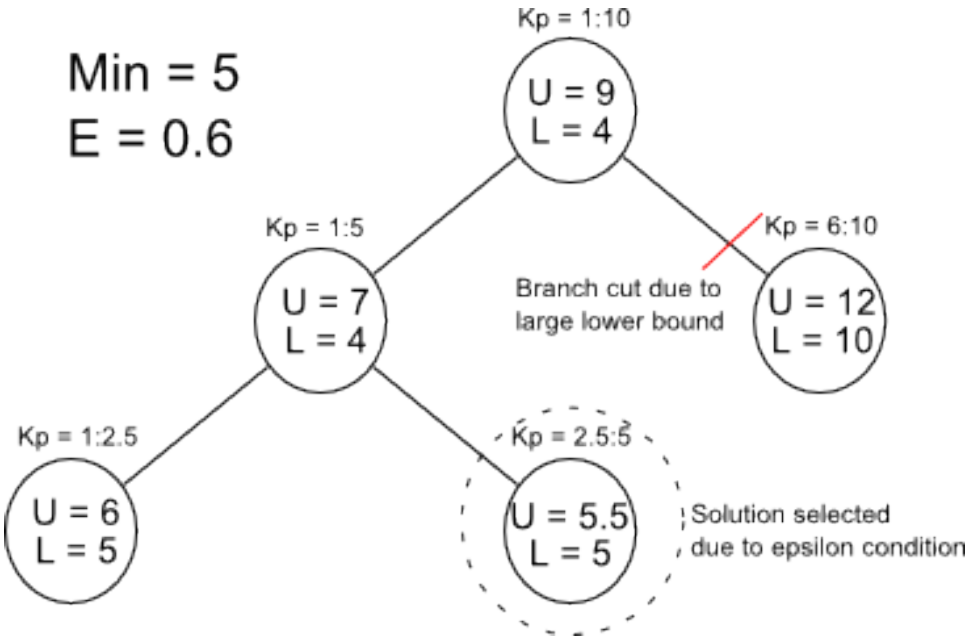

Figure A.1: Simple Branch and Bound Example with Simple K_p Control

optimization, but must tightly bound the solution as their parameter set is reduced. As its name implied, Branch and Bound (BaB) optimization works by continuously breaking the optimization into smaller subproblems until the upper bound U_i and lower bound L_i of a given branch fulfill the condition $U_i - L_i < \epsilon$.

In each step of the optimization, the upper and lower bound of all branches of the optimization are calculated. Branches are pruned from the process if their bound L_i is larger than the current overarching optimization upper bound U. Branches with the smallest lower bounds form the region of the next branch. Figure A.1 gives an example of the technique for a simple proportional gain controller.

The primary difficulty associated with using BaB is proper selection of the bounding functions, but these will not be discussed within the scope of this thesis. Since the ∞ norm calculation already produces an upper bound for the objective value γ, BaB is a promising solution route for our particular PID optimization. Readers interested

in learning more about Branch and Bound optimization can consult textbooks such as Scholz (2012).

Bibliography

A. C. Antoulas, D. C. S. and Gugercin, S. (2001). A survey of model reduction methods for large-scale systems. In *Structured Matrices in Mathematics, Computer Science, and Engineering I*, volume 280 of *Contemporary Mathematics*, pages 193–219. American Mathematical Society.

Astrom, K., Panagopoulos, H., and Hagglund, T. (1998). Design of pi controllers based on non-convex optimization. *Automatica*, **34**(5), 585 – 601.

Audet, C. and Dennis Jr, J. (2004). A pattern search filter method for nonlinear programming without derivatives. *SIAM Journal on Optimization*, **14**(4), 980–1010.

Audet, C. and J. E. Dennis, J. (2003). Analysis of generalized pattern searches. *SIAM Journal of Optimization*, **13:3**, 889–903.

Bhavikatti, S. (2007). *Finite Element Analysis*. New Age International (P) Ltd.

Bischoff, S. and Kobbelt, L. (2002). Ellipsoid decomposition of 3d-models. In *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pages 480–488.

Blanchini, F., Lepschy, A., Miani, S., and Viaro, U. (2004). Characterization of pid and lead/lag compensators satisfying given h¡ sub¿8¡/sub¿ specifications. *Automatic Control, IEEE Transactions on*, **49**(5), 736–740.

Boggs, P. T. and Tolle, J. W. (1995). Sequential quadratic programming.

Boyd, S. (2004). *Convex Optimization*. Cambridge University Press.

Boyd, S. (2008). Branch and bound methods. Lecture Notes from EE364b, Stanford University.

Bro-Nielsen, M. (1998). Finite element modeling in surgery simulation. *Proceedings of the IEEE*, **86**(3), 490 –503.

Doyle, J., Glover, K., Khargonekar, P., and Francis, B. (1988). State-space solutions to standard h2 and h[inf]; control problems. In *American Control Conference, 1988*, pages 1691 –1696.

Doyle, J., Glover, K., Khargonekar, P., and Francis, B. (1989). State-space solutions to standard h2 and h infin; control problems. *Automatic Control, IEEE Transactions on*, **34**(8), 831–847.

Fanson, R. and Patriciu, A. (2010). Model based deformable object manipulation using linear robust output regulation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 496 –501.

Francis, B. A. (1987). *A course in H [infinity] control theory*. Berlin; New York: Springer-Verlag.

Gahinet, P. and Apkarian, P. (1994). A linear matrix inequality approach to h[infinity] control. *International Journal of Robust and Nonlinear Control*, **4**(4), 421–448.

Gibson, S. F. and Mirtich, B. (1997). A survey of deformable modeling in computer graphics. *MERL, TR-97*, **19**.

Henrich, D. and Worn, Hrn, H. (2000). *Robot Manipulation of Deformable Objects*. Springer.

Henrich, D., Ogasawara, T., and Worn, H. (1999). Manipulating deformable linear objects - contact states and point contacts. In *Assembly and Task Planning, 1999. (ISATP '99) Proceedings of the 1999 IEEE International Symposium on*, pages 198 –204.

Hirai, S. and Wada, T. (2000). Indirect simultaneous positioning of deformable objects with multi-pinching fingers based on an uncertain model. *Robotica*, **18**(1), 3–11.

Ho, M.-T. (2003). Synthesis of hinfinty pid controllers: A parametric approach. *Automatica*, **39**(6), 1069 – 1075.

How, J. (Fall 2007). Course materials for 16.31 feedback control systems.

Hu, J., Bohn, C., and Wu, H. (2000). Systematic h[inf] weighting function selection and its application to the real-time control of a vertical take-off aircraft. *Control Engineering Practice*, **8**(3), 241 – 252.

Hughes, T. J. R. (2000). *The Finite Element Method*. Dover.

Kang, H. and Wen, J. (2000). Autonomous suturing using minimally invasive surgical

robots. In *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*, pages 742–747.

Khabbaz, F. H. and Patriciu, A. (2011). Stitching path planning using circular needles-tissue interaction model. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 1134–1139. IEEE.

Koustoumpardis, P. and Aspragathos, N. (2007). Neural network force control for robotized handling of fabrics. In *Control, Automation and Systems, 2007. ICCAS '07. International Conference on*, pages 566 –571.

Li, R.-C. and Bai, Z. (2006). Structure-preserving model reduction. In J. Dongarra, K. Madsen, and J. Waniewski, editors, *Applied Parallel Computing. State of the Art in Scientific Computing*, volume 3732 of *Lecture Notes in Computer Science*, pages 323–332. Springer Berlin Heidelberg.

Lundstrm, P., Skogestad, S., and Wang, Z.-Q. (1991). Performance weight selection for h-infinity and u-control methods. *Transactions of the Institute of Measurement and Control*, **13**, 241–252.

Mallapragada, V.G. Sarkar, N. P. T. (2009). Robot-assisted real-time tumor manipulation for breast biopsy. *IEEE Transactions on Robotics*, **25**, 316–324.

McInerney, T. and Terzopoulos, D. (1996). Deformable models in medical image analysis: a survey. *Medical image analysis*, **1**(2), 91.

Moser, A. N. (1993). Designing controllers for flexible structures with h-infinity/ mu -synthesis. *Control Systems, IEEE*, **13**(2), 79–89.

Nageotte, F., Zanne, P., De Mathelin, M., and Doignon, C. (2005). A circular needle path planning method for suturing in laparoscopic surgery. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 514–519.

Panagopoulos, H., Astrom, K., and Hagglund, T. (2002). Design of pid controllers based on constrained optimisation. *Control Theory and Applications, IEE Proceedings* -, **149**(1), 32 –40.

Patil, S. and Alterovitz, R. (2010). Toward automated tissue retraction in robot-assisted surgery. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2088–2094.

Point Grey Research, w. (2011). Point grey bumblee 2 information page.

Polycarpou, A. C. (2006). *Introduction to the finite element method in electromagnetics*. Morgan & Claypool Publishers.

Poulose, B. K., Kutka, M. F., Mendoza-Sagaon, M., Barnes, A. C., Yang, C., Taylor, R. H., and Talamini, M. A. (1999). Human vs robotic organ retraction during laparoscopic nissen fundoplication. *Surgical Endoscopy*, **13**, 461–465.

Quanser (2012). Quanser web site, www.quanser.com.

Safonov, M. and Chiang, R. (1989). A schur method for balanced-truncation model reduction. *Automatic Control, IEEE Transactions on*, **34**(7), 729 –733.

Scholz, D. (2012). *Deterministic Global Optimization*. Springer.

Sciavicco, L. and Siciliano, B. (2000). *Modeling and Control of Robot Manipulators, Second Edition*. Springer-Verlag.

Shibata, M. and Hirai, S. (2006). Soft object manipulation by simultaneous control of motion and deformation. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2460 –2465.

Slotine, J. and LI, W. (1991). *Applied Nonlinear Control*. Prentice Hall.

Smolen, J. and Patriciu, A. (2009). Deformation planning for robotic soft tissue manipulation. In *Advances in Computer-Human Interactions, 2009. ACHI '09. Second International Conferences on*, pages 199 –204.

Soylemez, M., Munro, N., and Baki, H. (2003). Fast calculation of stabilizing pid controllers. *Automatica*, **39**(1), 121–126.

Tan, N., Kaya, I., Yeroglu, C., and Atherton, D. P. (2006). Computation of stabilizing pi and pid controllers using the stability boundary locus. *Energy Conversion and Management*, **47**(18), 3045–3058.

Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. (1987). Elastically deformable models. In *ACM Siggraph Computer Graphics*, volume 21, pages 205–214. ACM.

Tokumoto, S., Fujita, Y., and Hirai, S. (1999). Deformation modeling of viscoelastic objects for their shape control. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 767 –772 vol.1.

Torabi, M., Hauser, K., Alterovitz, R., Duindam, V., and Goldberg, K. (2009). Guiding medical needles using single-point tissue manipulation. In *Proc IEEE Int. Conf. on Robotics and Automation*, pages 2705–2710.

Wada, T., Hirai, S., and Kawamura, S. (1998). Indirect simultaneous positioning operations of extensionally deformable objects. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 2, pages 1333 –1338 vol.2.

Wada, T., Hirai, S., Kawamura, S., and Kamiji, N. (2001). Robust manipulation of deformable objects by a simple pid feedback. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 85 – 90 vol.1.

Wakamatsu, H., Arai, E., and Hirai, S. (2006). Knotting/unknotting manipulation of deformable linear objects. *International Journal of Robotics Research*, **25**(4), 371–395.

Yamamoto, S. and Hashimoto (1991). Present status and future needs: the view from japanese industry. In *Proceedings of fourth intemational conference on Chemical process control, Texas, USA*.

Zhou, K. (1998). *Essentials of Robust Control*. Prentice Hall.

Ziegler, J. G. and Nichols, N. B. (1993). Optimum settings for automatic controllers. *Journal of Dynamic Systems, Measurement, and Control*, **115**(2B), 220–222.