Modeling of Linear Systems with Parameter Variations: Applications in Hard Disk and Ball Screw Drives

by

Daniel Sepasi

B.Sc., Mechanical Engineering, Sharif University of Technology, 2005
M.A.Sc., Mechanical Engineering, University of British Columbia, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES
(Mechanical Engineering)

The University Of British Columbia
(Vancouver)

August 2011

Abstract

This thesis considers variations in the parameters of the dynamics of linear systems, and tackles modeling of Linear Time-Invariant (LTI) and Linear Parameter Varying (LPV) plants. The variations in the dynamics make the controller design challenging, and to successfully overcome this challenge, two methods are proposed in this thesis.

One method generates a connected model set. The idea of the multi-dimensional principal curves methodology is employed to detect the nonlinear correlations between parameters of the given set of system dynamics. The connected model set is simple and tight, leading to both nonconservatism and reduced computational complexity in subsequent controller design, and hence, to improve the controller performance.

The other method is developed to derive a family of discrete model sets for a given set of system response data. A relaxed version of the normalized cut methodology is developed and used in an algorithm to divide a given set of system responses into the smallest possible number of partitions in such a way that a desired performance objective is satisfied for all partitions by designing one controller for each partition.

Using the proposed method, a tight uncertainty model is derived for Hard Disk Drive (HDD) systems, and an H_∞ controller is synthesized. The dynamics of HDDs is studied from a controller design point of view. Especially, the variations in the dynamics due to the change in temperatures and limited precision in the production line are examined.

Also, the variations in the dynamics of Ball Screw Drive (BSD) systems due to the structural flexibility, runout, and workpiece mass variation are studied. These three factors are explicitly incorporated in LPV models. To build the LPV models, it is determined how the system parameters are affected by two variables, namely, the measurable table position and the uncertain mass of the table. We design robust gain scheduling controllers which are scheduled by the table position and are robust over the table mass.

Preface

This thesis entitled "Modeling of Linear Systems with Parameter Variations: Applications in Hard Disk and Ball Screw Drives" presents the research performed by Daniel Sepasi[1]. The research conducted in this thesis was supervised by Dr. Ryozo Nagamune and co-supervised by Dr. Farrokh Sassani. In this section, we briefly explain the contents of the papers that are published or submitted for publications from this thesis [3, 4, 96–98]. We also clarify the relative contributions of co-authors in the papers.

- **M. Sepasi, F. Sassani, and R. Nagamune, *"Parameter uncertainty modeling using the multi-dimensional principal curves"*, Journal of Dynamic Systems, Measurement and Control, 2010, vol. 132, Issue 5, pp. 054501-054507.** This paper proposes a technique to model parametric uncertainties associated with linear time-invariant systems. The method is based on nonconvex optimization, involving a linear matrix inequality, a local optimization technique, and multi-dimensional principal curves. The proposed technique is explained in Chapter 2 of this thesis. The author of this thesis was the principal researcher of this publication. Drs. Ryozo Nagamune and Farrokh Sassani assisted with formulating the problem and writing the paper.

- **E. Azadi Yazdi, M. Sepasi, F. Sassani and R. Nagamune,**

[1]The author's given name was changed from Mohammad to Daniel during his Ph.D. program.

***"Automated multiple robust track-following control system design in hard disk drives"*, IEEE Transactions on Control System Technology, DOI: 10.1109/TCST.2010.2053541.**[2] This paper proposes a new design procedure for track-following control systems in hard disk drives. The procedure is automated, in the sense that, for given experimental frequency response data of the suspension arm dynamics and a model structure, it automatically constructs a model set with parametric uncertainties. Subsequently, for the transfer function set it automatically designs a partition of the uncertainties and corresponding multiple robust controllers. The first step of the procedure, i.e. model set construction, is developed by the author of the thesis, and the second step, i.e. multiple robust controllers design, is developed by Dr. E. Azadi Yazdi. Experiments on actual hard disk drives demonstrate the usefulness and efficiency of the proposed procedure. The experiments are performed by Dr. E. Azadi Yazdi and the author of the thesis. The results of this paper is partly presented in Chapter 4 of the thesis. Drs. Ryozo Nagamune and Farrokh Sassani provided practical insight to the problem, and contributed significantly to the writing of this paper.

- **D. Sepasi, R. Nagamune, and F. Sassani, *"Tracking control of flexible ball screw drives with runout effect and mass variation"*, Accepted in IEEE Transactions on Industrial Electronics, DOI: 10-1720-TIE.R2.**[3] In this paper, tracking controllers for a ball screw drive are designed, which consider flexibility and runout, as well as workpiece mass variation. The flexibility, runout, and mass vari-

[2] A brief version is also published in: E. Azadi Yazdi, M. Sepasi, F. Sassani and R. Nagamune, *"Automated multiple robust track-following control system design in hard disk drives"*, 2010 ASME Dynamic Systems and Control Conference, Boston, MA.

[3] A brief version is also published in: M. Sepasi, F. Sassani, and R. Nagamune, *"Tracking control of flexible ball screw drives with runout effect compensation"*, 2010 ASME Dynamic Systems and Control Conference, Boston, MA.

ation are explicitly incorporated in a linear parameter-varying (LPV) model. To build an LPV model, it is determined through the principal curve method how the system parameters are affected by two time-varying variables, namely, the measurable position and the uncertain mass. For the LPV model, we design robust gain scheduling controllers which are scheduled by the measurable position and are robust over the uncertain mass. The designed controllers are implemented on a ball screw drive system. Drs. Ryozo Nagamune and Farrokh Sassani supervised the research and assisted with conducting the experiment and writing the paper.

Table of Contents

List of Tables

List of Figures

Notation

Symbol	Description
$\mathbf{1}_n$	Vector with n elements of one
$\mathbf{0}_n$	Vector with n elements of zero
$\mathbb{R}$	Set of all real numbers
$\mathbb{R}^n$	Set of all n-dimensional real vectors
$\mathbb{R}^{n \times m}$	Set of all $n \times m$-dimensional real matrices
$\mathbb{C}$	Set of all complex numbers
$\mathrm{Tr}(A)$	The trace of a square matrix A
$A \succeq 0$	Positive semidefiniteness
$A \geq 0$	Element-wise non-negativity
$\boldsymbol{\mathcal{S}}_n$	Set of all n-dimensional symmetric matrices

Glossary

BSD	Ball Screw Drive
CNC	Computer Numerically Controlled
DM	Diffusion Map
DOB	Disturbance Observer
FFT	Fast Fourier Transform
FRF	Frequency Response Function
HDD	Hard Disk Drive
LDV	Laser Doppler Vibrometer
LFT	Linear Fractional Transformation
LLE	Local Linear Embedding
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
MAE	Mean Absolute Error
MDS	Multidimensional Scaling
MSD	Mass Spring Damper
NCut	Normalized Cut
NLPCA	Nonlinear Principal Component Analysis
PES	Position Error Signal
SDP	Semidefinite Programming
VCM	Voice Coil Motor

Acknowledgments

I would like to express my deepest appreciation to my advisors, Drs. Ryozo Nagamune and Farrokh Sassani, for their inspiration, friendship, encouragement, patience and unconditional support. I am extremely fortunate for having the opportunity to work with them and having the freedom to explore science in their laboratory. I am really thankful for their support over the last years.

I am indebted to Drs. Yusuf Altintas and Mu Chiao for sharing his lab equipment. I would like to convey my gratitude to our kind graduate secretary of Mechanical Engineering Department, Ms. Yuki Matsumura. In addition, I would like to extend my sincere thanks to Mr. Glenn Jolly, for his essential technical assistance during this research. Additionally, I would like to thank Mr. Perry Yabuno for timely acquisition of electro-mechanical devices required for the fabrication of the set-up.

Additionally, I am grateful to my dear colleagues in the Control Engineering Laboratory: Dr. Ehsan Azadi, Mr. Marius Postma, and Mr. Massih Hanifzadegan. Their friendship and support made the lab a motivating environment during the period of my work. I am so lucky to have such supportive friends: Vahid Bazargan, Pirooz Darabi, Saghar Mohajeri, Sina Radmard, Amir Rasuli, Behnam Razavi, and Hamidreza Yamini.

Lastly, I wish to express my genuine gratitude to my wonderful family for their never-ending love, support, and guidance. My fathers perfectionism and seek for the truth and my mothers energy, encouragement and moral support

have always been a great motivation for me. I also thank my brother, my durable source of inspiration, for his support in every step of my life.

Dedication

To my family,

an insufficient token of my appreciation of their unwavering love and faithfulness

Chapter 1

Introduction

1.1 Motivation

To achieve satisfactory performance for control systems during their operation, controllers must be designed for any conceivable situation. In different operating conditions, the dynamics of the system varies, and these variations need to be taken into account in high performance controller design. In many servo systems, such as Hard Disk Drives (HDD) and Ball Screw Drives (BSD), it is often the case that the governing dynamic equations do not vary, but there can be variations associated with parameters in the equations.

In this thesis, we assume that *uncertain* and *scheduling* variables cause variations in the parameters of the plant dynamics with fixed structures. In other words, the parameters are functions of these two types of variables. An uncertain variable is a parameter whose value is unknown and unmeasurable during the operations, but its variation is known to be bounded within a specific range. A varying parameter is called a scheduling variable if its value is available in real time while the dynamics varies. Such value can be either measured or estimated, and it can be used for scheduling controllers.

Since parameter variations affect the system dynamics, they need to be taken into consideration in controller design. A well-established theoretical

tool to deal with the variations in plant dynamics is robust control [120]. Using the robust control theory, one can design controllers that guarantee robust stability and performance for a model set with uncertainties. To avoid unnecessary conservatism which is inherent in robust control system design, it is of interest to derive a tight and accurate model set [13, 53, 69].

Moreover, in many applications, the variations of the system dynamics depend on the scheduling variables. The scheduling variables can be used to improve the closed-loop performance by adjusting controller parameters. To effectively address these issues in controller design, adaptive controllers [43, 108] have been utilized. We focus on an adaptive control method using the gain scheduling approach [91]. To successfully design a gain scheduling controller, it is essential to estimate the relations between the parameters of the system dynamics and the scheduling variables.

One servo system that is studied in this thesis is an HDD. One of the most important characteristics determining the quality of an HDD is the areal storage density. It is essential to reduce the tracking error in order to increase the areal density of HDDs. To achieve a desired performance we require a precise model of the system dynamics, based on which the position control of the read/write head is designed [15, 33, 45, 70]. The dynamics can vary due to many factors such as variations in the fabrication environment, temperature change, and mechanical imperfections due to the elapse of time. Such variations are studied in detail in Chapter 4.

Another servo system investigated in this thesis is a BSD. BSDs are mostly used for high precision motion applications, such as in CNC machines and wire bonding. In these applications, the objective is to accurately position the workpiece relative to the tool. The quality of the machining product depends greatly on the tracking performance of the machine over a desired trajectory for the workpiece position. In order to achieve a satisfactory tracking performance during the operations, servo controllers must be designed to take into account any possible situation during the operations.

Therefore, it is critical to derive a model that precisely presents the BSD dynamics, which has variations. These variations occur due to many unavoidable factors such as changes in the BSD table position and mass during operations. Such variations are studied in detail in Chapter 5.

This thesis develops two modeling methodologies to tackle the challenges caused by the variations in the plant dynamics. The effectiveness of the proposed methodologies are investigated using HDD and BSD plants. The dynamics of these two servo systems are also studied in detail.

1.2 Problem definition and methodology

The following is a modeling problem from a controller design point of view. Here, the problem is stated in a very general form.

Problem 1.2.1. *For a given system response data set with the same governing dynamic equation but possibly different parameter values, find a model set, which represents the given data set accurately, in such a way that a satisfactory closed-loop performance can be achieved by designing the corresponding controller set.*

Several special cases of the above problem have been addressed in the literature (See Section 1.4). Here are a number of special cases regarding

- system response data:

  D1. the given system response data is in the frequency domain.

  D2. the given system response data is in the time domain.

- closed-loop performance objective:

  P1. there is no given desired controller objective. However, the model is derived to be tight to avoid unnecessary conservatism inherent to any controller, which leads to the closed-loop performance enhancement.

Table 1.1: Three special cases in this thesis.

	P1	P2
M1	Chapter 2	Future work
M2	–	Chapter 3

P2. a desired controller objective is given, and the model set is derived in such a way that this objective is satisfied for all the given system response data by designing the corresponding controller set.

- the model set:

M1. the derived model set is connected, i.e., not only systems in the given set but also "intermediate" systems are considered.

M2. the derived model set is discrete, i.e., only systems in the given set are considered.

In this thesis, the special cases are addressed as shown in Table 1.1. Two methodologies are developed to address two special cases. One method [97] considers the special cases M1, P1, D1, and D2. This method generates a tight connected model set to avoid unnecessary conservatism. The other method considers the special cases M2, P2, D1, and D2. This method derives a family of discrete model sets, in such a way that a given satisfactory closed-loop performance can be achieved by designing a family of robust controllers, each of which is in charge of one discrete model set. Both of these methods can deal with the time and frequency responses of systems.

This section briefly describes the methodology, provides a simple example, and explains problem formulation for each of these methods deriving connected and discrete model sets. Here, to prevent confusing notations and excessive introductory material, the modeling problems are stated in an informal form. The problems are revisited and reformulated precisely in later

chapters of the thesis.

1.2.1 Connected model set

To understand the variations in the dynamics of a large number of systems with a common structure, we often take a number of samples, study their dynamics, and deduce a model representing not only the samples but also intermediate systems between the sampled ones. Here are two scenarios to clarify these applications, when the variations are caused by scheduling and uncertain variables.

1. Assume that the dynamics of a plant varies by changing the temperature during operations. For the controller design, we need to derive a model representing the plant dynamics for the entire considered range of temperature. To this end, we can take system responses for some temperature samples, and identify the system based on these samples. However, the temperature changes continuously in reality, and therefore, we need to drive a connected model set which approximately interpolates the dynamics of the sampled models. In the case where the temperature is measurable, it is considered as a scheduling variable, and the system is Linear Parameter Varying (LPV).

2. Suppose that we want to design a controller for products of a production line. We take system responses of some product samples to study the dynamics. These plants are considered as Linear Time-Invariant (LTI). Due to some factors, e.g., limited tolerances in the production line, there are variations associated with the dynamics of these samples. It is desired to derive a connected model set to cover these variations and represent the sampled products as well as the unsampled ones. Since we often can not measure nor estimate the sources of variations, we deal with an uncertainty modeling problem.

In both scenarios, we need to model parameter variations. Such modeling is important for designing robust and gain scheduling controllers. A smaller model set generally leads to controllers with less conservative performances. Therefore, it is of interest to derive a set of models with the following characteristics:

- it precisely represents each plant of the given sample set.
- it is a connected set since it is also required to represent the intermediate plants between the sampled ones.
- a small number of independent variables are used to parameterize the set to simplify the model set expression, leading to both nonconservatism and reduced computational complexity in the subsequent controller design.

A simple numerical example is studied to illustrate the advantages of variations modeling in controller design. Assume there are five samples whose dynamics are governed by a common transfer function in the frequency domain as

$$[G(\theta^*)](s) = \frac{b_0}{(s+b_1)(s+b_2)}, \quad \theta^* := [b_0, b_1, b_2]^T, \tag{1.1}$$

and the parameters are

$$\theta^* = \begin{bmatrix} 0.02 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 0.01 & 0 \\ 0 & 1.5 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}, \tag{1.2}$$

where λ_1 and λ_2 are two independent varying terms,

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0.1 \\ 0 \end{bmatrix} \right\}. \tag{1.3}$$

Figure 1.1 shows the Frequency Response Function (FRF) data of these systems.

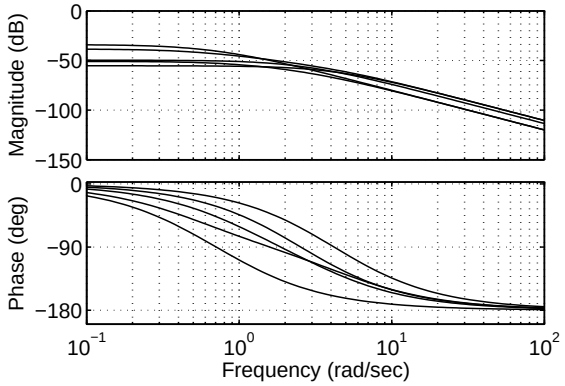


Figure 1.1: Five different dynamical systems with same structure

Now, assume that the FRF data of the samples, depicted in Figure 1.1, are given without any knowledge about the information in (1.1), (1.2), and (1.3). The objective is to derive a connected model set to estimate all samples as well as the unsampled systems. First, we estimate the order of a proper rational transfer function to retrieve the structure in (1.1). The order can be selected by systematic methods such as the Akaike's information criterion [1], inspection of the FRF data, or by trial and error. In this example, the order of the transfer function can be easily estimated by inspection,

$$[G(\theta)](s) = \frac{a_0}{s^2 + a_1 s + a_2}, \quad \theta := \begin{bmatrix} a_0, a_1, a_2 \end{bmatrix}^T. \tag{1.4}$$

Then, using the least-squares method, such as commands `invfreqs.m` or `fitfrd.m` in the Matlab software, the set $\mathbf{\Theta}$ consisting of all estimated parameter vectors is computed as

$$\mathbf{\Theta} := \{\theta_\ell\}_{\ell=1}^{5} = \left\{ \begin{bmatrix} 0.03 \\ 8.5 \\ 17.5 \end{bmatrix}, \begin{bmatrix} 0.03 \\ 5.5 \\ 2.5 \end{bmatrix}, \begin{bmatrix} 0.01 \\ 4.5 \\ 3.5 \end{bmatrix}, \begin{bmatrix} 0.01 \\ 1.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.021 \\ 5.2 \\ 6.4 \end{bmatrix} \right\}. \tag{1.5}$$

Figure 1.2(a) illustrates the parameter set, in which each vector in $\mathbf{\Theta}$ is shown by a dot. In this figure, the 3-dimensional grey box lies in between the minimum and the maximum values of all the parameters. Such a box covers all the parameter variations and therefore, a controller can be designed for the same region. On the other hand, a manifold with fewer dimensions, one or two here, that can approximate all vectors in $\mathbf{\Theta}$ may exist. Subsequently, a controller designed for this manifold can achieve a better performance in comparison with the case that the controller covers the entire grey box.

We have applied Principal Component Analysis (PCA) to find a linear manifold approximating the parameter vectors. The resultant plane is shown in Figure 1.2(b). For this simple example, an exact but unknown 2-dimensional manifold is obtained from (1.1), (1.2) , and (1.4) as

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0.02 \\ 5 \\ 6 \end{bmatrix} + \begin{bmatrix} 0.01 \\ 2 \\ 4 \end{bmatrix} \lambda_1 + \begin{bmatrix} 0 \\ 1.5 \\ 4.5 \end{bmatrix} \lambda_2 + \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix} \lambda_1 \lambda_2. \tag{1.6}$$

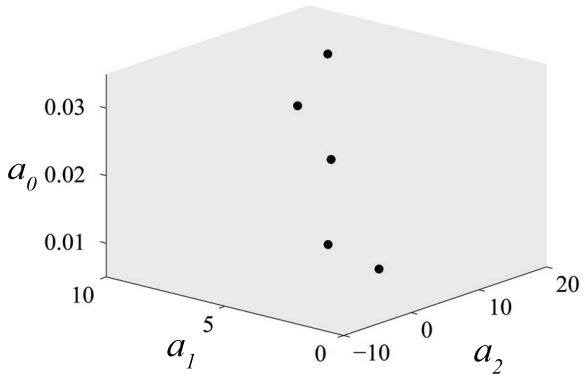
Notice that the manifold in (1.6) is nonlinear in variable terms, and therefore, linear methods, such as PCA, can not estimate the parameter set as precisely as nonlinear ones, such as Nonlinear PCA (NLPCA).

The goal is to detect a manifold, which interpolates the samples to cover the variations associated with not only the samples but also the intermediate plant dynamics. Detecting such a manifold is essential to reduce the conservativeness of the controllers, and hence, to improve the performance.

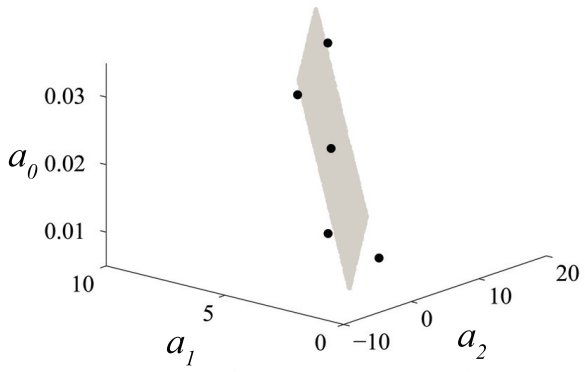
By generalizing the simple example above, we now formulate a problem of constructing a connected model set.

Problem 1.2.2. *Assume a set of transfer functions with a common structure, but possibly different parameter values, is given. The goal is to derive a connected set of transfer functions, which represents each element of the given set accurately. The connected model set should be tight*[1]*, leading to*

[1]For instance, "tight model set" refers to a model set with few dimensions in the



(a) The gray box covers all the variations in a conservative way.



(b) The gray plane shows a manifold approximating the parameter set as calculated by linear PCA.

Figure 1.2: Model parameters (black dots) and approximating manifolds.

nonconservatism in subsequent controller design.

1.2.2 Discrete model set

When large parameter variations exist, it may be infeasible to design a single robust controller to satisfy a desired performance objective for the entire range of variations. One way to overcome this infeasibility is to divide the parameter set into a finite number of partitions, and design one controller for each partition. Since each controller deals with smaller parameter variations, the performance can be enhanced. Generally, the overall performance strongly depends on the characteristics of the partitions, i.e., their size and the way in which partitions are separated. Therefore, it is of interest to divide the parameter set optimally, and derive a family of discrete model sets with the following characteristics:

- it represents each plant of the given sample set by one element.
- a given performance objective can be satisfied for all partitions by designing one controller for each partition.

As a motivating example, let us consider a control problem for a set $\boldsymbol{\mathcal{G}}$ including $L = 12$ mass-spring-damper (MSD) systems with force input and displacement output. The dynamics of such systems can be represented in a normalized form in the frequency domain as

$$G_\ell(s) := \frac{\omega_{n,\ell}^2}{s^2 + 2\zeta_\ell \omega_{n,\ell} s + \omega_{n,\ell}^2}, \ \ell = 1, \ldots, L, \tag{1.7}$$

where ω_n and ζ are modal parameters. The corresponding parameter set $\boldsymbol{\Theta} := \{(\omega_{n,\ell}, \zeta_\ell)\}_{\ell=1}^{L}$ is shown in Figure 1.3.

The control objective is to reject the displacement disturbance for all the systems in $\boldsymbol{\mathcal{G}}$ using feedback control. The disturbance rejection requirement

previous example.

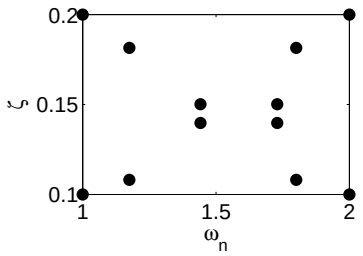


Figure 1.3: Parameter set of (1.7).

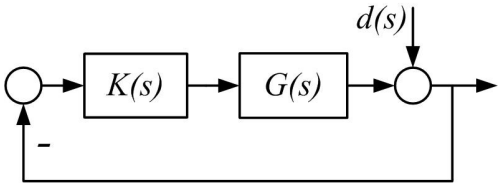


Figure 1.4: Mass-spring-damper closed-loop block diagram.

for a system G and a controller K can be expressed in terms of the sensitivity function as

$$\Gamma_{MSD}(G,K) := \left\| \frac{W}{1+GK} \right\|_\infty < 1, \tag{1.8}$$

for a given weighting function W. In this example, we assume the weighting function is[2]

$$W(s) = \frac{0.2s + 391.9}{s + 0.3919}. \tag{1.9}$$

Figure 1.4 shows the closed-loop block diagram, where d is the displacement disturbance signal.

One straightforward approach to controller design for multiple systems is to design one stabilizing feedback controller K_ℓ for each system G_ℓ. However, this approach can be demanding, especially for a large number of systems, because it leads to time-consuming design of a large number of controllers. Therefore, our goal is to design a small number of controllers, in such a way

[2]The weighting function is a design specification which is given based on the performance requirement [120].

that each system is controlled by one of the designed controllers.

In this example, we want to design the smallest possible number of controllers. First, we try to design a single controller by solving the following optimization problem,

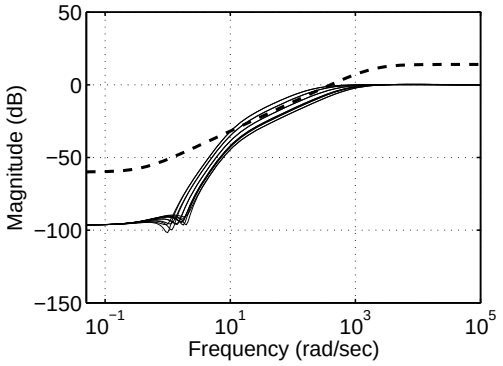
$$\gamma := \min_{K \in \boldsymbol{\mathcal{K}}} \max_{\ell=1,\ldots,L} \Gamma_{MSD}(G_\ell, K), \tag{1.10}$$

where $\boldsymbol{\mathcal{K}}$ shows a set of controllers, which robustly stabilize all G_ℓ. Here, we follow an output-feedback controller design procedure proposed in [50], which is based on the Linear Matrix Inequality (LMI) technique. The optimum γ that is achieved by applying only one controller is 2.172. Since $\gamma > 1$, the disturbance rejection requirement (1.8) is violated for some plants, see Figure 1.5(a)[3].

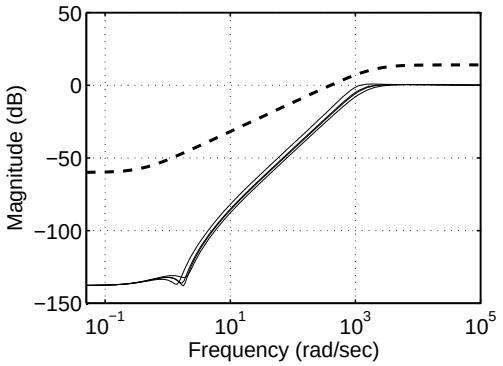
One way to possibly fulfill this requirement is to divide the set $\boldsymbol{\mathcal{G}}$ into two partitions $\boldsymbol{\mathcal{G}}^{(1)}$ and $\boldsymbol{\mathcal{G}}^{(2)}$, where $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{G}}^{(1)} \cup \boldsymbol{\mathcal{G}}^{(2)}$ and $\boldsymbol{\mathcal{G}}^{(1)} \cap \boldsymbol{\mathcal{G}}^{(2)} = \emptyset$. Then, solving the optimization problem (1.10) for each partition, the optimum γ_1 and γ_2 are achieved. The value of $\gamma_{opt} := \max\{\gamma_1, \gamma_2\}$ determines if the control objective is satisfied. Since the dynamics of systems are governed by a common transfer function, partitions in parameter domain $\{\boldsymbol{\Theta}^{(q)}\}_{q=1}^{2}$ are equivalent to $\{\boldsymbol{\mathcal{G}}^{(q)}\}_{q=1}^{2}$, where $\boldsymbol{\Theta}^{(q)}$ includes parameter vectors of systems in $\boldsymbol{\mathcal{G}}^{(q)}$.

Two intuitive ways of partitioning $\boldsymbol{\Theta}$, among many others, are shown in Figures 1.6(a) and 1.6(b). Here, elements of different partitions are shown by different types of markers. The optimum values of γ_{opt} are 1.243 and 1.045 respectively for partition sets shown in Figures 1.6(a) and 1.6(b), and thus, these two partition sets violate the requirement. On the other hand, a partition set shown in Figure 1.6(c) provides $\gamma_{opt} = 0.759 < 1$, see Figure 1.5(b). In fact, this value of γ_{opt} is the global minimum for the case of two partitions. This optimal partition set (Figure 1.6(c)) is derived by "full search",

[3]Based on the requirement (1.8), the magnitude of W^{-1} is compared with those of the sensitivity functions to examine the violation



(a) One controller is designed for all systems, and the performance objective is not satisfied for some systems, $\gamma_{opt} > 1$

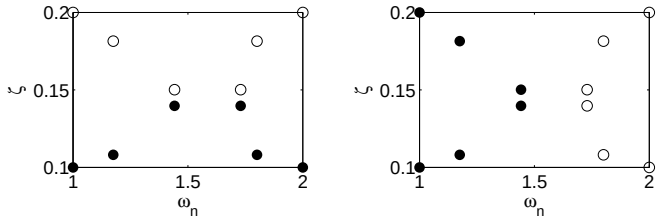


(b) Two controllers are designed for an optimum partition set, and the performance objective is satisfied for all systems, $\gamma_{opt} < 1$

Figure 1.5: Violation or satisfaction of the disturbance rejection requirement. The dashed line shows the magnitude plot of W^{-1} and solid lines show sensitivity functions of the systems

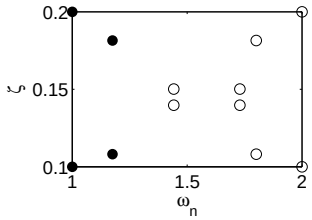
i.e., assessing the closed-loop performance of all 2509 possible partition sets, combinations of twelve points in two partitions.

In many practical applications, the parameter set can be large in terms of dimension (more complex transfer function) and the number of elements (more systems to be controlled). In such cases, the "full search" might be-



(a) A partition set leading to $\gamma_{opt} = 1.242$

(b) A partition set leading to $\gamma_{opt} = 1.045$



(c) The global optimum partition set leading to $\gamma_{opt} = 0.759$

Figure 1.6: Three ways of partitioning.

come impractical even for a small number of partitions. The goal is to obtain an optimum partition set in a systematic manner to satisfy the closed-loop performance objective.

By generalizing the simple example above, we now formulate a problem of constructing a family of discrete model set.

Problem 1.2.3. *Assume a set of transfer functions and a desired closed-loop performance are given. The transfer functions have a common structure with possibly different parameter values. The goal is to divide the given set into the smallest possible number of partitions in such a way that the desired performance objective is satisfied for all partitions by designing one controller for each partition.*

1.3 Objectives of the thesis

Both of the connected and discrete model set identification problems have been previously addressed in the literature. Nevertheless, the solutions proposed previously often suffer from a lack of generality and/or conservatism. Therefore, the solutions to these problem are still being developed by researchers.

Moreover, the dynamics of HDDs and BSDs have been thoroughly studied and reported in the literature. However, there is still a lack of attention to the dynamics of these systems from a controller design point of view.

The objectives of this thesis are to

O1. develop a modeling technique to derive connected model sets for Linear Parameter Varying (LPV) and Linear Time-Invariant (LTI) systems,

O2. develop a systematic algorithm to derive a family of discrete model sets, in such a way that a desired closed-loop performance can be achieved by designing the corresponding controller set,

O3. study the dynamics of HDDs from a controller design point of view,

O4. model BSDs and design a robust gain scheduling controller to precisely control the position of the machine table.

In this thesis, general linear plant dynamics are considered. The systems are general in a sense that they can be single-input single-output or multi-input multi-output, and discrete-time or continuous-time. The proposed methods will be compared to existing methods available and reported in the literature. The developed methods are also numerically and/or experimentally validated, using typical benchmark problems from the control literature, such as mass-spring-damper systems, as well as practical control problems, namely, the hard disk drive and ball screw drive servo systems.

1.4 Literature review

1.4.1 Connected model set

As discussed earlier, we may need to derive a connected model set to cover all the parameter variations. There have been some investigations into parameter variations modeling in various systems to generate connected model sets [11, 14, 52]. Most of the proposed methods assume that how each independent variable affects the parameters in the transfer functions is known [74, 76]. This assumption is reasonable when we know the physical laws governing the system dynamics, e.g., the structures of (1.1) and (1.2) in the motivating example. However, this assumption is not valid in general.

The parameter variation modeling problem that we consider is closely related to the parameter reduction. Hence, we review the literature in the area of parameter reduction, and utilize the related best methodology as the basis for modeling formulation.

Parameter dimension reduction is the transformation of a high-dimensional parameter set into a meaningful representation of reduced dimensionality. This representation has a dimensionality that corresponds to the "intrinsic dimensionality" of the parameter set. The intrinsic dimensionality of the parameter set is the minimum number of parameters needed to account for the observed properties of the model [34] without losing any critical information about the plant. To this end, the possible correlation of original parameters must be identified.

In some applications, parameters with little contribution to the input-output behavior can be ignored in the estimation process [64, 116]. One application of this method of parameter reduction can be found in [24] where PCA and sensitivity analysis were used to reduce the number of parameters in a model representing a complex metabolic network. Sun and Hahn [101] applied three techniques to reduce the parameter set of fundamental models. They extended the described methods to nonlinear systems. However, in

many cases including our application, i.e., high precision controller design, it is generally undesirable to neglect any parameter of the system model.

Dontchev et al. [27] proposed a numerical method for reduction of *a priori* bounds on the values of uncertain plant parameters. The method successively eliminates parts of the parameter domain which are inconsistent with the plant measurement. The goal is to achieve the smallest possible area which contains all the information about the uncertainty of the parameter set. This method is computationally expensive.

A large number of nonlinear techniques for dimensionality reduction have been proposed [7, 26, 41, 61, 105]. These techniques have the ability to deal with complex nonlinear data. One can divide such techniques into supervised and unsupervised learning methods.

Most commonly, supervised learning generates a global model that maps inputs to desired outputs. This means that the data set consists of pairs of input objects and desired outputs. Many well-known techniques for supervised learning methodology are based on the linear discriminant analysis [31]. Since, in our applications, the data set is a parameter set and not a set of input-output data, the supervised learning method is not suitable here. On the other hand, the modeling problem of the parameter variations can be solved by implementing an unsupervised learning technique such as PCA [42].

Nonlinear techniques for parameter reduction can also be divided into two main categories:

- **Local techniques.** These techniques attempts to preserve local properties of the original data set. Four most common methodologies in this category are, Local Linear Embedding (LLE) [90], Laplacian eigenmaps [7], Hessian LLE [26], and local tangent space analysis [119].

  There are two main disadvantages associated with this type: (a) Local properties do not necessarily follow the global structure, as noted in [9, 89], specially in the presence of noise. (b) Since the distribution of

the original data set is not necessarily uniform, the neighbor selection must be adaptive; otherwise the performance of parameter reduction would be degraded.

- **Global techniques.** These techniques attempt to preserve global properties of the original data set. Four well-known global nonlinear techniques are, Multidimensional Scaling (MDS) [20], Isomap [104], Diffusion Map (DM) [58], and NLPCA [55].

  – *MDS* maps the original data to a low-dimensional representation while preserving the distance between the data points in a pairwise fashion. The quality of the MDS is expressed in the stress function

  $$\sum_{i,j}(\|x_i - x_j\| - \|y_i - y_j\|)^2, \tag{1.11}$$

  where $\|x_i - x_j\|$ and $\|y_i - y_j\|$ are the Euclidean distances between the high-dimensional and low-dimensional data points respectively. By modifying its cost function, one may put more emphasis on preserving distances which were originally small, such as Sammon cost function [92].

  Although it is well known that MDS has been successful in many applications, it does not take into consideration the distribution of the neighboring points, and it is based on Euclidean distance. Therefore, MDS for data near to a curved manifold, such as "Swiss roll" data set, does not perform satisfactorily.

  – *Isomap* is a methodology that resolves the drawback of MDS by attempting to retain a curvilinear distance between data points instead of a Euclidean distance. Curvilinear distance is the distance between points measured over the manifold.

  – *DM* first constructs a graph of original data, $x \in X$. The weights of the edges in the graph are computed using a Gaussian kernel

function, leading to a matrix with entries, p_{ij}. This matrix is used to calculate the diffusion distance:

$$D(x_i, x_j) := \sum_k \frac{(p_{ik} - p_{jk})^2}{\psi(x_k)}, \tag{1.12}$$

where $\psi(x_k)$ is a term attributing more weight to part of the graph with high density. The main idea is that the DM is based on many paths through the graph. This makes the DM more robust than, e.g., the methods based on the curvilinear distance, such as the Isomap.

– *NLPCA* is an extension of PCA. Traditional linear PCA is a data analysis technique identifying patterns and expressing the data with independent variables in lower dimensional space. In other words, if vector $\theta \in \mathbb{R}^{n_\theta}$ represents the observations, there is a transformation matrix P to produce independent variables stored in $\lambda \in \mathbb{R}^{n_\lambda}$, $n_\lambda \leq n_\theta$,

$$\lambda = P^T \theta. \tag{1.13}$$

There are some nonlinear extension of PCA that have been proposed over the past two decades. Studies on NLPCA can be divided into the utilization of neural network [54], principal curves [38], and kernel approaches [94, 95].

Although Isomap and DM show a capability of outperforming some other techniques to reduce the dimension of data, in uncertainty modeling aspect they are not so powerful as NLPCA. Therefore, we propose an optimization problem based on the concept of PCA.

Here, we prefer to adopt the principal curves method over neural network and kernel PCA for the following reasons:

- The computational cost of neural network formulations increases considerably if the number of observations (here, the number of elements in

the parameter set) rises. On the other hand, based on some literature such as [25], principal curve computation is not normally subject to this computational concern.

- Kernel PCA performs PCA in a feature space of arbitrarily large (possibly infinite) dimensionality. The size of the kernel matrix increases quadratically with an increase in the number of samples. Also, compared to principal curves, kernel PCA is harder to interpret in the input space [57].

Principal curves presented by Hastie and Stuetzle [37, 38] are smooth one-dimensional curves that pass through the middle[4] of multi-dimensional distributions or data sets. The shape of the principal curve is determined by the structure of the data set, and it provides a nonlinear summary of the data. The principal curve is formally defined to be smooth self-consistent curve for a data set. In other words, any point on the curve is the average of all of the data which project onto that point. Although, in the original definition, the principal curve is defined as a one-dimensional curve, in this study the concept of the principal curve is extended to develop a multi-dimensional curve.

1.4.2 Discrete model set

As discussed above, the variations in the system dynamics increase the size of the model set, and consequently degrade the closed-loop performance, and thus need to be taken into consideration. In many applications, achieving the desired performance of the controller can be demanding due to the size of the model set [120]. There are two main approaches to overcome this challenging problem.

The more common approach solves the problem by modifying the controller synthesis methodology for a given model set. Depending on the essence

[4]The interpretation of "middle" is given in Section 2.3.1.

of the variations in the model set, different controller synthesis methodologies are available, such as the switching ([72, 115]), multiple robust ([18, 117]), adaptive ([32, 88]), and model predictive controllers [29]. The common idea of these methods is to design a number of controllers, such that, each of which covers one part of the variations. However, these methods lead to a more conservative closed-loop system in comparison with the case where a multiple-model set is derived based on the closed-loop performance.

The other approach attempts to derive a multiple-model set. This approach has a history of two generations [63, 65]. The first generation was represented by Magill [73] and Lainiotis [60]. Blom and Bar-Shaloms pioneered an interacting multiple-model algorithm [8] and introduced the second generation. The interacting multiple-model has earned an enviable reputation for multiple-model estimation via a significant number of successful applications (see, e.g., [6]). Since then, different aspects of multiple-model set derivation have been investigated [66, 87].

All the above literature in the multiple-model approach considers switching between local models. To the best of our knowledge, there is no literature on derivation of a family of discrete model sets based on the desired closed-loop performance, as defined in the Problem 3.2.1.

In cases where the system variations are only due to the parameter variations, the parameter set can be divided into a finite number of partitions. Then, one controller is designed for each partition, and since it deals with less parametric variations the performance can be enhanced. Generally, the overall performance strongly depends on the characteristics of the partitions, i.e., their size and the way that they are formed. Therefore, it is of interest to divide the parameter set optimally.

In general, dividing a given parameter set into partitions is based on the "similarity" among the parameters. A comprehensive introduction and survey in clustering can be found in [47]. Classical algorithms optimize simple objectives, for instance K-means minimizes the spread over centroids.

However, due to their simplifying assumptions about partition structure and intuitive interpretations, they may provide poor clustering solutions.

One well-recognized method for clustering is the Normalized Cut (NCut) [99]. However, it involves an optimization with a nonlinear objective and a combinatorial nature of the feasible set, which leads to an NP-hard problem. Spectral relaxation is a legitimate approach with significant practical successes [99] for relaxing the NCut optimization. Nevertheless, Guattery and Miller discussed some valid drawbacks of spectral relaxation [35]. Semidefinite programming (SDP) has been powerful in approximating similar difficult clustering optimization [51, 113], which results in a tighter relaxation of NCut in comparison with the spectral relaxation [122].

In this thesis, a relaxed version of NCut is applied in an algorithm to achieve an optimum set of partitions in such a way that a given performance objective is satisfied for all partitions by designing one controller for each partition. Further details about this approach is provided in Chapter 3.

1.5 The layout of the thesis

This thesis is organized as follows. Chapter 2 is devoted to Objective (O1) of the thesis, presented in Section 1.3. To be more specific, this chapter proposes a method to derive connected model sets. In Chapter 3 an algorithm is developed to achieve Objective (O2). The dynamics of HDD read/write head positioning systems is investigated in Chapter 4, where Objective (O3) is accomplished. The proposed method in Chapter 2 is then used to model the dynamics of HDD systems to design track-following controllers. The variations in the dynamics of BSDs are studied in Chapter 5 to achieve Objective (O4). Three main sources of variations are considered in tracking controller design. Finally, Chapter 6 concludes the thesis, summarizes its major contributions, and provides directions for future research.

Chapter 2

Connected Model Set for a Set of System Response Data [1]

2.1 Introduction

This chapter proposes a systematic method to obtain a connected model set for a given set of system response data. It is assumed that the governing equations of these system are the same but associated with parameter variations. This method achieves Objective (O1) in Section 1.3.

The proposed practical procedure of this method is briefly illustrated in Figure 2.1, and is described with mathematical expressions in this chapter. This algorithm develops a mapping from experimental system response data of a batch of sampled plants to a connected model set. The resultant set is a set of real rational transfer functions, of which parameters are parameterized by a small number of uncorrelated variables that capture the differences in the dynamics of the sampled plants.

The transfer function structure can be selected by systematic methods such as the Akaike's information criterion [1], inspection of the FRF data,

[1] Most of this chapter is based on the following publication: M. Sepasi, F. Sassani, and R. Nagamune, *"Parameter uncertainty modeling using the multi-dimensional principal curves"*, Journal of Dynamic Systems, Measurement, and Control, 2010, Vol. 132, No. 5, pp. 054501-054508.

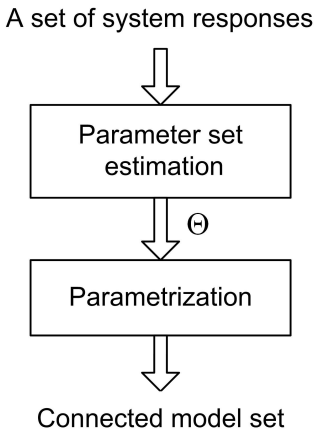


Figure 2.1: Connected model set derivation.

or by trial and error. The corresponding parameter set can be estimated through many powerful tools, such as the System Identification [68] and Signal Processing [67] Toolboxes in the Matlab software. Therefore, the procedure of obtaining the parameter set from the system responses is ignored in this chapter, and in the problem formulation, it is assumed that the parameter set is given.

After deriving a connected parameter set, we generate the corresponding connected model set, which can be simply done since the transfer function structure is available. Therefore, in the problem formulation the goal is to obtain a connected parameter set. An example is explained in Section 2.4.2, in which the connected model set is derived for a general case where we deal with both uncertain and scheduling variables.

This chapter is organized as follows. Section 2.2 formulates a connected model set derivation problem. In Section 2.3, a method is proposed to solve the formulated problem. Section 2.4 provides numerical and practical examples to validate the proposed method.

2.2 Connected set derivation problem

In Section 1.2.1, we have formulated a problem of generating a connected model set in a general term; See Problem 1.2.2. Here, using mathematical notation, we will reformulate it more rigorously.

Problem 2.2.1. *Assume a set of L parameter vectors of transfer functions with a common structure is given as*

$$\boldsymbol{\Theta} := \{\theta_\ell \in \mathbb{R}^{n_\theta}, \ell = 1, \ldots, L\}. \tag{2.1}$$

The goal is to derive a tight connected parameter set

$$\widehat{\boldsymbol{\Theta}}(f) := \{f(\lambda) \in \mathbb{R}^{n_\theta}, \quad \lambda \in \mathbb{R}^{n_\lambda} \ : \ n_\lambda \leq n_\theta\}, \tag{2.2}$$

which represents each element of the given set $\boldsymbol{\Theta}$. The function f is the parameterizing operator, and λ represents uncorrelated variables.

Remark. There are three remarks regarding the above problem;

1. The parameter λ can include uncertain and scheduling independent variables.

2. A small number of uncorrelated variables λ are used to parameterize the set $\boldsymbol{\Theta}$ to obtain a tight model set, leading to both nonconservatism and reduced computational complexity in subsequent controller design.

3. The number of independent variables λ as well as the structure of the function f are assumed to be known. However, if they are unknown, they can be obtained by trial and error. See the example explained in Section 2.4.1.

2.3 Connected set synthesis algorithm

The goal of this algorithm is to derive the set $\widehat{\mathbf{\Theta}}$ in (2.2), which interpolates L members of $\mathbf{\Theta}$ in (2.1), and represent the intermediate plants. Also, it is desired that the resultant set $\widehat{\mathbf{\Theta}}$ has the smallest possible size[2]. To this end, possible correlations among parameter vectors are determined by parametrization with a small number of uncorrelated variables.

In the following, first we explain the synthesis algorithm for a general function f. Then, in Section 2.3.2, a special case of this function is studied.

2.3.1 General structure of the parameterizing function

Parameters λ represent the sources of variations, which can be scheduling variables (e.g. temperature), denoted by λ^s

$$\mathbf{\Lambda}^s := \{\lambda^s \in \mathbb{R}^{n_s}\}, \tag{2.3}$$

or uncertain variables (e.g. product differences), denoted by λ^u which are normalized[3],

$$\mathbf{\Lambda}^u := \{\lambda^u \in \mathbb{R}^{n_u} : \lambda^u \in [-1, 1]\}, \tag{2.4}$$

or a combination of both.

Here, we develop the methodology based on the principal curves for the most general case, which takes into account both scheduling and uncertain

[2]Our intention of "the size of the parameter set" was clarified through an example in Section 1.2.1. In different problems, the word "size" can refer to the dimension, area, etc.

[3]It is common in robust controller design methods to normalize the uncertain variables. Therefore, we normalize the uncertain variables λ^u, contrary to the scheduling variables λ^s.

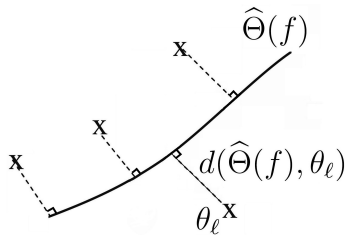


Figure 2.2: Visual explanations of distance notation.

variables

$$\mathbf{\Lambda} := \mathbf{\Lambda}^s \oplus \mathbf{\Lambda}^u = \{\lambda \in \mathbb{R}^{n_\lambda} : n_\lambda = n_s + n_u\}. \tag{2.5}$$

The distance between $\widehat{\mathbf{\Theta}}$ and a sample parameter vector θ_ℓ in $\mathbf{\Theta}$ is defined by

$$d(\widehat{\mathbf{\Theta}}(f), \theta_\ell) := \min_\lambda \|f(\lambda) - \theta_\ell\|, \tag{2.6}$$

with a proper norm $\|\cdot\|$. See the visual explanations of the notation in Figure 2.2. Such distance is desired to be minimized for all L samples. The function f which yields the minimum distance is obtained by a minimax optimization problem

$$\min_{f \in \mathcal{F}} \max_{\ell=1,\dots,L} d(\widehat{\mathbf{\Theta}}(f), \theta_\ell), \tag{2.7}$$

where $\mathcal{F}$ represents the given class of functions.

Remark. Here, the accuracy of the model is equally important for all samples. For example, instead of optimization (2.7), if we solve

$$\min_f \sum_{\ell=1}^{L} d(\widehat{\mathbf{\Theta}}(f), \theta_\ell) \tag{2.8}$$

to derive the function f, we may face a situation that the optimum f^* provides a model that has a large error for one sample θ_ℓ. Since the error for

other samples are small, the cost function of the optimization problem (2.8) is minimized. However, this result may not be acceptable from system modeling point of view. Therefore, we prefer the minimax optimization formulation rather than minimization of other cost functions, such as mean absolute error and mean square error.

Since the optimization problem (2.7) is nonconvex in general, we propose an iterative procedure to solve it as follows.

Algorithm 2.3.1.

Inputs: The set $\mathbf{\Theta} = \{\theta_\ell \in \mathbb{R}^{n_\theta}; \ell = 1, \ldots, L\}$*, the structure of the function* f*, and values of* $\{\lambda_\ell^s\}_{\ell=1}^L$*,* n_u *and* n_s*.*

1. *Initialize the function* f*.*

2. *Discretize the set* $\mathbf{\Lambda}^u$ *by selecting* N *values for* λ^u*. Then, solve the optimization (2.6) for discretized* λ_n^u

$$\min_{\lambda_n^u} \|f(\begin{bmatrix} \lambda_n^u \\ \lambda_\ell^s \end{bmatrix}) - \theta_\ell\|, \text{ for } n = 1, \ldots, N, \tag{2.9}$$

   for each ℓ *and denote the solution by* $\lambda_\ell^{u^*}$*.*

3. *Solve the optimization (2.7)*

$$\min_f \max_{\ell=1,\ldots,L} \|f(\begin{bmatrix} \lambda_\ell^{u^*} \\ \lambda_\ell^s \end{bmatrix}) - \theta_\ell\| \tag{2.10}$$

   to update the function f*.*

4. *Iterate Steps 2 and Step 3 until change in optimal values between iterations is small or a satisfactory error is achieved.*

The initialization of the function f in the first step of the above algorithm can be done by any suitable algorithm. We suggest to use linear PCA, which provides the principal directions of correlation between the parameters.

Depending on the structure of the function f, the third step can be a convex or nonconvex optimization problem.

2.3.2 Special structure of the parameterizing function

In some cases, the nonlinear function f can be chosen such that it can be decomposed into two multiplicative matrices of the form $f(\lambda) = CV(\lambda)$. Matrices C and V are the coefficient and the variable matrices, respectively. The details of this decomposition will be explained in a subsequent section through mathematical expressions.

There are two main advantages of selecting a function f such that the varying terms can be extracted by the above decomposition.

- Varying terms can be extracted from the system and the model can simply be transformed into a Linear Fractional Transformation (LFT) form shown in Figure 2.3, where K is the controller, G is the linear varying system, Λ is a matrix representing the variations, and P is the corresponding linear invariant system. For the LFT form, there are many robust control techniques available.

- The third step of the Algorithm 2.3.1 becomes a convex optimization problem

$$\min_{C} \max_{\ell=1,\ldots,L} \|CV(\begin{bmatrix} \lambda_\ell^{u^*} \\ \lambda_\ell^{s} \end{bmatrix}) - \theta_\ell\|, \tag{2.11}$$

  and can be solved by well-developed techniques, such as LMI.

As an example, assume that the function $f : \mathbb{R}^{n_\lambda} \longmapsto \mathbb{R}^{n_\theta}$ is in the form of

$$f(\lambda) = \begin{bmatrix} 0.02 \\ 5 \\ 6 \end{bmatrix} + \begin{bmatrix} 0.01 \\ 2 \\ 4 \end{bmatrix} \lambda_1 + \begin{bmatrix} 0 \\ 1.5 \\ 4.5 \end{bmatrix} \lambda_2 + \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix} \lambda_1\lambda_2,$$

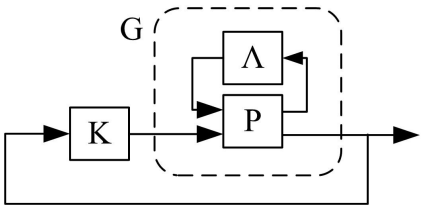


Figure 2.3: A closed-loop configuration for the special case

where $n_\lambda = 2$ and $n_\theta = 3$. Then, the matrices $C \in \mathbb{R}^{3\times 6}$ and $V \in \mathbb{R}^{6\times 1}$ can be decomposed as

$$C = \begin{bmatrix} 0.02 & 0.01 & 0 & 0 & 0 & 0 \\ 5 & 2 & 1.5 & 0 & 0 & 0 \\ 6 & 4 & 4.5 & 0 & 0 & 3 \end{bmatrix}, \quad V(\lambda) = \begin{bmatrix} 1 \\ \lambda_1 \\ \lambda_2 \\ \lambda_1^2 \\ \lambda_2^2 \\ \lambda_1\lambda_2 \end{bmatrix}.$$

In general, the variable matrix $V(\lambda)$ is in the form of

$$V(\lambda) := \begin{bmatrix} v_0(\lambda) \\ v_1(\lambda) \\ \vdots \\ v_m(\lambda) \end{bmatrix}, \tag{2.12}$$

where m is the order of the polynomial function f, and v_i is the *Veronese map* of degree i [36]. A veronese map $v_i : \mathbb{R}^{n_\lambda} \longmapsto \mathbb{R}^{J_i(n_\lambda)}$ is a map from $[\lambda_1, \cdots, \lambda_{n_\lambda}]$ to all $J_i(n_\lambda)$ monomials of degree i in $\lambda_1, \ldots, \lambda_{n_\lambda}$. In the above

example,

$$
\begin{aligned}
v_0 &= 1, & J_0(2) &= 1,\\
v_1(\lambda) &= \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}, & J_1(2) &= 2,\\
v_2(\lambda) &= \begin{bmatrix} \lambda_1^2 \\ \lambda_2^2 \\ \lambda_1\lambda_2 \end{bmatrix}, & J_2(2) &= 3.
\end{aligned}
$$

The size of the general matrix C is $n_\theta \times p$, where $p := \sum_{i=0}^{m} J_i(n_\lambda)$.

The special case, $m = 0$, corresponds to a single transfer function. Parameter m expresses the order of the uncertainty model. One may achieve a more accurate model by increasing the polynomial order, m. Notice that the parameter m assigns the size of the V matrix and consequently the size of the Λ block in Figure 2.3. Therefore, increasing m leads to a higher computational cost in controller design and possibly to an unacceptable conservative performance.

We would like to obtain a coefficient matrix C such that the connected set $\widehat{\mathbf{\Theta}}$ satisfactorily interpolates all the elements of $\mathbf{\Theta}$. For this purpose, we follow the Algorithm 2.3.1 with the special convex form in the third step shown in (2.11).

2.4 Examples

In this section, the developed algorithm is validated through numerical and practical examples. First, we study the results of our method applied to the motivating example explained in Section 1.2.1. Then, the uncertainty associated with a multi-input multi-output BSD is modeled using the proposed method.

2.4.1 Connected model set for a numerical example

We show the performance of the proposed algorithm by applying to the example explained in Section 1.2.1. Let us assume that the variations of the FRF data (Figure 1.1) are caused by only uncertain variables. The goal is to model these uncertainties. For this purpose, we employ a polynomial function f in the formulation (2.2).

First, we apply linear PCA in order to model the uncertainties. Linear PCA is one possible method in dealing with parameter variations for robust control design [56, 75, 76]. To evaluate the final resultant connected set $\widehat{\mathbf{\Theta}}$, we study how precisely it approximates the original parameter vectors. To this end, we obtain the nearest point in this connected set to each of the original parameter vector. The resultant parameter vectors are

$$\widehat{\mathbf{\Theta}}_{PCA} = \{\hat{\theta}_\ell\}_{\ell=1}^{L} = \left\{ \begin{bmatrix} 0.03 \\ 8.701 \\ 17.061 \end{bmatrix}, \begin{bmatrix} 0.028 \\ 5.523 \\ 2.222 \end{bmatrix}, \begin{bmatrix} 0.018 \\ 4.350 \\ 4.185 \end{bmatrix}, \begin{bmatrix} 0.009 \\ 2.166 \\ 0.168 \end{bmatrix}, \begin{bmatrix} 0.021 \\ 5.215 \\ 6.388 \end{bmatrix} \right\}.$$

By comparing this resultant parameter set with the original one shown in Section 1.2.1, Equation (1.5), it can be seen that there are some deviations which might not be acceptable for many applications. Therefore, one may want to derive a nonlinear uncertainty model rather than a linear one in order to cover the variations of the parameter set $\mathbf{\Theta}$ more accurately.

By applying the proposed method, we derive different nonlinear uncertainty models for different values of n_λ, number of uncertain terms, and m, the order of the polynomial function. To compare the results quantitatively, the error is defined as

$$\epsilon := \sum_{\ell=1}^{L} \|\theta_\ell - \widehat{\theta}_\ell\|, \tag{2.13}$$

where θ_ℓ is the original parameter vector from (1.5), and $\hat{\theta}_\ell$ is the nearest point in the connected set $\widehat{\mathbf{\Theta}}$ to θ_ℓ.

Table 2.1: Errors for different models

	$n_\lambda = 1$	$n_\lambda = 2$
$m = 1$	1	0.12
$m = 2$	0.9	9×10^{-9}

Error values ϵ for different models are shown in Table 2.1, where the error of the linear PCA is normalized to 1. From (1.6), we know that the actual values are $n_\lambda = 2$ and $m = 2$, which concur with the results shown in the table.

2.4.2 Uncertainty modeling of a dual-input dual-output Ball Screw Drive (BSD)

As a practical example, we consider to model the uncertainties associated with a machine tool with a ball screw drive. A picture of this machine is illustrated in Figure 5.1. This machine is a dual-input dual-output system. A motor controls the position of a table along a shaft. For different locations of the table, the mass distribution of the system varies, and as a result, the characteristics of the plant change during the operation.

It is assumed that this dual-input dual-output plant is governed by the following structure:

$$G(s) = \begin{bmatrix} M_1(s) & M_2(s) \\ M_2(s) & M_3(s) \end{bmatrix}.$$

To control the position of the table for all locations along the shaft, one way is to build a continuous uncertainty model such that it covers all the variations. For this purpose, we take some sample FRF data illustrating the system characteristics for different locations of the table,

$$\{(\omega_w, \widehat{M}_{iw\ell}) : \quad \omega_w \in \mathbb{R}, \quad \widehat{M}_{iw\ell} \in \mathbb{C}, \quad i = 1, 2, 3,\ w = 1, \ldots, W,\ \ell = 1, \ldots, L\},$$

where $\widehat{M}_{iw\ell}$ represents the system response of $M_i(s)$ at the ℓ-th location at

the frequency of ω_w, and W is the number of frequencies.

Figure 2.4 shows the dynamic responses (dotted lines) of the plant for three different table locations. For the given FRF data, no *a priori* knowledge about the system transfer functions M is assumed except being time-invariant and linear. Therefore, we estimate the general rational form of the transfer functions whose orders are assumed to be identical for all the samples, $\{G_\ell(s)\}_{\ell=1}^{L}$. This implies that the variations shown in Figure 2.4 are due to parameter variations only.

The first step is to obtain the order of the numerator and denominator polynomials of the transfer functions M. The results are as follows: $M_1(s)$ has 2 zeros and 4 poles, $M_2(s)$ has 1 zeros and 5 poles, and $M_3(s)$ has 3 zeros and 5 poles, which are selected by inspecting the shapes of the FRF data.

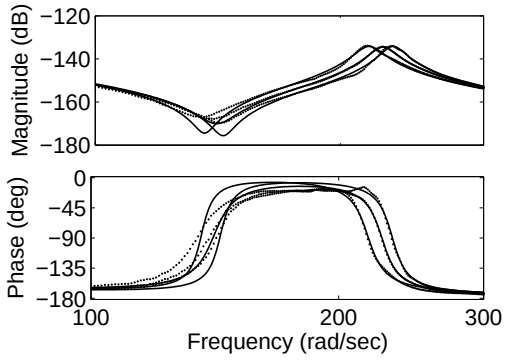
Now, the least squares optimization is used to estimate the parameter sets $\mathbf{\Theta}_i$ by minimizing the error as follows for $i = 1, 2, 3$.

$$\mathbf{\Theta}_i := \{\theta_\ell = \arg\min_{\theta} \sum_{w=1}^{W} |\widehat{M}_{iw\ell} - [M_i(\theta)](j\omega_w)|^2, \quad \ell = 1, \ldots, L\}, \qquad (2.14)$$

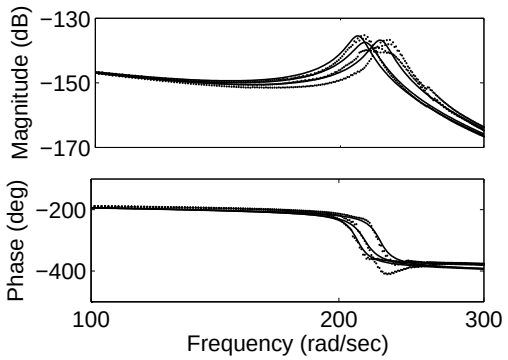
where W is the number of frequencies for the FRF data. The responses of the transfer functions corresponding to these parameter sets are shown with solid lines in Figure 2.4.

Then, we define the parameter set $\mathbf{\Theta} := \{\mathbf{\Theta}_1 \oplus \mathbf{\Theta}_2 \oplus \mathbf{\Theta}_3\}$, and model the uncertainty of this parameter set. First, we apply linear PCA ($m = 1$) with one uncertain variable ($n_\lambda = n_u = 1$). The nearest responses in the connected set to the original ones are shown by dashed lines in Figure 2.5. As can be seen in the figure, this model captures the main features but it is not capable to capture the entire ranges of variations. This is concluded because the dashed lines lie in between the solid lines.

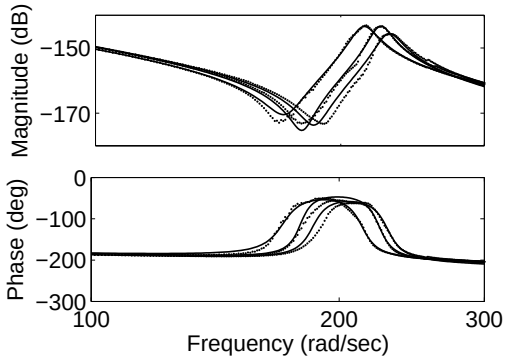
By increasing either n_λ or m, the model captures the variations more satisfactorily. The numerical comparison is made in Table 2.2. Similar to Table 2.1, to evaluate the resultant uncertainty models, we compare the



(a) M_1



(b) M_2



(c) M_3

Figure 2.4: Frequency responses of the ball screw machine (dotted line), and the estimated transfer function (solid line).

Table 2.2: Ball screw uncertainty model normalized errors for different values of n_λ and m

	$n_\lambda = 1$	$n_\lambda = 2$	$n_\lambda = 3$
$m = 1$	1	2.3×10^{-9}	7.6×10^{-9}
$m = 2$	0.15	2.6×10^{-11}	2×10^{-15}
$m = 3$	0.12	1×10^{-15}	1.5×10^{-15}

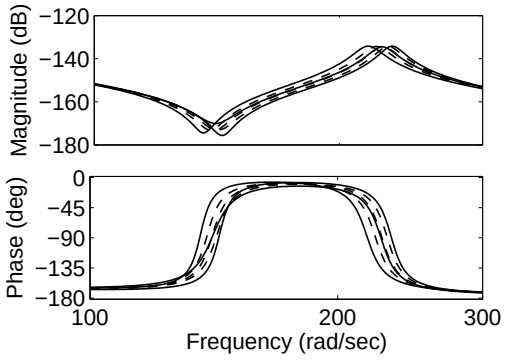
errors defined in (2.13) between the parameter sets. The error of the linear PCA is normalized to unity. Depending on the complexity and accuracy requirements in a particular application, the values of n_λ and m should be selected accordingly.

As can be seen in Table 2.2, if the linear PCA is applied, $m = 1$, we are not able to achieve the normalized error less than the order of 10^{-9}, and by increasing the polynomial order to $m = 2$, a significant improvement might be accomplished. For instance, in the case of $n_\lambda = 3$, the error decreases by a factor of 10^6. The resultant frequency responses for $n_\lambda \geq 2$ almost overlap the estimated dynamics responses.

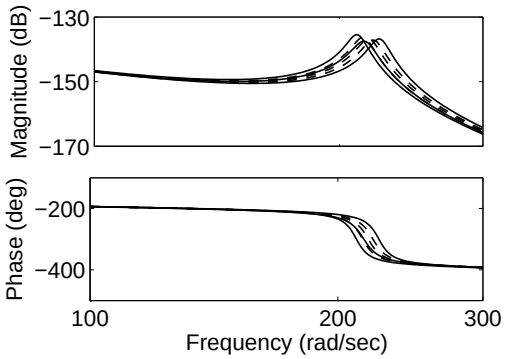
2.5 Conclusions

This chapter proposed a systematic method to derive a connected set for a given set of system response data associated with parameter variations. This method achieved Objective (O1) in Section 1.3. The algorithm illustrated in Figure 1.1 was described with mathematical expressions in this chapter. This algorithm developed a mapping from experimental system response data of a batch of sampled plants to a connected set. The resultant set was a set of real rational transfer functions, of which parameters were parameterized by a small number of uncorrelated variables that captured the differences in the dynamics among the sampled plants.

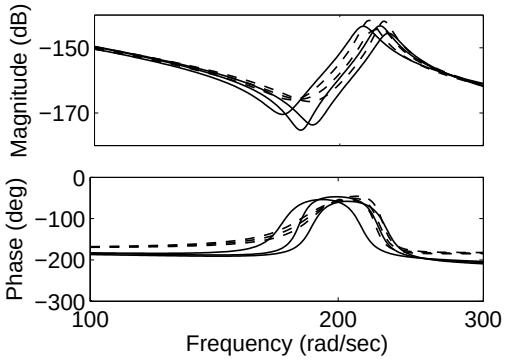
The method was based on a minimax optimization problem for the multi-dimensional principal curves. We applied our algorithm to an illustrative and



(a) M_1



(b) M_2



(c) M_3

Figure 2.5: The estimated transfer function (solid line), and the model set calculated by applying linear PCA (dashed line). The results for $n_\lambda \geq 2$ almost overlap the solid lines.

a practical example, and obtained accurate models in both cases, and hence circumvented the possibility of conservative performance of the closed-loop control systems.

Chapter 3

Family of Discrete Model Sets for a Set of System Response Data [1]

3.1 Introduction

This chapter addresses Objective (O3) in Section 1.3. The algorithm that we propose to achieve this objective is briefly illustrated in Figure 3.1, and is described in detail with mathematical expressions in this chapter. This algorithm develops a mapping from the system response data to a family of discrete model sets. It is assumed that the governing equations of the given system responses are the same but associated with parameter variations. The goal is to divide the given set of system responses into the smallest possible number of partitions, and generate a family of discrete model sets, in such a way that the given performance objective is satisfied for all partitions by designing one controller for each partition.

The common transfer function structure for the system responses can be selected by systematic methods such as the Akaike's information criterion

[1] This chapter is based on the following articles which is under preparation: D. Sepasi, R. Nagamune and F. Sassani, *"Performance-oriented multiple model set estimation using normalized cut"*

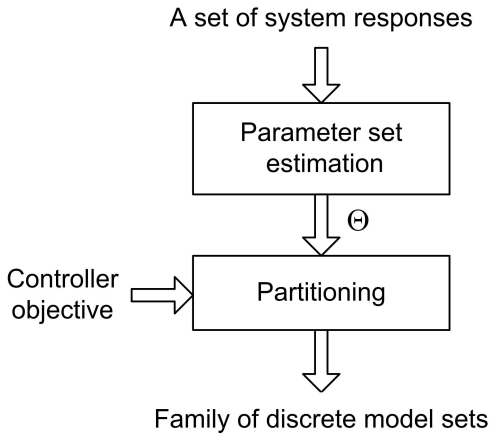


Figure 3.1: Family of discrete model set derivation.

[1], inspection of the system responses, or by trial and error. The corresponding parameter set can be estimated through many powerful tools, such as the System Identification [68] and Signal Processing [67] Toolboxes in the Matlab software. Therefore, these two procedures, i.e., deriving the transfer function set from the system responses and obtaining the parameter set, are not discussed in this chapter.

This chapter is organized as follows. Section 3.2 formulates a problem of deriving a family of discrete model sets. In Section 3.3, a method is proposed to solve the formulated problem. Section 3.4 provides numerical examples to validate the proposed method as well as the effectiveness of combining the method developed in this chapter with that in Chapter 2.

3.2 Problem of deriving a family of discrete model sets

In Section 1.2.2, we have formulated a problem of generating a connected model set in a general term; See Problem 1.2.3. Here, using mathematical notation, we will reformulate it more rigorously.

Problem 3.2.1. *Assume a set of L transfer functions and a desired closed-loop performance are given. The transfer functions have a common structure with possibly different parameter vectors,*

$$\boldsymbol{\mathcal{G}} := \{[G(\theta)](s), \quad \theta \in \boldsymbol{\Theta}\}, \tag{3.1}$$

where

$$\boldsymbol{\Theta} := \{\theta_\ell \in \mathbb{R}^{n_\theta}, \ell = 1, \ldots, L\} \tag{3.2}$$

is the related parameter set. The goal is to divide the set (3.1) into the smallest possible number of partitions, Q, and derive a partition set,

$$\widehat{\boldsymbol{\mathcal{G}}} := \{\boldsymbol{\mathcal{G}}^{(q)}\}_{q=1}^{Q}, \ \boldsymbol{\mathcal{G}}^{(q)} := \{[G(\theta)](s) : \theta \in \boldsymbol{\Theta}^{(q)}\}, \tag{3.3}$$

such that a certain closed-loop performance is satisfied by designing one controller $K^{(q)}$ for each partition $\boldsymbol{\mathcal{G}}^{(q)}$.

Remark. There are two remarks regarding the above problem.

1. For simplicity, the problem is formulated in a continuous-time setting. However, the discrete-time case can be treated analogously.

2. The resultant pairs of $\{\boldsymbol{\mathcal{G}}^{(q)}, K^{(q)}\}$ may be used for switching systems. However, we do not investigate potential issues related to switching between the controllers.

3.3 Synthesis of a family of discrete model sets

In this section, we explain the main idea of how to tackle Problem 3.2.1 and derive the optimum partition set[2]. The proposed algorithm is summarized as a flowchart in Figure 3.2. Each step is briefly addressed first, and then, detailed explanations are provided in the following sections.

[2]Although this partition set is called optimum, it may not be the global optimum.

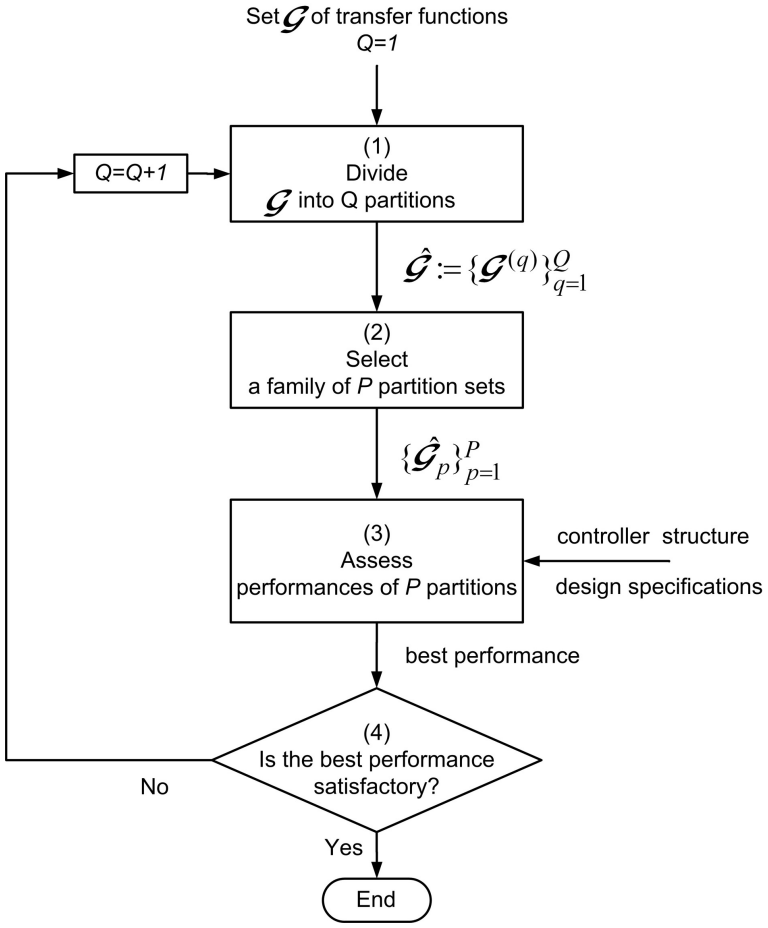


Figure 3.2: Main idea of the procedure to seek the optimum partition set.

(1,2) In the flowchart, there are two distinct steps labeled as (1) and (2). However, the developed algorithm performs these two steps simultaneously as follows. For the given transfer function set, we employ a procedure, which is explained in Section 3.3.2, to obtain a family of P

"best"[3] partition sets,

$$\{\widehat{\boldsymbol{\mathcal{G}}}_p\}_{p=1}^{P} = \{\widehat{\boldsymbol{\mathcal{G}}}_1, \ldots, \widehat{\boldsymbol{\mathcal{G}}}_P\}. \tag{3.4}$$

Note that each of these sets of partitions, $\widehat{\boldsymbol{\mathcal{G}}}_p$, consists of Q partitions as shown in (3.3), $\widehat{\boldsymbol{\mathcal{G}}}_p = \{\boldsymbol{\mathcal{G}}_p^{(1)}, \ldots, \boldsymbol{\mathcal{G}}_p^{(Q)}\}$.

(3) The closed-loop performances of these P partition sets are assessed as explained in Section 3.3.3. Therefore, the partition set, which provides the best performance, is obtained.

(4) If the best performance in Step (3) is satisfactory, we select the corresponding partition set as the optimum one. Otherwise, we go back to step (1) with $Q = Q + 1$.

Since the dynamics of all systems are governed by a common transfer function, partitions in the model set i.e., $\{\boldsymbol{\mathcal{G}}^{(q)}\}_{q=1}^{Q}$, are equivalent to the partitions in the parameter domain, i.e., $\{\boldsymbol{\Theta}^{(q)}\}_{q=1}^{Q}$, where $\boldsymbol{\Theta}^{(q)}$ includes parameter vectors of the systems in $\boldsymbol{\mathcal{G}}^{(q)}$. Hence, we divide the parameter set $\boldsymbol{\Theta}$ into Q partitions $\{\boldsymbol{\Theta}^{(q)}\}_{q=1}^{Q}$ such that there exists a controller $K^{(q)} \in \boldsymbol{K}$ to satisfy the performance objective of the systems in partition q, where $\boldsymbol{K}$ represents a class of stabilizing feedback controllers.

In this section, first, we provide some background material, which is necessary before explaining the main algorithm. A procedure, which provides the family of P best partition sets, is explained in Section 3.3.2. Obtaining the optimum partition set, which leads to the best closed-loop performance is described in Section 3.3.3. The entire procedure is summarized as an algorithm in Section 3.3.4.

[3]Our meaning of the word "best", and how to select the value of P will be explained in detail in Section 3.3.2.

3.3.1 Background material

Notations used in this section are standard. If each element of the vector d is shown by a nonnegative number d_i, the vector $d^{1/2}$ is defined as a vector with elements of $d_i^{1/2}$.

Definition: The nonempty sets $\mathbf{\Theta}^{(1)}, \ldots, \mathbf{\Theta}^{(Q)}$ form *partitions* of the data points $\mathbf{\Theta}$ if $\mathbf{\Theta}^{(i)} \cap \mathbf{\Theta}^{(j)} = \emptyset$ and $\mathbf{\Theta}^{(1)} \cup \ldots \cup \mathbf{\Theta}^{(Q)} = \mathbf{\Theta}$.

Given a set of data points $\mathbf{\Theta}$, the *similarity graph* S models the data set in the form of a graph for which the vertices represent the data points. If the *similarity* s_{ij} between two data points, θ_i and θ_j, is greater than a certain threshold, the corresponding two vertices are connected in the graph by an edge with the weight of s_{ij}. For a visual explanation see Figure 3.3, where data points and edges are shown by dots and lines, respectively. For notational brevity, only weight s_{12} is labeled.

The similarity s_{ij} represents the local neighborhood relationships between θ_i and θ_j. Generally, the higher value s_{ij} holds, the more similar the corresponding points are. The goal of clustering is to find partitions of the graph S such that the edges within the partitions have high weights while the edges between the partitions have small weights.

Three mostly used similarity graphs are neighborhood [48], k-nearest neighbor [30], and fully connected [80] graphs. A comprehensive survey on these graphs is given in [110], based on which, the fully connected similarity graph with Gaussian similarity weights is employed in our algorithm.

Definition: The *affinity matrix* $A = (a_{ij})_{i,j=1,\ldots,L}$ is defined as

$$\begin{aligned} a_{ij} = s_{ij} &:= e^{-\frac{\|\theta_i - \theta_j\|^2}{2\sigma^2}} \quad \text{if} \quad i \neq j, \\ a_{ii} &= 0, \end{aligned} \tag{3.5}$$

with a proper norm $\|\cdot\|$.

The constant σ governs all the similarities throughout the entire data set, and regulates the width of the neighborhoods in the Gaussian similarity

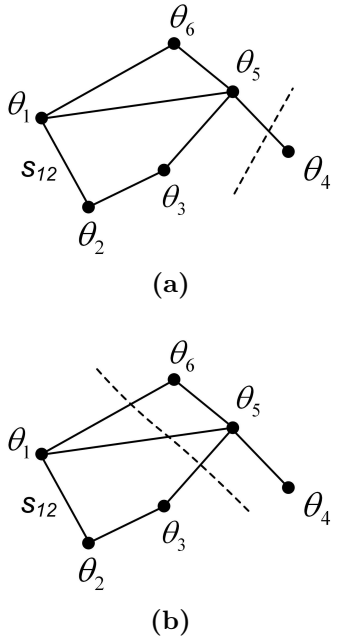


Figure 3.3: Two ways of partitioning a graph. Vertices and edges are shown by dots and solid lines, respectively. The dashed lines are partition boundaries.

function. The parameter σ is in the order of the average distance of a point to its r-th nearest neighbors, where r is chosen to be the nearest integer to $log(L) + 1$ [10].

Definition: The *degree* of a vertex θ_i is defined as $d_i := \sum_{j=1}^{L} a_{ij}$, and the *degree matrix* D is a diagonal $L \times L$ matrix with $\{d_1, \ldots, d_L\}$ on the diagonal.

3.3.2 Parameter set partitioning and derivation of the best partition sets

This section describes Steps (1) and (2) in Figure 3.2 in detail. These steps attempt to minimize a cost function by dividing the estimated parameter set $\mathbf{\Theta}$ into Q partitions. The outputs of this step are P partition sets, which

provide the smallest cost function values as will be explained later.

An intuitive objective of data clustering is to partition the data in such a way that the sum of the pairwise similarities of the points from different partitions is minimized. However, such a naive objective may yield a trivial solution. For example, some partitions may include only one point as shown in Figure 3.3(a).

The Normalized Cut (NCut) methodology normalizes the previous naive cost function, which was the sum of the pairwise similarities of the points from different partitions, with the total weighted degree of the points in each partition. Consequently, partitions such as the one illustrated in Figure 3.3(b) have higher chance to be derived in comparison with the case that a naive objective was considered.

Definition: The *indicator matrix* is defined as $\hat{Y} \in \mathbb{R}^{L \times Q}$ with columns of $\hat{y}_j := [\hat{y}_{1j}, \hat{y}_{2j}, \ldots, \hat{y}_{Lj}]^T$, where

$$\hat{y}_{ij} = \begin{cases} 1: \text{ if } i \in \boldsymbol{\Theta}_j \\ 0: \text{ if } i \notin \boldsymbol{\Theta}_j \end{cases}. \tag{3.6}$$

For example, for the partition sets shown in Figure 3.3, we have

$$\text{For (a): } \hat{Y} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad \text{For (b): } \hat{Y} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}. \tag{3.7}$$

The weight of the NCut can be expressed as [113]

$$W_{NCut}(\hat{Y}) := \sum_j \frac{\hat{y}_j^T (D - A)\hat{y}_j}{\hat{y}_j^T D \hat{y}_j} = Tr[(\hat{Y}^T D \hat{Y})^{-1}(\hat{Y}^T (D - A)\hat{Y})]. \tag{3.8}$$

Consequently, the clustering problem can be defined as the following optimization problem

$$\min_{\hat{Y}} \; W_{NCut}(\hat{Y}). \tag{3.9}$$

It is an optimization problem with a nonlinear objective function, shown in (3.8), and a combinatorial nature of the feasible set $\hat{Y}$, which leads to an NP-hard problem.

One way to relax the NCut optimization problem (3.9) is to define an indicator matrix $Y \in \mathbb{R}^{L\times Q}$ which is the "normalized" $\hat{Y}$ by an unknown factor, such that $Y = D^{1/2}\hat{Y}(\hat{Y}^T D\hat{Y})^{-1/2}$ [113]. If $Z := YY^T$, the optimization (3.9) can be expressed in the relaxed version as (The proof is provided in Appendix A)

$$\begin{aligned} \max_{Z} \; & \mathrm{Tr}(WZ) \\ & Zd^{1/2} = d^{1/2}, \mathrm{Tr}(Z) = Q, \\ & Z \geq 0, I \succeq Z \succeq 0, \end{aligned} \tag{3.10}$$

where I is the identity matrix with the right size, and

$$W := D^{-1/2}AD^{-1/2}, \tag{3.11}$$

where D and A are the degree and affinity matrices, respectively. The constraint $Zd^{1/2} = d^{1/2}$ indicates $\Sigma U^T d^{1/2} = U^T d^{1/2}$ where Σ and U are derived from eigenvalue decomposition, $Z = U\Sigma U^T$. Hence, the i-th eigenvalue Σ_{ii} equals to one when $(U^T d^{1/2})_i \neq 0$, and can be arbitrary when $(U^T d^{1/2})_i = 0$. Since at the optimal solution $Z^* = YY^T$, the eigenvalues of Z^* corresponding to Y are 1, and the indicator matrix Y is the Q eigenvectors in U corresponding to the Q largest eigenvalues in Σ.

In order to introduce a standard Semidefinite Programming (SDP), which

is equivalent to optimization (3.10), we define

$$\widetilde{Z} := \begin{bmatrix} Z & \\ & 1-Z \end{bmatrix}. \tag{3.12}$$

As discussed above, to obtain the indicator matrix Y, the eigenvectors of Z^* should be extracted. The eigenvalue decomposition of $\widetilde{Z}^*$ is

$$\begin{aligned} \widetilde{Z}^* = \begin{bmatrix} Z^* & \\ & 1-Z^* \end{bmatrix} &= \begin{bmatrix} U\Sigma U^T & \\ & U(1-\Sigma)U^T \end{bmatrix} \\ &= \begin{bmatrix} U & \\ & U \end{bmatrix} \begin{bmatrix} \Sigma & \\ & (1-\Sigma) \end{bmatrix} \begin{bmatrix} U^T & \\ & U^T \end{bmatrix}. \end{aligned} \tag{3.13}$$

Therefore, eigenvectors of Z^*, i.e., U, can be selected from eigenvectors of $\widetilde{Z}^*$.

The optimization problem (3.10) can be written in the standard SDP form as described in Appendix B. A standard dual pair of the SDP optimization shown in Appendix B can be modeled as a Linear Matrix Inequality (LMI),

$$\begin{aligned} &\min_x \ t^T x \\ &F(x) := \mathcal{F}x - \begin{bmatrix} W & \\ & 0 \end{bmatrix} \succeq 0, \end{aligned} \tag{3.14}$$

where $x \in \mathbb{R}^M : \ M = 1.5(L^2+L)+1$,

$$t = \begin{bmatrix} \mathbf{1}_{2L+1} \\ \mathbf{0}_{1.5L^2-0.5L} \end{bmatrix}, \tag{3.15}$$

and $\mathcal{F} : \mathbb{R}^M \to \boldsymbol{\mathcal{S}}_{2L}$ is a linear operator as

$$\begin{aligned}\mathcal{F}x = &\sum_{i=1}^{L} \begin{bmatrix} (\widetilde{D}^i + (\widetilde{D}^i)^T)/2 & \\ & 0 \end{bmatrix} x_i + \begin{bmatrix} I/Q & \\ & 0 \end{bmatrix} x_{L+1} \\ &+ \sum_{i=1}^{L} \begin{bmatrix} B^i & \\ & B^i \end{bmatrix} x_{i+L+1} + \sum_{m=1}^{L-1} \sum_{n=m+1}^{L} \begin{bmatrix} C^{mn} & \\ & C^{mn} \end{bmatrix} x_{n+(m+1)L-\frac{m^2+m}{2}+1} \\ &+ \sum_{i=1}^{L} \sum_{j=1}^{L} \begin{bmatrix} & H^{ij} \\ (H^{ij})^T & \end{bmatrix} x_{j+iL+\frac{L^2+L}{2}+1},\end{aligned} \tag{3.16}$$

where elements of matrices $\widetilde{D}^i, B^i, C^{mn}$ and H^{ij} are zero except that

- the i-th row of $\widetilde{D}^i$ is $d^{1/2}/d_i^{1/2}$,
- $B^i_{ii} = 1$,
- $C^{mn}_{mn} = C^{mn}_{nm} = 1$, and
- $H^{ij}_{ij} = 1$,

for $i, j = 1, \ldots, L$ and $1 \leq m < n \leq L$.

Now, the optimal solution x^* is used to achieve the eigenvectors of $\widetilde{Z}^*$. In the above primal-dual pair of SDP, optimizations (B.1) and (3.14), the strong duality is satisfied in the sense that for any optimal primal solution $\widetilde{Z}^*$ and any optimal dual x^* we have[4]

$$\widetilde{Z}^* F(x^*) = 0. \tag{3.17}$$

Therefore, the large eigenvalues of $\widetilde{Z}^*$ correspond to the small eigenvalues of $F(x^*)$.

Consequently, eigenvectors of Z^* can be selected from eigenvectors of $F(x^*)$ by extracting its L eigenvectors, whose largest entries concentrate at

[4]In general, for any standard primal-dual pair of SDP, such a relationship (3.17) between the optimal solutions holds [86].

the first half (see the decomposition in (3.13)) without knowing the primal solution $\widetilde{Z}^*$. Since the Q smallest eigenvalues of $F(x^*)$ correspond to the Q largest eigenvalues of Z^*, Y can be simply selected from U by conducting eigenvalue decomposition of $F(x^*)$.

Then, the un-normalized indicator matrix $\hat{Y}$ should be approximated. To this end, we generate the matrix $D^{-1/2}Y \in \mathbb{R}^{L\times Q}$, and consider rows of this matrix as L points in $\mathbb{R}^Q$. These points represent the original points $\{\theta_\ell\}_{\ell=1}^L$ in a different domain. By applying the K-means technique, these points are divided into Q partitions. The K-means method attempts to minimize the cost υ, which is the sum of the point-to-centroid distances in each partition, summed over all Q partitions. The K-means method is well developed in the Statistics Toolbox of the Matlab software [49].

One of the outputs of the K-means algorithm is the centroid for each partition. One can put an accurate interpretation on the distances of each point, rows of the matrix $D^{-1/2}Y$, to every centroid in order to *rank* the points of each partition. Therefore, we can determine how close each point is to the other members of its partition and to those of other partitions. As a result, the partition sets, indicated by $\hat{Y}$, can be assorted by the cost υ,

$$\{\{\hat{Y}_1, \upsilon_1\}, \{\hat{Y}_2, \upsilon_2\}, \ldots\} \ : \ \upsilon_i < \upsilon_j \text{ if } i < j. \tag{3.18}$$

We ignore the partition sets with high costs, and keep P partition sets with the least costs. One may choose one of the following ways to select a value for P.

1. If there is a jump between υ_i and υ_{i+1}, we select $P = i$.

2. The value of P should be large enough to possibly cover all the satisfactory partition sets.

After selecting P, the partition sets $\{\widehat{\boldsymbol{\mathcal{G}}}_p\}_{p=1}^P$ corresponding to the indicator matrices $\hat{Y}_p$ are entitled as the output of step (2) in Figure 3.2. The procedure developed in this section is illustrated in Figure 3.4.

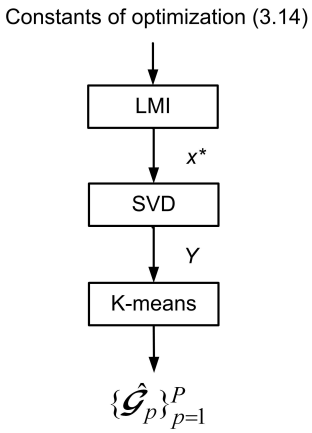


Figure 3.4: A short-form flowchart of the procedure explained in Section 3.3.2.

3.3.3 Optimum partition set selection

This section describes Step (3) in Figure 3.2 in detail. This step attempts to obtain the optimum partition set, which leads to the best closed-loop performance, among the given P partition sets from Step (2). We suggest the following two approaches. Note that one controller $K^{(q)}$ satisfies the performance objective of all the systems in partition $\boldsymbol{\mathcal{G}}^{(q)}$.

1. Synthesize a controller set for each of the P partition sets, and the system-controller pairs are derived as $\{\{\boldsymbol{\mathcal{G}}_p^{(q)}, K_p^{(q)}\}_{q=1}^{Q}\}_{p=1}^{P}$. The set with the best closed-loop performance is chosen as the optimum partition set. Assume that the closed-loop performances for system-controller pairs can be denoted by $\{\{\gamma_p^{(q)}\}_{q=1}^{Q}\}_{p=1}^{P}$, and that the better performance is achieved for the smaller γ. Therefore, the best performance can be selected as

$$\gamma_{opt} := \min_{p} \max_{q} \gamma_p^{(q)}. \tag{3.19}$$

   The partition set corresponding to the pair with the best closed-loop performance, γ_{opt}, is chosen as the optimum partition set.

2. Synthesize a controller set for the partition set with the minimum cost, i.e, υ_1, and generate $\{\boldsymbol{\mathcal{G}}_1^{(q)}, K_1^{(q)}\}_{q=1}^{Q}$ and $\{\gamma_1^{(q)}\}_{q=1}^{Q}$. Then, we define

$$\gamma_1^* := \max_q \gamma_1^{(q)}. \tag{3.20}$$

The performance of this controller set $\{K_1^{(q)}\}_{q=1}^{Q}$ is analyzed on the other $P-1$ partition sets. Therefore, we obtain $\{\gamma_p^*\}_{p=2}^{P}$ for the pairs of $\{\{\boldsymbol{\mathcal{G}}_p^{(q)}, K_1^{(q)}\}_{q=1}^{Q}\}_{p=2}^{P}$. Similarly, the best performance is selected as

$$\gamma_{opt} := \min_{p=1,\dots,P} \gamma_p^*. \tag{3.21}$$

In the first approach, P controller sets are synthesized while in the second approach, one controller set is synthesized, and analyzed for the remaining $P-1$ partition sets. The user chooses one approach based on the advantages of each for a specific application. In general, controller analyzing is more straightforward than synthesizing. However, the first approach may achieve a partition set with a better closed-loop performance since controllers are synthesized for each partition set separately.

3.3.4 Optimum partition set derivation algorithm

To improve the performance of the algorithm, the parameter set $\boldsymbol{\Theta}$ should be normalized such that all parameters are in the range of $[-1, 1]$. Also, we may need to scale the normalized $\boldsymbol{\Theta}$ to address the controller design limit. We multiply the i-th component of θ by a weight ρ_i. For example, if the parameter vector includes the natural frequency and the damping ratio of a system, and the variations in the natural frequency is more problematic in controller design, the weight for the natural frequency is higher than that for the damping ratio. The weight $\rho \in \mathbb{R}^{n_\theta}$ is either given as *a priori* knowledge or obtained by trial and error.

Here, we summarize the proposed methods in Sections 3.3.2 and 3.3.3 as

an algorithm.

Algorithm 3.3.1.

Inputs: Parameter set $\mathbf{\Theta} = \{\theta_\ell \in \mathbb{R}^{n_\theta}; \ell = 1, \ldots, L\}$*, the number of partitions* Q*, and the scale factor* $\rho \in \mathbb{R}^{n_\theta}$*.*

1. *Normalize* $\mathbf{\Theta}$ *such that all parameters are in the range of* $[-1, 1]$*.*
2. *Scale* $\mathbf{\Theta}$ *by* ρ *to address the controller design limit.*
3. *Generate constant matrices for optimization problem (3.14).*
4. *Follow the flowchart in Figure 3.4 to obtain* P *best partition sets.*
5. *Follow the procedure in Section 3.3.3 to derive the optimum partition set.*

3.4 Numerical examples

In this section, the developed method is validated through numerical examples. First, we discuss the results of applying Algorithm 3.3.1 to the motivating example explained in Section 1.2.2. Then, we apply the method developed in Chapter 2 to the discrete sets to derive connected sets. Five different approaches are compared to highlight the effectiveness of the developed methods.

3.4.1 Illustrative example

It has been shown in Section 1.2.2 that, to achieve the desired closed-loop performance of systems (1.7), the parameter set shown in Figure 3.5 needs to be divided into at least two subsets. We apply Algorithm 3.3.1 step by step as follows to derive the optimum partition set.

1. Parameters ω_n and ζ are normalized to the range of $[-1, 1]$.

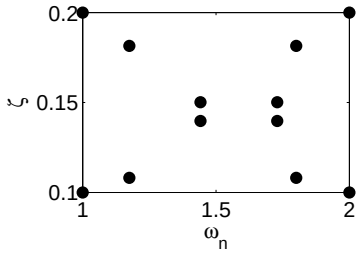


Figure 3.5: The parameter set of the example in Section 1.2.2.

2. The weight ρ is chosen such that the parameter ω_n is multiplied by 2, because variations in ω_n deteriorate the controller performance more in comparison with that in ζ in this example.

3, 4. The flowchart in Figure 3.4 is followed, and the sorted pairs of cost-partition are derived as explained in (3.18). The normalized values of the costs υ for $P = 12$ partition sets are shown in Figure 3.6. Figure 3.7 shows three partition sets with the least costs.

5. As it can be seen, the global optimum partition set shown in Figure1.6(c), which obtained by checking the closed-loop performances of all the possible partition sets, is among three partition sets derived in the previous step. Obviously, after checking the closed-loop performance, the global optimum partition set is chosen.

The proposed algorithm takes 254 s of completion time using a computer with a 2.93 GHz processor and 2 GB of RAM to extract the optimum partition set, while examining the closed-loop performances of the entire 2509 possible partition sets takes 10246 s. It shows the efficiency of the proposed method. In many practical applications, the parameter set is large in terms of the dimension (more complex transfer function) and the number of elements (more systems to be controlled). In such cases, the "full search" might become impractical even for a small number of partitions.

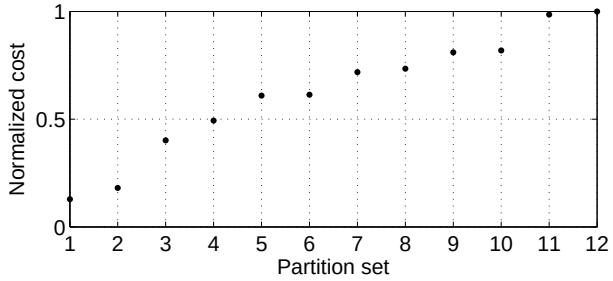


Figure 3.6: The normalized values of the costs of the different partition sets.

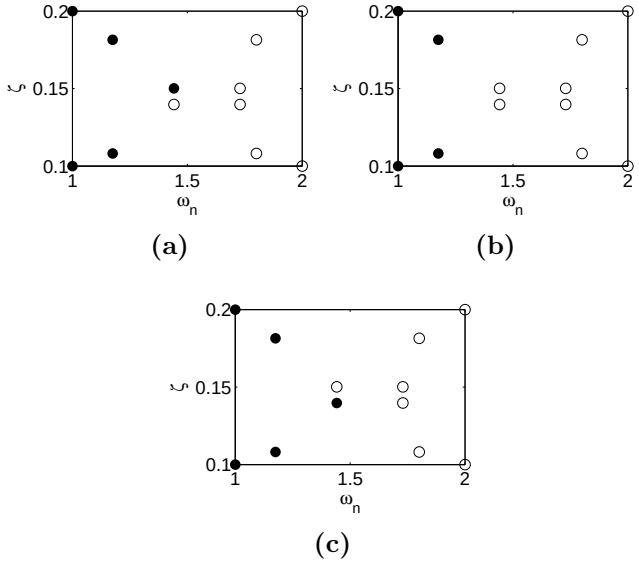


Figure 3.7: Three cluster sets with least cost functions.

3.4.2 Closed-loop performance comparison of connected sets

In this section, the effectiveness of combining the algorithm in this chapter with that presented in Chapter 2 is studied. We consider five different approaches to divide the given parameter set in the previous example into two partitions, where each of which is a connected set as follows.

1. The entire parameter variation region, $\{\omega_n \in [1,2],\ \zeta \in [0.1, 0.2]\}$ shown by a rectangle in Figure 3.5, is divided into two subregions in ω_n direction, i.e., $\omega_{n1} \in [1, 1.5]$ and $\omega_{n2} \in [1.5, 2]$, and a controller is synthesized for each subregion. These two controllers cover the entire region, which leads to a conservative solution.

2. The entire parameter variation region is divided into two subregions in ζ direction, i.e., $\zeta_1 \in [0.1, 0.15]$ and $\zeta_2 \in [0.15, 0.2]$, and a controller is designed for each subregion. Similar to the previous approach, this approach is conservative.

3. Algorithm 2.3.1 is applied to each of two partitions, which are generated intuitively and shown in Figure 3.8(a), to derive connected sets. The user inputs to the algorithm are $n_s = 0,\ n_u = 1,\ N = 50$, and the functions f are chosen as second order polynomials for both partitions.

4. Similar to the previous approach, the connected sets are derived for the partitions, which are generated intuitively and shown in Figure 3.8(b).

5. Algorithm 2.3.1 is applied to the optimum partition set, Figure 3.7(b). For both partitions, $n_s = 0,\ n_u = 1,\ N = 50$, while the functions f are chosen as second and third order polynomials for the partitions shown by dots and circles, respectively.

Robust controllers are designed for the resultant connected sets using the Matlab Robust Control Toolbox software [5]. The resultant closed-loop performances γ for above approaches are shown in Table 3.1 for both partitions for each approach. Note that the controller performances deteriorate in comparison with the performances shown in the example in Section 1.2.2. The main reason is that the controllers guarantee the performances γ for infinite number of systems (connected sets) here. On the other hand, in the previous case (discrete sets), the performances γ are guaranteed for a finite number of systems.

According to Table 3.1, the best performance is achieved for the forth approach. However, in this approach, the partitions are derived intuitively, which might be impractical for applications with the higher dimension of the parameter domain. The second best performance is obtained by the proposed method, which shows the effectiveness of the developed method. For a bigger parameter domain, which can be the case in practical applications, naive approaches (similar to the first and second ones above) lead to more conservative controllers, and intuitive clustering (similar to the third and forth approaches) becomes more difficult.

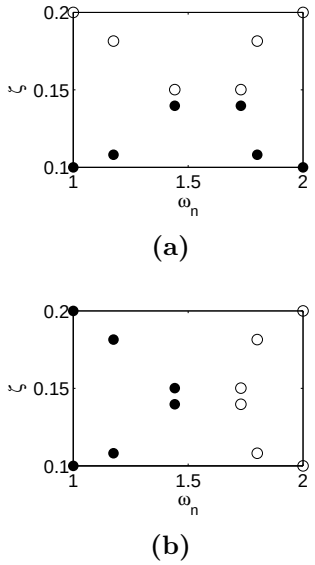


Figure 3.8: Two intuitive ways of clustering.

3.5 Conclusions

We proposed a technique to derive a family of discrete model sets, in the form of a partition set, for a given set of system response data. It was assumed that the given response data set was governed by a common transfer function with variations in parameters. A systematic algorithm was developed

Table 3.1: The achieved closed-loop performance γ for different approaches.

Approach	γ_1	γ_2	$\max\{\gamma_1, \gamma_2\}$
1	3.3168	2.8941	3.3168
2	3.8516	3.8526	3.8526
3	3.3124	3.1582	3.3124
4	3.2249	3.2231	3.2249
5	3.2999	2.0135	3.2999

to estimate a family of discrete model sets such that a certain closed-loop performance objective is fulfilled for all the given systems by designing a corresponding controller set. A relaxed version of Normalized Cut was applied in an algorithm to divide the system set into a few partitions. The effectiveness of the proposed method was verified through an illustrative example. Also, the effectiveness of combining this method and the one developed in Chapter 2 was shown.

Chapter 4

Modeling and Robust Track-Following Controller Design for Hard Disk Drives [1]

4.1 Introduction

Hard Disk Drives (HDDs) have been used widely in many consumer electronics, such as commercial computer systems, digital music players, and video-cameras for more than 50 years. They have been continuously evolving to achieve higher storage capacity and miniaturized sizes. In magnetic disks, data is stored on a recording medium (in industry commonly referred to as *the media*), which is responsive to the presence of strong magnetic fields, but stable in their absence.

Figure 4.1 shows the schematic diagram of a typical single-stage HDD. Main components are a spindle motor, one or more disks with data written on their surfaces, suspensions, heads/sliders, and a Voice-Coil Motor (VCM),

[1] This chapter is based on the following articles: E. Azadi Yazdi, M. Sepasi, F. Sassani and R. Nagamune, *"Automated multiple robust track-following control system design in hard disk drives"*, 2010 ASME Dynamic Systems and Control Conference, Boston, MA, and E. Azadi Yazdi, M. Sepasi, F. Sassani and R. Nagamune, *"Automated multiple robust track-following control system design in hard disk drives"*, to appear in IEEE Transactions on Control System Technology.

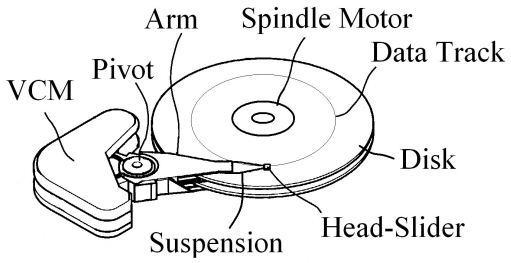


Figure 4.1: A schematic diagram of an HDD.

which rotates the arms around a pivot. Figure 4.2 shows a solid model, which includes the arms, suspensions, and heads. During operations, the disk may spin at speeds as high as 10,000 RPM by the spindle motor and generates high velocity airflow between the disk surface and the head. This high speed airflow has the effects of air bearing. A dynamic balance keeps the slider at a flying height of several nanometers over the disk surface. The VCM positions the head at the right data track, and thereby data can be read from or written to the disk.

The main HDD characteristic, which is the focus of most literature in this area, is areal storage density. It is essential to decrease the tracking error of the read/write head in order to increase the areal density of HDDs. A practical approach to achieve small tracking error is to add a secondary MEMS actuator to the servo assembly, and manufacture dual-stage HDDs (see, e.g., [106], [102], and [12]). Moreover, to achieve a good performance, we require a precise positioning control of the read/write head, such as the designs explained in [70], [33], [15], and [45].

To be able to design a high performance controller, the dynamics of HDDs should be examined thoroughly. Their dynamics can vary due to many factors such as variations in the fabrication environment, the temperature change during the operations, and mechanical deteriorations due to the elapse of time. This chapter addresses the Objective (O3) in Section 1.3. Here, we study the variations in the dynamics of HDDs, and derive a math-

ematical model for tracking controller design based on the FRF data.

This chapter is organized as follows. The experimental setup is briefly described in Section 4.2. The dynamics of this setup is explained in Section 4.3. The variations in the dynamics due to the manufacturing limits and the temperature change are discussed in detail. Modeling of the system is presented in Section 4.4. Section 4.5 explains the design of a robust controller, and demonstrates its track-following performance.

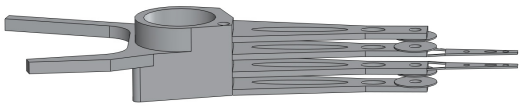


Figure 4.2: The solid model, which includes the arms, suspensions, and heads of an HDD.

4.2 HDD experimental setup

As a prelude to the demonstration and verification of modeling and controller design methods in the following sections, we will first describe an HDD experimental setup at the University of British Columbia, shown in Figure 4.3. The equipment, listed below, is quite standard for HDD servo experiments:

- laser doppler vibrometer (LDV) OFV-5000 and OFV-551 (Polytec);
- anti-vibration table RS3000 (Newport);
- amplifier TA105-A14 (TRUST Automation Inc.);
- FFT dynamic signal analyzer 35670A (Agilent Technologies);
- controller board DS1103 (dSPACE Inc.);
- five hard disk drives N256 (Maxtor);
- A blower heater to change the HDD temperature.

The input to the system is the voltage to the VCM while the output is the position of the head tip, which is measured by the LDV.

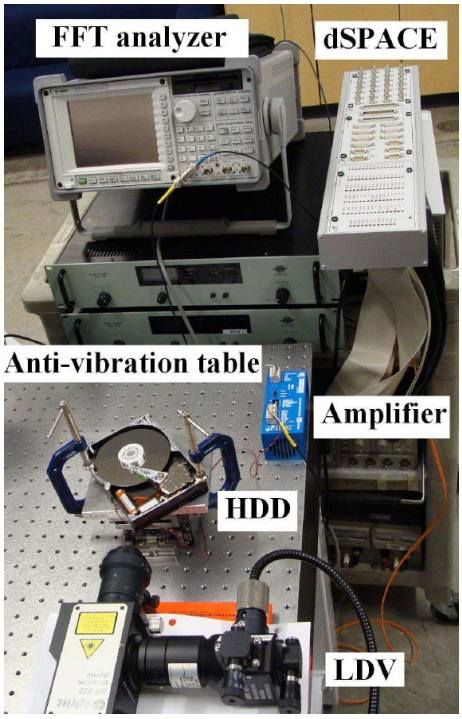


Figure 4.3: HDD experimental setup.

4.3 Dynamics of HDDs

It has been well-recognized that not only performance of track-following controllers but also their robustness are of great importance in HDDs [19, 77]. We study two types of factors, which introduce variations in dynamics. One type causes time-invariant variations, such as the product variability. The transfer function between the VCM voltage and the read/write head position may have differences between products due to the limited precision in the manufacturing line. The other type causes time-varying variations, such as the ones due to the temperature change during the HDD operations. The

temperature change occurs owing to the heat caused by the spinning spindles and the cooling air generated by the fan to avoid overheating.

To design a precise robust control system the knowledge about the plant dynamics as well as its variations is essential. However, such knowledge is frequently unavailable *a priori*. Therefore, some experiments must be carried out with the system in order to estimate the lacking information. In this section, the variations in the system dynamics are studied.

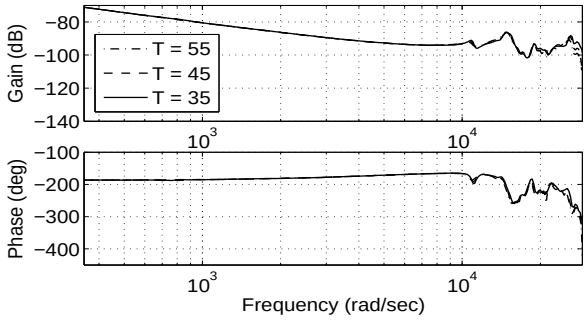
4.3.1 Variations in HDD dynamics due to temperature

It is well known that the temperature influences the dynamics of the systems by affecting such factors as the geometry and material properties. In all probability, the temperature of the arm varies during the operation, and therefore, it is essential to consider the influence of the temperature on the HDD dynamics. Temperature effects are studied for an HDD micro-actuator in [46]. However, the temperature effects on HDD dynamics are not sufficiently addressed in the literature from a controller design point of view (for one of the few examples see [82]).

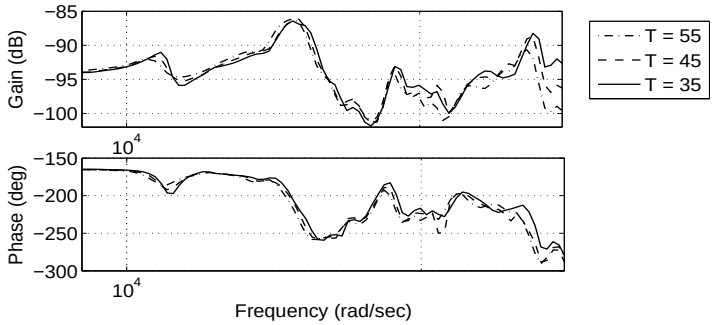
We derived the FRF data for three distinct temperatures of the arm, 35, 45, and 55 degree Celsius. This range of temperature change is realistic for a standard HDD during operations [45]. In the experiments, the temperatures are measured at the position of the pivot. However, one can assume that the temperature of the arm is fairly uniform [79]. Figure 4.4 shows the FRF data for different temperatures. As it can be seen, the frequency and damping ratio of each mode change slightly over temperature.

4.3.2 Variations in HDD dynamics due to the manufacturing limits

To investigate the variations between the dynamics of different HDDs from one production line, the frequency responses of five sample HDDs are studied.



(a) FRF data for HDD at different temperatures.



(b) Zoomed FRF data for HDD at different temperatures.

Figure 4.4: Variations due to the changes in the temperature.

Two sets of FRF data for each of the five HDDs are derived, and thus we have ten FRF data sets in total. As can be seen in Figure 4.5, all ten FRF data sets have similar gain and phase curves, but manufacturing variations obviously exist.

4.4 Modeling of HDDs

Based on the results shown in Section 4.3, there are variations in the HDD dynamics, which need to be taken into consideration. By comparing the results shown in Figures 4.4 and 4.5, it can be concluded that the variations in dynamics due to the manufacturing limits are more significant than those due to the temperature. Therefore, in the modeling and controller design we

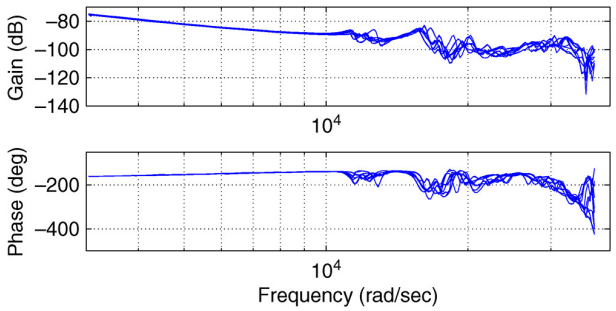


Figure 4.5: 10 FRF data (2 FRF are taken for each HDD).

ignore the effect of the time-varying temperature on the system dynamics.

Since only variations due to the time-invariant source, i.e. limited tolerance in the production line, are considered, the HDD dynamics is modeled as an LTI system. The sources of this type of variations may neither be measured nor estimated. Therefore, we deal with an uncertain LTI plant. It is desired to derive a connected model set to cover these variations and to represent the sampled products as well as the unsampled ones with a tight model.

By inspection of the FRF data, we have selected the model structure as

$$[G(\theta)](s) := \frac{a}{s^2} \prod_{i=1}^{4} \frac{s^2 + 2\zeta_{ni}\omega_{ni}s + \omega_{ni}^2}{s^2 + 2\zeta_{di}\omega_{di}s + \omega_{di}^2} \tag{4.1}$$

with $\theta \in \mathbb{R}^{17}$ consisting of a and $\{\zeta_{ni}, \omega_{ni}, \zeta_{di}, \omega_{di}\}_{i=1}^{4}$. Now, it is of interest to generate a connected set of transfer functions

$$\boldsymbol{\mathcal{G}}(\widehat{\boldsymbol{\Theta}}) := \{[G(\theta)](s) : \theta \in \widehat{\boldsymbol{\Theta}}\} \tag{4.2}$$

or equivalently, a connected parameter set $\widehat{\boldsymbol{\Theta}}$, such that we have the following:

- each FRF data in Figure 4.5 is represented properly by one element in $\boldsymbol{\mathcal{G}}(\widehat{\boldsymbol{\Theta}})$;
- the members of the set $\widehat{\boldsymbol{\Theta}}$ are parameterized with a small number of

uncorrelated parameters.

The method developed in Chapter 2 is employed to estimate the connected parameter set $\widehat{\mathbf{\Theta}}$. The inputs to the Algorithm 2.3.1 are

- $n_s = 0$,
- $n_u = 1$ and $N = 100$.

The parameterizing functions f is chosen by trial and error as[2]

$$f(\lambda) := f_0 + f_1\lambda + f_2\lambda^2 \tag{4.3}$$

where $\lambda \in \mathbf{\Lambda} \subset \mathbb{R}$ and $f_j \in \mathbb{R}^{17}$, $j = 0, 1, 2$. The uncertainty set $\mathbf{\Lambda}$ is

$$\mathbf{\Lambda} := \{\lambda \in \mathbb{R} \ : \ \lambda \in [-1, 1]\}. \tag{4.4}$$

Since the function f is chosen as a polynomial, we applied the special case of the algorithm explained in Section 2.3.2 to estimate the coefficient vectors f_0, f_1, and f_2. The resulting coefficient vectors are provided in Table 4.1. Figure 4.6 shows the Bode plots of ten samples from the set $\boldsymbol{\mathcal{G}}(\widehat{\mathbf{\Theta}}(f))$. The figure illustrates that the obtained model set captures well the major characteristics of the 10 FRF data sets.

4.5 Robust controller design for HDDs

To design robust track-following controllers, various methodologies have been proposed. These methodologies include, for example, H_∞ control [28], adaptive robust control [103], and sliding mode control [44]. In pursuing to improve robustness, it is inevitable to worsen the tracking performance, due to the trade-off relationship between robustness and performance inherent in

[2]In order to obtain a more accurate model set, one may increase the order of the polynomial, but the subsequent controller design would be numerically more demanding. The second order parametrization provides sufficient resolution in this study.

Table 4.1: Numerical values of the coefficients of polynomial functions in (4.3).

parameter	f_0	f_1	f_2
a	1360.4	-389.1	181.1
ζ_{n_1}	0.612	0.0101	-0.6
ω_{n_1}	13345	127	-719
ζ_{d_1}	0.2104	0.0544	-0.0825
ω_{d_1}	11077	-1	15
ζ_{n_2}	0.1746	0.2306	0.5195
ω_{n_2}	14276	573	-144
ζ_{d_2}	0.0329	0.013	0.0346
ω_{d_2}	15650	148	-604
ζ_{n_3}	0.0196	-0.01	-0.007
ω_{n_3}	17746	-469	1300
ζ_{d_3}	0.059	0.011	-0.044
ω_{d_3}	18432	-420	695
ζ_{n_4}	0.466	0.0705	-0.1871
ω_{n_4}	31419	3366	2889
ζ_{d_4}	0.055	-0.0129	-0.0251
ω_{d_4}	30740	1926	-1272

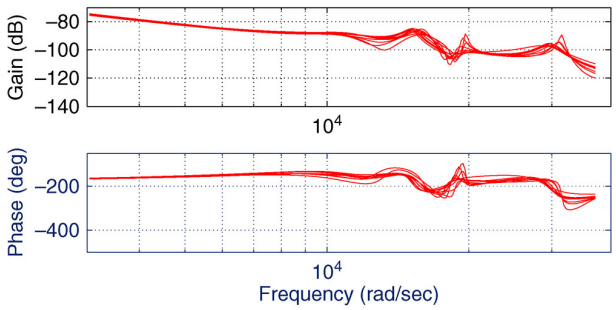


Figure 4.6: 10 random samples from the connected model set.

controller design. Therefore, in HDD applications, it is essential to compromise this conflict between high performance requirements and robust performance limitations. To meet such requirements by overcoming the limitations, a robust controller design technique has been proposed in [117].

We employ the proposed method in [117] for the derived connected model set (4.2) to meet the design specifications by designing an H_∞ robust controller. A typical performance specification for HDD track-following control is as follows.

Robust sensitivity shaping: The sensitivity function for a plant $G(s)$ and a controller $K(s)$ is defined as

$$S(s) := \mathcal{F}(G, K) := \frac{1}{1 + G(s)K(s)}. \tag{4.5}$$

For HDD servo control, the function S represents the transfer function from the track reference signal to the Position Error Signal (PES), and from the output disturbance to the read/write head tip position. By interpreting the shape of the Bode plot of S, the tracking performance can be determined. In particular, the low frequency gain and the bandwidth of S indicate degree of tracking accuracy and tracking speed, respectively. Therefore, constraints on the FRF $S(j\omega)$ is one of the key specifications for track-following.

Let us define a class of fixed-structure controllers robustly stabilizing $\boldsymbol{\mathcal{G}}(\widehat{\boldsymbol{\Theta}}(f))$ as $\boldsymbol{K}$. How to shape S can be expressed as a weighted H_∞ problem: design a controller $K \in \boldsymbol{K}$, which satisfies the inequality constraint

$$\max_{G \in \boldsymbol{\mathcal{G}}(\widehat{\boldsymbol{\Theta}}(f))} \|\mathcal{F}(G, K)\|_\infty < \gamma. \tag{4.6}$$

Here, γ is a given positive scalar, the function $\mathcal{F}$ contains weighting functions, and $\|\mathcal{F}\|_\infty$ denotes the H_∞ norm of $\mathcal{F}$, i.e. the maximum singular value of $\mathcal{F}$ over all frequencies. Since the transfer function $\boldsymbol{\mathcal{G}}(\widehat{\boldsymbol{\Theta}}(f))$ is parameterized by the variable $\lambda \in \boldsymbol{\Lambda}$, we can rewrite the left-hand side of (4.6) in terms of λ as

$$\max_{\lambda \in \boldsymbol{\Lambda}} \|\mathcal{F}(G(\lambda), K)\|_\infty < \gamma. \tag{4.7}$$

where $\boldsymbol{\Lambda}$ is defined in (4.4).

For the connected transfer function set $\boldsymbol{\mathcal{G}}(\widehat{\boldsymbol{\Theta}}(f))$, the controller is synthe-

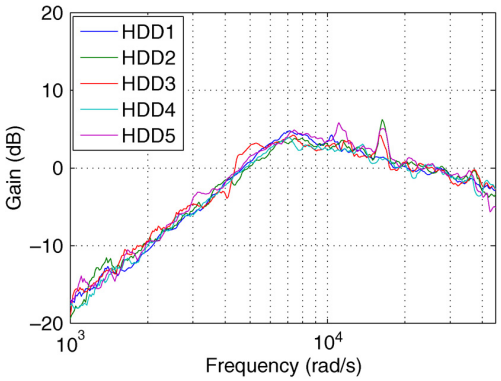


Figure 4.7: Frequency-domain response of closed-loop systems.

sized to fulfill the above explained track-following specification. The values of the given pre-specified parameters, e.g., γ, and tuning parameters, e.g., weighting functions, are provided in [4][3]. The resultant sensitivity function of the closed-loop system is shown in Figure 4.7. As can be seen, the experiments show generally

- low gain peaks, which leads to large stability and less oscillatory time-domain responses,
- low gain at low frequencies, which leads to disturbance attenuation in these frequencies, and
- high bandwidth, which leads to high speed tracking.

For one of the five HDDs, the open-loop and the closed-loop responses are shown in Figure 4.8. From this time-domain signal, it is evident that the controller not only suppresses the head vibration but also eliminates the drift of the head position caused by effects, such as friction and air-flow turbulence. Note that there is an offset of around 11 μm.

[3]The design of the robust controller for HDDs is not within the scope of this thesis.

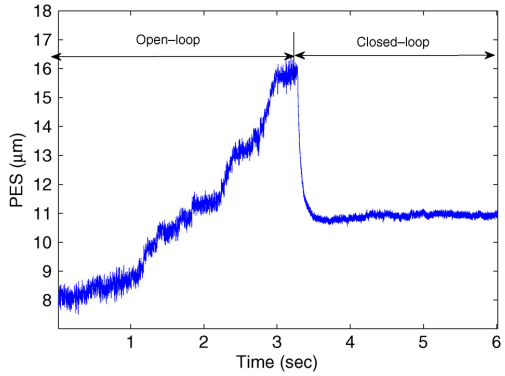


Figure 4.8: Time-domain response of open-loop and closed-loop systems.

4.6 Conclusions

In this Chapter, we studied the dynamics of HDD systems, especially, the variations in the dynamics due to the change in the temperatures and limited tolerances in the production line. It was shown that the influence of the temperature on the system dynamics was not significant, and hence was ignored in the modeling and controller design. A tight connected model set was derived based on a set of experimental FRF data. Then, the controller synthesis was described. The experimental results were demonstrated in the frequency and time domains.

Chapter 5

Modeling and Robust Tracking Controller Design for Flexible Ball Screw Drives with Runout Effect and Mass Variation[1]

5.1 Introduction

Most machine tools rely on precision Ball Screw Drives (BSD) to accurately position the workpiece relative to the tool. The quality of the machining outcome depends significantly on the tracking performance of the workpiece position over the desired trajectory. In order to minimize the tracking errors at all times during machining processes, feedback servo controllers must be designed carefully for any conceivable condition [16].

To achieve small tracking errors in various conditions, the servo controllers should compensate for the variations in the dynamics of the systems. Such variations occur because of nonlinearities and uncertainties inherent to real

[1]This chapter is based on the following articles: M. Sepasi, F. Sassani and R. Nagamune, *"Tracking Control of Flexible Ball Screw Drives with Runout Effect Compensation"*, 2010 ASME Dynamic Systems and Control Conference, Boston, MA, and M. Sepasi, F. Sassani and R. Nagamune, *"Tracking Control of Flexible Ball Screw Drives with Runout Effect and Mass Variation"*, to appear in IEEE Transactions on Industrial Electronics.

plants. Two of the most common sources of nonlinearities in BSD systems are the structural flexibility [123] and runout [78]. The former is a natural characteristic of physical materials, and the latter results from the variability in ball screw manufacturing and assembly. In addition, during operations, the table mass, which refers to the combined equivalent mass of the ball screw table and the mass of the workpiece attached to it, is normally varying and not measurable, and hence, classified as uncertain. Such uncertainty in the mass leads to the system parametric uncertainty, which along with the nonlinearities in the dynamics, makes the servo controller design a challenging process.

Extensive research has been conducted on servo control methods applied to precision motion mechanisms. Classical controllers are found to be widely used [17, 111] because of their high adaptability, simplicity, and ease of understanding, designing and tuning. Despite the popularity of the classical controllers, their performances are limited due to the uncertainties and nonlinearities in the systems. To effectively address these issues in controller design, sliding mode controllers [84, 112] and adaptive controllers [43] have been utilized. This chapter focuses on adaptive control, and uses the gain scheduling approach.

Since the variations due to the nonlinearities of the BSDs depend on the table position [85], and the speed of the table is bounded in reality, the tracking performance can be improved by adjusting controller parameters using the table position information. Therefore, one may consider the table position as a scheduling variable and design a gain scheduling controller [91] accordingly. Furthermore, the intended control system must have a good tracking performance in the presence of uncertainty in the system dynamics. The table mass uncertainty influences the transfer function parameters, and thus, makes them uncertain. It is critical to detect the correlations between the parameters and the table mass in order to develop a tight uncertain set for the controller synthesis. It is a non-trivial task to detect such correlations

through physical laws alone due to the complex coupling effects in the BSDs [83]. In such cases, experimental and numerical analysis methods can be used in a complementary manner.

This chapter addresses the Objective (O4) in Section 1.3. Here, we study the variations in the dynamics of BSDs and develop a systematic method to make a mathematical model and design tracking controllers for BSDs based on their FRF data. We consider structural flexibility and runout of the shaft, as well as the table mass variation. Although one specific experimental setup is used for investigation of the plant dynamics and validating the proposed method, the discussion and the methodology are general enough to be applied to other BSDs.

This chapter is organized as follows. The experimental setup is described in Section 5.2. The dynamics of the BSD systems is explained in Section 5.3. The position-dependent and mass-dependent variations in the dynamics are discussed in detail. Modeling of the system is presented in Section 5.4. Section 5.5 explains the design of a number of controllers, and demonstrates their track-following performances in the presence of flexibility, runout, and mass variations.

5.2 BSD experimental setup

The investigation of the variations in the dynamics and the demonstration and verification of modeling and controller design methods are carried out on an experimental BSD system at the University of British Columbia. The components of the setup, which is depicted in Figure 5.1, are listed below.

- A brushless DC motor.
- A linear encoder with a resolution of 50 nm.
- A ball screw with 20 mm shaft diameter, 20 mm screw pitch, and 360 mm stroke.
- A 20 kg table sliding on roller bearing guideways.

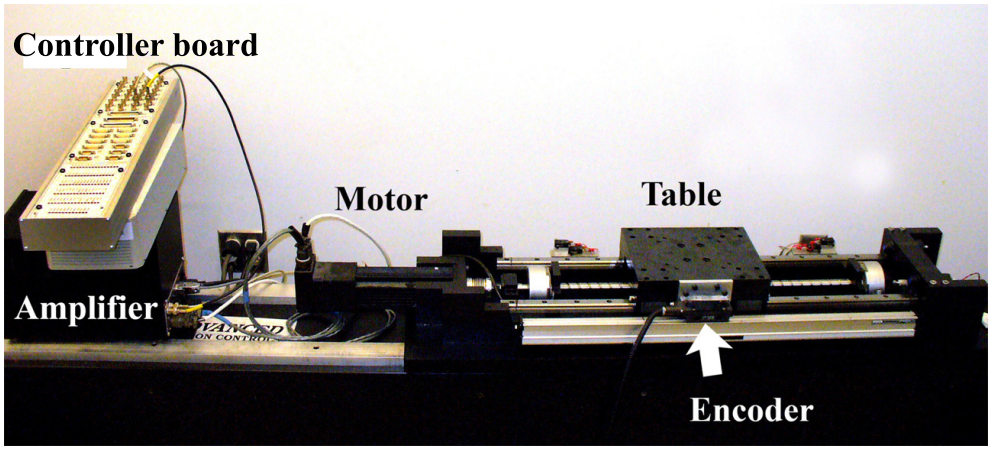


Figure 5.1: The ball screw experimental setup.

- A controller board.
- An amplifier.
- An FFT dynamic signal analyzer. (not shown in the picture)

All experiments reported in the following sections are carried out within the range of [0.12, 0.3] m along the ball screw shaft.

5.3 Variations in the dynamics of BSDs

A schematic diagram of the mechanical structure of the BSD is shown in Figure 5.2. The objective is to control the position of the table ℓ by applying the motor torque τ, while the disturbance d is applied. The dynamics of the BSD varies by changing the position and the mass of the table due to some factors, such as different mass distribution of the plant. From the controller design viewpoint, it is essential to know the manner in which these factors influence the dynamics of the system, and consequently, the positioning of the table. To this end, we take FRF data for different table positions and masses.

5.3.1 Position-dependent variations

The actuation torque τ, transmitted to the table, passes through the active length of the ball screw shaft, i.e. the part of the shaft between the motor and the table. The equivalent stiffness of the ball screw within its active length mainly depends on the bearing, the shaft itself, and the ball screw-nut interface. The position of the table, ℓ, affects the active length of the shaft and the corresponding stiffness, and hence, the dynamics of the system.

On the other hand, the runout phenomenon mainly occurs due to the tolerances in the bearings and in the shaft manufacturing. Figure 5.3 visually explains this phenomenon. Figure 5.3(a) shows the ideal position of the ball screw shaft which is aligned with the motor shaft, while Figure 5.3(b) shows one possible configuration of the ball screw shaft in reality. Because of this phenomenon, the table positioning dynamics depends on the rotational angle of the shaft. The cyclic dependency is experimentally revealed later.

The position of the table is measurable with a bounded rate of change within the plant limits

$$\dot{\ell}(t) \in [-0.1, 0.1]\ m/s. \tag{5.1}$$

Hence, table position ℓ is employed as a scheduling variable. Consequently, we design a gain scheduling controller, which is adapted online by the value of this variable. To achieve a high performance controller, we need to extract accurately how this scheduling variable influences the dynamics.

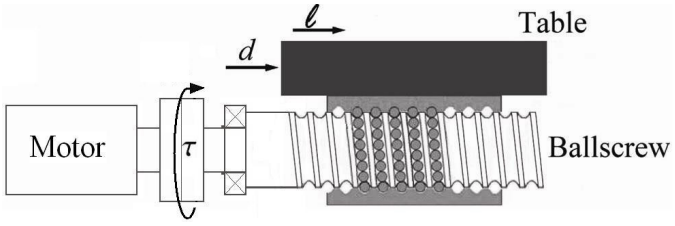
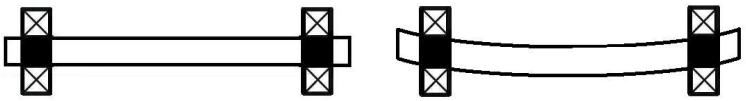


Figure 5.2: A schematic diagram of a ball screw assembly.



(a) Ideal configuration.

(b) Exaggerated configuration of a case which leads to runout.

Figure 5.3: Run out effect in the ball screw shaft.

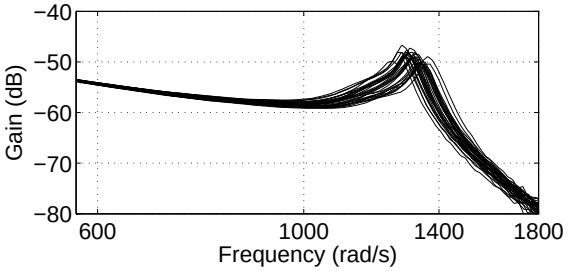


Figure 5.4: Frequency responses for 37 different positions of the table along the shaft when no mass is added to the table.

To obtain an accurate representation of the dynamics needed for designing controllers, the FRF data of the system is examined for several positions of the table along the ball screw shaft. In these experiments, the input to the open-loop system is the motor voltage while the output is the table position. The FRF data for the entire range of table motion, $[0.12, 0.3]$ m, is derived for every 5 mm, and thus, for 37 frequency points in total. Figure 5.4 shows the resultant 37 frequency responses when there is no added mass to the table. Here, the system response varies by changing the table location mostly due to the nonlinearities.

5.3.2 Mass-dependent variations

The mass of the workpiece changes significantly as material is removed during the cutting operations. The situation is further complicated by the fact that

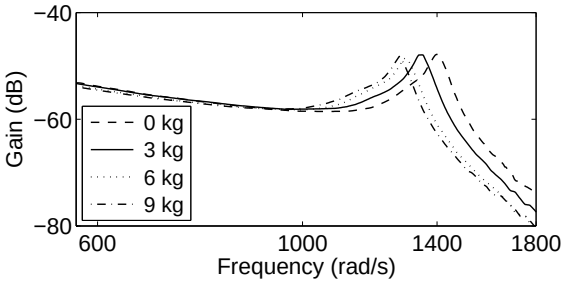


Figure 5.5: Frequency responses for four different masses added to the table when the position is at 0.25 m.

the rate of mass change cannot be easily predicted because it is different for each tool, workpiece, machine tool and cutting process combination.

The mass of the table is considered uncertain but expected to remain within a bounded range during the machining operation. However, the rate of change is not bounded. Therefore, the tracking gain scheduling controller must be robust enough to handle this time-varying uncertainty. To achieve a high performance robust controller, we need to extract accurately how this uncertain variable affects the parameters of the model.

To obtain an accurate representation of the dynamics needed for designing controllers, the FRF data of the system is examined for different masses attached to the table. The FRF data for four different masses (0, 3, 6, 9 kg) is taken. Figure 5.5 shows the FRF data when the table position is at 0.25 m and different masses are placed on the table. Likewise, the frequency responses vary according to table mass changes.

5.4 Linear parameter varying model of BSDs

As explained earlier, we have samples of FRF data for 37 different table positions and 4 different masses, totally $D = 148$ samples. In this section, first we identify an LTI system for each sampled FRF data. These systems have

the same structures but different parameters. Then, we apply the method developed in Chapter 2 to estimate the correlations between the system parameters and the variables, which are scheduling and uncertain variables. Next, an uncertain Linear Parameter Varying (LPV) model is derived to cover the entire variations over the table position and mass. Moreover, we apply the method developed in Chapter 3 to divide the mass variations into a few subsets, and consequently, a multiple model is derived.

5.4.1 Linear time-invariant system identification based on frequency response

The structure of the transfer function for LTI models may be selected by inspection of the FRF data, application of systematic methods such as the Akaike information criterion [1], or by trial and error. A model, which captures the first mode dynamics, has been developed in [109], where the ball screw system was represented by a uniform beam and some rigid bodies connected through springs. Then, the beam model was transformed into a two degree of freedom system.

Here, suppose that the model structure is given and fixed as

$$\begin{aligned} G(s) &:= G_1(s)G_2(s), \\ G_1(s) &:= \frac{1}{Js^2 + \beta\ s}, \quad G_2(s) := \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n\ s + \omega_n^2}, \end{aligned} \tag{5.2}$$

where J and β are the equivalent mass and viscous damping factor respectively, while k, ζ and ω_n are modal parameters. The transfer function $G_1(s)$ models the low frequency characteristics while $G_2(s)$ captures the first structural mode of the ball screw. All parameters are estimated using a nonlinear least squares optimization formulation [76]. For the sake of convenience in the formulation, instead of J, parameter $J^* := 1/J$ is used. The estimated parameters show that the viscous damping β remains almost constant at the

value of $\beta = 604.01$ over different masses m and different table positions ℓ,

$$m \in [20, 29]\ kg, \quad \ell \in [0.12, 0.3]\ m. \tag{5.3}$$

On the other hand, other parameters, k, ζ, ω_n and J^*, vary depending on ℓ and m. Hence, the parameter set $\mathbf{\Theta}$ is introduced as

$$\mathbf{\Theta} := \{\theta_d := [k_d,\ \zeta_d,\ \omega_{nd},\ J_d^*]^T,\ d = 1, \ldots, D\}. \tag{5.4}$$

Figure 5.6(a) shows the values of estimated parameters at the table positions of every 20 mm when no mass is added to the table. Since the sampled table position interval is equal to the ball screw pitch, the runout effect is not observable here. Hence, variations in estimated parameters are mostly due to the structural flexibility. Further, the FRF data is studied for more table positions, i.e. for every 5 mm. In this case, quasi-sinusoidal variations due to the runout can be observed, as shown in Figure 5.6(b).

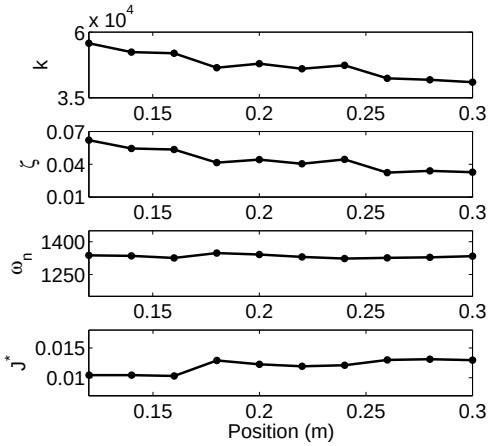
A normalized uncertain term μ is defined as

$$\sqrt{\frac{1}{m}} \in [x, y] = \left\{ \frac{x+y}{2} + \frac{x-y}{2}\mu \right\} \ : \\ x := \sqrt{\frac{1}{\max(m)}},\ y := \sqrt{\frac{1}{\min(m)}},\ \mu \in [-1, 1]. \tag{5.5}$$

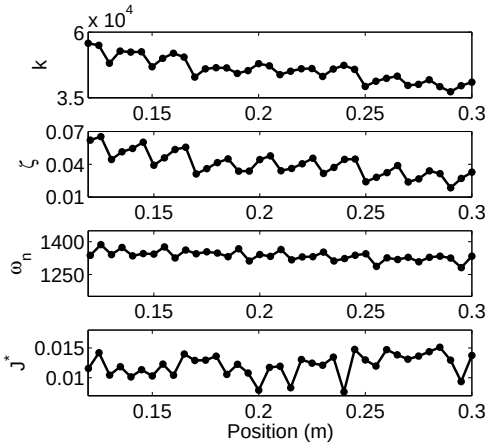
Value of μ can be explicitly derived from

$$\mu = \left[\sqrt{\frac{1}{m}} - \frac{x+y}{2} \right] \frac{2}{x-y}. \tag{5.6}$$

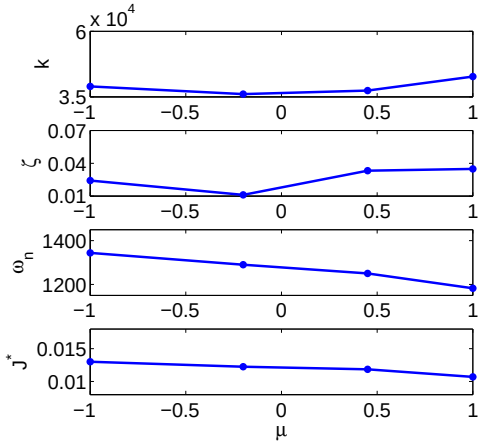
The variations in the parameters of the system dynamics due to μ are shown in Figure 5.6(c). By this transformation, $m = 20$ and $m = 29$ correspond to $\mu = -1$ and $\mu = 1$, respectively. The figure demonstrates that the effect of table mass on the dynamics is as significant as the effect of the table position



(a) 10 different table positions when no mass is added.



(b) 37 different table positions when no mass is added.



(c) 4 different masses are added while the table position is at 0.25 m.

Figure 5.6: Estimated transfer function parameters.

ℓ. It is worthwhile to note that, contrary to parameters J^*, ζ and ω_n, the parameter k is theoretically independent of the mass. However, due to some coupling inherent to the physical systems and unmodeled dynamics [83], k is influenced by the mass of the table, and hence, some variations can be observed in Figure 5.6(c).

The way how the table position and mass affect the varying parameters is expressed by a function $\tilde{f}$,

$$\widehat{\mathbf{\Theta}}(\tilde{f}) := \{\tilde{f}(\ell, m) \in \mathbb{R}^4,\ \ell \in [0.12, 0.3]\ m, \quad m \in [20, 29]\ kg\}, \tag{5.7}$$

where $\widehat{\mathbf{\Theta}}$ estimates the parameter set $\mathbf{\Theta}$. The goal is to obtain an accurate and yet simple function $\tilde{f}$.

The uncertainty parameter μ is introduced such that it is proportional to $\sqrt{1/m}$, see (5.5). This is inspired since, based on the physical laws, the parameters ζ and ω_n are proportional to $\sqrt{1/m}$ and the parameter J^* is proportional to $1/m$, and hence, they are proportional to μ and μ^2, respectively. Consequently, the set $\widehat{\mathbf{\Theta}}$ in (5.7) can be reformulated as

$$\widehat{\mathbf{\Theta}}(f) := \{f(\ell, \mu) \in \mathbb{R}^4,\ \ell \in [0.12, 0.3]\ m, \quad \mu \in [-1, 1]\}, \tag{5.8}$$

where the function f is readily parameterized as second order polynomials of μ.

5.4.2 Uncertain LPV modeling

In general, parameters of an LPV model vary with respect to independent variables called scheduling variables. As discussed above, the parameters of the linear model (5.2) are varying in time due to the scheduling variable. Therefore, we construct an LPV model to represent the dynamics of the system. In order to derive the LPV model, we interpolate the local LTI models. This interpolation is performed through the function f in (5.8).

Remark. It is not always possible to directly interpolate local models to

derive the overall LPV model [107]. However, according to [71], such derivation is implementable if the parameters of the identified local models show a smooth variation over the scheduling parameter (position ℓ here), and there is no sign change due to the non-uniqueness of the balancing transformation [62]. Here, the system is qualified for the direct interpolation (see Figure 5.6).

Moreover, our LPV system is uncertain. Therefore, it is essential to model the parametric uncertainty associated with the table mass. Such a model should be as tight as possible to reduce the unnecessary conservatism inherent to robust controllers.

Based on the knowledge about the BSD dynamics and the variations shown in Figure 5.6 the function f is expressed as

$$f(\ell, \mu) = \theta_0 + P_\ell(\ell) + P_\mu(\mu) + \alpha \sin(\omega\ell + \phi), \tag{5.9}$$

where P_ℓ and P_μ are polynomials. The polynomials $P_\ell(\ell)$ approximate the variations due to the table position change shown in Figure 5.6(a), while the sinusoidal term approximates the quasi-sinusoidal variations shown in Figure 5.6(b). The polynomials $P_\mu(\mu)$ model the variations due to the mass change shown in Figure 5.6(c). The parameters ϕ and ω are assumed to be the same for k, ζ, ω_n and J^*. The phase ϕ is manually selected as $\phi = 1\ rad$ by trial and error, and ω is calculated, $\omega = 2\pi/(\text{screw pitch}) = 100\pi\ rad/m$. The estimation of the other parameters in the function f, i.e., θ_0, coefficients in polynomials P_ℓ and P_μ, and α, will be explained later.

For the controller design procedure proposed is Section 5.5, we need to derive the LPV model in the state space form. Therefore, the system (5.2) with uncertain time-varying parameters k, ζ, ω_n and J^* and constant β is expressed in a quadruple of the state space data $(A_G(\ell, \mu), B_G(\ell, \mu), C_G, D_G)$.

The stae-space matrices are obtained for an observable canonical form as

$$
\begin{aligned}
A_G(\ell,\mu) &= \begin{bmatrix} 0 & 0 & 0 & J^*(\ell,\mu) \\ 1 & -\beta J^*(\ell,\mu) & 0 & 0 \\ 0 & 0 & 0 & -\omega_n(\ell,\mu)^2 \\ 0 & 0 & 1 & -2\zeta(\ell,\mu)\omega_n(\ell,\mu) \end{bmatrix}, \\
B_G(\ell,\mu) &= \begin{bmatrix} 0 & 0 & k(\ell,\mu)\omega_n(\ell,\mu)^2 & 0 \end{bmatrix}^T, \\
C_G &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}, \\
D_G &= 0.
\end{aligned} \tag{5.10}
$$

Single uncertain LPV model

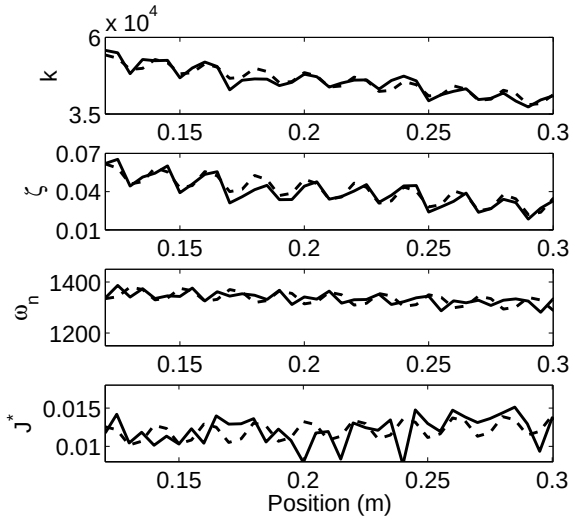
First, we derive a single uncertain LPV model, which estimates the system dynamics for the entire range of table position and mass variation. The method developed in Chapter 2 is employed to estimate parameters of the functions f in (5.9). The inputs to the Algorithm 2.3.1 are

- $n_s = 1$ and $\lambda^s = [0.12 : 0.005 : 0.3]$,
- $n_u = 1$ and $N = 20$.

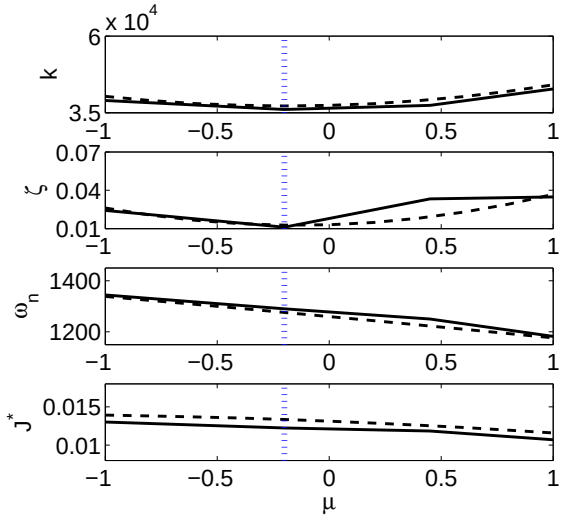
The resultant estimated parameters are given in Table 5.1, and the corresponding values of k, ζ, ω_n and J^* are shown in Figure 5.7 with dash lines over the entire range of the table positions for all the values of the added mass. Some model errors inherent with the modeling can be observed. These errors diminish by utilizing a more detailed (complex) model, e.g. higher order polynomials P_ℓ and P_μ, at the cost of higher complexity of controller design and implementations.

Multiple uncertain LPV model

In order to improve the closed-loop performance, we employed the method developed in Chapter 3 to divide the mass variation region $m \in [20, 29]$ kg



(a) Different table positions when no mass is added, $\mu = -1$.



(b) Different masses are added while the table position is at 0.25 m.

Figure 5.7: Transfer function parameters (solid lines), values in the estimated single model (dash lines), boundary of partitions (vertical dotted lines).

Table 5.1: Estimated parameters of the polynomial and sinusoidal functions.

			P_ℓ coefficient	P_μ coefficients	
	θ_0	α	ℓ	μ	μ^2
k	58849.8	2367.4	-77822.8	1885.3	4815.0
ζ	0.0567	0.0084	-0.132	0.0053	0.01841
ω_n	1299.5	-28.64	-256	-81.13	-2.28
J^*	0.0104	0.0013	0.0081	-0.00116	-3.742×10^{-5}

into two subregions. By choosing $\rho_i = 1$ for $i = 1, ..., 4$ in the Algorithm 3.3.1, the optimum partition set, $\{\Theta^{(q)}\}_{q=1}^{2}$, is derived, which consists of the following portions of mass variations

$$m^{(1)} \in [20, 23]\ kg,\ m^{(2)} \in (23, 29]\ kg. \tag{5.11}$$

Then, normalized parameters $\mu^{(1)}$ and $\mu^{(2)}$ can be introduced using (5.5) for $m^{(1)}$ and $m^{(2)}$, respectively. Vertical dotted lines in Figure 5.7 show the boundary of the partitions.

The parametrization function $f^{(q)}$ can be expressed as

$$f^{(q)}(\ell, \mu) = \theta_0 + P_\ell(\ell) + P_\mu^{(q)}(\mu) + \alpha \sin(\omega\ell + \phi). \tag{5.12}$$

The polynomials $P_\mu^{(q)}(\mu)$ model the variations due to the mass change within partition q shown in Figure 5.7b, or equivalent Figures 5.8(b) and 5.8(c). According to these figures, the order of the polynomials P_ℓ, $P_\mu^{(1)}$ and $P_\mu^{(2)}$ are selected as 1, 1 and 2, respectively. Subsequently, the Algorithm 2.3.1 is applied independently to each partition with the similar inputs as for single model derivation.

The estimated parameters for function $f^{(q)}$ are given in Tables 5.2 and 5.3 for partitions $\mu^{(1)}$ and $\mu^{(2)}$, respectively. The resultant estimated transfer

Table 5.2: Estimated parameters of the polynomial and sinusoidal functions for partition $\mu^{(1)}$.

			P_ℓ coeff.	P_μ coeff.
	θ_0	α	ℓ	μ
k	59613.2	2367.8	-77823.2	-1665.7
ζ	0.0577	0.0081	-0.133	-0.0073
ω_n	1353	-28.1	-255.4	-25.1
J^*	0.0113	0.0011	0.00809	-2.05×10^{-4}

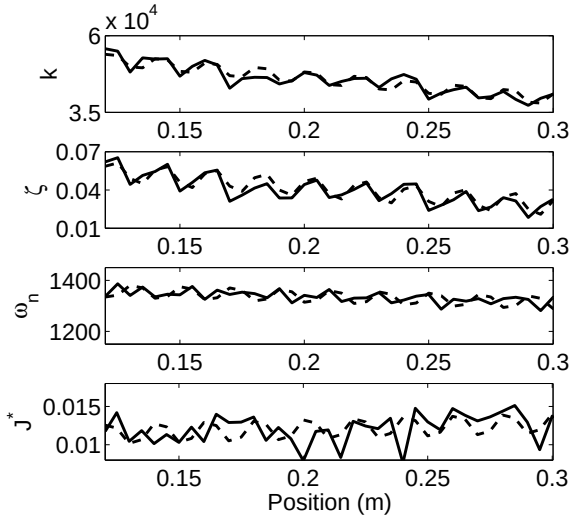
Table 5.3: Estimated parameters of the polynomial and sinusoidal functions for partition $\mu^{(2)}$.

		P_ℓ coefficient		P_μ coefficients	
	θ_0	α	ℓ	μ	μ^2
k	60458.1	2367.5	-77798.3	3450.2	1150.8
ζ	0.0754	0.009	-0.131	0.01213	-0.01211
ω_n	1291.17	-28.4	-256.3	-44	-14.67
J^*	0.0111	0.0014	0.00811	-0.001	-2.22×10^{-4}

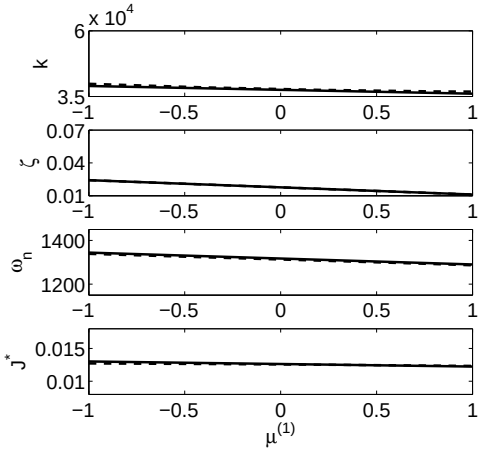
function parameters are presented in Figure 5.8 with dash lines. As it can be seen, the multiple model leads to less errors in the model in comparison with the single model, see Figure 5.7.

5.5 Controller design for the BSD

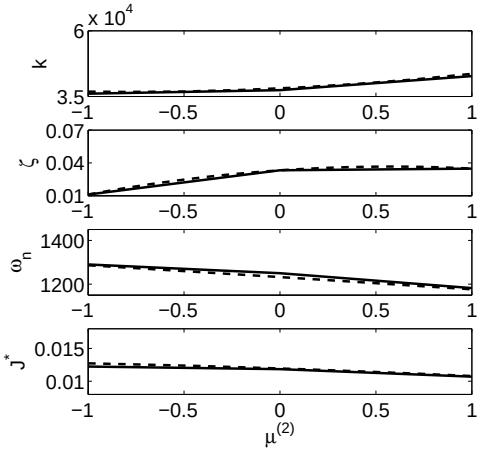
A gain scheduling output-feedback controller proposed by Apkarian and Adams [2] is applied to the LPV models derived earlier. The goal is to enforce the stability and achieve a good tracking performance of the closed-loop system. The design of the controller is an iterative procedure. It does not give a global solution to the problem, but it has been demonstrated in practice to result in acceptable solutions [22, 39, 118].



(a) Different table positions when no mass is added, i.e., $\mu^{(1)} = -1$.



(b) Different table masses in $\mu^{(1)}$ region.



(c) Different table masses in $\mu^{(2)}$ region.

Figure 5.8: Transfer function parameters (solid lines) and values in the estimated multi model (dash lines).

To examine the effectiveness of the robustness over the mass variation, controllers with and without robustness are designed, and referred to as the *robust controllers* and *non-robust controllers*, respectively. The non-robust controllers are designed for the case where there is no added mass to the table. While, the robust controllers take into consideration the mass variations.

Figure 5.9 shows the closed-loop block diagram used for both robust and non-robust controller synthesis schemes. In this configuration, a constant weight W_u is chosen based on the physical plant specifications to limit the control input u. The weighting function W_e in the state space form (A_e, B_e, C_e, D_e) is included to shape the frequency response, and is tuned to reach the desired performance of the controller K, which attempts to track the reference signal r in the presence of the disturbance d.

The aim of tuning parameters in W_e is to shape the closed-loop sensitivity function to meet the following criteria:

1. Minimize the gain in low frequencies to achieve a good disturbance rejection capability and tracking performance.

2. Maximize the cross-over frequency to obtain a fast response and good tracking performance.

3. Minimize the gain peak to provide a large stability margin and less oscillatory time-domain response.

For each controller synthesis, a distinct second order weighting function W_e is tuned to obtain the best achievable performance. However, the constant W_u is selected as

$$W_u = 2.5 \tag{5.13}$$

to limit the control input in all controller designs.

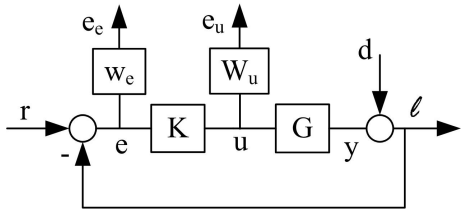


Figure 5.9: The closed-loop block diagram.

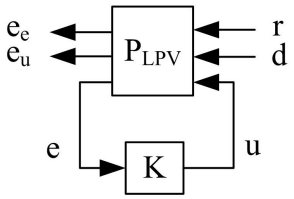


Figure 5.10: Synthesis closed-loop configurations for the case that the uncertainty is ignored in the plant G.

5.5.1 Non-robust gain scheduling controller design

The LPV model in (5.10) is a general model in which variations in table position and mass are considered. To address a special case where the mass uncertainty is ignored, the parametrization (5.9) is written in the form

$$f(\ell) = \theta_0 + P_\ell(\ell) + \alpha \sin(\omega\ell + \phi). \tag{5.14}$$

Accordingly, the quadruple of the state space data for system G in the block diagram in Figure 5.9 is

$$
\begin{aligned}
A_G(\ell) &= \begin{bmatrix} 0 & 0 & 0 & J^*(\ell) \\ 1 & -\beta J^*(\ell) & 0 & 0 \\ 0 & 0 & 0 & -\omega_n(\ell)^2 \\ 0 & 0 & 1 & -2\zeta(\ell)\omega_n(\ell) \end{bmatrix}, \\
B_G(\ell) &= \begin{bmatrix} 0 & 0 & k(\ell)\omega_n(\ell)^2 & 0 \end{bmatrix}^T, \\
C_G &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}, \\
D_G &= 0.
\end{aligned} \tag{5.15}
$$

The dynamics of this LPV system can be written in the form

$$
\begin{aligned}
\dot{x} &= A_G(\ell(t))x + B_G(\ell(t))u, \\
y &= C_G x + D_G u.
\end{aligned} \tag{5.16}
$$

Figure 5.10 is a Linear Fractional Transformation (LFT) form [121] of the configuration shown in Figure 5.9. The model $P_{LPV}(\ell)$ in Figure 5.10 is generated by combining the system (5.16) with the weighting functions. The mathematical expression of the configuration shown in this figure is

$$
\begin{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{x}_e \end{bmatrix} \\ \begin{bmatrix} e_e \\ e_u \end{bmatrix} \\ \begin{bmatrix} e \end{bmatrix} \end{bmatrix} = \underbrace{\begin{bmatrix} A_L & B_{1L} & B_{2L} \\ C_{1L} & D_{11L} & D_{12L} \\ C_{2L} & D_{21L} & D_{22L} \end{bmatrix}}_{P_{LPV}(\ell)} \begin{bmatrix} \begin{bmatrix} x \\ x_e \end{bmatrix} \\ \begin{bmatrix} d \\ r \end{bmatrix} \\ \begin{bmatrix} u \end{bmatrix} \end{bmatrix}, \tag{5.17}
$$

where x and x_e are the state vectors of the plant (5.16) and the weighting

function W_e in the state space form (A_e, B_e, C_e, D_e), respectively, and

$$
\begin{aligned}
A_L &= \begin{bmatrix} A_G(\ell) & 0 \\ -B_e C_G & A_e \end{bmatrix}, \\
B_{1L} &= \begin{bmatrix} 0 & 0 \\ -B_e & B_e \end{bmatrix}, \\
B_{2L} &= \begin{bmatrix} B_G(\ell) & -B_e D_G \end{bmatrix}^T, \\
C_{1L} &= \begin{bmatrix} -D_e C_G & C_e \\ 0 & 0 \end{bmatrix}, \\
C_{2L} &= \begin{bmatrix} -C_G & 0 \end{bmatrix}, \\
D_{11L} &= \begin{bmatrix} -D_e & D_e \\ 0 & 0 \end{bmatrix}, \\
D_{12L} &= \begin{bmatrix} -D_e D_G & W_u \end{bmatrix}^T, \\
D_{21L} &= \begin{bmatrix} -1 & 1 \end{bmatrix}, \\
D_{22L} &= -D_G(\ell).
\end{aligned}
$$

The resultant controller is in the state space form $(A_K(\ell), B_K(\ell), C_K(\ell), D_K(\ell))$ for which the order is six, equal to the sum of the orders of G and W_e. The matrices are

$$
\begin{aligned}
A_K =& N^{-1}[\hat{A}_K - X(A_L - B_{2L}\hat{D}_K C_{2L})Y_0 \\
& - \hat{B}_K C_{2L} Y_0 - X B_{2L} \hat{C}_K], \\
B_K =& N^{-1}(\hat{B}_K - X B_{2L} \hat{D}_K), \\
C_K =& \hat{C}_K - \hat{D}_K C_{2L} Y_0, \\
D_K =& \hat{D}_K,
\end{aligned} \tag{5.18}
$$

where $N = I - XY_0$, and matrices Y_0 and X are Lyapunov variables [93,

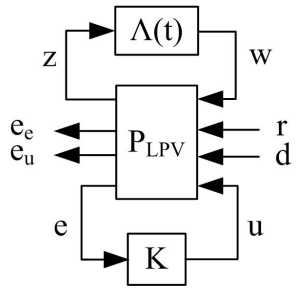


Figure 5.11: Synthesis closed-loop configurations for the uncertain plant G.

page 902]. Matrices $\hat{A}_K, \hat{B}_K, \hat{C}_K, \hat{D}_K$, and X are in an affine fashion as

$$\begin{aligned}
\hat{A}_K(\ell) &= \hat{A}_{K_0} + \ell\hat{A}_{K_1} + \sin(\omega\ell + \phi)\hat{A}_{K_2}, \\
\hat{B}_K(\ell) &= \hat{B}_{K_0} + \ell\hat{B}_{K_1} + \sin(\omega\ell + \phi)\hat{B}_{K_2}, \\
\hat{C}_K(\ell) &= \hat{C}_{K_0} + \ell\hat{C}_{K_1} + \sin(\omega\ell + \phi)\hat{C}_{K_2}, \\
\hat{D}_K(\ell) &= \hat{D}_{K_0} + \ell\hat{D}_{K_1} + \sin(\omega\ell + \phi)\hat{D}_{K_2}, \\
X(\ell) &= X_0 + \ell X_1 + \sin(\omega\ell + \phi)X_2,
\end{aligned} \tag{5.19}$$

which can be written in a generic form as

$$\mathcal{O} = \mathcal{O}_0 + \ell\mathcal{O}_1 + \sin(\omega\ell + \phi)\mathcal{O}_2, \tag{5.20}$$

where $\mathcal{O} = \{\hat{A}_K(\ell), \hat{B}_K(\ell), \hat{C}_K(\ell), \hat{D}_K(\ell), X(\ell)\}$ An iterative procedure is explained in [2] to obtain the optimum values for Y_0, $\mathcal{O}_0$, $\mathcal{O}_1$, and $\mathcal{O}_2$ for a non-robust gain scheduling controller.

5.5.2 Robust gain scheduling controller design

Figure 5.11 is also a Linear Fractional Transformation (LFT) form [121] of the configuration Figure 5.9 when the system G is uncertain. To derive such a form, the uncertainty of the plant G is extracted into an upper LFT form[2].

[2]Extracting the LFT form is well developed in the Matlab software [5].

The dynamics of the certain part, which is a function of ℓ, can be written in the form

$$\begin{aligned}
\dot{x} &= A(\ell(t))x + B_1(\ell(t))w + B_2(\ell(t))u, \\
z &= C_1(\ell(t))x + D_{11}(\ell(t))w + D_{12}(\ell(t))u, \\
y &= C_2(\ell(t))x + D_{21}(\ell(t))w + D_{22}(\ell(t))u,
\end{aligned} \tag{5.21}$$

and the uncertainty block is

$$\Lambda(t) := \mu(t)I, \tag{5.22}$$

where the size of I is the same as the order of the polynomial $P_\mu(\mu)$. Then, the system (5.21) combines with the weighting functions, generating the model $P_{LPV}(\ell)$ in Figure 5.11. The mathematical expression of the configuration shown in Figure 5.11 is

$$\begin{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{x}_e \end{bmatrix} \\ \begin{bmatrix} z \\ e_e \\ e_u \end{bmatrix} \\ \begin{bmatrix} e \end{bmatrix} \end{bmatrix} = \underbrace{\begin{bmatrix} A_L & B_{1L} & B_{2L} \\ C_{1L} & D_{11L} & D_{12L} \\ C_{2L} & D_{21L} & D_{22L} \end{bmatrix}}_{P_{LPV}(\ell)} \begin{bmatrix} \begin{bmatrix} x \\ x_e \end{bmatrix} \\ \begin{bmatrix} w \\ d \\ r \end{bmatrix} \\ \begin{bmatrix} u \end{bmatrix} \end{bmatrix}, \tag{5.23}$$

where x and x_e are the state vectors of the plant (5.10) and the weighting function W_e in the state space form (A_e, B_e, C_e, D_e), respectively, and

$$
\begin{aligned}
A_L &= \begin{bmatrix} A(\ell) & 0 \\ -B_e C_2(\ell) & A_e \end{bmatrix}, \\
B_{1L} &= \begin{bmatrix} B_1(\ell) & 0 & 0 \\ -B_e D_{21}(\ell) & -B_e & B_e \end{bmatrix}, \\
B_{2L} &= \begin{bmatrix} B_2(\ell) & -B_e D_{22}(\ell) \end{bmatrix}^T, \\
C_{1L} &= \begin{bmatrix} C_1(\ell) & 0 \\ -D_e C_2(\ell) & C_e \\ 0 & 0 \end{bmatrix}, \\
C_{2L} &= \begin{bmatrix} -C_2(\ell) & 0 \end{bmatrix}, \\
D_{11L} &= \begin{bmatrix} D_{11}(\ell) & 0 & 0 \\ -D_e D_{21}(\ell) & -D_e & D_e \\ 0 & 0 & 0 \end{bmatrix}, \\
D_{12L} &= \begin{bmatrix} D_{12}(\ell) & -D_e D_{22}(\ell) & W_u \end{bmatrix}^T, \\
D_{21L} &= \begin{bmatrix} -D_{21}(\ell) & -1 & 1 \end{bmatrix}, \\
D_{22L} &= -D_{22}(\ell).
\end{aligned}
$$

Similar to Section 5.5.1, the resultant controller is in the state space form $(A_K(\ell), B_K(\ell), C_K(\ell), D_K(\ell))$ for which the order is six. The optimum values for the matrices are obtained by following a procedure given in [2] for a robust gain scheduling controller.

5.5.3 Disturbance observer design

By tuning the weighting function W_e during controller design, we try to achieve the best possible tracking performance for each controller. Moreover, in the implementations of all the designed controllers, a disturbance observer

(DOB) has been employed in order to further reduce the tracking error by attenuating the low frequency disturbances [23, 59, 81]. The addition of the DOB does not change the open-loop dynamics as well as the transfer function between r (the reference signal) and ℓ (the output signal). Therefore, the dynamics of the DOB does not need to be considered in the controller design procedure, which is explained in Section5.5.2.

Figure 5.12 shows the block diagram of the implemented motion control system. The following transfer functions can be obtained for this block diagram

$$\begin{aligned}\ell(s) =& Q(s)d(s) + G(s)T(s)u(s),\\ Q(s) :=& \frac{G_0(s)(1-F(s))}{G_0(s) - G_0(s)F(s) + G(s)F(s)},\\ T(s) :=& \left[1 - F(s) + \frac{F(s)}{G_0(s)}G(s)\right]^{-1}.\end{aligned} \tag{5.24}$$

The transfer function G_0 includes the low frequency dynamics of the nominal plant. Here, G_0 has the structure of $G_1(s)$ in (5.2) with nominal values of J and β. The block F represents a stable low-pass filter with the following characteristics. Its bandwidth is limited by the frequency where the unmodeled dynamics is significant, it has a DC gain of unity, and its relative degree is greater or equal to that of G_0. Here, a filter with all these characteristics is selected as

$$F(s) = \frac{a_0}{(\tau s)^2 + a_1\tau s + a_0}, \tag{5.25}$$

where $\tau = 1/(20\pi)$, $a_0 = 120$, and $a_1 = 200$. The structure of the filter is chosen from [100], and the parameters are tuned to fulfill above characteristic requirements.

The low frequency disturbance d is attenuated since at low frequencies $F(s) \approx 1$, which means that $Q(s)$ vanishes. Moreover, the DOB has small

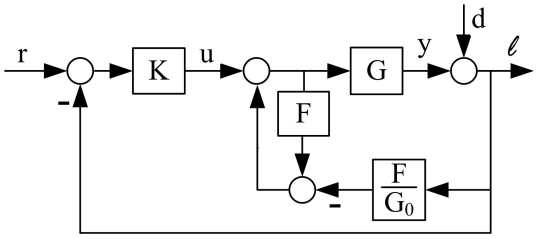


Figure 5.12: Controller with disturbance observer scheme.

influence on the plant dynamics, i.e. open-loop transfer function between u and ℓ, because $T(s) \approx 1$ for the entire frequency range. The proximity of $T(s)$ to unity is due to the facts that the gain of $F(s)$ attenuates at high frequencies, and that $G(s) \approx G_0(s)$ at low frequencies. This phenomenon is proven by simulation in Figure 5.13. This figure shows various frequency responses of $T(s)$ for different values of parameters k, ζ, ω_n and J^*. Based on the magnitude and the phase of the transfer function $T(s)$, the effect of $T(s)$ on the system dynamics can be neglected over the entire frequency range.

Remark. The performance of the DOB depends strongly on the accuracy of the estimation of low frequency dynamics, e.g. the nominal values of J and β. Since the table mass is uncertain, the DOB is designed for the conservative case. In other words, the cutoff frequency of the low pass filter $F(s)$ is chosen based on the plant with the smallest crossover frequency.

5.6 Controller results

In order to compare the tracking performance of the controllers, a trajectory for the machine table position is generated. Considering the test bed limits, the following values are chosen in the trajectory generation: stroke of 0.18 m, velocity of 0.09 m/s, acceleration of 0.7 m/s^2, and jerk of 50 m/s^3. The trajectory moves the table along the shaft from one end, farthest from the motor, to the other and then returns to the starting position.

Four sets of experiments are carried out in this section.

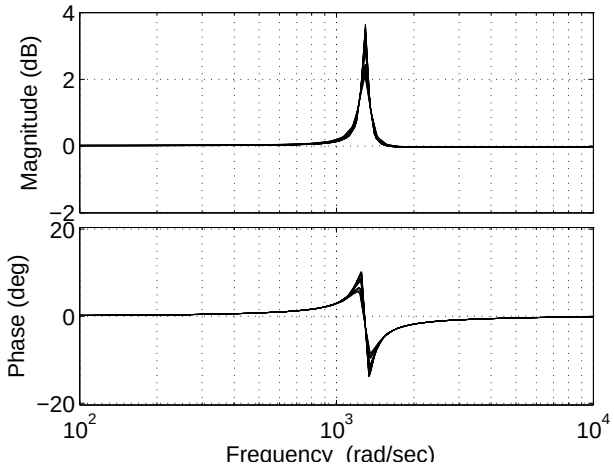


Figure 5.13: Frequency responses of perturbed $T(s)$.

1. Three non-robust controllers are designed for the BSD without mass variations. Hereby, we study the importance of including the position dependent nonlinearities in the modeling and controller design steps.

2. We perturb the system parameters, and design new controllers to study the performance sensitivity to the estimated parameters provided in Table 5.1.

3. A robust gain scheduling controller for the BSD with mass variations is synthesized to study the effectiveness of the robustness in presence of mass variations.

4. A controller set for the derived multiple model is designed to study the effect of dividing the range of the mass variation on the closed-loop performance.

All the above is discussed in detail in the following sections.

5.6.1 Single controller for the BSD without mass variations

Considering the parametrization (5.20), the resultant controller, denoted by K_{runout}, compensates for the flexibility and runout effects. To synthesize a controller, denoted by $K_{no-runout}$ which ignores the runout effect, the sinusoidal terms in the parametrization (5.20) are excluded, and only the polynomials are considered

$$\mathcal{O} = \mathcal{O}_0 + \ell\mathcal{O}_1. \tag{5.26}$$

The tracking performances of two controllers, non-robust K_{runout} and $K_{no-runout}$, as well as a PID controller are investigated with no added mass to the table. In the design of K_{runout} and $K_{no-runout}$, the state space form (A_e, B_e, C_e, D_e) of the tuned W_e is

$$\left(\begin{bmatrix} -10 & -10 \\ 8 & 0 \end{bmatrix}, \begin{bmatrix} 64 \\ 0 \end{bmatrix}, \begin{bmatrix} 2.3 & 82.3 \end{bmatrix}, \begin{bmatrix} 0.25 \end{bmatrix} \right). \tag{5.27}$$

The experimental results are shown in Figure 5.14. The control input signal of the PID controller is slightly higher than those of the other two controllers. The spikes in the errors at the beginning and in the middle, when the direction of motion reverses, occur mostly due to the momentary static friction and a backlash-like effects in the nut assembly [21].

A numerical comparison based on the Mean Absolute Error (MAE) is given in Table 5.4. Tracking performance of the PID controller is the worst due to the plant nonlinearities, specially the structural flexibility. Recall that the flexibility is taken into account in both $K_{no-runout}$ and K_{runout}, but the runout effect is considered in K_{runout} only. Two cases are reported in this table. "Full run" addresses the data for the entire time span of the run, i.e. $t \in [0,\ 4.4]\ s$, while "Tracking" refers to the time span of $t \in [0.25,\ 2.2] \bigcup [2.52,\ 4.4]\ s$, which ignores the error spikes. The track-

Table 5.4: Controllers tracking error results (μm).

	PID	$K_{no-runout}$	K_{runout}
Full run	32.6	21.2	17.1
Tracking	21.0	11.6	9.3

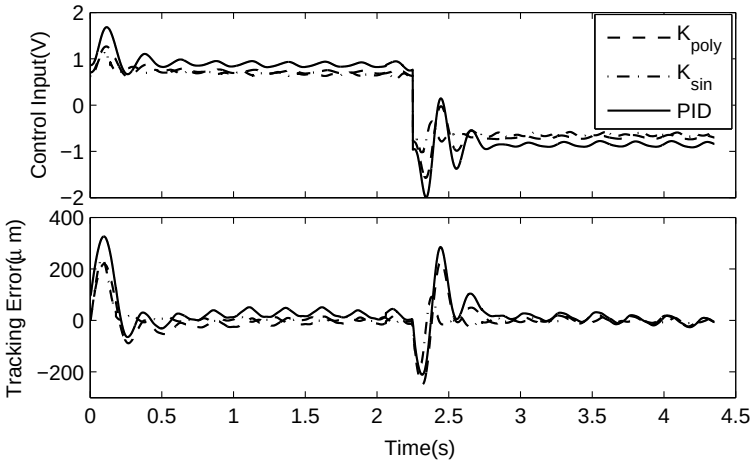


Figure 5.14: Tracking errors and control inputs.

ing performance of K_{runout} is 55.7 % and 19.8 % better than that of the PID and $K_{no-runout}$ controllers, respectively. It justifies the use of K_{runout} in high precision application even though the design procedure is more involved.

5.6.2 Performance sensitivity of the BSD without mass variations

Now, we study the sensitivity of the closed-loop performance to the accuracy of the estimated model. The estimated nominal values of parameters k, ζ, ω_n, J^* (in Table 5.1) are perturbed one at a time by ± 5 %. Different non-robust K_{runout} controllers, with common weighting functions (5.13) and (5.27), are designed based on the perturbed models. Tracking errors are pro-

Table 5.5: Controllers tracking error (μm), and the percentage increase of errors in comparison with K_{runout} tracking error in Table 5.4.

	k	ζ	ω_n	J^*
-5 %	9.9 (6.4%)	9.4 (1.1%)	11.4 (22.6%)	9.2 (-1.1%)
$+5$ %	11.3 (21.5%)	9.5 (2.1 %)	10.7 (15.1%)	11.1 (19.3%)

vided in Table 5.5, when nominal values increase and decrease by 5 %. Also, a percentage showing how much the tracking error of K_{runout} increases by perturbing each parameter is shown in parentheses.

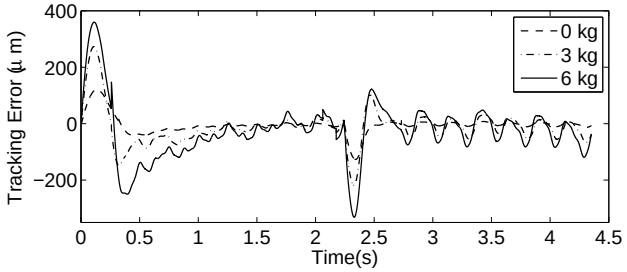
Small tracking error changes can be neglected since they are related to the experimental setup repeatability. By perturbing some parameters, the performance degrades more than 20%. Therefore, it is critical to either estimate the model accurately, or if not possible, to design robust controllers.

5.6.3 Single controller for the BSD with mass variations

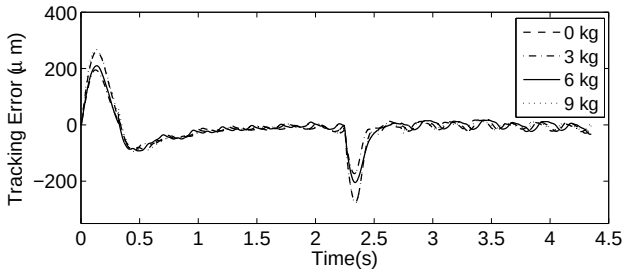
In this section, we focus on the role and importance of the robustness of the controllers. A new robust controller K_{runout} is designed, where the state space form (A_e, B_e, C_e, D_e) of the tuned W_e is selected as

$$\left(\begin{bmatrix} -3 & -1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 16 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.3 & 21.7 \end{bmatrix}, \begin{bmatrix} 0.2 \end{bmatrix} \right). \tag{5.28}$$

Figure 5.15 shows the tracking performances of the closed-loop plant when the runout effect is taken into consideration, and different masses are added to the table. A comparison between the tracking errors is given in Table 5.6. Recall that the non-robust K_{runout} is synthesized for the plant without added mass while the robust K_{runout} is robust over the table mass variation. Because of not considering robustness, the non-robust K_{runout} yields the best



(a) Non-robust controller



(b) Robust controller

Figure 5.15: Tracking errors of robust and non-robust K_{runout} controller for different added masses to the table.

Table 5.6: Tracking error results of the controllers in the third scenario.

Added mass (kg)	Robust K_{runout} (μm)	Non-robust K_{runout} (μm)
0	20.2	9.3
3	19.1	29.4
6	19.0	54.3
9	20.9	Unstable

performance with an MAE of 9.3 μm. However, its performance deteriorates by adding some mass to the table, and eventually, the closed-loop system becomes unstable when a mass of 9 kg is added. On the other hand, the performance of the robust K_{runout} remains uniformly acceptable over different added masses.

Table 5.7: Tracking errors and performance improvement (%) calculated by $100(MAE_{single} - MAE_{multiple})/MAE_{single}$.

Added mass (kg)	0	3	6	9
Single (μm)	20.2	19.1	19.0	20.9
multiple (μm)	15.1	14.5	15.9	16.6
Improvement (%)	25.2	24.1	16.3	20.6

5.6.4 Multiple controllers for the BSD with mass variations

In this section, we study the effect of dividing the range of the mass variation on the closed-loop performance. A robust controller is designed for each LPV model in the derived multiple model. In controller design for both partitions, an identical W_e is selected as

$$\left(\begin{bmatrix} -5 & -2.5 \\ 4 & 0 \end{bmatrix}, \begin{bmatrix} 32 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.44 & 31.23 \end{bmatrix}, \begin{bmatrix} 0.21 \end{bmatrix} \right). \tag{5.29}$$

The results for the robust multiple controller show similar patterns to those shown in Figure 5.15(b) but with less tracking error. A numerical comparison is made in Table 5.7. These results imply that the multiple controller had potential for improving the track-following performance of the BSD by more than 16%.

5.7 Conclusions

The variations in the dynamics of BSD systems due to the structural flexibility, runout, and workpiece mass variation were studied in Chapter 5. Tracking controllers for a BSD were designed, which considered flexibility and runout, as well as mass change. These three factors were explicitly incorporated in LPV models. To build the LPV models, it was determined

how the system parameters were affected by two scheduling and uncertain variables, namely, the measurable table position and the uncertain mass of the table. For the LPV models, we designed controllers which were scheduled by the table position and were robust over the table mass. The performances of the designed controllers were examined on the BSD experimental setup.

It was experimentally demonstrated that the tracking performance improved significantly by taking into account the runout effect in modeling and controller design. Also, it was shown that more than 20% performance degradation occurs by perturbing some parameters by 5%. Therefore, it is critical to either estimate the model accurately, or if not possible, to design robust controllers. In addition, it was verified that the consideration of robustness against mass variation in the design stage was necessary for maintaining the stability and a uniform tracking performance. Also, multiple model derivation and the performance of the corresponding controller set were demonstrated and discussed thoroughly. It was shown that the multiple controller had potential for improving the track-following performance of the BSD by more than 16%.

Chapter 6

Conclusions, Contributions and Future Research Directions

6.1 Conclusions

This thesis considered variations in the dynamics of linear systems, and tackled modeling of Linear Time-Invariant (LTI) and Linear Parameter Varying (LPV) plants. These variations were assumed to be parametric, and caused by two types of variables, uncertain and scheduling. The variations in the dynamics make the controller design challenging, and, to successfully overcome this challenge, two methods were proposed in this thesis.

The method developed in Chapter 2 generated a connected model set based on a given set of system response data. This method interpolated the given system dynamics to cover the variations associated with not only these systems but also the intermediate plant dynamics. The connected model set was constructed to become simple and tight, leading to both nonconservatism and reduced computational complexity in subsequent controller design, and hence, to improve the performance. We applied our algorithm to an illustrative and a practical example, and obtained accurate model in both cases.

In Chapter 3, a method was developed to derive a family of discrete model sets for a given set of system response data. The idea was to divide the given set into the smallest possible number of partitions in such a way that a desired closed-loop performance was satisfied for all partitions by designing one controller for each partition. The effectiveness of the proposed method was verified through an illustrative example. Also, the effectiveness of combining this method and the one developed in Chapter 2 was shown through an example.

In Chapter 4, we studied the dynamics of Hard Disk Drive (HDD) systems, especially, the variations in the dynamics due to the change in temperatures and limited precision in the production line. A tight uncertainty model was derived based on a set of experimental frequency response data, and an H_∞ controller was synthesized. The experimental results were demonstrated and discussed in the frequency and time domains.

The variations in the dynamics of Ball Screw Drive (BSD) systems due to the structural flexibility, runout, and workpiece mass variation were studied in Chapter 5. Tracking controllers for a BSD were designed, which considered flexibility and runout, as well as mass change. These three factors were explicitly incorporated in LPV models. To build the LPV models, it was determined how the system parameters were affected by two scheduling and uncertain variables, namely, the measurable table position and the uncertain mass of the table. We designed controllers which were scheduled by the table position and were robust over the table mass. The performances of the designed controllers were examined on the BSD experimental setup.

It was experimentally demonstrated that the tracking performance improved significantly by taking into account the runout effect in modeling and controller design. Also, it was shown that more than 20% performance degradation occurs by perturbing some parameters by 5%. Therefore, it is critical to either estimate the model accurately, or if not possible, to design robust controllers. In addition, it was verified that the consideration

of robustness against mass variation in the design stage was necessary for maintaining the stability and a uniform tracking performance. Also, multiple model derivation and the performance of the corresponding controller set were demonstrated and discussed thoroughly. It was shown that the multiple controller had potential for improving the track-following performance of the BSD by more than 16%.

6.2 Summary of contributions

The contributions of this thesis are as follows.

- *The connected model set derivation method*
  - The idea of the principal curves methodology in a multi-dimensional fashion is employed to detect the nonlinear correlations between parameters of the system dynamics. Therefore, the model can be parameterized by the minimum number of independent variables. The number of independent variables can be detected readily by trial and error using the developed method.
  - This method does not need any information about the way that uncertain and scheduling variables affect the physical parameters of the system, such as natural frequency and damping ratio, in contrast to most of the literature in this field [74, 76]. Therefore, the method is applicable to any form of the transfer functions, e.g., the general form.
  - The developed method is applicable to LTI and LPV systems. Such applications are demonstrated through examples in this thesis.
- *The family of discrete model sets derivation method*

- A relaxed version of the normalized cut methodology is developed and used in an algorithm to divide a given set of LTI system responses into the smallest possible number of partitions in such a way that a desired performance objective is satisfied for all partitions by designing one controller for each partition. To the best of our knowledge, there is no literature on derivation of a family of discrete model sets based on the desired closed-loop performance as described in this thesis.

- *Controller design for flexible BSDs with runout effect and mass variation*

  - The dynamics of BSDs is studied, where the position and mass dependent variations are examined in detail from a controller design point of view. LPV models are derived to represent the dynamics of the BSDs, which happens to be time-varying and uncertain. The modeling results prove the effectiveness of the proposed method. To the best of our knowledge, there is no literature considering the effects of the structural flexibility, runout, and mass variations in the BSD systems simultaneously.
  - Tracking controller design method is proposed for BSDs, which consists of a disturbance observer and a robust gain scheduling controller. Tracking performances of a number of different controllers are compared, and it is demonstrated the importance of including flexibility, runout effect, and mass variations in modeling and controller design. Also, experimental results show more than 16% improvement in the tracking performance by implementing a multiple controller.

6.3 Future research directions

This section recommends a number of potential future research directions.

6.3.1 Uncertainty modeling for stochastic robust controller

The method, which is developed to derive connected model sets, is successful in uncertainty modeling for HDD and BSD systems. One assumption is inherently made that these systems are deterministic, and not stochastic. Recently, stochastic robust control has received an increasing attention. Many results about conventional robust control are extended in stochastic setting since Hinrichsen [40] proposed the stochastic H_∞ control. For instance, Xu and Chen [114] proposed the sufficient condition for the solvability of robust H_∞ control problem for uncertain stochastic delay systems. One topic of research which emanates from this thesis is to extend the developed method to model the uncertainty for the robust controller design, when stochastic perturbations exist.

6.3.2 Performance oriented connected model set derivation

The algorithm, which is explained in Chapter 2, attempts to circumvent the possibility of conservative performance of the closed-loop control systems. We do not take into account if there is a desired closed-loop performance. Refer to the special case in Table 1.1 which is denoted by "Chapter 2". Since the closed-loop performance depends on the characteristics of the model set, considering the performance in deriving the connected model sets can be beneficial. Therefore, One research topic is to consider the special case denoted by "Future work" in Table 1.1.

By considering the controller performance, we may need to divide the connected model set, and derive a family of connected model sets. The

algorithm, which is explained in Chapter 2, derives a connected model set with the minimum size, and does not take into consideration the possibility of dividing the set, and deriving a family of connected model sets. On possible research direction is to extend the developed method in such a way that the model set is divided into a number of subsets if necessary based on the closed-loop performance.

6.3.3 Advanced performance oriented family of discrete model sets derivation

In the algorithm explained in Chapter 3, first we generate a number of partition sets based on a clustering cost function, and then, the best partition set is chosen based on the closed-loop performance. The performance of this method can be improved by combining these two steps, and consequently, an optimization can be formulated to provide the best partition set in one step. The new method may be faster and computationally simpler and results in a better family of discrete model sets.

6.3.4 Switching controllers for BSDs

In Chapter 5, we derive a multiple closed-loop system by dividing the mass variation to improve the tracking performance. However, deriving the multiple system should agree with the machining operation. In this study, we divide the range of mass variations into two partitions, and assume that the table mass stays within one of these partitions during machining. However, in the machining operations, we may need to violate this boundary. For example, the machining starts with a mass within the range of $(23, 29]$ kg, and during cutting operations the mass reduces to the range of the other partition $[20, 23]$ kg[1]. In this case, we need to consider switching between two controllers. Therefore, it is essential to design switching controllers, which

[1]These numbers are from the results in Chapter 5.

guarantee stability and robust tracking, when switching occurs.

6.3.5 BSD table mass estimation in real time

We assume that the value of the BSD table mass is not available in real time during the operations. This lack of information causes loss of performance due to the mass variations. One extension of the method proposed in this thesis is to estimate the table mass in real time. Therefore, the tracking performance can be improved by updating the mass-dependent parameters of the controllers as a function of the table mass, similar to the position-dependent gain scheduling controllers described in Chapter 5.

Bibliography

[1] H. Akaike, *A new look at the statistical model identification*, Automatic Control, IEEE Transactions on **19** (2002), no. 6, 716–723. → pages 7, 23, 40, 78

[2] P. Apkarian and R.J. Adams, *Advanced gain-scheduling techniques for uncertain systems*, IEEE Transactions on Control Systems Technology **6** (1998), no. 1, 21–32. → pages 86, 92, 94

[3] E. Azadi Yazdi, M. Sepasi, F. Sassani, and R. Nagamune, *Automated multiple robust track-following control system design in hard disk drives*, To appear in IEEE Transactions on Control System Technology. → pages iv

[4] ———, *Automated multiple robust track-following control system design in hard disk drives*, ASME Dynamic Systems and Control Conference (Boston, MA), September 2010, pp. 4163(1)–4163(6). → pages iv, 69

[5] G. Balas, R. Chiang, A. Packard, and M. Safonov, *Robust control toolbox*, Matlab Users Guide **3** (2005). → pages 56, 92

[6] Y. Bar-Shalom, X.R. Li, T. Kirubarajan, and J. Wiley, *Estimation with applications to tracking and navigation*, Wiley Online Library, 2001. → pages 21

[7] M. Belkin and P. Niyogi, *Laplacian eigenmaps and spectral techniques for embedding and clustering*, Advances in Neural Information Processing System **1** (2002), 585–592. → pages 17

[8] H.A.P. Blom and Y. Bar-Shalom, *The interacting multiple model algorithm for systems with Markovian switching coefficients*, Automatic Control, IEEE Transactions on **33** (1988), no. 8, 780–783. → pages 21

[9] M. Brand, *Charting a manifold*, Advances in Neural Information Processing System (2003), 985–992. → pages 17

[10] MR Brito, EL Chavez, AJ Quiroz, and JE Yukich, *Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection*, Statistics & Probability Letters **35** (1997), no. 1, 33–42. → pages 45

[11] G.D. Buckner, H. Choi, and N.S. Gibson, *Estimating model uncertainty using confidence interval networks: Applications to robust control*, Journal of Dynamic Systems, Measurement, and Control **128** (2006), 626–635. → pages 16

[12] B. Chen, J. Miao, and F.E.H. Tay, *Fabrication and characterization of DRIE-micromachined electrostatic microactuators for hard disk drives*, Microsystem Technologies **13** (2007), no. 1, 11–19. → pages 60

[13] J. Chen and G. Gu, *Control-oriented System Identification: An H_∞ Approach*, John Wiley & Sons, 2000. → pages 2

[14] L.K. Chen and A.G. Ulsoy, *Identification of a driver steering model, and model mncertainty, mrom driving simulator data*, Journal of Dynamic Systems, Measurement, and Control **123** (2001), 623–629. → pages 16

[15] T.L. Chen and R. Horowitz, *Design, fabrication and dynamic analysis of a PZT-actuated siliconsuspension*, Proceedings of American Control Conference, vol. 2, 2001. → pages 2, 60

[16] G. Cheng and K. Peng, *Robust composite nonlinear feedback control with application to a servo positioning system*, IEEE Transactions on Industrial Electronics **54** (2007), no. 2, 1132–1140. → pages 71

[17] J.U. Cho, Q.N. Le, and J.W. Jeon, *An FPGA-based multiple-axis motion control chip*, IEEE Transactions on Industrial Electronics **56** (2009), no. 3, 856–870. → pages 72

[18] J. Choi, R. Nagamune, and R. Horowitz, *Multiple robust controller design based on parameter dependent lyapunov functions*, Proceedings of 17th International Symposium on Mathematical Theory of Networks and Systems, 2006. → pages 21

[19] R. Conway and R. Horowitz, *A μ-synthesis approach to guaranteed cost control in track-following servos*, The International Federation of Automatic Control, 2008, pp. 833–838. → pages 62

[20] T.F. Cox and M.A.A. Cox, *Multidimensional scaling*, CRC Press, 2001. → pages 18

[21] JF Cuttino, TA Dow, and BF Knight, *Analytical and experimental identification of nonlinearities in a single-nut, preloaded ball screw*, Journal of Mechanical Design **119** (1997), 15–19. → pages 98

[22] F.A. Cuzzola, *A multivariable and multi-objective approach for the control of hot-strip mills*, Journal of Dynamic Systems, Measurement, and Control **128** (2006), 856–868. → pages 86

[23] M. Defoort and T. Murakami, *Sliding-mode control scheme for an intelligent bicycle*, IEEE Transactions on Industrial Electronics **56** (2009), no. 9, 3357–3368. → pages 95

[24] D. Degenring, C. Froemel, G. Dikta, and R. Takors, *Sensitivity analysis for the reduction of complex metabolism models*, Journal of Process Control **14** (2004), no. 7, 729–745. → pages 16

[25] D. Dong and TJ McAvoy, *Nonlinear principal component analysisBased on principal curves and neural networks*, Computers and Chemical Engineering **20** (1996), no. 1, 65–78. → pages 20

[26] D.L. Donoho and C. Grimes, *Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data*, Proceedings of the National Academy of Sciences **100** (2003), no. 10, 5591–5596. → pages 17

[27] AL Dontchev, RP Polis, and VM Veliov, *A dual method for parameter identification under deterministicuncertainty*, IEEE Transactions on Automatic Control **45** (2000), no. 7, 1341–1346. → pages 17

[28] C. Du, SS Ge, and FL Lewis, *H_∞ compensation of external vibration impact on servo performance of hard disk drives in mobile applications*, International Journal of Adaptive Control and Signal Processing **22** (2008), no. 4, 374–387. → pages 66

[29] J. Du, C. Song, and P. Li, *Application of gap metric to model bank determination in multilinear model approach*, Journal of Process Control **19** (2009), no. 2, 231–240. → pages 21

[30] S.A. Dudani, *The distance-weighted k-nearest-neighbor rule*, IEEE Transactions on Systems, Man and Cybernetics (1976), no. 4, 325–327. → pages 44

[31] R.A. Fisher, *The use of multiple measurements in taxonomic problems*, Annals of Eugenics **7** (1936), no. 2, 179–188. → pages 17

[32] Y. Fu and T. Chai, *Nonlinear multivariable adaptive control using multiple models and neural networks*, Automatica **43** (2007), no. 6, 1101–1110. → pages 21

[33] H. Fujita, K. Suzuki, M. Ataka, and S. Nakamura, *A microactuator for head positioning system of hard disk drives*, IEEE Transactions on Magnetics **35** (1999), no. 2 Part 1, 1006–1010. → pages 2, 60

[34] K. Fukunaga, *Introduction to statistical pattern recognition*, Academic Press, 1990. → pages 16

[35] S. Guattery and G.L. Miller, *On the quality of spectral separators*, SIAM Journal on Matrix Analysis and Applications **19** (1998), no. 3, 701–719. → pages 22

[36] J. Harris, *Algebraic Geometry: A First Course*, Springer, 1992. → pages 30

[37] T. Hastie, *Principal curves and surfaces*, Tech. report, 1984. → pages 20

[38] T. Hastie and W. Stuetzle, *Principal curves.*, Journal of the American Statistical Association **84** (1989), no. 406, 502–516. → pages 19, 20

[39] R. Hibino, M. Osawa, K. Kono, and K. Yoshizawa, *Robust and simplified design of slip control system for torque converter lock-up clutch*, Journal of Dynamic Systems, Measurement, and Control **131** (2009), 011008–011017. → pages 86

[40] D. Hinrichsen and A.J. Pritchard, *Stochastic H_∞*, SIAM Journal on Control and Optimization **36** (1998), no. 5, 1504–1538. → pages 108

[41] GE Hinton and RR Salakhutdinov, *Reducing the dimensionality of data with neural networks*, Science **313** (2006), no. 5786, 504–507. → pages 17

[42] H. Hotelling, *Analysis of a complex of statistical variables into principal components*, Journal of Educational Psychology **24** (1933), no. 6, 417–441. → pages 17

[43] C. Hu, B. Yao, and Q. Wang, *Coordinated adaptive robust contouring control of an industrial biaxial precision gantry with cogging force compensations*, IEEE Transactions on Industrial Electronics **57** (2010), no. 5, 1746–1754. → pages 2, 72

[44] Q. Hu, C. Du, L. Xie, and Y. Wang, *Discrete-time sliding mode control with time-varying surface for hard disk drives*, Control Systems Technology, IEEE Transactions on **17** (2009), no. 1, 175–183. → pages 66

[45] T. Imamura, M. Katayama, Y. Ikegawa, T. Ohwe, R. Koishi, and T. Koshikawa, *MEMS-based integrated head/actuator/slider for hard disk drives*, IEEE/ASME Transactions on Mechatronics **3** (1998), no. 3, 166–174. → pages 2, 60, 63

[46] ———, *MEMS-based integrated head/actuator/slider for hard disk drives*, Mechatronics, IEEE/ASME Transactions on **3** (2002), no. 3, 166–174. → pages 63

[47] A.K. Jain, M.N. Murty, and P.J. Flynn, *Data clustering: a review*, ACM computing surveys (CSUR) **31** (1999), no. 3, 264–323. → pages 21

[48] J.W. Jaromczyk and G.T. Toussaint, *Relative neighborhood graphs and their relatives*, Proceedings of the IEEE **80** (2002), no. 9, 1502–1517. → pages 44

[49] B. Jones, *Matlab Statistics Toolbox*, The MathWorks, Inc. Natick, MA, USA (1997). → pages 50

[50] S. Kanev, C. Scherer, M. Verhaegen, and B. De Schutter, *Robust output-feedback controller design via local BMI optimization*, Automatica **40** (2004), no. 7, 1115–1127. → pages 12

[51] S.E. Karisch and F. Rendl, *Semidefinite programming and graph equipartition*, Topics in Semidefinite and Interior-Point Methods (1998), 77–95. → pages 22

[52] P.V. Kokotovic, L. Menini, T. Nicosia, L. Zaccarian, and C.T. Abdallah, *Current Trends in Nonlinear Systems and Control*, Springer, 2006. → pages 16

[53] RL Kosut, MK Lau, SP Boyd, I.S. Inc, and CA Santa Clara, *Set-membership identification of systems with parametric and nonparametric uncertainty*, IEEE Transactions on Automatic Control **37** (1992), no. 7, 929–941. → pages 2

[54] M.A. Kramer, *Nonlinear principal component analysis using autoassociative neural networks*, AIChE Journal **37** (1991), no. 2, 233–243. → pages 19

[55] U. Kruger, J. Zhang, and L. Xie, *Developments and applications of nonlinear principal component analysis-a review*, Lecture Notes in Computational Science and Engineering **58** (2007), 1. → pages 18

[56] A. Kwiatkowski and H. Werner, *Parameter reduction for LPV systems via principle components analysis*, Proceedings of the International Federation of Automatic Control, 2005. → pages 32

[57] J.T.Y. Kwok and I.W.H. Tsang, *The pre-image problem in kernel methods*, IEEE Transactions on Neural Networks **15** (2004), no. 6, 1517–1525. → pages 20

[58] S. Lafon and A.B. Lee, *Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization*, IEEE Transactions on Patern Analysis and Machine Intelligence (2006), 1393–1403. → pages 18

[59] C.Y. Lai, F.L. Lewis, V. Venkataramanan, X. Ren, S.S. Ge, and T. Liew, *Disturbance and friction compensations in hard disk drives using neural networks*, IEEE Transactions on Industrial Electronics **57** (2010), no. 2, 784–792. → pages 95

[60] D. Lainiotis, *Optimal adaptive estimation: Structure and parameter adaption*, Automatic Control, IEEE Transactions on **16** (1971), no. 2, 160–170. → pages 21

[61] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan, *Learning the kernel matrix with semidefinite programming*, The Journal of Machine Learning Research **5** (2004), 27–72. → pages 17

[62] A. Laub, M. Heath, C. Paige, and R. Ward, *Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms*, IEEE Transactions on Automatic Control **32** (2002), no. 2, 115–122. → pages 82

[63] R. Li et al., *Hybrid estimation techniques*, Control and Dynamic Systems **76** (1996), 213–287. → pages 21

[64] R. Li, MA Henson, and MJ Kurtz, *Selection of model parameters for off-line parameter estimation*, IEEE Transactions on Control Systems Technology **12** (2004), no. 3, 402–412. → pages 16

[65] R. Li and V.P. Jilkov, *Survey of maneuvering target tracking. Part V. Multiple-model methods*, Aerospace and Electronic Systems, IEEE Transactions on **41** (2005), no. 4, 1255–1321. → pages 21

[66] X.R. Li, *Engineer's guide to variable-structure multiple-model estimation for tracking*, Multitarget-multisensor tracking: Applications and advances. **3** (2000), 499–567. → pages 21

[67] J.B. Little, Mass.) MathWorks (Natick, and L. Shure, *Signal processing toolbox for use with MATLAB: user's guide*, MathWorks, 1988. → pages 24, 40

[68] L. Ljung, *System identification toolbox*, The Matlab users guide. → pages 24, 40

[69] ———, *System Identification: Theory for the User.*, Prentice-Hall, 1999. → pages 2

[70] Y. Lou, P. Gao, B. Qin, G. Guo, E.H. Ong, A. Takada, and K. Okada, *Dual-Stage servo with on-slider PZT microactuator for hard disk drives*, IEEE Transactions on Magnetics **38** (2002), no. 5, 2183. → pages 2, 60

[71] M. Lovera and G. Mercere, *Identification for gain-scheduling: a balanced subspace approach*, American Control Conference, 2007, pp. 858–863. → pages 82

[72] B. Lu and F. Wu, *Switching LPV control designs using multiple parameter-dependent Lyapunov functions*, Automatica **40** (2004), no. 11, 1973–1980. → pages 21

[73] D. Magill, *Optimal adaptive estimation of sampled stochastic processes*, Automatic Control, IEEE Transactions on **10** (1965), no. 4, 434–439. → pages 21

[74] R. Nagamune and J. Choi, *Parameter reduction of nonlinear least-squares estimates via nonconvex optimization*, 2008 American Control Conference, Seattle, Washington, USA, 1298–1303. → pages 16, 106

[75] ———, *Parameter reduction of nonlinear least-squares estimates via the singular value decomposition*, Proceedings of the International Federation of Automatic Control, 12383–12388. → pages 32

[76] ———, *Parameter Reduction in Estimated Model Sets for Robust Control*, Journal of Dynamic Systems, Measurement, and Control **132** (2010), 021002.1–021002.10. → pages 16, 32, 78, 106

[77] R. Nagamune, X. Huang, and R. Horowitz, *Robust control synthesis techniques for multirate and multi-sensing track-following servo systems in HDDs*, ASME Journal of Dynamic Systems, Measurement and Control. → pages 62

[78] K. Nagaoka, A. Matsubara, T. Fujita, and T. Sato, *Analysis method of motion accuracy using nc system with synchronized measurement of tool-tip position*, International Journal of Automation Technology **3** (2009), no. 4, 394–400. → pages 72

[79] P. Naphon and S. Maharchon, *Temperature distribution of read/write head soldering with ribbon cable of HDD*, International Communications in Heat and Mass Transfer **37** (2010), no. 4, 379–384. → pages 63

[80] A. Ng, M. Jordan, and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, Advances in Neural Information Processing Systems, 2001, pp. 849–856. → pages 44

[81] K. Ohnishi, N. Matsui, and Y. Hori, *Estimation, identification, and sensorless control in motion control system*, Proceedings of the IEEE **82** (1994), no. 8, 1253–1265. → pages 95

[82] K. Ohno, Y. Abe, and T. Maruyama, *Robust following control design for hard disk drives*, Control Applications, 2001.(CCA'01). Proceedings of the 2001 IEEE International Conference on, IEEE, 2002, pp. 930–935. → pages 63

[83] C. Okwudire, *Modeling and control of high speed machine tool feed drives*, Ph.D. Thesis, University of British Columbia (2009). → pages 73, 81

[84] C. Okwudire and Y. Altintas, *Minimum tracking error control of flexible ball screw drives using a discrete-time sliding mode controller*, Journal of Dynamic Systems, Measurement, and Control **131** (2009), 051006–1–051006–12. → pages 72

[85] C.E. Okwudire and Y. Altintas, *Hybrid modeling of ball screw drives with coupled axial, torsional, and lateral dynamics*, Journal of Mechanical Design **131** (2009), 071002. → pages 72

[86] M.V. Ramana, L. Tunçel, and H. Wolkowicz, *Strong duality for semidefinite programming*, SIAM Journal on Optimization **7** (1997), no. 3, 641–662. → pages 49

[87] Li Rong et al., *Multiple-model estimation with variable structure-Part VI: Expected-mode augmentation*, IEEE transactions on aerospace and electronic systems **41** (2005), no. 3, 853–867. → pages 21

[88] P. Rosa, C. Silvestre, J.S. Shamma, and M. Athans, *Multiple-model adaptive control with set-valued observers*, Proceedings of the 48th IEEE Conference on Decision and Control., IEEE, 2010, pp. 2441–2447. → pages 21

[89] S. Roweis, L.K. Saul, and G.E. Hinton, *Global Coordination of Local Linear Models*, Proceedings of Advances in Neural Information Processing Systems, MIT Press, 2002. → pages 17

[90] S.T. Roweis and L.K. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science **290** (2000), no. 5500, 2323–2326. → pages 17

[91] WJ Rugh and J.S. Shamma, *A survey of research on gain-scheduling*, Automatica **36** (2000), no. 10, 1401–1425. → pages 2, 72

[92] J.W. Sammon, *A nonlinear mapping for data structure analysis*, IEEE Transactions on Computers **18** (1969), no. 5, 401–409. → pages 18

[93] C. Scherer, P. Gahinet, and M. Chilali, *Multiobjective output-feedback control via LMI optimization*, IEEE Transactions on Automatic Control **42** (1997), no. 7, 896–911. → pages 91

[94] B. Scholkopf, A. Smola, and K.R. Muller, *Nonlinear component analysis as a kernel eigenvalue problem*, Neural Computation **10** (1998), no. 5, 1299–1319. → pages 19

[95] B. Scholkopf and A.J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT Press, 2002. → pages 19

[96] D. Sepasi, F. Sassani, and R. Nagamune, *Tracking control of flexible ball screw drives with runout effect and mass variation*, To appear in IEEE Transactions on Industrial Electronics. → pages iv

[97] M. Sepasi, F. Sassani, and R. Nagamune, *Parameter uncertainty modeling using the multi-dimensional principal curves*, Journal of Dynamic

Systems, Measurement and Control **132** (2010), 054501–054507. → pages 4

[98] ———, *Tracking control of flexible ball screw drives with runout effect compensation*, ASME Dynamic Systems and Control Conference (Boston, MA), September 2010, pp. 4039(1)–4039(6). → pages iv

[99] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2002), no. 8, 888–905. → pages 22

[100] H. Shim and Y.J. Joo, *State space analysis of disturbance observer and a robust stability condition*, Decision and Control, 2007 46th IEEE Conference on, IEEE, 2007, pp. 2193–2198. → pages 95

[101] C. Sun and J. Hahn, *Parameter reduction for stable dynamical systems based on Hankel singular values and sensitivity analysis*, Chemical Engineering Science **61** (2006), no. 16, 5393–5403. → pages 16

[102] N. Tagawa, K.I. Kitamura, and A. Mori, *Design and fabrication of MEMS-based active slider using double-layered composite PZT thin film in hard disk drives*, IEEE Transactions on Magnetics **39** (2003), no. 2, 926–931. → pages 60

[103] H.D. Taghirad and E. Jamei, *Robust performance verification of adaptive robust controller for hard disk drives*, Industrial Electronics, IEEE Transactions on **55** (2008), no. 1, 448–456. → pages 66

[104] J.B. Tenenbaum, *Mapping a manifold of perceptual observations*, Advances in Neural Information Processing System (1998), 682–688. → pages 18

[105] J.B. Tenenbaum, V. Silva, and J.C. Langford, *A global geometric framework for nonlinear dimensionality reduction*, 2000, pp. 2319–2323. → pages 17

[106] H. Toshiyoshi, M. Mita, and H. Fujita, *A MEMS piggyback actuator for hard-disk drives*, Journal of Microelectromechanical Systems **11** (2002), no. 6, 648–654. → pages 60

[107] R. Tóth, F. Felici, P.S.C. Heuberger, and P.M.J. Van den Hof, *Discrete time lpv i/o and state space representations, differences of behavior and pitfalls of interpolation*, Proceedings of the European Control Conference, 2007, pp. 5418–5425. → pages 82

[108] C.C. Tsai, H.C. Huang, and S.C. Lin, *Adaptive neural network control of a self-balancing two-wheeled scooter*, IEEE Transactions on Industrial Electronics **57** (2010), no. 4, 1420–1428. → pages 2

[109] K.K. Varanasi and S.A. Nayfeh, *The dynamics of lead-screw drives: Low-order modeling and experiments*, Journal of Dynamic Systems, Measurement, and Control **126** (2004), 388–396. → pages 78

[110] U. Von Luxburg, *A tutorial on spectral clustering*, Statistics and Computing **17** (2007), no. 4, 395–416. → pages 44

[111] R. Wai, J. Lee, and K. Chuang, *Real-Time PID control strategy for Maglev transportation system via particle swarm optimization*, IEEE Transactions on Industrial Electronics (2010), no. 99, 1. → pages 72

[112] Y. Xia, M. Fu, H. Yang, and G.P. Liu, *Robust sliding-mode control for uncertain time-delay systems based on delta operator*, IEEE Transactions on Industrial Electronics **56** (2009), no. 9, 3646–3655. → pages 72

[113] E. Xing, E.P. Xing, M. Jordan, and M.I. Jordan, *On semidefinite relaxations for normalized k-cut and connections to spectral clustering*, Tech. report, 2003. → pages 22, 46, 47, 125

[114] S. Xu and T. Chen, *Robust H_∞ control for uncertain stochastic systems with state delay*, IEEE Transactions on Automatic Control **47** (2002), no. 12, 2089–2094. → pages 108

[115] P. Yan and H. Ozbay, *On switching controllers for a class of linear parameter varying systems*, Systems & Control Letters **56** (2007), no. 7-8, 504–511. → pages 21

[116] K.Z. Yao, B.M. Shaw, B. Kou, K.B. McAuley, and DW Bacon, *Modeling ethylene/butene copolymerization with multi-site catalysts: parameter estimability and experimental design*, Polymer Reaction Engineering **11** (2003), no. 3, 563–588. → pages 16

[117] E.A. Yazdi and R. Nagamune, *Multiple robust H_∞ controller design using the nonsmooth optimization method*, International Journal of Robust and Nonlinear Control **20** (2010), no. 11, 1197–1212. → pages 21, 67, 68

[118] F. Zhang, K.M. Grigoriadis, M.A. Franchek, and I.H. Makki, *Linear parameter-varying lean burn air-fuel ratio control for a spark ignition engine*, Journal of Dynamic Systems, Measurement, and Control **129** (2007), 404–414. → pages 86

[119] Z. Zhang and H. Zha, *Principal manifolds and nonlinear dimensionality reduction via tangent space alignment*, Journal of Shanghai University (English Edition) **8** (2004), no. 4, 406–424. → pages 17

[120] K. Zhou and J.C. Doyle, *Essentials of robust control*, vol. 104, Prentice Hall New Jersey, 1998. → pages 2, 11, 20

[121] K. Zhou, J.C. Doyle, and K. Glover, *Robust and optimal control*, Prentice Hall Englewood Cliffs, NJ, 1996. → pages 90, 92

[122] T. Zhou and D. Tao, *Fast gradient clustering*, NIPS 2009 Workshop on Discrete Optimization in Machine Learning: Submodularity, Sparsity & Polyhedra (NIPS: DISCML), 2009, pp. 1–6. → pages 22

[123] O. Zirn, C. Jaeger, and T. Scholler, *Model based control of machine tool manipulators*, IEEE International Symposium on Industrial Electronics, 2008, pp. 1267–1274. → pages 72

Appendix A: Relaxed Form of the Optimization (3.9)

It is desired to show that the combinatorial optimization problem

$$\begin{aligned} \min_{\hat{Y}} \ & \mathrm{Tr}[(\hat{Y}^T D\hat{Y})^{-1}(\hat{Y}^T(D-A)\hat{Y})] \\ & \hat{Y}^T\hat{Y} = I_Q, \ \hat{Y}_{ij} \in \{1,0\} \end{aligned} \tag{A.1}$$

can be relaxed to a non-combinatorial one as

$$\begin{aligned} \max_{Z} \ & \mathrm{Tr}(WZ) \\ & Zd^{1/2} = d^{1/2}, \mathrm{Tr}(Z) = Q, \\ & Z \geq 0, I \succeq Z \succeq 0, \end{aligned} \tag{A.2}$$

where $W := D^{-1/2}AD^{-1/2}$, A and D are $L \times L$ diagonal matrices, and d is a vector whose entries are diagonal elements of D. The cost function of the optimization (A.1) can be expanded and rewritten as

$$\mathrm{Tr}[(\hat{Y}^T D\hat{Y})^{-1}(\hat{Y}^T D\hat{Y}) - (\hat{Y}^T D\hat{Y})^{-1}(\hat{Y}^T A\hat{Y})]. \tag{A.3}$$

Therefore, the optimization (A.1) is equivalent to

$$\max_{\hat{Y}} \ \mathrm{Tr}[(\hat{Y}^T D\hat{Y})^{-1}(\hat{Y}^T A\hat{Y})],$$
$$\hat{Y} \in \mathbb{R}^{L\times Q},\ \hat{Y}_{ij} \in \{1,0\}. \tag{A.4}$$

The trace operator is invariant under cyclic permutations, i.e.,

$$\mathrm{Tr}(KPMN) = \mathrm{Tr}(MNKP),$$

for matrices K, P, M, and N with proper sizes. Hence, the known and unknown parts of the cost function in (A.4) can be separated as

$$\mathrm{Tr}(A \cdot \hat{Y}(\hat{Y}^T D\hat{Y})^{-1}\hat{Y}^T).$$

Then, by normalizing the unknown part, the cost can be written as

$$\mathrm{Tr}(D^{-1/2}AD^{-1/2} \cdot D^{1/2}\hat{Y}(\hat{Y}^T D\hat{Y})^{-1}\hat{Y}^T D^{1/2}).$$

Using the definition of W, the above can be written as

$$\mathrm{Tr}(WZ),$$

where

$$Z := D^{1/2}\hat{Y}(\hat{Y}^T D\hat{Y})^{-1}\hat{Y}^T D^{1/2}.$$

The matrix Z as defined above has the following properties [113]:

$$Zd^{1/2} = d^{1/2}, \mathrm{Tr}(Z) = Q,$$
$$Z \geq 0, I \succeq Z \succeq 0. \tag{A.5}$$

By deriving the optimizer for the relaxed optimization problem (A.2), we approximate the optimizer of the original optimization problem (A.1).

Appendix B: Standard SDP Form of the Optimization (3.10)

The optimization problem (3.10) can be written in the standard SDP form as

$$\begin{aligned} \max_{\widetilde{Z}} \ & \mathrm{Tr}\left(\begin{bmatrix} W & \\ & 0 \end{bmatrix} \widetilde{Z}\right), \\ & \mathcal{E}\widetilde{Z} = t, \\ & \widetilde{Z} \succeq 0, \end{aligned} \tag{B.1}$$

where W is defined in (3.11), $\mathcal{E} : \boldsymbol{\mathcal{S}}_{2L} \to \mathbb{R}^M$ is a linear operator, where $M = 1.5(L^2 + L) + 1$, and

$$t = \begin{bmatrix} \mathbf{1}_{2L+1} \\ \mathbf{0}_{1.5L^2 - 0.5L} \end{bmatrix}, \tag{B.2}$$

The set $\boldsymbol{\mathcal{S}}_{2L}$ represents a set of symmetric matrices with the order of $2L$.

The detailed version of the first constraint in the optimization (B.1), $\mathcal{E}\widetilde{Z} = t$, is

$$\mathrm{Tr}\left(\begin{bmatrix} (\widetilde{D}^i + (\widetilde{D}^i)^T)/2 & \\ & 0 \end{bmatrix} \widetilde{Z}\right) = 1,\ i = 1, \ldots, L, \tag{B.3}$$

$$\mathrm{Tr}\left(\begin{bmatrix} I/Q & \\ & 0 \end{bmatrix} \widetilde{Z}\right) = 1, \tag{B.4}$$

$$\mathrm{Tr}\left(\begin{bmatrix} B^i & \\ & B^i \end{bmatrix} \widetilde{Z}\right) = 1,\ i = 1, \ldots, L, \tag{B.5}$$

$$\mathrm{Tr}\left(\begin{bmatrix} C^{mn} & \\ & C^{mn} \end{bmatrix} \widetilde{Z}\right) = 0,\ 1 \leq m < n \leq L, \tag{B.6}$$

$$\mathrm{Tr}\left(\begin{bmatrix} & H^{ij} \\ (H^{ij})^T & \end{bmatrix}^T \widetilde{Z}\right) = 0,\ i, j = 1, \ldots, L, \tag{B.7}$$

where the matrices $\widetilde{D}^i, B^i, C^{mn}$ and H^{ij} are similar to the ones defined for the optimization in (3.14).

In the constrains, (B.3) and (B.4) represent $Zd^{1/2} = d^{1/2}$ and $Tr(Z) = Q$ in (3.10), respectively. The constraintes (B.5) and (B.6) guarantee that the diagonal blocks of $\widetilde{Z}$ are Z and $(1 - Z)$, and (B.7) keeps the other entries in $\widetilde{Z}$ zeros.