Title
Stability and Control for Machining of Thin-Walled Structures — A Time-Varying Delayed Distributed Parameter System

Author
Esfandi, Niloufar

Publication Date
2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Stability and Control for Machining of Thin-Walled Structures — A Time-Varying Delayed Distributed Parameter System

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

Niloufar Esfandi

2017

ABSTRACT OF THE DISSERTATION

Stability and Control for Machining of
Thin-Walled Structures — A Time-Varying Delayed
Distributed Parameter System

by

Niloufar Esfandi
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2017
Professor Tsu-Chin Tsao, Chair

Motivated by machining of thin-walled parts this dissertation investigates the modeling and control of dynamic systems, which feature finite-dimensional and distributed-parameter structures interacting with one another over time-varying contacts and time-delayed feedback. The research studies and creates methods for modeling and control of such systems and applies them to a proposed innovative use of industrial robot manipulators and end effectors to act as an active steady rest in the machining of thin-walled parts.

On modeling the combined discrete-continuous structural and machine dynamics with time varying contacts, finite element method (FEM) is employed to discretize and extract dynamic modes of the continuous structure. The time delayed feedback in the machining process is approximated by the semi-discretization method, where only the delay part of the otherwise continuous-time system is approximated by discrete-time delays and proper sampling and holding functions. To enable digital control design, sampling and zero-order hold circuit modeling is applied to the actuators in conjunction with the semi-discretization for the delayed feedback. Followed by lifting technique on the periodic-varying system and order reduction, discrete-time control oriented models for the entire machining system with the actively controlled robots and end effectors are constructed.

Linear optimal control and robust control are investigated and applied to the proposed robot assisted machining system models. In the linear quadratic state estimator feedback control, a new approach is proposed for the formulation of the cost function weighting matrices. The approach essentially places weighting on outputs and their derivatives since output has clear physical meaning in connection to the process performance. The optimal linear state estimators, aka Kalman filters, are designed taking an approach that exploits the dual properties between the linear optimal regulators and state estimators. In the robust control, parameters in the machining process, particularly the cutting stiffness that depends on the tool conditions and part material, are isolated as uncertainties in formulating the control oriented models for μ-synthesis based robust control design.

The above control design methods are applied to the turning of a thin-walled circular wheel with the actively controlled robotic steady rest. The simulation results show that the feedback controlled robotic active steady rest not only can stabilize an otherwise unstable machining system, caused by the well-known machining chatter due to the delayed feedback, but also can improve the machined forms and precision. The increased stability margins with higher speeds and larger depths of cut than possible without the proposed approach suggest the potential of improving the productivity and quality in the machining of thin-walled parts with the proposed control methods for the robotic steady rests.

The dissertation of Niloufar Esfandi is approved.

Lieven Vandenberghe

Tetsuya Iwasaki

James S. Gibson

Tsu-Chin Tsao, Committee Chair

University of California, Los Angeles

2017

Table of Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to express my deep gratitude to my advisor Professor T-C Tsao for his amazing wisdom and exceptional guidance throughout my years at UCLA. He turned me from someone who had a lot of passion for controls but could hardly read a single paper from beginning to end in one sitting, into a patient researcher who actually writes papers and gave me the opportunity to gain invaluable hands-on experience with mechatronic systems and control in his lab. For that I am very grateful.

I would like to thank all of my other committee members, Professors Vandenberghe, Iwasaki and Gibson, for their guidance throughout my research. Special thanks to Professor Gibson for spending a lot of time with me investigating the details of the LQR tuning method and its parallels.

I would like to thank my lab-mates at the Mechatronics and Controls Laboratory at UCLA through the years who were great friends and colleagues the company of whom I enjoyed. A special thanks goes to Herrick Chang for helping me get started in the lab when I first joined the group. He taught me how to create mechatronic systems from scratch and use the instruments in the lab. I would also like to specifically thank Chris Kang and James Simonelli for being great senior lab members and answering all my questions about technical issues with the plants and guiding me on how to succeed in the program in general. Thank you to Sandeep Rai for usually having a complete opposite perspective on things as I had. That helped me form a more well-rounded idea about projects I worked on. Thank you to Stephen Prince for motivating me to graduate on time.

Finally, I would like to express a heartfelt thank you to my family for always encouraging me and being there for me. To my brother Navid, thank you for cheering me up and taking my stress away by your humor and easy-going approach to life. Thank you mom and dad for making so many sacrifices to help me get where I am now. You have always believed in me and set the bar high for me. I am forever grateful for your love and support through these years and always.

Vita

2009	B.S. (Mechanical Engineering), K. N. Toosi University of Technology, Tehran, Iran
2013	M.S. (Mechanical Engineering), University of California Los Angeles (UCLA), Los Angeles, CA.
2012–2017	Teaching Assistant, Mechanical and Aerospace Engineering Department and Life Sciences Department, University of California, Los Angeles (UCLA), Los Angeles, CA.
2012–2017	Graduate Student Researcher, Mechanical and Aerospace Engineering Department, University of California, Los Angeles (UCLA), Los Angeles, CA.

Publications

N. Esfandi and T. C. Tsao, "Robot Assisted Machining of Thin-walled Structures", International Federation of Automatic Control (IFAC), 20th World Congress, July 2017

CHAPTER 1

Introduction

This dissertation studies the modeling and control of a class of combined systems that includes distributed parameter systems, the time varying interactions of sub time invariant systems, infinite dimensional systems and delayed systems. These characteristics in a system can block the use of conventional stability analysis and control design as those are only good for linear time invariant systems with finite dimensional state vectors and no delays. In this dissertation strategies for modeling, analysis and control of such systems is studied and presented on a real life physical problem that demonstrates all of these characteristics.

Systems can be grouped into two categories. If the system under study is assumed to be concentrated at a single spatial point it is called a lumped parameter system and if it is assumed to occupy a certain spatial domain it is considered a distributed parameter system. The lumped parameter system is modeled by ordinary differential equations while the distributed parameter system is modeled by partial differential equations [OS89]. Some examples of distributed parameter systems are fluid systems, heat exchangers, magneto-hydrodynamic systems, and vibration systems [OS89]. Distributed parameter systems thus are infinite dimensional.

Mechanically flexible systems are distributed parameter systems and theoretically they are infinite dimensional [Bal78]. In practice they can be approximately modeled with very large dimensional systems. In other words, often times a physical structure is essentially an infinite dimensional system in the continuous spatial domain with no closed form solution for its natural modes of vibration. That can be represented as a field problem in which one should find the distribution of one or more dependent variables. In mathematics a field

problem is represented by differential equations. For a complicated shape writing or solving these differential equations in the continuous spatial domain becomes impossible.

Time-delay systems belong to the class of functional differential equations which are infinite dimensional as opposed to ordinary differential equations (ODEs) [Ric03]. Delayed systems can be modeled through delay differential equations which are infinite dimensional, since they belong to the group of partial differential equations (PDEs). PDEs have infinite dimensional state vectors. Examples of time delayed systems include heating systems as there is typically a delay between heating applied and temperature changes, biological functions as it takes time between sensory inputs to the body and their transmission to the brain as well as the brains decision sent through neurons to the body and also hormonal releases take time to induce the desired change in the body. Sensors/actuators and networks can have delays, long transmission lines inevitably cause a transport delay. Machining chatter is another example of delayed systems due to its dynamics depending on not only the current but also the previous revolution of the cutting process.

Many systems have moving parts causing the interaction between different parts to be time varying. Examples of that include the interaction of a bridge with the vehicles crossing it [TTP01], interaction of tracks and the trains, and interactions of the cutter and workpiece in machining. Conventional control design and analysis likes to deal with linear time invariant (LTI) systems. So this time varying nature of systems and processes with moving parts which are each its own dynamical system cannot easily be analyzed with conventional stability analysis and a control design is not as straight forward.

Figures 1.1 and 1.2 show block diagram and illustration of a possible scenario involving infinite dimensional dynamics, delay dynamics and time varying interactions between the different systems involved. To be able to use conventional control methods and stability analysis, this system would need to be approximated by a finite dimensional LTI model.

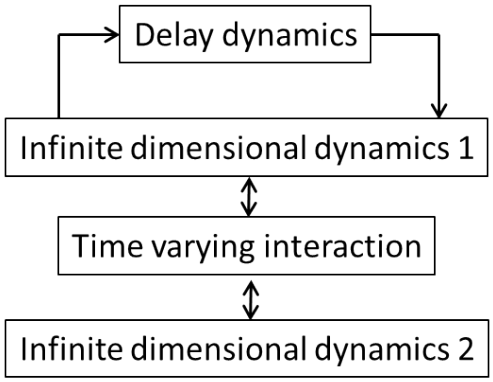


Figure 1.1: Block diagram of a possible system with delays, distributed parameter subsystems and time-varying interactions

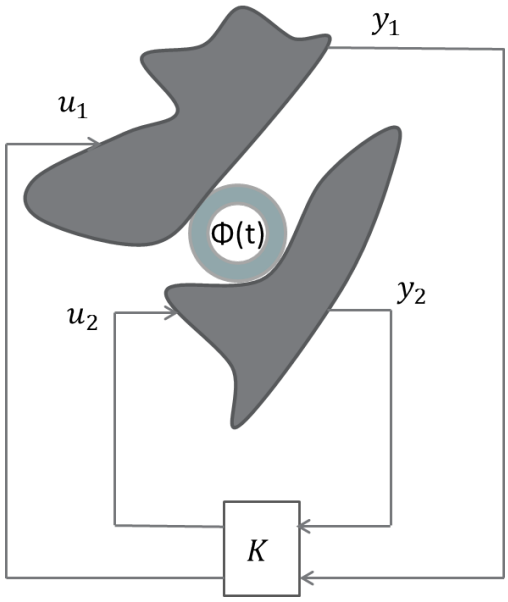


Figure 1.2: Schematic of a possible system with delays, distributed parameter subsystems and time-varying interactions

In this work I modify and combine some of the existing methods of approximation for dealing with each of those issues and develop some novel methods to address the issues better.

Here I give a preview of the solutions and methods used in this work. Finite element

method and analysis was utilized to give a finite dimensional approximation of the infinite dimensional model of structures in the spatial domain. Semi-discretization was used to deal with the delay terms in delay differential equations and to avoid an infinite dimensional solution to the DDE while avoiding the discretization of the non-delayed part of the dynamics in order to have a more exact solution. The time varying components of the interacting systems were isolated and dealt with separately from the underlying time invariant systems. Finally a novel method for tuning LQR controllers is introduced that helps make the process more intuitive even for a high dimensional MIMO system.

In this dissertation these concepts and methods are applied to an example case of a real world problem that demonstrates all of these issues: the machining of thin-walled structures. The structure being machined can have complicated shapes with no closed form solution for its modes and natural frequencies. The interaction between the cutter and the workpiece is time varying. The cutting process including the regenerative chatter phenomenon, which is prominent in thin-walled machining, is represented mathematically with delay differential equations. After using approximations and model reduction, control design for the system is studied. A new idea on how to apply controls to this system is introduced. In this work the use of industrial robot arms to apply controls to the workpiece as opposed to the more popular method of controlling the cutting tool is suggested and studied. The novel tuning method developed in this work for LQR controllers is then applied to this example system to get the most desirable results for the application.

The remainder of this dissertation is organized as follows: Chapter 2 describes the methods used for modeling the thin-walled structure, machining process and its interactions with the proposed robot arm manipulator to later be used to apply controls to the workpiece. It then discusses semi-discretization and its application to the modeling of delay in the machining process and how a linear time invariant model can be generated using this final step. In Chapter 3 the method developed in Chapters 2 is applied to the example case of machining a thin-walled tube. Chapter 4 discusses an optimal control design for the machining system to be applied to the workpiece through the robot arm manipulator. Chapter 5 introduces a

novel modeling and semi-discretization method that is robust control oriented and allows for design of a controller for the system that is robust to uncertainties specifically the cutting stiffness value. Finally in Chapters 6 and 7 a novel method for tuning LQR controllers that is more intuitive and very useful with systems of high dimension is introduced and its application to several example systems is included. Chapter 8 is the conclusion of the dissertation with some suggestions for future directions.

CHAPTER 2

Machining of Thin-Walled Structures

2.1 Introduction

Thin-walled structures can be found in diesel engine cylinder liners, automotive wheel rims, gas turbine blades and rings, and many other parts in automotive and aerospace systems. Machining thin-walled workpieces is a challenging manufacturing process as the workpiece flexibility makes it prone to deformation and chatter during the machining by the way the cutting process interacts with the structural dynamics of the mechanical components involved, i.e. tool, workpiece, and holders. The forced deflection and regenerative feedback interaction is modeled as delay-differential equations and has been well understood and studied [Mer65, HT74, SN78, OH98, Ste01, IST07]. The delay could be fixed or varying while the differential part could be constant or varying ordinary or partial differential equations.

Special treatments required to address the difficulties in machining thin-walled parts can be categorially divided into two classes of methods: passive and active methods. In the passive methods, off-line process planning and optimization is done [RLH04, CXT09, GMJ16] to create the tool-path and cutting parameters that compensate for the deflection and/or optimize the material removal rate that avoids chatter [TC95, SGY09]. Another approach designs special work holders to stiffen the flexible workpieces, like plastic jaws, pendulum jaws [Aka89], 6-jaw chucks and magnetic chucks [FM99], vacuum clamping systems [GA74], and other special clamps [DA67, Cam89]. Another method for avoiding chatter determines spindle speeds that afford relatively large stability margin, thus higher productivity with higher material removal rate, based on the machine tools structural dynamics [Tlu86]. Im-

provements have been made on determining accurate stability lobe diagram for complex machining systems, such as the thin-walled structure, by employing the semi-discretization [IS04] of the delay term in the governing delay-differential equations [HIS06, CFE14, FEA14].

In the active methods, some address workpiece deformation while the others address chatter. Non-circular tool path in boring a thin-walled tube has been demonstrated to obtain round profiles by compensating for the workpiece cylindrical errors caused by the cutting force applied to the workpiece held by a three-jaw chuck [HT00]. The clamping deformation issue has also been addressed by model-based error compensation for thin-walled cylindrical parts through a turning tool holder with integrated sensors and actuators [HK10]. Method to automatically adjust the spindle speed to find the sweet spot for chatter-free machining using in-process vibration sensing has been developed [DTS92, ST92]. Periodic variation of the spindle speed is another method that mitigates chatter instability. Sexton, et al[SMS77] were among the first to simulate the result of variable spindle speed (VSS) machining. The variation of speed can be sinusoidal [LDK90, TMK93, SI97] while some researchers [YEN02] used multi-level random spindle speed variation to prevent chatter which they claim is more universally applicable than sinusoidal spindle speed variation. VSS work is based on modeling the tool as a spring mass damper model suggesting that the flexibility of the tool is the main cause of chatter. In thin-walled machining however the dynamics of the system is dominated by the flexibility of the workpiece. In many cases the tool is modeled as rigid since its flexibility is negligible compared to that of workpiece. Nonetheless VSS has also been demonstrated by experiment to be effective on non-circular turning of thin-walled tube [HT98]. There have also been efforts in finding an active control solution to the problem of chatter and instability. These works [AT94, FE11] focus on control of the cutting tool to achieve the desired stable cutting conditions.

In machining thin-walled structure, it is obvious that stiffening the workpiece near the cutting location would mitigate the deformation and chatter problems. So how do we avoid custom designing and making special work holders for different thin-walled parts? The suggested approach of this paper is to utilize the dexterous robotic manipulators with fast

actuator end effectors and feedback control to provide active steady rests for the workpieces. This concept of robot assisted machining of thin-walled structure is in contrast to the other active methods where the actuator is placed on the tool side. It also differs from using a robot to carry a cutting tool, which has been applied to light load machining where the need of dexterity outweighs rigidity, such as deburring, polishing, and drilling [ZCX06, NRR02, GS95, GS97, BL11]. Since robots are not as rigid and accurate as machine tools for precision machining of thin-walled structure, using the robots, which would otherwise be standing by during machining until called for part loading and unloading, to assist with handling the deformation and chatter challenges in machining thin-walled workpiece seems quite plausible. This concept shifts the technical challenge of creating custom made hardware work holders to control software enabled active steady rests. In this work we focus on the control for the stability issue, i.e. avoiding the onset of chatter and then take a look at the implications on performance.

The rest of the chapter is organized as follows: Section 2.2 describes the machining system model including workpiece, robotic manipulator and cutting process models and model reduction techniques utilized and briefly describes the control design.

2.2 Machining System Model

To achieve stable and precise machining of thin-walled workpieces, this section will address the modeling and control of the robot assisted machining system. The overall control system is depicted by Figure 2.1 which consists of the tool side dynamics and control, the work side dynamics and robot control and the cutting process. In the figure, the workpiece and blocks below it represent the conventional machine tool control system, where the tool path is generated by the controller at the bottom block. In this research we propose to augment the machine tool control with actively controlled robots, depicted as those blocks above the workpiece. The contact locations for the robot end effector to the workpiece structure and the cutting force to the workpiece structure can be fixed or time varying, as are represented by

$\gamma_{tw}(t)$ and $\gamma_{rw}(t)$ respectively with γ being the shape function of the point on the workpiece where the force from the tool or the robot is applied. The tool can be assumed rigid or it could be modeled flexible as well with γ_{wt} representing the point on the tool where the reaction of cutting force is applied. We will talk more about shape functions in the following sections.

In principle, the thin-walled workpiece structural dynamics can be modeled as an infinite dimensional system in the continuous spatial coordinates. The dynamic cutting forces include delayed feedback for the relative displacement between the workpiece and the cutting tool. Furthermore, the moving contacts of the workpiece with the robot arm and the cutting tool result in time varying a system. To facilitate reduced order modeling for controller design, this section presents the modeling of the dynamic system and its reduction to a lower-dimensional linear system.

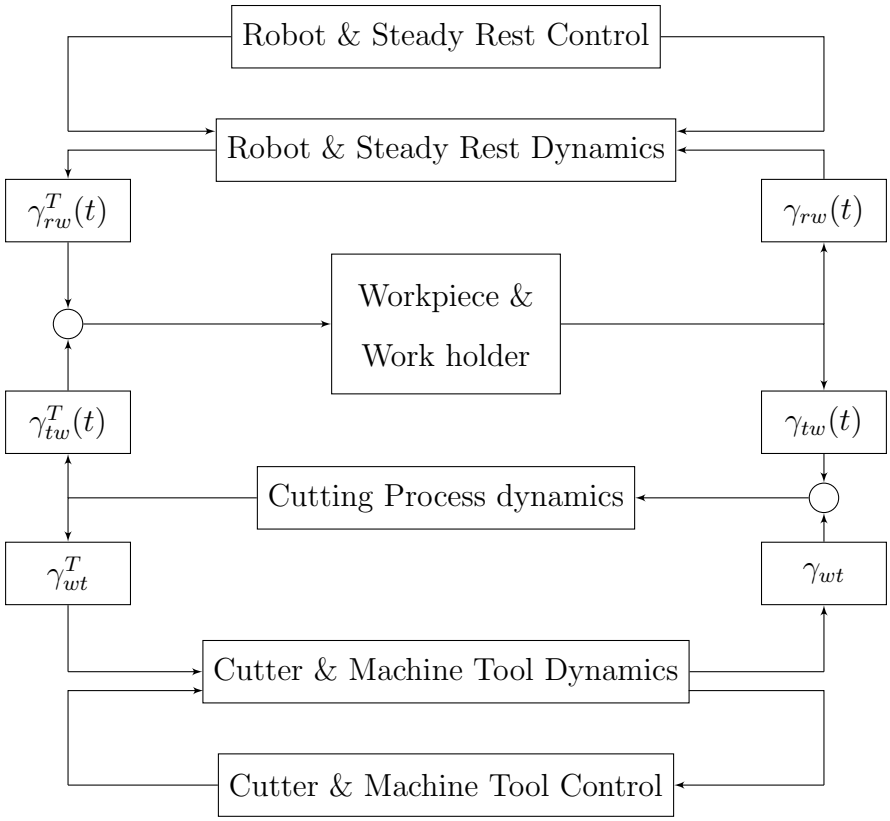


Figure 2.1: Breakdown of the machining system. (γs represent shape function matrices explained in Section 2.2.1)

2.2.1 Finite Element Model for Continuum Dynamics

In principle, the thin-walled workpiece structural dynamics can be modeled as infinite dimensional systems in the continuous spatial coordinates. The dynamic cutting forces include delayed feedback for the relative displacement between the workpiece and the cutting tool. Furthermore, the moving contacts of the workpiece with the robot arm and the cutting tool result in time varying a system. To facilitate reduced order modeling for controller design, this chapter presents the modeling of the dynamic system and its reduction to a lower-dimensional linear system.

In the first step finite element method is used to get a large finite dimensional model of the workpiece. After extracting the mass and stiffness matrices from the FEA, we perform

$N_1 = \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta)$	$N_5 = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta)$
$N_2 = \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta)$	$N_6 = \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta)$
$N_3 = \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta)$	$N_7 = \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta)$
$N_4 = \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta)$	$N_8 = \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta)$

Table 2.1: Shape functions for an eight-node trilinear hexahedral element

model reduction. Damping is introduced to each mode with damping ratio increasing in proportion to the modal frequencies.

In many cases the shape of the workpiece is such that a closed form solution for the modes of vibration does not exist or is difficult to calculate. Finite element method (FEM) provides an approximate but reliable means to extract the information needed. FEM is a method for numerical solution of field problems. Individual finite elements can be thought of as small pieces of a structure. In each finite element a field quantity is permitted to have only a simple spatial variation while the actual variation in the area is almost certainly more complicated. Thus FEM provides an approximate solution.

The models are created using the software Abaqus. There are different choices of elements for the FEA. In this study I used eight-node trilinear hexahedral elements in natural coordinates, and Lagrange interpolation to do the finite element analysis. Figure 2.2 shows a typical hexahedral element with the natural coordinates (ξ, η, ζ) centered at the element's geometric center. Natural coordinates assume 1 and -1 at the walls and zero at the center on all three axes regardless of the actual distance from center to a wall and the fact that it may differ from one axis to another. Table 2.1 shows the shape functions of the element type used.

A shape function matrix is used to go from natural coordinates on an element to the global coordinates of the workpiece. Every point on the workpiece belongs to a certain element and on that element it can be pinpointed using the natural coordinates. Then using the shape function matrix it is transferred to the global coordinates. For example Figure 2.3

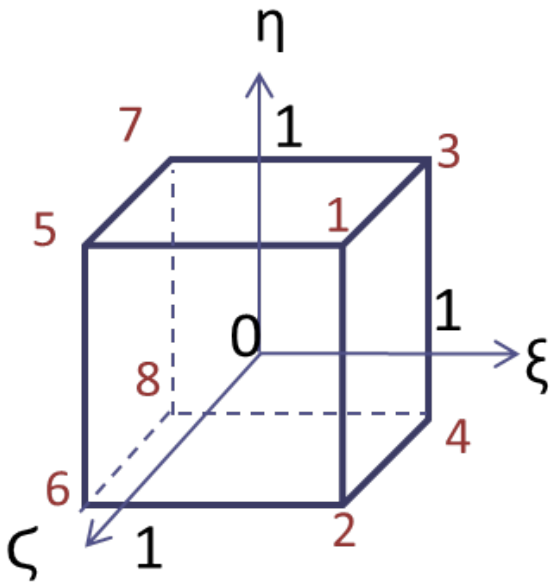


Figure 2.2: A typical element node numbering and placement of natural coordinates

shows the tool center point (TCP) or the tools force application point on a typical element and the coordinates of that point in the natural coordinate system of that element. The assumption is that the force applied anywhere other than the nodes needs to be distributed on the nodes for an FE analysis. By plugging in the coordinates of that point into shape function formulations of Table 2.1, one can get 8 numbers each of which signifies the fraction of the force applied to the green point on Figure 2.3, that is going to be applied to the respective node.

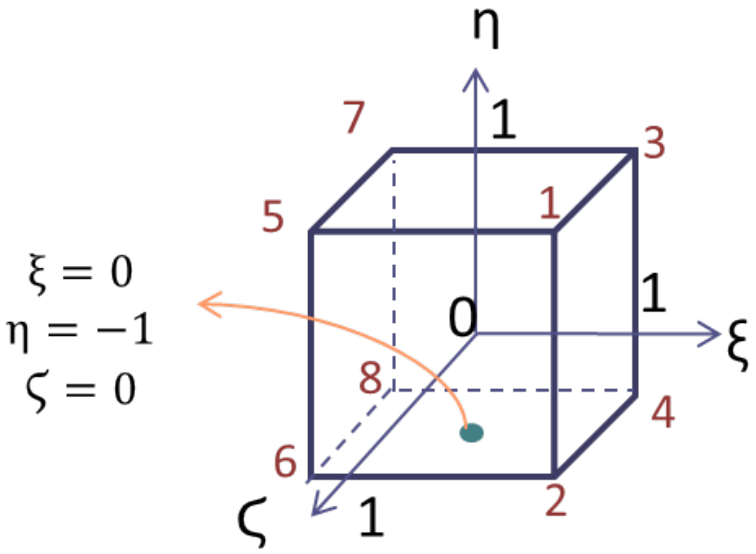


Figure 2.3: Natural coordinates of the tool tip

So the force vector F showing the force applied to all the nodes of the structure from a point force f can be calculated from

$$F = \gamma_1^T f \tag{2.1}$$

It is noteworthy to mention that the shape function matrix can also be used to go from a vector of all displacements for all nodes to the displacement of a specific point on the structure. So if for instance X is the vector of nodal displacements of a structure and we would like to calculate the displacement of TCP,

$$x_{TCP} = \gamma X \tag{2.2}$$

2.2.2 Modal Analysis

Assuming a linearly elastic material, multi-dimensional form of Newton's second law for the finite element structure is [CMP02]

$$[m]\{\ddot{d}\} + [c]\{\dot{d}\} + [k]\{d\} = \{r_{ext}\} \tag{2.3}$$

Global form of the above equation for a multi-element structure is [CMP02]

$$[M]\{\ddot{D}\} + [C]\{\dot{D}\} + [K]\{D\} = \{R_{ext}\} \tag{2.4}$$

Global mass, damping and stiffness matrices are constructed from element matrices [m], [c], [k]. They can be constructed by conceptual expansion of element matrices to structure size and summation of the overlapping terms. For more details on this the reader is referred to Cook's book [CMP02].

Vibratory motion consists of nodal amplitudes $\{\bar{D}\}$ that vary sinusoidally with time. Assuming zero static equilibrium displacements $\{\bar{D}\}$ represents excursions from the unstressed configuration. Thus nodal displacements associated with vibrations are

$$\{D\} = \{\bar{D}\}sin\omega t \tag{2.5}$$

With damping matrix omitted the eigenproblem for undamped free vibration becomes

$$([K] - \omega^2[M])\{\bar{D}\} = \{0\} \tag{2.6}$$

Where ω^2 is an eigenvalue and ω is a natural frequency. $\{\bar{D}\}$ contains only the DOF that may assume nonzero values. That is nodes that are constrained are not included in the displacement vector. Thus [K] is positive definite. With precautions [M] is also positive definite.[1] As a result the number of ω_i's is equal to number of DOF in $\{\bar{D}\}$.
If eigenvectors are normalized with respect to mass matrix we will have Equations (2.7)

$$\{\bar{D}\}_i^T[M]\{\bar{D}\}_i = 1 \quad and \quad \{\bar{D}\}_i^T[K]\{\bar{D}\}_i = \omega_i^2 \tag{2.7}$$

[1] If element mass matrices are consistent or lumped with strictly positive diagonal coefficients, Cook's book [CMP02] p. 385.

A modal matrix ϕ can be defined whose columns are the eigenvectors normalized with respect to the mass matrix in addition to a diagonal spectral matrix $[\omega^2]$ whose entries are the squared natural frequencies of vibration.

$$\phi = \begin{bmatrix} \bar{D}_1 & \bar{D}_2 & \cdots & \bar{D}_n \end{bmatrix}, \quad [\omega^2] = \begin{bmatrix} \omega_1^2 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \omega_n^2 \end{bmatrix} \tag{2.8}$$

Consequently

$$\phi^T[M]\phi = I, \quad \phi^T[K]\phi = [\omega^2] \tag{2.9}$$

Any arbitrary displacement vector {D} can be expressed as a linear combination of the eigenvectors. Thus

$$\{D\} = \phi\{Z\}, \quad \{\dot{D}\} = \phi\{\dot{Z}\}, \quad \{\ddot{D}\} = \phi\{\ddot{Z}\} \tag{2.10}$$

Where {Z}=$[Z_1 Z_2 \dots Z_n]^T$ and the Z_i are generalized DOF often called modal coordinates or modal displacements.
Substituting Equations (2.10) into (2.4) one obtains

$$\{\ddot{Z}\} + [C_\phi]\{\dot{Z}\} + [\omega^2]\{Z\} = \{R_\phi\}, \quad \{R_\phi\} = \phi^T\{R_{ext}\} \tag{2.11}$$

One benefit of normalizing eigenvectors with respect to mass matrix is that as a result Equation (2.11) does not include mass matrix or stiffness matrix. This will simplify the calculations and derivations using the modal method to study systems.

With all modes retained Equation (2.11) is an exact representation of Equation (2.4). Yet for many practical problems only the lower eigen-frequencies need to be retained. So one can approximate the equations using the following reduction[CMP02]

$$\{D\} \approx \sum_{i=1}^{m} \{\phi\}_i Z_i \tag{2.12}$$

where {D} and $\{\phi_i\}$ are n×1 and typically $m \ll n$.

The reason behind keeping only the lower eigen-frequencies is that structural internal damping is higher for higher frequencies. Proportional or Rayleigh damping in modal form is a diagonal matrix obtained through $[C_\Phi] = \alpha[I] + \beta[\omega^2]$. Also in many cases modal damping $[C_\Phi]$ is defined as a diagonal matrix whose i^{th} diagonal coefficient is $2\xi_i\omega_i$ where ξ_i is the damping ratio [CMP02]. This means modes with higher natural frequencies would have higher damping leading to those modes being insignificant.

As mentioned before Abaqus is the FEM software of choice in this study. Abaqus can perform frequency tests. It can generate elemental mass and stiffness matrices that can later be processed to calculate global mass and stiffness matrices. It will output the eigenvalues (ω^2) and one can code it in such a way that Abaqus will output displacement of all nodes for each eigenvalue. Putting these displacements in a vector for each mode will result in the respective eigenvector. Also Abaqus gives the user different options for normalization of the eigenvectors one of which is mass normalization. In order to make sure this normalization will result in Equations (2.9), a simple test was designed. A small beam with only two elements was created in Abaqus and the internal code was modified to output eigenvectors with mass normalization, mass matrix for all elements and stiffness matrix for all elements. In the case of two elements it is easy to construct the global mass and stiffness matrices by summation of overlapping terms and omitting the encastred node values from the element that holds them. Special attention needs to be paid to the ordering of the nodes that Abaqus comes up with. Eigenvectors should be reordered manually to match the node order of the mass and stiffness matrices. After all these preliminary steps, $\{\bar{D}\}_i^T[M]\{\bar{D}\}_i$ and $\{\bar{D}\}_i^T[K]\{\bar{D}\}_i$ were calculated using MATLAB and compared to their expected values from Equation (2.7). Tables 2.2 and 2.3 show the values and their respective error for the first three eigenvalues.

e-vec no. (i)	$\{\bar{D}\}_i^T[M]\{\bar{D}\}_i$	Relative error
1	0.99970	-2.9885e-04
2	0.99983	-1.6587e-04
3	0.99997	-2.9544e-05

Table 2.2: Relative error of $\{\bar{D}\}_i^T[M]\{\bar{D}\}_i$ compared to 1 for the first 3 eigenvectors

e-vec no. (i)	$\{\bar{D}\}_i^T[K]\{\bar{D}\}_i$	ω_i^2	Relative error
1	8.87905e+04	8.88030E+04	-1.4024e-04
2	2.95660e+05	2.95704E+05	-1.4857e-04
3	3.44593e+06	3.44595E+06	-5.3725e-06

Table 2.3: Relative error of $\{\bar{D}\}_i^T[K]\{\bar{D}\}_i$ compared to ω_i^2 for the first 3 eigenvectors

As can be seen in Tables 2.2 and 2.3 the relative errors are small and the mass normalization method that abaqus performs is quite reliable. This means by using Equation (2.11) there will be no need for calculating the global mass and stiffness matrices for the larger and more complicated models of interest for study.

2.3 Robotic Arm

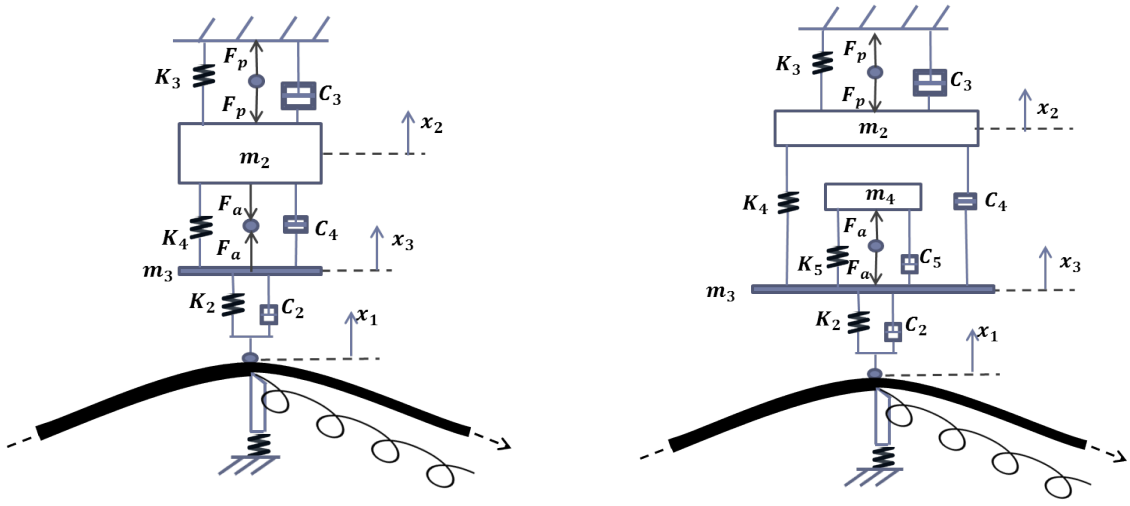
The objective of the robot assisted machining is twofold. First is to reinforce the thin workpiece and compensate for the deformation due to cutting force by the robot's mechanical structure and joint actuator control. Second is to avoid onset of chatter instability, caused by a delayed dynamic feedback interaction of the workpiece structural dynamics and the cutting tool, by actively controlled actuator held by the robot's end effector. The above mentioned robot augmentation provides an "active steady rest" to the workpiece.

An industrial robot arm that can be added to the machining system independently is what has been the idea behind this study. It is convenient since industrial robots are already

available in machining sites and are used for other purposes such as loading and unloading workpieces into the machines but they are idle during the machining process. Thus adding a controller mechanism to the already existing industrial robots would need minimal effort and would incur minimal expense.

The effect the robotic arm is to have on the workpiece can be modelled by a few springs, masses and dampers as shown in Figure 2.4. The robot in contact with the thin-walled work piece may exert forces to constraint the work piece motion. The forces may come from the robot joint actuators through the end effector (F_p) and actively controlled actuators mounted on the end effector (F_a). The robot joint actuators may not be adequate for active chatter suppression because of limited bandwidth with the force transmission through the end effector. Figure 2.4 shows two possible methods that the actuators may exert actively controlled forces on the workpiece, on both the workpiece and robot or on the workpiece and a proof-mass. In the first method force of the actuator to be applied to the workpiece is generated against the robot structure and static forces can be applied. In the second method the force of the actuator to be applied to the workpiece is generated against an added mass, as such, only zero mean dynamic forces can be applied. This work considers the first method.

In Figure 2.4a, the large mass m_2 and its spring and damper (K_3, C_3) are to represent the industrial robotic arm's mass and stiffness. The smaller spring mass damper system (m_3, K_4, C_4) is equivalent of an actuator system. The actuation system can be used as a controls means to eliminate chatter. F_a in Figure 2.4a is where the actuator force is applied. This force is going to be the control output. The contact point of the arm with the workpiece is modelled with another set of spring and damper with pre-compression on the spring. For implementation, the robot arm could be in contact with the workpiece through a roller since the workpiece is usually spinning. Displacements x_2 and x_3 are for the arm frame and the actuator's equivalent mass respectively as shown in Figure 2.4a. x_1 is the displacement of the point on the workpiece where the robot contacts the workpiece and can be found form the finite element model of the workpiece through shape function γ_{rw}.



(a) Piezo-electric actuation

(b) Proof-mass actuation

Figure 2.4: Schematic of the robot arm equivalent models

From simple dynamics

$$F_a - K_4(x_3 - x_2) - C_4(\dot{x}_3 - \dot{x}_2) + K_2(x_1 - x_3 + \alpha_1) + C_2(\dot{x}_1 - \dot{x}_3) = m_3\ddot{x}_3 \tag{2.13}$$

$$-F_a + K_4(x_3 - x_2) + C_4(\dot{x}_3 - \dot{x}_2) - K_3x_2 - C_3\dot{x}_2 = m_2\ddot{x}_2 \tag{2.14}$$

2.4 Cutting Dynamics

Regenerative chatter is caused by the cutting force variation due to the wavy workpiece surface cut one revolution ago. The phenomenon can be described by incorporating time delay in the model equations. In essence chatter happens when the machine tool, workpiece and cutting process combine to form a dynamically unstable system. This can adversely affect the machining productivity by limiting the allowable depth of cut or feed rate. Stability of the machining processes is usually depicted by the so-called stability lobe diagrams, which plot the maximum stable axial depths of cut versus the spindle speed. These diagrams provide a guide to the machinist to select the optimal machining parameters in order to

achieve maximum material removal rate without chatter. Machine tool structures have been made with high structural stiffness to combat chatter instability but in many situations including the thin-walled workpiece machining, it is the low rigidity of the workpiece that causes the cutting instability [HT00].

In this study we assume the tool is rigid and the workpiece is thin and inevitably flexible. Cutting dynamics have been well studied and modeled ([SN78]). Equation (2.15) is used to model the force law applied to the workpiece by the tool.

$$F(t) = F_{c0} + \kappa.(x_{tcp}(t) - \mu x_{tcp}(t-T)) \tag{2.15}$$

F_{c0} is the constant initial force applied to the workpiece by the tool and its amount is determined by the desired depth of cut. Successive cuts usually overlap in turning so the force not only depends on the present, but also the past values of workpiece or tool displacement. Here to represent the overlapping of consecutive cuts is the factor μ that is between 0 and 1. x_{tcp} is the displacement of the tool center point. κ is the cutting stiffness number or specific cutting force. In this study to consider the worst case scenario in chatter creation μ=1 was chosen.

The process force from the cutting tool is to be calculated from (2.15) and is applied as an external force to the finite element model of the workpiece.

$$r_{ext_1} = F(t) \text{ and } \{R_{ext_1}\} = \gamma_{tw}^T r_{ext_1} \tag{2.16}$$

Here γ_{tw} is the shape function matrix for the point on the workpiece where the tool is applying the cutting force which is also called the tool center point (TCP). Displacement of TCP can be calculated by $x_{tcp} = \gamma_{tw}\{D\}$. Assuming there is an active fixture or a robotic arm in contact with another point on the structure with shape function matrix γ_{rw}, there could be a second external force denoted by r_{ext_2}. That turns (2.11) into

$$\begin{aligned} \{\ddot{Z}\} + [\omega^2]\{Z\} &= \phi^T\{R_{ext}\} \\ &= \phi^T\gamma_{tw}^T\big(F_0 + \kappa\gamma_{tw}\phi(Z(t) - Z(t-T)\big) + \phi^T\gamma_{rw}^T r_{ext_2} \end{aligned} \tag{2.17}$$

2.5 State Space Representation

A state space representation for the full system of the workpiece and the arm would have 4 states for the arm (2 states for m_2 and 2 states for m_3) and as many states for the workpiece as twice the number of modes considered important in the modal domain. So the state vector is a combination of modal domain states for the workpiece and actual displacement and velocity states for the arm frame and the end effector. But this is not going to cause any problems since the coupling between the modal model of the workpiece and the dynamic model of the arm is through the force that the arm exerts on the workpiece and that force is a function of the physical states of the arm and the end effector as well as the displacement of the point on the workpiece where the arm is exerting force. That point displacement can be extracted from the modal domain using the ϕ matrix and γ_{rw} as discussed before. After combining Equations (2.17), (2.13), (2.14) into state space, the following matrix equation describes the full system.

$$
\begin{bmatrix} \{\dot{Z}\}(t) \\ \{\ddot{Z}\}(t) \\ \dot{x}_2(t) \\ \ddot{x}_2(t) \\ \dot{x}_3(t) \\ \ddot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & 0 & 0 & A_{25} & A_{26} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & A_{45} & A_{46} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ A_{61} & A_{62} & A_{63} & A_{64} & A_{65} & A_{66} \end{bmatrix} \begin{bmatrix} \{Z\}(t) \\ \{\dot{Z}\}(t) \\ x_2(t) \\ \dot{x}_2(t) \\ x_3(t) \\ \dot{x}_3(t) \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\phi^T \gamma_{tw}^T(t) \kappa \gamma_{tw}(t) \phi & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \{Z\}(t-T) \\ \{\dot{Z}\}(t-T) \\ x_2(t-T) \\ \dot{x}_2(t-T) \\ x_3(t-T) \\ \dot{x}_3(t-T) \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1/m_2 \\ 0 \\ 1/m_3 \end{bmatrix} F_a(t) + \begin{bmatrix} 0 \\ \phi^T \gamma_{tw}^T(t) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} F_{c0} + \begin{bmatrix} 0 \\ -\phi^T \gamma_{rw}^T(t) \\ 0 \\ 0 \\ 1/m_3 \\ 0 \end{bmatrix} K_2 \alpha_1 \qquad (2.18)
$$

where

$$A_{21} = -[\omega^2] + \phi^T \gamma_{tw}^T(t)\kappa\gamma_{tw}(t)\phi - \phi^T \gamma_{rw}^T(t)K_2\gamma_{rw}(t)\phi$$
$$A_{22} = -\phi^T \gamma_{rw}^T(t)C_2\gamma_{rw}(t)\phi \quad A_{25} = \phi^T \gamma_{rw}^T(t)K_2$$
$$A_{26} = \phi^T \gamma_{rw}^T(t)C_2 \qquad A_{43} = (-K_4 - K_3)/m_2$$
$$A_{44} = (-C_4 - C_3)/m_2 \qquad A_{45} = K_4/m_2$$
$$A_{46} = C_4/m_2 \qquad A_{61} = (1/m_3)K_2\gamma_{rw}(t)\phi$$
$$A_{62} = (1/m_3)C_2\gamma_{rw}(t)\phi \qquad A_{63} = K_4/m_3$$
$$A_{64} = C_4/m_3 \qquad A_{65} = (-K_2 - K_4)/m_3$$
$$A_{66} = (-C_2 - C_4)/m_3$$

and α_1 is the pre-compression of the spring K_2. After naming the matrices and using X for the state vector:

$$\dot{X} = A(t)X(t) + Q(t)X(t-T) + B_1F_a(t) + B_2(t)F_{c0}(t) + B_3(t)K_2\alpha_1 \tag{2.19}$$

As can be noticed in Equation (2.18), Components of A, Q, B_2 and B_3 matrices include the shape function matrices which are time varying due to moving contact point between the workpiece and the tool and robotic arm. Also there is a delay term due to the modeling of cutting dynamics. This leads to delay-differential equations with time varying coefficients.

In the next sections the method to deal with these issues in order to get a linear time invariant (LTI) system on which conventional stability and control methods can be applied, is discussed.

2.6 Semi-Discretization for Delayed Dynamics

2.6.1 Introduction to Semi-Discretization

Insperger and Stepan[IS02] introduce semi-discretization as an efficient numerical method for the stability analysis of linear delayed systems which is based on a special kind of discretization technique with respect to the past effect only. It results in an approximate system which

is delayed and also time periodic. It can be transformed analytically into a high-dimensional linear discrete system.

Figure 2.5 shows a simple block diagram representation of semi-discretization. The top diagram is the true continuous time system with continuous time plant represented by G(s) and has a feedback delay represented by the continuous time e^{-sT_1} block. Semi-discretization would approximate this continuous time delay with a discrete delay block as z^{-N_p} shown in the bottom diagram and the sampling happens inside the loop now. Essentially the delay T_1 is divided into N_p equal intervals. The hold denoted by the block H will allow the signal to be the input to the continuous time plant G(s). This hold could be the same zero order hold as the one that holds the control input u^* or it could be a different holding scheme. Thus in this method the plant is not discretized. Another advantage of this method is its compatibility with digital control since for digital control it is necessary to sample output and the control will generate a discrete signal.

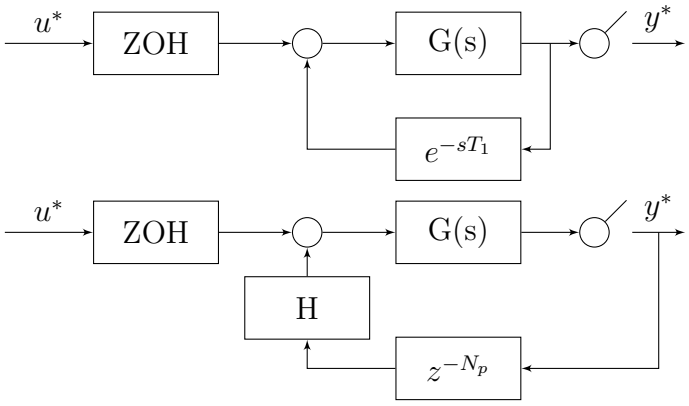


Figure 2.5: Semi-discretization in block diagram form

2.6.2 Semi-Discretization Applied to Thin-Walled Machining

T in the Equations (2.18) and (2.19) is the time it takes the workpiece to complete one cycle. By looking at either of the equations one can observe that it is a delay-differential equation. Having the $X(t-T)$ term prevents us from being able to use standard stability criteria and control designs. To turn this time-varying delayed system into a linear time invariant one on

which stability criteria would work and common control designs can be implemented, semi-discretization method was used. For semi-discretization that period T is divided into k equal subdivisions of time. In each subdivision of time the time varying matrices of Equation (2.18) are approximated by constant matrices using the following equations.

$$T = k\Delta t \tag{2.20}$$

$$\begin{aligned} A(t) &\approx A(t + \Delta t/2) = A_i \\ Q(t) &\approx Q(t + \Delta t/2) = Q_i \\ B(t) &\approx B(t + \Delta t/2) = B_i \\ i &= 1, ..., k \end{aligned} \tag{2.21}$$

For the delay term a weighted average of two of the past values is suggested to be used.

$$X(t - T) \approx X(t + \Delta t/2 - T) \approx w_a X_{i-k} + w_b X_{i-k+1} \tag{2.22}$$

In this study w_a and w_b are both taken to be $\frac{1}{2}$.

These steps help make up for the phase difference that can be caused by the zero order hold shown in Figure 2.5.

Substituting Equations (2.21) and (2.22) into Equation (2.19) results in a time invariant differential equation during one time interval Δt.

$$\begin{aligned} \dot{X} &= A_i X(t) + Q_i(\frac{1}{2}X_{i-k} + \frac{1}{2}X_{i-k+1}) + B_1 F_{a_i} + B_{2_i} F_{c0} + B_{3_i} K_2 \alpha_1 \\ &= A_i X(t) + F_{i\ const} \end{aligned} \tag{2.23}$$

Where the forces grouped in $F_{i\ const}$ are constant during one Δt.
For a given initial condition $(X_i, X_{i-k}, X_{i-k+1})$, the state of the system at the end of the time interval i, i.e. X_{i+1}, can be calculated by solving the differential Equation (2.23), leading to

$$X_{i+1} = e^{A_i \Delta t} X_i + (e^{A_i \Delta t} - I) A_i^{-1} \cdot F_{i\ const} \tag{2.24}$$

By putting k+1 consecutive solutions of sub-intervals into one state vector, one can derive matrix Equation (2.25).

$$\begin{bmatrix} X_{i+1} \\ X_i \\ \vdots \\ X_{i-k+1} \end{bmatrix} = \begin{bmatrix} P_i & 0 & \cdots & 0 & S_i & S_i \\ I & 0 & & \cdots & & 0 \\ 0 & I & 0 & \cdots & & 0 \\ \vdots & & \ddots & & & \\ & & & & & \\ 0 & & \cdots & & I & 0 \end{bmatrix} \begin{bmatrix} X_i \\ X_{i-1} \\ \vdots \\ \\ X_{i-k+1} \\ X_{i-k} \end{bmatrix} + \begin{bmatrix} V_i B_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} F_{a_i} + \begin{bmatrix} V_i B_{2_i} \\ 0 \\ \vdots \\ 0 \end{bmatrix} F_{c0} + \begin{bmatrix} V_i B_{3_i} \\ 0 \\ \vdots \\ 0 \end{bmatrix} K_2 \alpha_1 \tag{2.25}$$

Where

$$\begin{aligned} P_i &= e^{A_i \Delta t} \\ S_i &= \frac{1}{2}(e^{A_i \Delta t} - I) A_i^{-1} Q_i \\ V_i &= (e^{A_i \Delta t} - I) A_i^{-1} \end{aligned} \tag{2.26}$$

Renaming the matrices we have:

$$\hat{X}_{i+1} = \bar{A}\hat{X}_i + \bar{B}_1 F_{a_i} + \bar{B}_2 F_{c0} + \bar{B}_3 K_2 \alpha_1 \tag{2.27}$$

The above steps helped eliminate the delay term in the dynamical equations of the system by absorbing it into the longer state vector. However we may still have time-varying coefficients in the matrix equations due to moving contact point between workpiece and tool or the robot that in many cases cannot be neglected. In such cases lifting of the inputs and

outputs would help reaching at an LTI system from a linear time-periodic system. This will be explained in more detail in chapter 3.

In the next chapter the methodology developed in this chapter will be applied to a machining example.

CHAPTER 3

Turning of Thin-Walled Tube: Modeling

The methodology described in Chapter 2 can be applied to complicated workpiece shapes. In most real applications the workpiece or the tool will be moving resulting in time varying γ's and consequently time-varying coefficients in Equations (2.18) and (2.25). An example that has a lot of practical use is the turning process of thin-walled tubes. Thin-walled cylindrical parts have many applications in industry such as engine liners, turbine casings and vehicle wheels. The same methodology described in the previous chapters can be used here up until deriving equations similar to 2.25 and 2.27. However an extra step needs to be taken here to go from the now time-varying equations to a set of time invariant ones. This is going to be a process of lifting of inputs and outputs which will be explained later in this chapter.

3.1 Finite Element Method Modeling

The workpiece is a thin-walled tube made of an aluminum alloy and has the sizes and characteristics shown in Table 3.1. It is going to be clamped at one end with a three jaw chuck. Figure 3.1 shows the hollow cylinder and its three jaw chuck held portions.

mass density	Young's modulus	Poisson's ratio	inner diameter	thickness
7800 kg/m^3	$21 \times 10^7 Pa$	0.3	24 cm	6 mm

Table 3.1: Properties of the tube

Knowing that closed form solutions for the modes of vibration of the clamped cylinder are not easily achievable, an FEM model of the workpiece was created in Abaqus. Figure 3.1

shows the hollow cylinder and its three jaw chuck held portions.

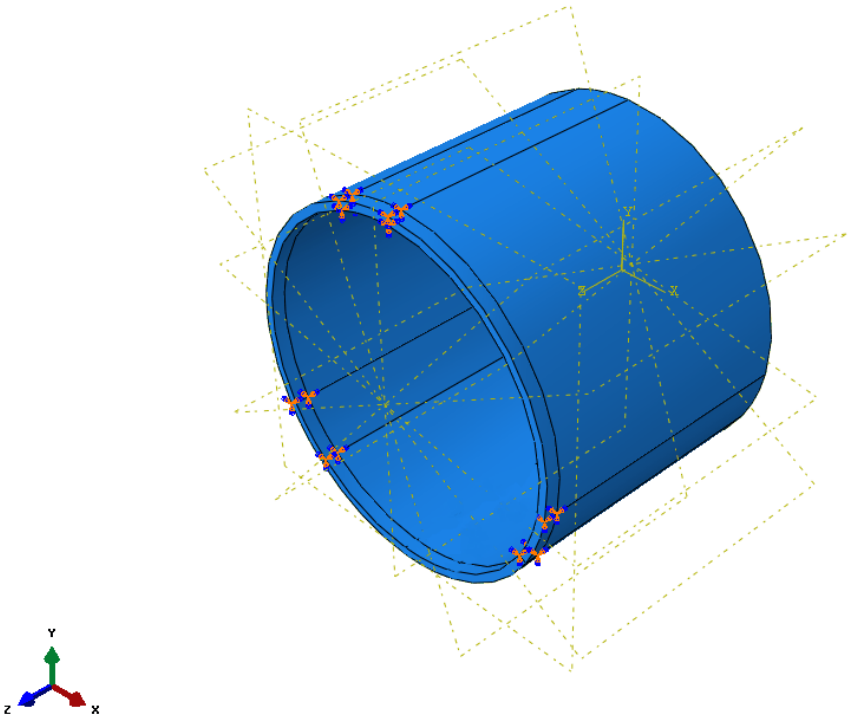


Figure 3.1: Model of the cylinder and the clamped sections

The workpiece was meshed as shown in Figure 3.2 with eight-node trilinear hexahedral elements. There are 4 rows of 18 elements. Three of the elements on the last row (8 nodes for each) are going to be encastred. As the nodes on these elements will have no degrees of freedom the total number of degrees of freedom for this model assuming 3 DOF for each free node will be 468 $((18\times2\times5-3\times8)\times3)$. After some manipulations on the node and element numbers to put them in an appropriate order, a frequency study was performed in Abaqus resulting in natural frequencies and modes of vibration i.e. eigenvalues and eigenvectors of the clamped cylindrical model. Figure 3.3 shows the first 7 modes of vibration. Table 3.2 shows the first 10 natural frequencies of the model. It was decided to keep only 7 modes for the purpose of modeling the workpiece given that the higher modes have higher frequencies and will be more damped and less significant as explained in Section 2.2.2. Each eigenvector

has a length of 468, as many as degrees of freedom.

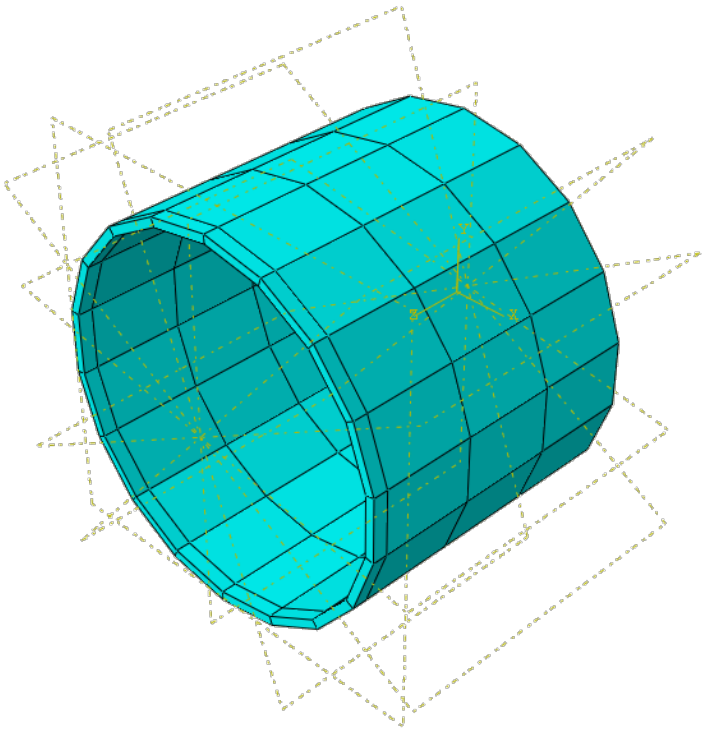


Figure 3.2: FEM elements of the cylindrical workpiece model

mode no.	1	2	3	4	5	6	7	8	9	10
ω^2	60435	60435	177627	177627	406833	418586	445321	687745	816184	816185

Table 3.2: First 10 modes of vibration for the clamped cylinder

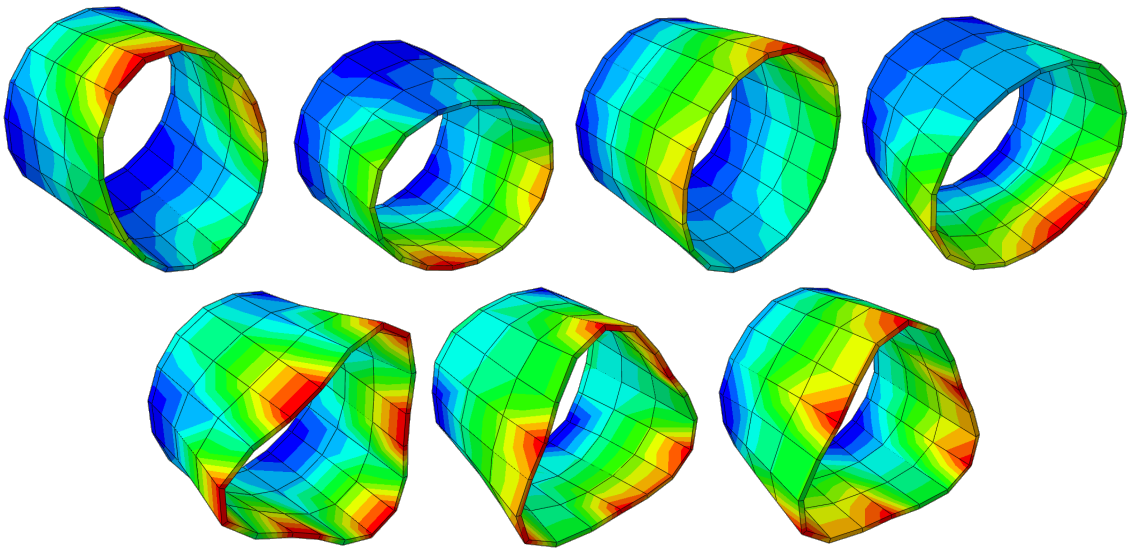


Figure 3.3: First 7 modes of vibration for the cylinder

It is assumed that the tool is performing inside cutting while the robot arm is placed on the outside surface of the thin-walled cylinder opposite the tool. In reality the cylinder would be rotating and the cutting tool would be stationary. For the sake of modeling however we assumed that the workpiece is stationary and the cutting tool and the robotic arm are rotating around the workpiece. In essence these two scenarios are equivalent because the rotational velocity is constant during the operation and there is no acceleration to cause inertial force components on the workpiece.

In the case of a cylinder there are shape function matrices rather than vectors because the force can have components in three dimensions. Since eight node trilinear hexahedral elements are used here shape functions can be extracted from Table 2.1. In current work it was assumed that the cutting only happens on the row of elements farthest from the chuck. It is in a sense the worst case scenario since flexibility of the workpiece is higher the farther away a point on it is from the chuck. Having 18 elements in this row there can be 18 different shape function matrices (γ's) defined for the centers of the sides of these elements that will be in contact with the tool and 18 for the centers of the sides that will be contact with the robotic arm manipulator. These will be some combination of the matrix β_1 or β_2 below with

zeros for the rest of the elements of γ.

$$\beta_1 = \begin{bmatrix} \Psi & \Psi & \sigma & \sigma & \Psi & \Psi & \sigma & \sigma \end{bmatrix}, \quad \beta_2 = \begin{bmatrix} \sigma & \sigma & \Psi & \Psi & \sigma & \sigma & \Psi & \Psi \end{bmatrix},$$

where

$$\Psi = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \quad and \ \sigma = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.1}$$

To make it clearer for example in element number 1, γ_{tw} would be $[\beta_1\ 0\ ...\ 0]$ meaning that nodes 1,2,5 and 6 of element 1, which are the nodes on the bottom, face each receive $\frac{1}{4}$ of the total force applied to the center of the bottom face of element 1 and the rest of the nodes including the ones on other elements receive no force at all.

As explained in Section 2.2.1 in computing γ the local natural coordinates of the element are used. In the case of the thin tube, the local coordinates of the elements are such that the local y axis would be in the radial direction with respect to the geometric center of the tube. The global coordinate system is fixed in space and actually not aligned with any of the local coordinate systems. So in this case we need a rotation matrix (R_i) defined for each element to be able to go from the local coordinates of that element to the global FEM coordinates of the piece. Following the same concept as explained in Section 2.2.2 the modal matrix ϕ and the spectral matrix ω^2 were created from the eigenvectors and eigenvalues of the workpiece FEM model.

Before considering the moving force situation a test on the static force was conducted to check if the FEM results from Abaqus would match the approximate results achieved by using the matrix equations only considering the first 7 natural frequencies of the cylindrical workpiece. For this purpose a radial 8 N force was applied to the cylinder as shown in Figure 3.4. The displacements in global y direction of one row of nodes on the outside of the cylinder were compared as calculated through reduced matrix equations and as generated from the FEM software. Table 3.3 shows the displacement values generated from each

method and the relative error. The values are close enough to call the approximation good. Later in Section 4.2 we will address the spillover effect in truncating the higher modes and whether a controller designed on fewer modes performs well on higher mode plant.

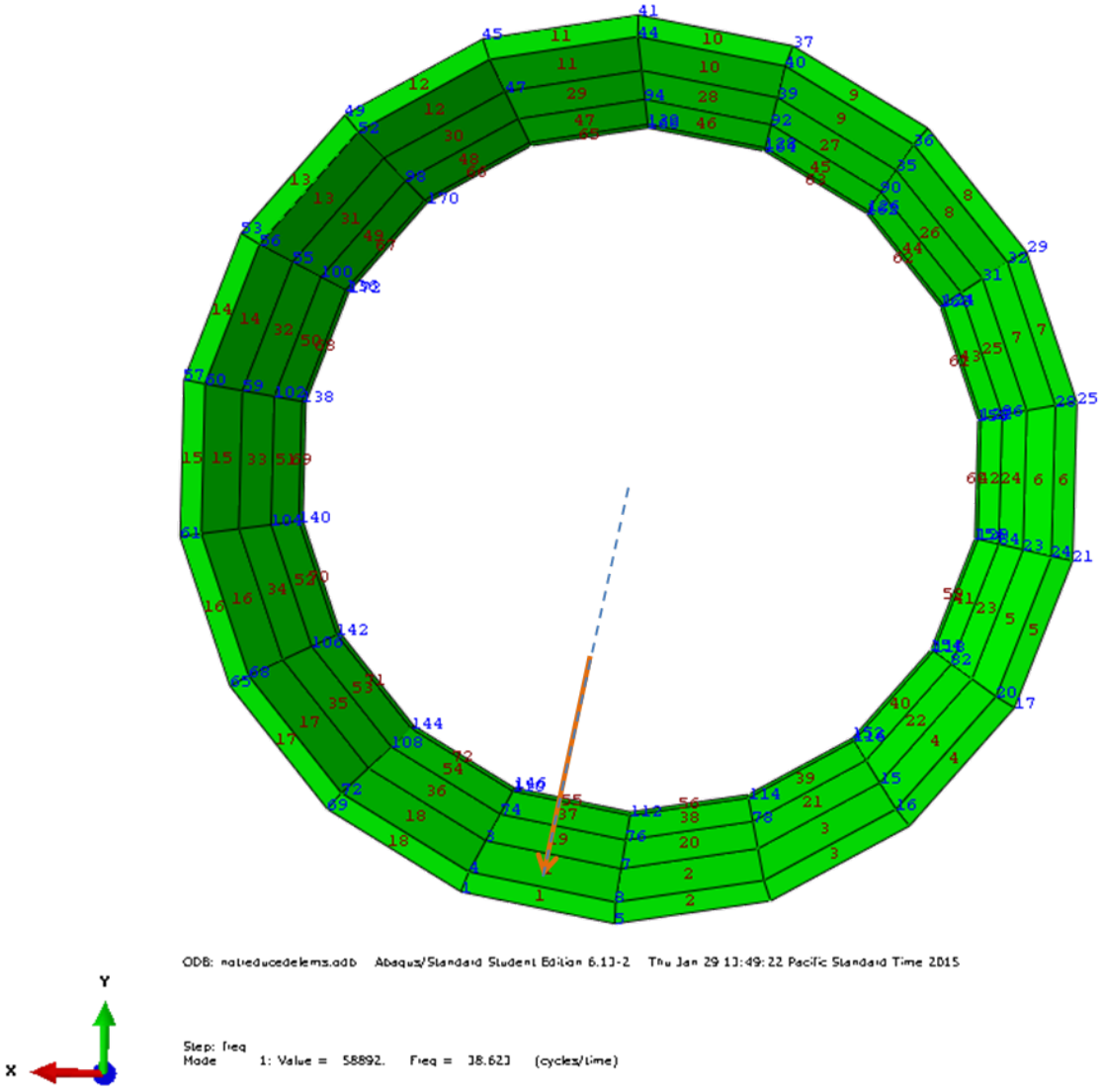


Figure 3.4: Test radial force applied to element 1 of the cylinder

Element no.	Displacement from		Relative error (Matlab-Abaqus)/Abaqus
	FEM (Abaqus) 1e-3 ×	Reduced order modal equations (Matlab) 1e-3 ×	
1	-0.119940000000000	-0.104559006823022	-0.128239062672818
2	-0.077427000000000	-0.077390566182240	-0.000470557011894
3	-0.034350000000000	-0.039539986810888	0.151091319094274
4	-0.015363000000000	-0.017863661733957	0.162771706955504
5	-0.015202000000000	-0.016183133304872	0.064539751669012
6	-0.016235000000000	-0.017483483400488	0.076900733014326
7	-0.006567400000000	-0.007970054196326	0.213578310492145
8	0.012246000000000	0.009956735907117	-0.186939743008571
9	0.027958000000000	0.024461446567872	-0.125064505047842
10	0.028110000000000	0.025916076103434	-0.078047808486886
11	0.010492000000000	0.012098933778290	0.153158004030692
12	-0.014754000000000	-0.010866754881578	-0.263470592274797
13	-0.033446000000000	-0.030522495391546	-0.087409693489615
14	-0.038742000000000	-0.036931936862647	-0.046720952386382
15	-0.037447000000000	-0.034840860035384	-0.069595427260278
16	-0.047233000000000	-0.043065872979678	-0.088224906745750
17	-0.079784000000000	-0.070614586528317	-0.114927973925641
18	-0.118800000000000	-0.100011644131153	-0.158151143677159

Table 3.3: Global y direction displacements of nodes on the outer first row subject to 8N radial force from FEM and from reduced order modal equations and their relative error

For simplicity it is assumed that the force is always applied to the center of the element face it is on and as the force or the cutting tool in this case moves on the workpiece it jumps

from face center of one element to the face center of the next element. If the elements are small enough this is a reasonable approximation.

In order to get a more realistic model, structural damping was added to the workpiece. Structural damping is the $[C_\phi]$ term in (2.11). The structural damping is modeled to increase for higher modes. We used modal damping defined as in [CMP02]: a diagonal matrix whose i^{th} diagonal coefficient is $2\xi_i\omega_i$ where ξ_i is the damping ratio for mode i. Some researchers assign higher damping ratios to higher modes but even if the damping ratio is kept at a constant average in modeling, the damping itself is increased for higher modes due to having higher ω_i. In this work a constant average number based on [AA99] was picked to create the modal damping matrix.

The same robotic arm model as described in Section 2.3 and shown in Figure 2.4a is used in this simulation. In this case study the tool is placed inside the cylinder performing inside turning while the robotic arm is placed on the outside surface of the cylindrical workpiece. Figure 3.5 shows a schematic of the placement of cutting tool and the robotic arm on the cylinder.

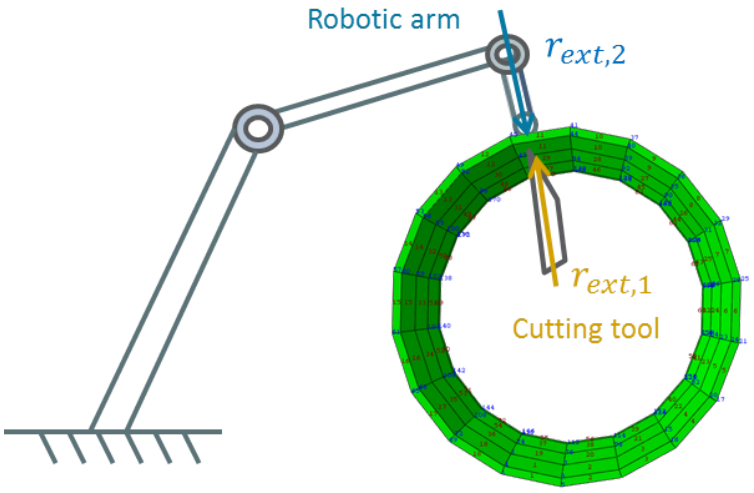


Figure 3.5: Placement of the cutting tool and robotic arm on the cylindrical workpiece

Same rotation matrices calculated for each element before can be used to distribute the force of the robotic arm on to the finite element nodes while shape function matrices for

the robotic arm should reflect that the arm is placed outside the cylinder. The state space realization of the whole system is created. The state vector consists of the 7 modal states for the workpiece and their derivatives as well as 4 states for the arm making it an 18 state system. A matrix equation similar to Equation (2.19) would describe the system with each $\gamma_{tw,i}$ and $\gamma_{rw,i}$ multiplied by its respective rotation matrix R_i. In this case system matrices vary with time as the tool and robot arm change position from element to element.

$$\dot{X} = A(t)X(t) + Q(t)X(t-T) + B_1F_a(t) + B_2(t)F_{c0}(t) \tag{3.2}$$

$A(t)$ and $Q(t)$ have terms containing $\gamma_{tw}(t), R(t)$ for the cutting tool position and $\gamma_{rw}(t), R(t)$ for the robot arm position. B_2 has $\gamma_t w(t)$ and $R(t)$ in it. So as the position of the cutting tool and robot arm change from element to element the $A(t)$, $Q(t)$ and $B_2(t)$ matrices vary. As described before in this case it was assumed that the cutting only happens on the row of elements farthest from the chucks and the tool and robot arm are fixed in place relative to each other such that the robot arm is always on the other side of the element with the cutting tool on it. As the workpiece turns the elements on which the cutting tool and the arm are located change. Since there are 18 elements on this row after 18 steps the workpiece is back at its original position. The process model is simplified by assuming the arm and the cutting tool move discretely from center of one element to the center of the next element. So in one revolution of the workpiece turning there will be 18 different "A" matrices. On the other hand the "A" matrix changes periodically with time meaning after 18 steps the same matrix as the first step is going to describe the system.

3.2 Semi-Discretization and Lifting of the Inputs and Outputs

Semi-discretization is used to eliminate the delay term by absorbing it into the states. Another assumption is that the semi-discretization is done by dividing the time it takes for the workpiece to complete one revolution by the number of elements the tool has to cut through in one revolution, eighteen in this case. The semi-discretized system will be time-periodic

rather than time-varying. In the case of tube turning the periodicity and the delay in the equations are equal and are represented with T in the equations. For semi-discretization that period T is divided into k equal subdivisions of time.

Using Equations (2.20), (2.21) and (2.22) with k taken as number of elements cut in one revolution and the same method explained in Section 2.6.2 a matrix equation like (2.25) was written with P_i, S_i and V_i defined as in (2.26). Equation (2.27) is rewritten as (3.3) below.

The semi-discretized system will have no delay term but will have time periodic "$\mathcal{A}$" and "$\mathcal{B}$" matrices.

$$\begin{bmatrix} X_{i+1} \\ X_i \\ \vdots \\ X_{i-k+1} \end{bmatrix} = \mathcal{A}_i \begin{bmatrix} X_i \\ X_{i-1} \\ \vdots \\ \\ X_{i-k+1} \\ X_{i-k} \end{bmatrix} + \mathcal{B}_1 F_{a_i} + \mathcal{B}_{2,i} F_{c0_i}$$

$$\widehat{X}_{i+1} = \mathcal{A}_i \widehat{X}_i + \mathcal{B}_1 F_{a_i} + \mathcal{B}_{2,i} F_{c0_i} \tag{3.3}$$

There is a periodic nature to these matrices and their sequence will repeat after 18 time steps or one revolution. If stability analysis is the only purpose, at this point one can just multiply all the $\mathcal{A}_i$'s in one revolution and study the eigenvalues of this new matrix which will be representative of the stability of the system. To design controllers however, lifting of the inputs and outputs is a method that can turn a linear time periodic system into a linear time invariant one. Lifting was first introduced by Meyer, et al [MB75] and has been used in different applications ever since. The derivation of (3.4) leads to relating the state of the system at $(i + k)$ to the state of the system at (i) given all the force inputs in this k step time period.

$$\widehat{X}_{i+1} = \mathcal{A}_i \widehat{X}_i + \mathcal{B}_1 F_{a_i} + \mathcal{B}_{2,i} F_{c0_i}$$

$$\widehat{X}_{i+2} = \mathcal{A}_{i+1}(\mathcal{A}_i \widehat{X}_i + \mathcal{B}_1 F_{a_i} + \mathcal{B}_{2,i} F_{c0_i}) + \mathcal{B}_1 F_{a_{i+1}} + \mathcal{B}_{2,i+1} F_{c0i+1}$$

.

.

.

$$\widehat{X}_{i+k} = \mathcal{A}_{i+k-1}\mathcal{A}_{i+k-2}...\mathcal{A}_i \widehat{X}_i +$$

$$\begin{bmatrix} \mathcal{A}_{i+k-1}\cdots\mathcal{A}_{i+2}\mathcal{A}_{i+1}\mathcal{B}_1 & \cdots & \mathcal{A}_{i+k-1}\mathcal{A}_{i+k-2}\mathcal{B}_1 & \mathcal{A}_{i+k-1}\mathcal{B}_1 & \mathcal{B}_1 \end{bmatrix} \begin{bmatrix} F_{a_i} \\ F_{a_{i+1}} \\ \vdots \\ F_{a_{i+k-1}} \end{bmatrix} + \tag{3.4}$$

$$\begin{bmatrix} \mathcal{A}_{i+k-1}\cdots\mathcal{A}_{i+2}\mathcal{A}_{i+1}\mathcal{B}_{2,i} & \cdots & \mathcal{A}_{i+k-1}\mathcal{B}_{2,i+k-2} & \mathcal{B}_{2,i+k-1} \end{bmatrix} \begin{bmatrix} F_{c0_i} \\ F_{c0_{i+1}} \\ \vdots \\ F_{c0_{i+k-1}} \end{bmatrix}$$

We will denote the system matrices derived this way with subscript "rev".

$$\widehat{X}_{i+k} = A_{rev}\, \widehat{X}_i + B_{1,rev} \begin{bmatrix} F_{a_i} \\ F_{a_{i+1}} \\ \vdots \\ F_{a_{i+k-1}} \end{bmatrix} + B_{2,rev} \begin{bmatrix} F_{c0_i} \\ F_{c0_{i+1}} \\ \vdots \\ F_{c0_{i+k-1}} \end{bmatrix} \tag{3.5}$$

For the output matrix assuming that there is a sensor on the tool measuring the displacements of the workpiece underneath the tool, C_{rev} was defined as in (3.6). This would generate a vector output of the displacements of each of the elements on the machined row at the time the tool was passing them. (3.5) and (3.6) define an LTI system.

$$C_{rev} = \begin{bmatrix} 0 & C_{i-1} & & & 0 \\ 0 & 0 & C_{i-2} & & \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & & 0 & C_{i-k} \end{bmatrix}, \tag{3.6}$$

where

$$C_j = \begin{bmatrix} R_j^{-1}\gamma_{rw,j}\phi \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T$$

Equations (3.5) and (3.6) show how this method is a lifting of inputs and outputs that would lead to a linear time invariant system. The LTI system described by (3.5) and (3.6) is very high dimensional. Having 18 states for the robot assisted machining after keeping only the first 7 modes for the workpiece, the semi-discretization and absorbing the delay into the states increases the number of states to 342 as k+1 state vectors need to be stacked on top of one another. Lifting the inputs and outputs does not increase the number of states but rather increases the size of the input and output vectors. Not all these 342 states are controllable and observable however, and for the control design purpose a minimal realization of the system is more appropriate. The minimal realization, which is a result of stable pole zero cancellations, reduces the number of states in the system to 90. Number of unstable modes before and after the minimal realization were compared to make sure no unstable modes get cancelled.

Having this representation of the machining process worked out, the control applied through the robotic arm for stabilizing purposes is to be explored in the next two chapters.

CHAPTER 4

Optimal Control Design for Machining of Thin-Walled Tube

After getting the model of the system to be linear time invariant, conventional control strategies can be applied to the problem. In this work linear quadratic state feedback regulator was the control scheme of choice. While access to the states of the system is not directly possible through sensing, it is possible through use of an observer or estimator. In this work we used linear quadratic state estimators also known as Kalman filters. Given a plant model and assumption of process and measurement noise covariance data, the Kalman estimator provides the optimal solution to the discrete estimation problem defined by

$$\begin{aligned} x(t+1) =& Ax(t) + Bu(t) + Gw(t) \\ y(t) =& Cx(t) + Du(t) + Hw(t) + v(t) \end{aligned} \tag{4.1}$$

With white process noise w and white measurement noise v satisfying

$$E(w) = E(v) = 0, \quad E(ww^T) = Q_e, \quad E(vv^T) = R_e \tag{4.2}$$

The estimator builds a state estimate $\hat{x}$ that minimizes the steady-state error covariance P given below.

$$P = \lim_{t\to\infty} E\big((x - \hat{x})(x - \hat{x})^T\big) \tag{4.3}$$

Using the state estimates from the Kalman estimator, linear quadratic state feedback regulator can be designed. Considering the discrete time state space plant model of the form

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{4.4}$$

The performance index is

$$J(x(0), u) = \sum_{k=0}^{\infty} \left(x^*(k) Q x(k) + u^*(k) R u(k) \right) \tag{4.5}$$

Assuming Q and R are self-adjoint and R is positive definite, the LQR design finds the optimal gain matrix K for the state feedback such that $u(t) = -Kx(t)$ minimizes the performance index (4.5) over the set of all control functions $\tilde{u}$ such that both $\tilde{u}$ and x are square-summable sequences of complex n-vectors. The K calculated from the LQR control design can be used to multiply the estimated state of the system generated by the Kalman filter. The focus of tuning for the LQR controller would be faster settling time given that the whole purpose of robot assisted machining is to be able to cut faster without running into chatter instability.

The described combination of linear quadratic estimator and regulator is the control design scheme of choice for this work.

In Section 3.1 the cylindrical workpiece machining process was modeled through FEM, modal analysis and truncation, semi-discretization and finally lifting of the inputs and outputs. All these steps led us to an LTI model of the system that can be appropriate for control design. One seemingly drawback of the last step, i.e. lifting of the inputs and outputs, that was necessary for turning the time-periodic system into a time invariant one, is that the control will have to see one batch of measured plant outputs at once and create one batch of control outputs at once. In other words there will be no feedback during one revolution of the workpiece turning and the control outputs for the 18 time steps which the workpiece takes to make one revolution are created in the beginning of that revolution and control is feedforward until the beginning of the next revolution. At this time the controller receives another batch of the sensor readings as feedback and can generate new inputs for the next revolution. So during each revolution the control is feedforward. After each revolution is complete however, the sensor readings are fed back to the controller who generates the control outputs for the next revolution.

4.1 Control Design for Stability

Before trying any controls on this system it is important to establish whether it is stabilizable with the control input being the force applied to the robot arm actuator F_{a_i} in (3.3). Semi-discretization preserves asymptotic stability of the original system [HIS06] and generally, discretization techniques preserve asymptotic stability for delay differential equations. A system is stabilizable if all the unstable modes are controllable. The eigenvalues of A_{rev} in (3.5) determine whether the system will experience unstable chatter with a set of given parameters. If all eigenvalues of the now discrete system are in magnitude less than 1 the system is stable and there is no chatter. However if there are eigenvalue(s) with magnitude greater than 1 the system has unstable chatter. At this point we can test to see if those unstable modes are controllable through force of the arm's actuator. For that B_{rev} was taken as the input matrix of the system. The method to check the controllability of the unstable modes used here is some variation of the PBH (Popov-Belevitch-Hautus) test. If the left eigenvectors corresponding to unstable eigenvalues of A_{rev} have nonzero products with B_{rev} the system is stabilizable. This test was carried out with many different sets of machining parameters and every time the system rendered either stable or stabilizable encouraging the conclusion that adding the robot arm and actuator can stabilize this system on the onset of chatter.

To thoroughly study the effect the robotic arm has on the cutting process, we can look at the effect it has in terms of adding to the stiffness of the thin-walled workpiece without applying any controls to the actuator and then study the stability with control applied to the arm's actuator. A stability lobe diagram is the plot of maximum cutting stiffness possible, that does not lead to chatter, versus the spindle speed. Higher cutting stiffness results in higher material removal rates and is desirable. A stability lobe diagram was created for the passive cylinder cutting with no robotic arm coupling which is shown in Figure 4.1 with a solid black line. Any point below the lobe lines is stable and chatter free. Points above the lobes lead to chatter and are unstable. Then the same plot was created for the

cylinder and the passive robot arm (no control applied) placed on it which is also shown in Figure 4.1 with red dashed line. The arm alone sometimes improves the stability of the system allowing higher cutting stiffnesses without chatter. The improvement however is very minor. The bottom part of Figure 4.1 zooms in on an example location on the lobes to show the positive although small effect of the passive robotic arm on stability of the system.

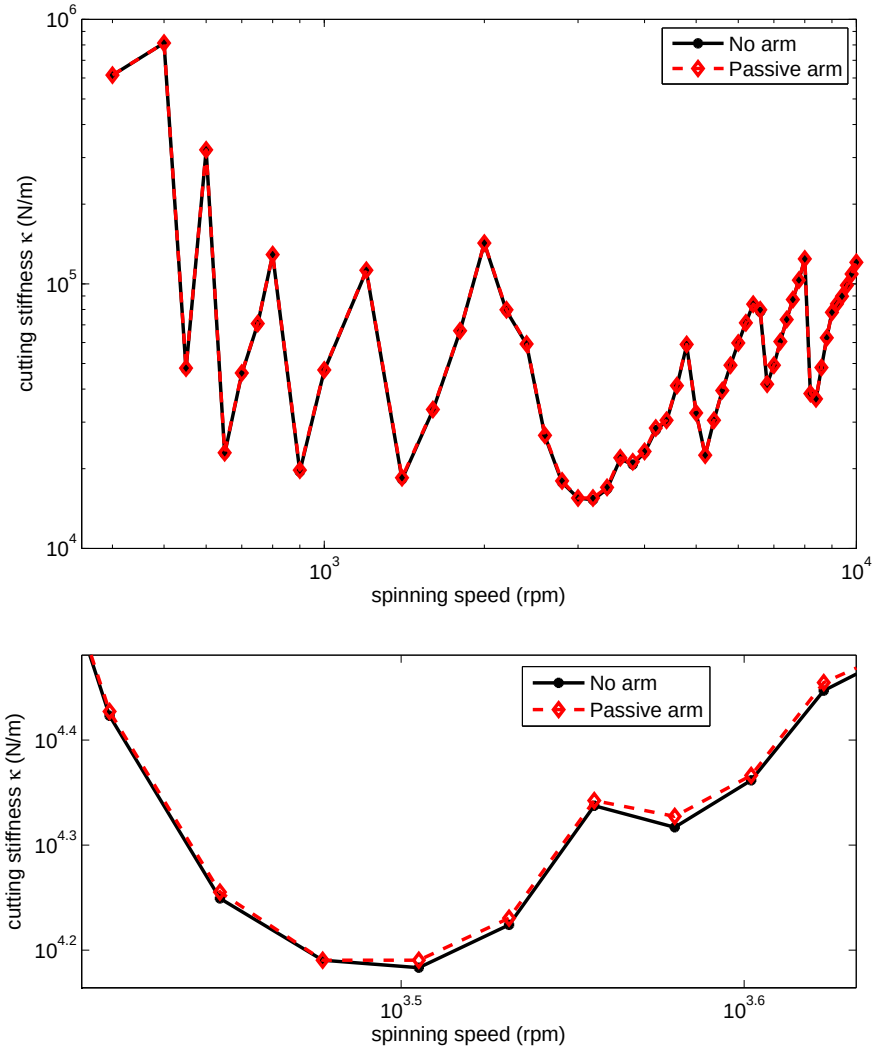


Figure 4.1: Top: stability lobe diagram for inside turning of a thin-walled cylinder with and without help of a passive robotic arm. Bottom: zoomed in to show the small positive effect of the passive arm

Changing the parameters of the system like the stiffness and damping of the robot arm

springs and dampers can affect the stability lobes diagram. Figure 4.2 shows the lobes of the machining system in a case where the robot arm manipulator is less damped and has different stiffnesses. Once again the no arm lobes are shown with solid line (this time violet) and the lobes for the machine coupled with passive robotic arm is shown with dashed line (this time black). As can be seen in Figure 4.2 the robotic arm coupling sometimes improves the stability margins and sometimes makes them worse.

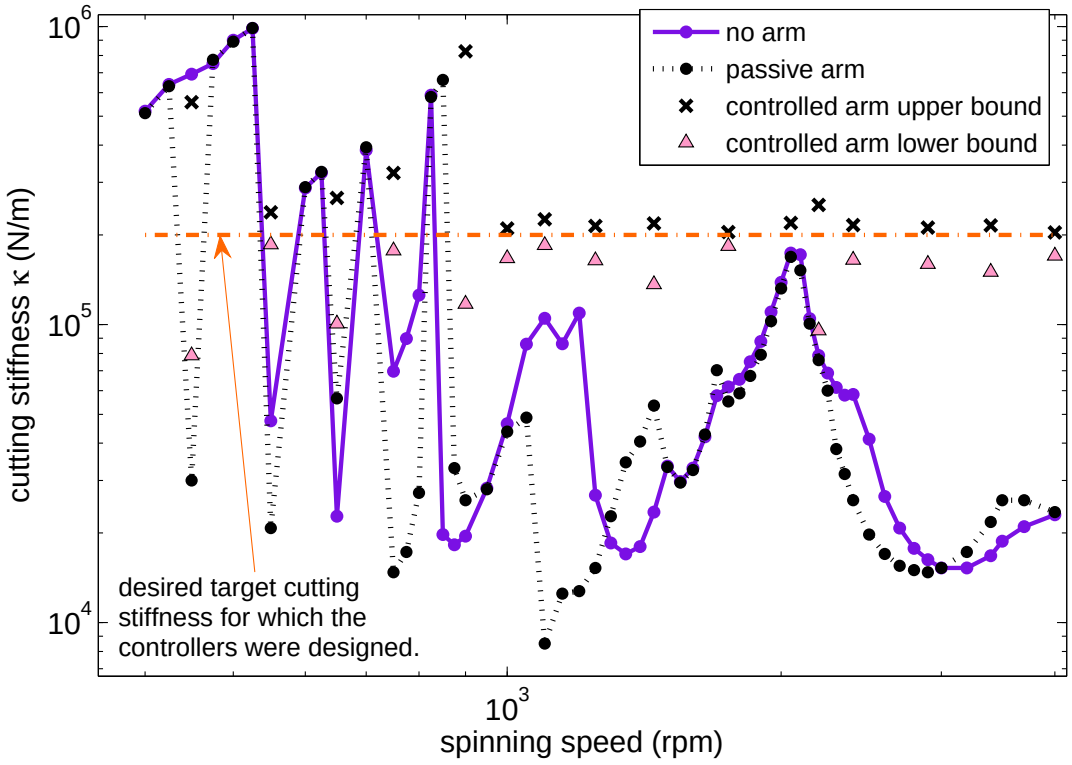


Figure 4.2: Stability lobe diagram for inside turning of a thin-walled cylinder with and without help of a robotic arm. Controller designed on a 7 modes plant model upper and lower bounds of stability shown

Lastly linear quadratic estimators and regulators were designed for specific spindle speeds to stabilize for desired cutting stiffness of $\kappa = 7 \times 10^4 N/m$ for the first set of machining/robot parameters and $\kappa = 2 \times 10^5 N/m$ for the second set. LQR is a state feedback control method. Although we do not have access to all the states through sensing, this system is

fully observable using the sensor on the cutting tool that can generate an output vector of displacements of each element on the machined row as the tool was passing them in one revolution. A linear quadratic estimator that has access to this output vector and the vector of control inputs can estimate the full state of the system, which in turn can be provided to the LQR controller to produce the next control output vector. The controllers designed stabilized the system for the target cutting stiffness shown with a dash-dot line on Figure 4.3 for the system with the first set of machine/robot parameters and on Figure 4.2 for the system with the second set.

Even though the controller is designed for that specific cutting stiffness the controlled system can most of the time handle higher cutting stiffnesses. The highest cutting stiffness (κ) achievable without chatter in the case with control applied through the arm actuator is also determined for each of the desired spindle speeds. For that purpose a series of iterations was run starting from the target κ and increasing κ iteratively, stopping where the system starts to demonstrate chatter instability. $\times$ points in Figure 4.3 and 4.2 signify for their respective spindle speeds the highest cutting stiffness between which and the target κ the system is stable with the controller.

Same kind of iteration was run to find the range the controller can stabilize for cutting stiffnesses smaller than the target it was designed for. The lower bounds are shown with pink triangle points on Figure 4.2 for the second set of machine/robot parameters. The controllers designed for the system with the first set of machine/robot parameters would stabilize for a wide range below the target that would mostly be zero or out of the limits of the y axis in Figure 4.3. In conclusion, the arm control method has sizable robustness around the target even though it was not a design consideration.

The control applied to the workpiece through the arm can raise the maximum stable cutting stiffness substantially in cases where without arm or without controlling the arm, only very low cutting stiffnesses would stay away from chatter.

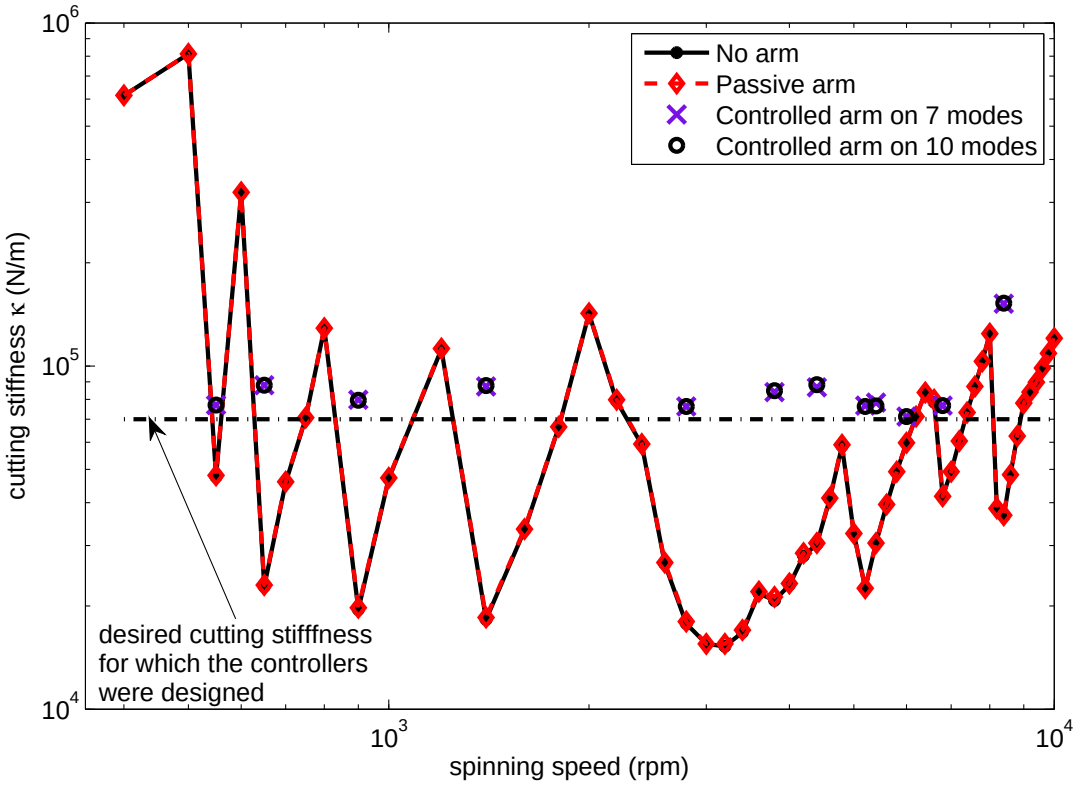


Figure 4.3: Stability lobe diagram for inside turning of a thin-walled cylinder with and without help of a robotic arm. Controllers designed on a 7 modes plant model were applied to both 7 modes and higher modes plant

4.2 Spillover Effect

The controllers were designed for each spinning speed based on a model of the workpiece with the first 7 modes of vibration kept. the $\times$ points on Figure 4.3 are for applying the controller to a truncated model of the plant which only keeps the first 7 modes. In order to assure the designed controllers would be able to stabilize the real plant, they were also applied to a plant with more modes kept to study if there is any spillover effect. Spillover phenomenon occurs when the unmodeled dynamics get excited. Even though viscous and Rayleigh damping as well as the modal proportional damping used to model internal structural damping for

the workpiece in this paper, all increase for higher modes, the effect may still exist. The circle points on Figure 4.1 are generated from applying the control designed for the 7 modes plant to a model of the plant with 10 modes kept. As can be seen these points are almost coincident with the $\times$ points of the 7 modes plant and the difference is very small and in any case does not affect the stability of the closed loop system. Given that even higher modes than those kept in the 10 modes plant would have even higher dampings one can conclude that the spillover effect does not affect the stability in this problem when the controllers are designed for a truncated plant keeping the first 7 modes.

4.3 Control Design for Performance

Thus far the effect of adding a controlled robotic arm to the machining system for better stability was studied. We showed that the addition of the active robotic arm to the system enables us to machine with higher cutting stiffnesses without experiencing chatter. In this section we look at the performance and the final finish of the cut surface. The control design in the previous sections was able to stabilize the system and prevent the onset of chatter for high cutting stiffnesses. A measure of performance used in this section is the final shape of the cut surface to represent the surface finish and whether unwanted deformations have affected the final shape of the workpiece. The circle points in Figure 4.4 show the steady state offset of points around the cylinder's cut surface after cutting. Here the ideal desired position after cut is represented by zero. The figure shows finished surface for a cut done at 900 rpm and cutting stiffness of $7 \times 10^4 N/m$ with a constant portion of cutting force equal to 49 N which in this case represents nominal depth of cut of 0.7mm. As seen in Figure 4.4 the finish is not uniform and there is as much as around 0.4 mm of offset from the desired finished position at some locations on the cylinder.

Having a more exact final cut shape and better surface finish is in a sense a tracking problem so adding an internal model to the control design must help improve the surface finish and the final cut shape. In the next step integral action was added to the control

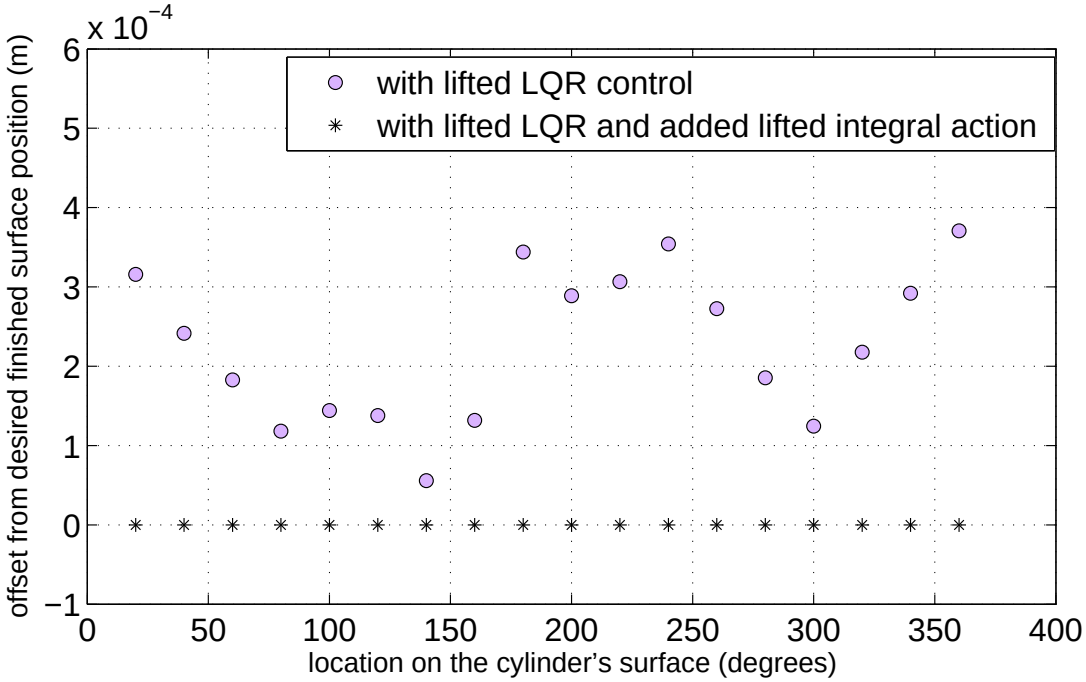


Figure 4.4: Final cut surface for actively controlled robotic arm augmented machining system for the thin-walled cylinder at 900 rpm and $\kappa = 7 \times 10^4 N/m$

in the lifted domain and we studied its effect on the final cut shape and surface finish. To add integral action to the controller, the output which is the sensor measurements of the displacements of the tool center point is fed back to an integrator and essentially the states of the system are augmented by this integral state. Then the same LQR design described earlier is applied to the new augmented system creating the state feedback gain as well as the integral gain for the output feedback. As mentioned all this is still in the lifted domain. The star points in Figure 4.4 show the steady state offset of the points around the cylinder after cutting with the integral action added. As shown, integral action in the lifted domain does achieve its purpose and brings the offset down to zero.

As mentioned before the focus of tuning for the LQR controllers designed for this application is to be on fast settling time so that the method can in fact increase productivity of the machining plant. The described system of thin-walled tube and the robotic arm has many states even after truncating higher modes. Semi-discretization and the lifting of inputs

and outputs further increases the number of states and the number of inputs and outputs to the system. Even after performing minimal realization the system has 90 states. And these states now do not have physical meaning. Tuning an LQR controller for a high dimensional MIMO system is not an easy and intuitive task. In Chapter 6 a new tuning method for linear quadratic regulators and estimators is going to be introduced which is based on the output characteristics and makes tuning more intuitive. Here the results of applying that tuning method to the control of the high dimensional model of the thin tube cutting is shown in Figure 4.5. The system with the controller tuned with the new method settles to zero in less than 15 seconds after applying the 49 N constant cutting force to the workpiece.

The method used for tuning penalizes the output and its derivatives through a weighting matrix. The cost function is:

$$J = \sum \left(\begin{bmatrix} y \\ \frac{dy}{dt} \\ \frac{d^2y}{dt^2} \end{bmatrix}^T \begin{bmatrix} W_1 I & 0 & 0 \\ 0 & W_2 I & 0 \\ 0 & 0 & W_3 I \end{bmatrix} \begin{bmatrix} y \\ \frac{dy}{dt} \\ \frac{d^2y}{dt^2} \end{bmatrix} + \begin{bmatrix} u \\ \frac{du}{dt} \end{bmatrix}^T \begin{bmatrix} RI & 0 \\ 0 & R_d I \end{bmatrix} \begin{bmatrix} u \\ \frac{du}{dt} \end{bmatrix} + [y_i]^T q_i I [y_i] \right) \quad (4.6)$$

where y is the output that is the vector of displacements under the cutting tool in one revolution, here. y_i is the integrator state since the controller includes an integrator.

with weighting matrices components:

W_1	W_2	W_3	R	R_d	q_i
6×10^5	2×10^{-2}	2×10^{-3}	1×10^{-4}	1×10^{-4}	2.1×10^9

Table 4.1: Tuning parameters

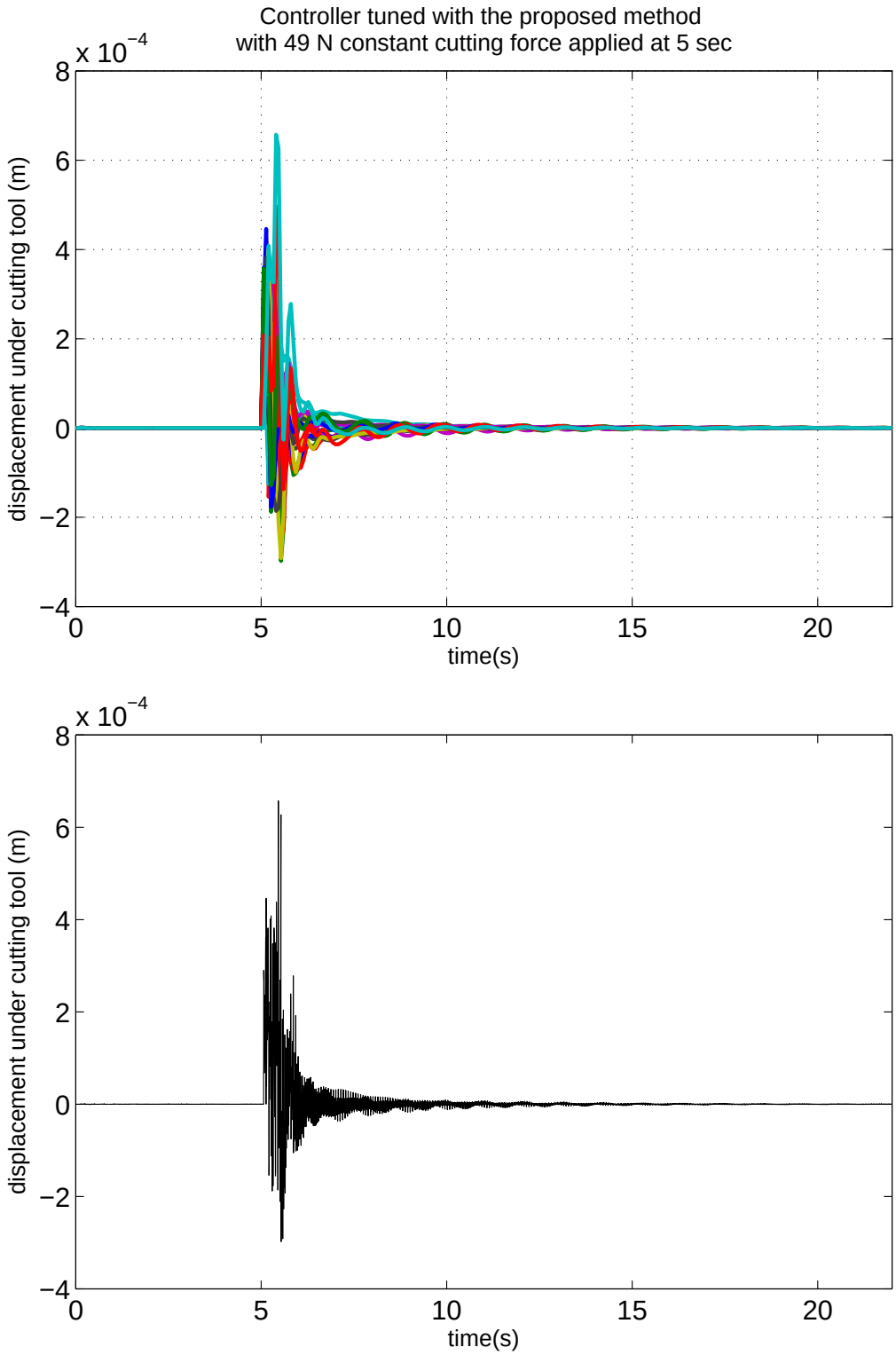


Figure 4.5: Fast settling time achieved through new method of tuning. Top: Displacement of different elements is shown by different colors. Bottom: The real-time output of the sensor on the cutting tool

This method of tuning will be derived and explained in more detail in Chapter 6.

4.4 Conclusion

In this chapter the optimal control method was applied to the LTI model of the inside turning of a thin-walled tube. The actively controlled robot manipulator coupling with the workpiece helps increase the productivity of the machining plant because through using the stabilizing controllers on the robot manipulator, the machining can proceed at higher speeds and depths of cut without chattering instability. In addition it was shown that the control could improve the quality of final cut surface. In conclusion, the robotic arm control augmentation to the machining system not only helps with productivity and chatter avoidance but also helps achieve desired final cut shape and surface finish for the workpiece.

While this chapter presented LQR as the method of choice for control of the robot assisted machining system, Chapter 5 will present design of controllers that are robust to changes in cutting stiffness as this number is a result of several machining conditions and can be uncertain. The difficulty however is the presence of the parameter κ in all the state matrices. We will discuss a methodology in Chapter 5 to separate the uncertainty from the rest of the plant and then proceed to design robust controllers.

CHAPTER 5

Robust Control Design for Thin-Walled Machining

In Chapter 3 the stability lobe diagram for the turning process of a thin tube was generated for specific material and robot manipulator characteristics. The lobes were created based on a model generated from the method discussed in Chapter 2 which includes finite element approximation, taking the plant to modal domain, truncating higher modes, doing a state expansion through semi-discretization to absorb the delay and lifting of the inputs and outputs which was detailed in Chapter 3. In Chapter 3 a linear quadratic regulator was designed as the controller of the system to apply stabilizing control input to the machining system through the robot arm. The control design in the previous chapter was based on a nominal model. But in reality there may be uncertainties in the system's model and we may need to have robustness to those uncertainties as a control design consideration. Depending on what part of the model is uncertain we would want to separate that part from the rest of the system in order to prepare a robust control oriented model.

The parameter κ which is the cutting stiffness number depends on several machining conditions and parameters and is not easily determined to an exact accuracy. Thus a robust control method that treats κ as uncertainty is reasonably desirable. For a general robust control design method the uncertainty needs to be isolated from the rest of the model. The state space of the model derived as described however, includes κ in a lot of elements of A and B matrices. Therefore to be able to isolate the uncertainty of cutting stiffness, a new method of modeling and semi-discretization is suggested in this chapter.

5.1 Robust Control Oriented Modeling and Semi-Discretization Method

In Figure 4.2 it was shown that the controller designed for the target κ with the LQR method could handle a range of κ values above and below the target. If we would like a larger margin however, we would need to design a controller with robustness being a design consideration.

In order to isolate κ we go back to Equation (2.18) and take the term containing κ out from element A_{21}. The new A matrix created this way will be denoted by A_1 which will have no term with κ in it. If uncertainty is isolated in a block the plant can be represented as in Figure 5.1 with a feedback loop containing a delay.

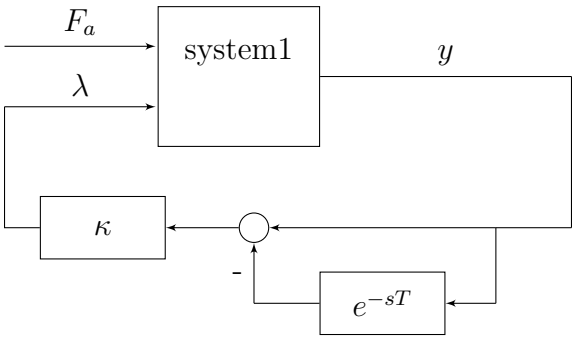


Figure 5.1: Block diagram of the cutting system with isolated cutting stiffness (κ)

The input signal labeled as λ and the output signal labeled as y in Figure 5.1 can be written as

$$\lambda(t) = \kappa\big(y(t) - y(t-T)\big) \tag{5.1}$$

$$y(t) = L(t).x(t) \tag{5.2}$$

Where

$$L = \begin{bmatrix} \gamma_{tw}\phi & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{5.3}$$

That leads to the state propagation equation

$$\dot{X}(t) = A_1(t)X(t) + B_1F_a(t) + B_2(t)F_{cut,0} + B_2(t)\kappa L(t)\big(X(t) - X(t-T)\big) \tag{5.4}$$

Consequently

$$\dot{X}(t) = A_1(t)X(t) + B_1F_a(t) + B_2(t)F_{cut,0} + B_2(t)\lambda(t) \tag{5.5}$$

Applying semi-discretization to this equation by replacing the e^{-sT} block in Figure 5.1 with z^{-k} where k is the number of discretization points ($T = k\Delta t$) yields

$$\dot{X}(t) = A_{1_i}X(t) + \underbrace{B_1F_{ai} + B_{2i}F_{cut,0} + B_{2i}\lambda_i}_{F_{i\,const}}, \quad \text{For one } \Delta t \tag{5.6}$$

$$\lambda_i = \kappa(y_i - y_{i-k}) \tag{5.7}$$

For one Δt, this has a closed form solution:

$$X_{i+1} = e^{A_{1_i}\Delta t}X_i + (e^{A_{1_i}\Delta t} - I)A_{1_i}^{-1}F_{i\,const} \tag{5.8}$$

Leading to

$$X_{i+1} = P_iX_i + S_iB_1F_{ai} + S_iB_{2i}F_{cut,0} + S_iB_{2_i}\lambda_i \tag{5.9}$$

$$P_i = e^{A_{1_i}\Delta t} \tag{5.10}$$

$$S_i = (e^{A_{1_i}\Delta t} - I)A_{1_i}^{-1} \tag{5.11}$$

At this point to get the state k time steps later from the current state, we perform a

lifting of the inputs and outputs:

$$X_{i+k} = A_r X_i + B_{1r} \begin{bmatrix} F_{ai} \\ F_{ai+1} \\ \vdots \\ F_{ai+k-1} \end{bmatrix} + B_{2r}(\begin{bmatrix} \lambda_i \\ \lambda_{i+1} \\ \vdots \\ \lambda_{i+k-1} \end{bmatrix} + \begin{bmatrix} F_{cut,0} \\ F_{cut,0} \\ \vdots \\ F_{cut,0} \end{bmatrix}) \tag{5.12}$$

$$\underbrace{\begin{bmatrix} y_i \\ y_{i+1} \\ \vdots \\ y_{i+k-1} \end{bmatrix}}_{\tilde{y}_i} = C_r X_i + D_{1r} \underbrace{\begin{bmatrix} F_{a_i} \\ F_{ai+1} \\ \vdots \\ F_{ai+k-1} \end{bmatrix}}_{F_{a_{lift,i}}} + D_{2r}(\underbrace{\begin{bmatrix} \lambda_i \\ \lambda_{i+1} \\ \vdots \\ \lambda_{i+k-1} \end{bmatrix}}_{\lambda_{lift,i}} + \begin{bmatrix} F_{cut,0} \\ F_{cut,0} \\ \vdots \\ F_{cut,0} \end{bmatrix}) \tag{5.13}$$

where

$$\begin{aligned} A_r &= P_{i+k-1} P_{i+k-2} \cdots P_i \\ B_{1_r} &= \begin{bmatrix} P_{i+k-1} P_{i+k-2} \cdots P_{i+1} S_i B1 \\ P_{i+k-1} P_{i+k-2} \cdots P_{i+2} S_{i+1} B1 \\ \vdots \\ P_{i+k-1} S_{i+k-2} B_1 \\ S_{i+k-1} B_1 \end{bmatrix}^T \\ B_{2_r} &= \begin{bmatrix} P_{i+k-1} P_{i+k-2} \cdots P_{i+1} S_i B_{2_i} \\ P_{i+k-1} P_{i+k-2} \cdots P_{i+2} S_{i+1} B_{2_{i+1}} \\ \vdots \\ P_{i+k-1} S_{i+k-2} B_{2_{i+k-2}} \\ S_{i+k-1} B_{2_{i+k-1}} \end{bmatrix}^T \end{aligned} \tag{5.14}$$

and

$$C_r = \begin{bmatrix} L_i \\ L_{i+1}P_i \\ \vdots \\ L_{i+k-1}P_{i+k-2}\cdots P_i \end{bmatrix}$$

$$D_{1_r} = \begin{bmatrix} 0 & L_{i+1}S_iB_1 & L_{i+2}P_{i+1}S_iB_1 & \cdots & L_{i+k-1}P_{i+k-2}P_{i+k-3}\cdots P_{i+1}S_iB_1 \\ 0 & 0 & L_{i+2}S_{i+1}B_1 & \cdots & L_{i+k-1}P_{i+k-2}P_{i+k-3}\cdots P_{i+2}S_{i+1}B_1 \\ \vdots & \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \vdots & & L_{i+k-1}S_{i+k-2}B_1 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}^T$$

$$D_{2_r} = \begin{bmatrix} 0 & L_{i+1}S_iB_{2_i} & L_{i+2}P_{i+1}S_iB_{2_i} & \cdots & L_{i+k-1}P_{i+k-2}P_{i+k-3}\cdots P_{i+1}S_iB_{2_i} \\ 0 & 0 & L_{i+2}S_{i+1}B_{2_{i+1}} & \cdots & L_{i+k-1}P_{i+k-2}P_{i+k-3}\cdots P_{i+2}S_{i+1}B_{2_{i+1}} \\ \vdots & \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \vdots & & L_{i+k-1}S_{i+k-2}B_{2_{i+k-2}} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}^T \tag{5.15}$$

With this modeling method there is no cutting stiffness term (κ) in the system matrices and the uncertainty is in a feedback loop that outputs λs:

$$\begin{bmatrix} \lambda_i \\ \lambda_{i+1} \\ \vdots \\ \lambda_{i+k-1} \end{bmatrix} = \kappa \begin{bmatrix} y_i - y_{i-k} \\ y_{i+1} - y_{i+k-1} \\ \vdots \\ y_{i+k-1} - y_{i-1} \end{bmatrix} = \kappa(\tilde{y}_i - \tilde{y}_{i-1}) \tag{5.16}$$

Putting the system in a standard format for robust control would lead to

$$\begin{bmatrix} X_{i+k} \\ \tilde{y}_i \end{bmatrix} = \begin{bmatrix} A_r & 0 \\ C_r & 0 \end{bmatrix} \begin{bmatrix} X_i \\ \tilde{y}_{i-k} \end{bmatrix} + \begin{bmatrix} B_{2_r} & B_{1_r} \\ D_{2_r} & D_{1_r} \end{bmatrix} \begin{bmatrix} \lambda_{lift,i} \\ F_{a_{lift,i}} \end{bmatrix}$$
$$\begin{bmatrix} \tilde{z}_i \\ \tilde{y}_i \end{bmatrix} = \begin{bmatrix} C_r & -I \\ C_r & 0 \end{bmatrix} \begin{bmatrix} X_i \\ \tilde{y}_{i-k} \end{bmatrix} + \begin{bmatrix} D_{2_r} & D_{1_r} \\ D_{2_r} & D_{1_r} \end{bmatrix} \begin{bmatrix} \lambda_{lift,i} \\ F_{a_{lift,i}} \end{bmatrix} \tag{5.17}$$

Figure 5.2 shows the plant and uncertainty in an LFT form suitable to design a robust controller.

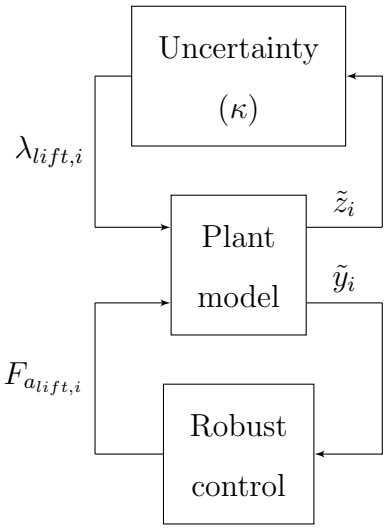


Figure 5.2: Block diagram of the generalized plant and the uncertainty for robust control design

As a novel modeling and semi-discretization technique for cutting process is being introduced in this chapter, we start with the simplest plant to represent cutting for which there is a closed form solution for stability lobes and compare the robust control oriented modeling and semi-discretization technique to it.

5.2 Simple Cutting Example

The simple plant consists of a rigid tool and a spring-mass-damper system to represent the workpiece.

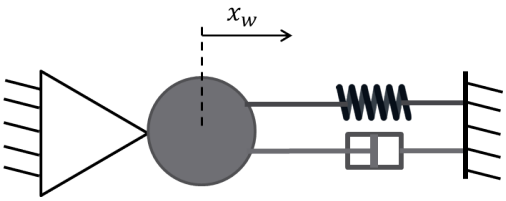


Figure 5.3: Schematic of the simple model for cutting

The cutting force can be represented by

$$F(t) = F_{cut,0} + \kappa(x_w(t) - \mu x_w(t-\tau) \tag{5.18}$$

The state of this system would be $x = \begin{bmatrix} x_w \\ \dot{x}_w \end{bmatrix}$.

In order to find the continuous time model based stability lobe diagrams, the plant whose state space representation is shown in (5.19) is put in a feedback loop with the uncertainty κ and a continuous time delay as shown in Figure 5.4.

$$\begin{bmatrix} \dot{x}_w \\ \ddot{x}_w \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m_w & -c/m_w \end{bmatrix} \begin{bmatrix} x_w \\ \dot{x}_w \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m_w \end{bmatrix} F$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_w \\ \dot{x}_w \end{bmatrix} \tag{5.19}$$

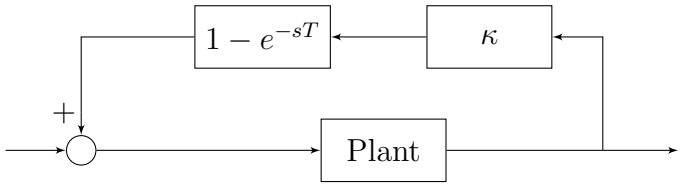


Figure 5.4: Block diagram of continuous time model of cutting

A Nyquist plot of the described closed loop continuous time model can be generated for each cutting speed with $\kappa = 1$ assumed. Different cutting speeds lead to different T values in the e^{-sT} term. From each Nyquist plot the point on the real axis that is leftmost and by increasing κ can move towards -1 and lead to an encirclement is identified. The inverse

of that real number is the maximum κ that the system can handle without going unstable (starting to chatter). Figure 5.5 shows an example Nyquist plot with the said point shown with a blue arrow.

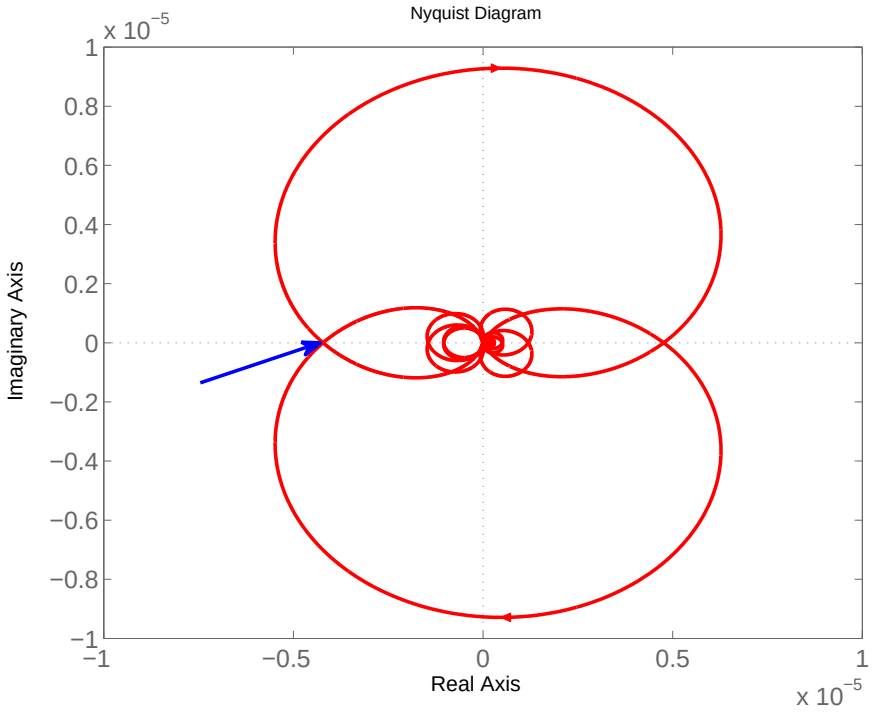


Figure 5.5: Nyquist plot of the continuous time cutting system at 400 rpm

Finding the critical κ for many points, we generate the stability lobe diagram for the continuous time model of the cutting process shown in Figure 5.6. Any discretization and approximation method can be compared to this diagram to evaluate the quality of the approximation.

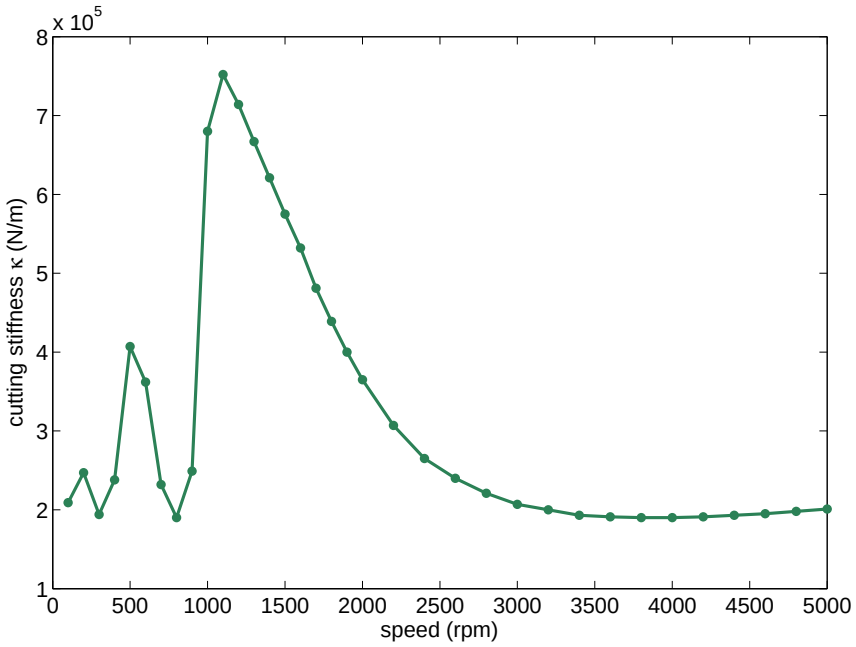


Figure 5.6: True model of the stability lobes diagram generated from the continuous time system

Next we plot the stability lobes generated from the robust control oriented method of modeling and semi-discretization described in this chapter. To create stability lobe diagrams with the robust control oriented method, a discrete time plant is created from system matrices A_r, B_{2_r}, C_r and D_{2_r} with sampling time equal to T which was the delay in the cutting process. In this simple model P and S are time invariant as there are no moving contact points and shape functions that vary with time. The plant is then put in a positive feedback loop with the delay term and the cutting stiffness as shown in Figure 5.7. The delay z^{-1} in the figure also has sampling time T.

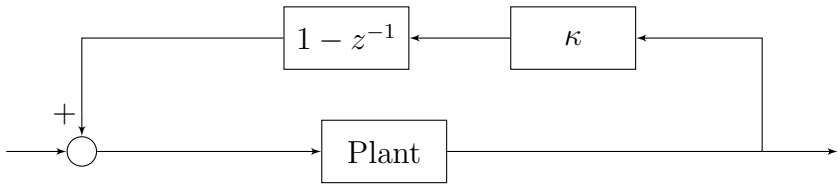


Figure 5.7: Block diagram of robust control oriented modeling and discretization method

The poles of this closed loop system can be used as indication of stability or instability. So for each machining speed an iteration is run from low κ to higher κ's, stopping where any of the poles get out of the unit circle. As mentioned T is divided into k equal Δts when semi-discretization is performed. The number k affects the precision of the model. As can be seen in Figure 5.8 the higher the number of discretization points k is the closer the modeling and semi-discretization matches the true model. The k is kept constant for different machining speeds when creating the stability lobe diagram. As a result, higher speeds render smaller Δt's making the approximation more exact in higher speeds.

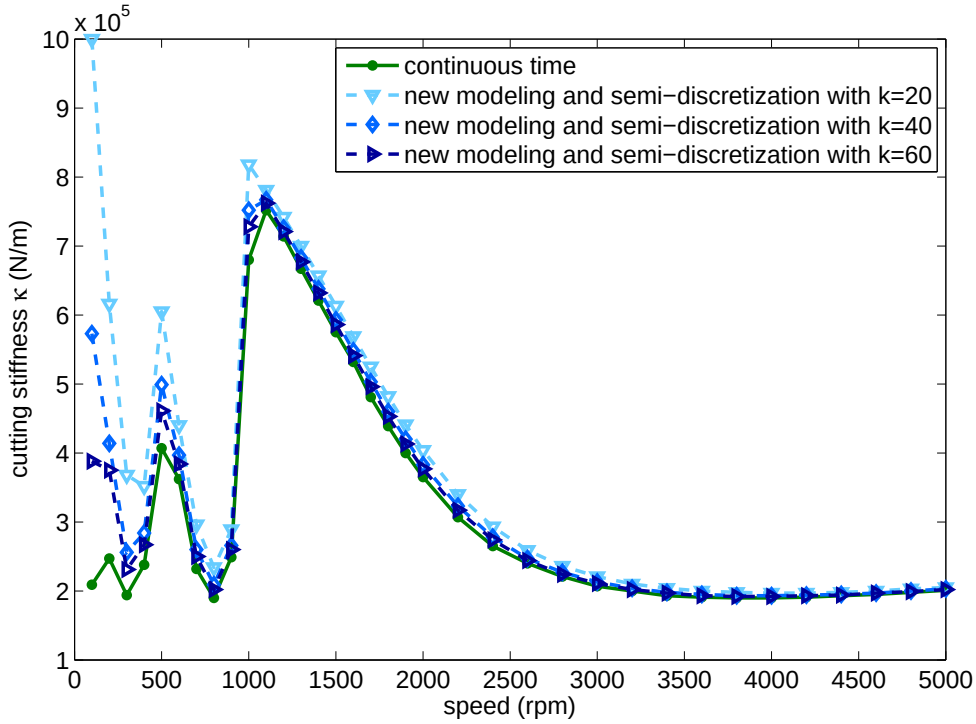


Figure 5.8: Comparison of the proposed robust control oriented modeling and semi-discretization method with true model

The test above with a simple machining model shows that for high enough number of discretization points the robust control oriented modeling and semi-discretization is a good approximation that matches the true model closely.

5.3 Robust Control Oriented Modeling and Semi-discretization Applied to Tube Turning

In this section we apply the method described in Section 5.1 to the tube turning problem. Using the same tube and robot arm detailed in Chapter 3 and robust control oriented modeling and semi-discretization with number of discretization points equlal to the elements cut in one revolution (eighteen) we generated the stability lobe diagram. Figure 5.9 shows the stability lobes diagram.

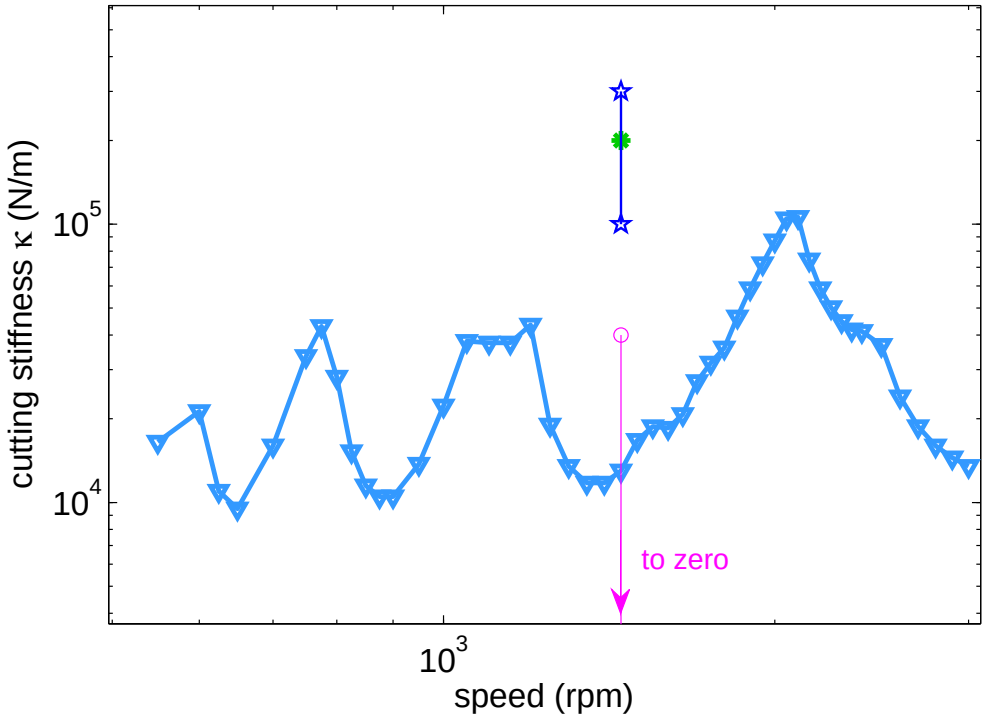


Figure 5.9: Stability lobes diagram of turning of thin tube with the proposed robust control oriented modeling and semi-discretization methods

Next we design a robust controller for the cutting system modeled through the robust control oriented method for a range of κ's around the target cutting stiffness, for which we designed LQR controllers in Chapter 3.

The robust controller is designed for speed of 1450 rpm. The control is still going to

be applied through the actuator on the robotic arm manipulator. κ is modeled as a real uncertainty with a given range. The green star on the plot of Figure 5.9 is the target for which we designed the LQR controller in Chapter 3. As mentioned with LQR we did not have power over the range around the target for which our controller would still be stabilizing and robustness was not a design consideration with that controller. The figure also shows a desired range of cutting stiffness, with dark blue pentagrams connected with a line, that we would like the robust controller to be able to stabilize for. This range could come from the knowledge of how uncertain the number κ we pick for the machining process really is, since it is a result of several machining parameters and conditions taken into account and summarized in one number which we denote by κ.

The method of choice for controller design is μ-synthesis using D-K iterations. There are many choices to pick the inputs and outputs of the system to have the robust controller minimize the structured singular value of the robust performance problem associated with that system. Not all choices would lead to stabilizing controllers, and signals to be penalized should be chosen with care and intuition for the problem at hand. In the choice picked in this work the goal of the controller would be to minimize μ or structured singular value for a robust performance problem associated with a system with input, a force applied from the cutting tool, and outputs $\tilde{y}$ and λ_{lift} and $F_{a_{lift}}$. The latter two outputs had a weighting of 0.1 so that there is more emphasis on $\tilde{y}$ when designing the robust controller. The D-K iteration procedure is a sequence of minimizations, first over the controller variable while holding the D variable associated with the scaled μ upper bound fixed, and then over the D variable while keeping the controller variable fixed. Matlab was used to perform the D-K iteration. For a range of $1 \times 10^5 \leq \kappa \leq 3 \times 10^5$, the robust controller was designed with the mentioned performance input and outputs and D-K iterations. The resulting controller was then tested on the model of the system for the upper and lower bound on κ as well as several κ's in the range and it was indeed stabilizing.

Another way to decide the range of cutting stiffness, κ for which we would want to stabilize the system using a robust controller would be to make the range include zero so

that with that controller in place, air cutting as well as cutting stiffnesses that were stable without the presence of the arm and the controller would still be stable with the controller in place. That way the machine operator would not need to worry about switching controllers or turning the controller on only after reaching a certain depth of cut or machining parameter. For this purpose, again for the spinning speed of 1450 rpm, a robust controller was designed to stabilize between the range of 0 to 4×10^4 N/m. Same robust performance inputs and outputs and weightings as before were kept. Using Matlab for the μ-synthesis using DK iterations a stabilizing controller was created and tested on the model for that range and was able to stabilize for a big range including zero at the lower bound. The range for which this controller was designed to stabilize is shown in Figure 5.9 with a magenta line.

5.4 Conclusion

In this chapter we showed how the new robust control oriented method of performing modeling and semi-discretization for the machining process can isolate the cutting stiffness which can be an uncertain parameter from the rest of the system and pave the way for a robust control synthesis.

CHAPTER 6

A New Method for Tuning Linear Quadratic Regulators and Estimators

6.1 Introduction

Linear Quadratic Regulator (LQR) problem is about finding an optimal feedback control law to minimize the quadratic performance index with respect to the control inputs and the states through weighting matrices R and Q respectively. However there appears to be no universal and straight forward relationship between the tuning matrices Q and R and the time or frequency properties of the closed loop response. Q and R are usually tuned through heuristic methods and especially in case of a high-dimensional multi input multi output (MIMO) systems tuning the Q and R to reach a specific performance characteristic can be very difficult. Tuning of LQR controllers has conventionally been done through trial and error with the two weighting matrices which do not necessarily have intuitive base especially for high-dimensional models and models attained through system ID where the states of the system do not have physical meaning.

In the past research has been conducted in the area of tuning LQR controllers and LQR has been used to tune PID controllers. Kumar, et al [KJ13] turn the PID control design into an LQR design by taking output error and its integral and its derivative as the states of a new system and penalizing them through the Q matrix of the LQR design. By assuming that the system is a second order whose closed loop polynomial has certain connection to its damping ratio and natural frequency, they solve for the elements of the Q matrix given a desired natural frequency and damping ratio for the closed loop. This method however

always needs to assume a second order model which is not always applicable or desired. Das, et al [DPH13] also used LQR theory for the design of optimal analog and digital PID controllers. They used Genetic Algorithm to find the optimal weighting matrices for the optimal state feedback regulator design while minimizing a second time domain integral performance index which consists of weighted sum of Integral of Time multiplied Squared Error and the controller effort. Hence they essentially introduce time into the cost function in order to achieve desired closed loop time response. However in their work also, only second order systems are considered. Kim, et al [KLL15] propose a method for tuning an LQ-PID controller for second order system to simultaneously meet the time and frequency domain design specifications. In their case also the method only works for second order systems. Hassani, et al [HL14] use quantum particle swarm optimization to tune an LQR. They apply the method to stabilize an inverted pendulum in simulation and compare that with some other methods such as tuning through trial and error and Genetic Algorithms. They conclude that their method gives better results in terms of rise time and settling time, however they cannot address the overshoot as good as trial and error method.

There are other researchers who focus their attention on compensating for the imprecise knowledge of the model and the process noise. Trimpe, et al [TMD14] came up with an automatic tuning algorithm that iteratively updates an LQR design in order to improve the closed loop performance. Since there is process noise the whole LQR problem is set up as a stochastic optimization. In each iteration the controller is tested on the system and the cost function is calculated from the measurements of output and controller effort. And the automatic process looks for better Q and R matrices for the LQR problem in the presence of model uncertainties and process noise. A drawback of the described tuning method is the critical dependence of its performance on the parameters for the optimizer, which is an inherent feature of this optimization technique. These are essentially tuning parameters for which there is no general guideline of choosing and one should tune them according to the problem at hand. The authors also mention that the optimizer only takes the latest cost evaluations into account for making a controller update (instead of the entire history of

cost evaluations). Ramos, et al [RSM05] use bounded data uncertainties (BDU) techniques to tune LQR controllers in order to improve system robustness in the face of modeling uncertainties. Marco, et al [MHB16] came up with an automatic LQR tuning method based on Gaussian process global optimization that improves the controller gains according to a pre-defined performance objective evaluated from experimental data.

The purpose of this chapter is to introduce a more intuitive way for tuning LQR controllers based on the desired output characteristics. Rather than relying on trial and error and guesswork on how the internal states affect the outputs, this method helps translate the output characteristics the control engineer expects of the system to tuning matrices Q and R. Furthermore this method can be combined with different internal models to satisfy other control needs for the system.

6.2 New Tuning Method for Linear Quadratic Regulators

The suggested method can be applied to continuous time as well as discrete time LQR control design. We start with a unified approach helping us to present the problem in mathematically similar representation regardless of whether we are in continuous or discrete time domain. For that purpose as well as other benefits to be discussed, in designing the discrete time version of the suggested tuning method we turned to the delta domain. Delta domain rather than z domain helps with numerical stability especially when implemented in a medium where fixed point calculations are the only type allowed. Additionally a unified approach for control design in continuous and discrete time domains can be adopted.

6.2.1 Delta Domain

Middleton and Goodwin were first to introduced the delta Operator [MG86],[MG90]. The delta Operator is defined as below

$$\delta = \frac{q-1}{\Delta} \tag{6.1}$$

Where q is the forward shift operator in time domain and will have a frequency domain representation as below:

$$\gamma = \frac{z-1}{\Delta} \tag{6.2}$$

There are other variations of δ such as:

$$\delta = \frac{1-q^{-1}}{\Delta}, \text{and} \quad \delta = \frac{2(1-q^{-1})}{\Delta(1+q^{-1})} \tag{6.3}$$

which are causal.

Taking the unified notation from Middleton and Goodwin [MG90] we use ρ the generalized derivative and some other notation as below.

(1) $y(t)$ denotes a function of time in continuous time, or a sequence in discrete time where $t \in \Omega$ and

$\Omega = \mathbb{R}$, continuous time

$\Omega = \{t : \frac{t}{\Delta} \in \mathbb{Z}\}$, discrete time

(2) ρ denotes d/dt in continuous time or δ in discrete time.

(3) $\mathbf{S}_{t_1}^{t_2} f(\tau)d\tau$ denotes the Riemann integral in continuous time:

$$\int_{t_1}^{t_2} f(\tau)d\tau \tag{6.4}$$

and lower Riemann sum in discrete time:

$$\Delta \sum_{\frac{t_1}{\Delta}}^{\frac{t_2}{\Delta}-1} f(k\Delta) \tag{6.5}$$

(4) $\mathbf{E}(A,t)$ denotes e^{At} in continuous time, and $(I + A\Delta)^{t/\Delta}$ in discrete time.

A state space representation of a system in the unified domain can be shown as follows

$$\rho X = AX + Bu \tag{6.6}$$

$$y = CX$$

6.2.2 LQR Problem and Tuning Method in Delta Domain

In a regular LQR problem the cost function is presented as Equation (6.7)

$$J_0 = \mathbf{S}_0^{\infty}(X^* Q_0 X + u^* R_0 u) \tag{6.7}$$

In the suggested method we want to penalize the output and its general derivatives which can be put into one vector as (6.8).

$$\tilde{y} = \begin{bmatrix} y \\ \rho y \\ \rho^2 y \\ \vdots \\ \rho^n y \end{bmatrix} \tag{6.8}$$

The new state space representation will be

$$\begin{aligned} \rho\tilde{X} &= \tilde{A}\tilde{X} + \tilde{B}v \\ \tilde{y} &= \tilde{C}\tilde{X} + \tilde{D}v \end{aligned} \tag{6.9}$$

where

$$\tilde{X} = \begin{bmatrix} X \\ u \\ \rho u \\ \vdots \\ \rho^{n-2}u \end{bmatrix}; \quad v = \rho^{n-1}u\,; \tag{6.10}$$

$$\tilde{A} = \begin{bmatrix} A & B & 0 & \cdots & & & 0 \\ 0 & 0 & I & 0 & \cdots & & 0 \\ 0 & 0 & 0 & I & 0 & \cdots & 0 \\ \vdots & & & & \ddots & & \\ \vdots & & & & & \ddots & \\ 0 & 0 & \cdots & 0 & 0 & & I \\ 0 & 0 & \cdots & & & \cdots & 0 \end{bmatrix}; \quad \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \\ \vdots \\ 0 \\ I \end{bmatrix} \tag{6.11}$$

Given the following equations

$$
\begin{aligned}
y &= CX \\
\rho y &= C\rho X = C(AX + Bu) = CAX + CBu \\
\rho^2 y &= C\rho^2 X = CA^2 X + CABu + CB\rho u \\
&\vdots \\
\rho^n y &= C(A^n X + A^{n-1}Bu + A^{n-2}B\rho u + \cdots + AB\rho^{n-2}u + B\rho^{n-1}u)
\end{aligned}
\tag{6.12}
$$

We will have

$$
\tilde{C} = \begin{bmatrix}
C & 0 & 0 & 0 & \cdots & 0 \\
CA & CB & 0 & 0 & \cdots & 0 \\
CA^2 & CAB & CB & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
CA^{n-2} & CA^{n-3}B & \ddots & \ddots & \ddots & 0 \\
CA^{n-1} & CA^{n-2}B & \cdots & \cdots & CAB & CB \\
CA^n & CA^{n-1}B & CA^{n-2}B & & \cdots & CAB
\end{bmatrix}
\tag{6.13}
$$

$$
\tilde{D} = \begin{bmatrix}
0 \\ 0 \\ \vdots \\ 0 \\ CB
\end{bmatrix}
\tag{6.14}
$$

The new longer state vector is to be penalized by $\tilde{Q}$ which includes gains for the control

output and its general derivatives. The new control output is v and is penalized by R_{n-1}.

$$\tilde{Q} = \begin{bmatrix} Q_0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & R & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & R_1 & 0 & 0 & \cdots & 0 \\ \vdots & 0 & 0 & R_2 & 0 & \cdots & 0 \\ & \vdots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & & \cdots & 0 & R_{n-2} \end{bmatrix} \tag{6.15}$$

Now the cost function would be

$$\tilde{J} = \mathbf{S}_0^{\infty}(\tilde{X}^* \tilde{Q} \tilde{X} + v^* R_{n-1} v + \tilde{y}^* W \tilde{y}) \tag{6.16}$$

Where

$$W = \begin{bmatrix} W_1 & 0 & & \cdots 0 & 0 \\ 0 & W_2 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ & & & \ddots & 0 \\ 0 & \cdots & & 0 & W_{n+1} \end{bmatrix} \tag{6.17}$$

is a diagonal matrix with the diagonal elements being penalty gains for output y and its derivatives.

The following change of variables helps represent the cost function (6.16) as a standard cost function similar to Equation (6.7).

$$R_m = R_{n-1} + \tilde{D}^* W \tilde{D} \tag{6.18}$$

$$k_1 = R_m^{-1} \tilde{D}^* W \tilde{C}$$

$$\hat{v} = v + k_1 \tilde{X} \tag{6.19}$$

$$Q_m = \tilde{Q} + k_1^* R_{n-1} k_1 + [\tilde{C} - \tilde{D} k_1]^* W [\tilde{C} - \tilde{D} k_1]$$

$$A_m = \tilde{A} - \tilde{B} k_1$$

$$\rho \tilde{X} = A_m \tilde{X} + \tilde{B} \hat{v}$$

$$J = \mathbf{S}_0^{\infty} (\tilde{X}^* Q_m \tilde{X} + \hat{v}^* R_m \hat{v}) \tag{6.20}$$

Now the cost function is in the form of a standard LQR problem. The LQR problem is solved through a standard method in the delta domain. Assume that K_{lqr} is the state feedback gain generated by the solution. we have

$$\hat{v} = -K_{lqr} \tilde{X} \tag{6.21}$$

And taking Equation (6.19) into consideration we have

$$v = -\tilde{K} \tilde{X} \tag{6.22}$$

$$\tilde{K} = K_{lqr} + k_1 \tag{6.23}$$

and essentially

$$\tilde{K} = [K, [K_0, K_1, \ldots, K_{n-2}]] \tag{6.24}$$

Where K is the feedback gain for the original states and $[K_0, K_1, \ldots, K_{n-2}]$ are gains for the original control output u and its first $n-2$ general derivatives. Depending on whether the implementation is in discrete time or continuous time, the solution can be implemented through delays or an added dynamics to the controller. This will be further clarified in the two example cases in Chapter 7.

Furthermore, we can add an internal model to the controller to attain specific goals. An integrator or a sinusoidal resonator are common examples of useful internal models to include

in the control. The addition of the internal models would require an output feedback. The states of the internal model can be augmented to the original states of the plant in Equation (6.6) resulting in a longer X vector and the Q_0 matrix will be augmented accordingly with gains to penalize the internal model states.

In Chapter 7 a continuous time and a discrete time application of the described tuning method is going to be shown. There is a dual problem of linear quadratic estimation for the above method which will be discussed in the next section.

6.3 Advantage of the Proposed Method Over Frequency Weighted LQ

Looking at the suggested method one might see a similarity between this and the frequency-weighted LQ problem that has been around for many years. The penalty on higher derivatives of the output in the proposed method can be interpreted as penalizing the high frequency parts of output. To briefly review, frequency weighted LQ suggests using a frequency weighted cost function. First taking the cost function to frequency domain turning

$$J = \int_{-\infty}^{\infty} \big(y^*(t)y(t) + \beta u^*(t)u(t)\big)dt \tag{6.25}$$

into

$$J = \frac{1}{2\pi} \int_{-\infty}^{\infty} \big(y^*(j\omega)y(j\omega) + \beta u^*(j\omega)u(j\omega)\big)d\omega \tag{6.26}$$

And then adding frequency weighting:

$$J = \frac{1}{2\pi} \int_{-\infty}^{\infty} \big(y^*(j\omega)\mathcal{W}_1^*(j\omega)\mathcal{W}_1(j\omega)y(j\omega) + \tag{6.27}$$

$$\beta u^*(j\omega)\mathcal{W}_2^*(j\omega)\mathcal{W}_2(j\omega)u(j\omega)\big)d\omega \tag{6.28}$$

Afterwards one should add extra states to the model of the system to represent $\mathcal{W}_1(s)$ and$\mathcal{W}_2(s)$. Augmenting the states of the original system with these extra states one can turn the problem into a regular LQR problem and solve for control gains.

The proposed method in this paper has two advantages over the conventional frequency weighted LQ. Firstly in terms of the effect on the output, the proposed method is more intuitive as we target the derivatives of the output directly and the effect is more straight forward in terms of time domain response than predicting how penalizing high frequency would affect the time response. The second advantage of the proposed method is that as long as the number of derivatives of the output signal we choose to penalize are fewer or equal to the relative degree of the plant there will be no need to extend the state vector of the system. However in the frequency weighted LQ problem we always have to extend the states of the system in order to penalize higher frequencies. So the proposed method can come up with a lower order filter to achieve the same goal.

6.4 Dual Problem for Linear Quadratic Estimators

Steady state Kalman filtering refers to optimal linear quadratic estimation and filtering for LTI systems on infinite time intervals. The optimal filtering problem is shown to be equivalent to the discrete time LQR problem.

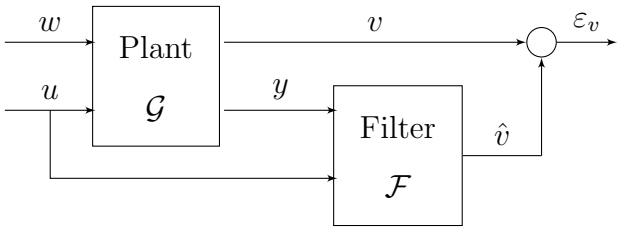


Figure 6.1: Block diagram for Kalman filtering

Figure 6.1 shows a block diagram of the Kalman filtering dual problem. The Kalman filtering problem is to choose the filter to minimize the 2-norm of the impulse response sequence of the transfer function from the disturbance signal to the estimation-error signal ε_v. Considering the following system in delta domain with constant A, B_1, C_2, D_{21} matrices

$$\rho x(t) = Ax(t) + B_1 w(t) \tag{6.29}$$

$$y(t) = C_2 x(t) + D_{21} w(t) \tag{6.30}$$

$$v(t) = \xi_v^* x(t) \tag{6.31}$$

The unknown signal w is a vector containing both plant disturbance (process noise) and sensor noise. The only measured signal is output y. The output v is to be estimated from y. Estimators in the class to be considered have the form

$$\hat{v}(t) = \mathbf{S}_{\tau=0}^{t+\Delta} a^*(t-\tau) y(\tau) = \mathbf{S}_{\tau=0}^{t+\Delta} a^*(t) y(t-\tau), \quad t \geq 0 \tag{6.32}$$

a^* is the impulse response of the estimator or the filter $\mathcal{F}$. Define the signals X, Y, and Z by

$$X(\tau) = \begin{cases} 0, & \tau = 0 \\ \mathbf{E}(A, \tau), & \tau > 0 \end{cases} \tag{6.33}$$

$$Y(\tau) = \begin{cases} D_{21}, & \tau = 0 \\ C_2 X(\tau) B_1, & \tau > 0 \end{cases} \tag{6.34}$$

$$Z(\tau) = \xi_v^* X(\tau) B_1, \quad \tau \geq 0 \tag{6.35}$$

Assuming $x(0) = 0$, the signals $y, v, \hat{v}$ and w satisfy

$$y = Y * w, \quad v = Z * w, \tag{6.36}$$

$$\hat{v} = a^* * y = (a^* * Y) * w \tag{6.37}$$

Then the estimation error signal ε_v satisfies

$$\varepsilon_v = \hat{v} - v = g^* * w \tag{6.38}$$

where

$$g^* = a^* * Y - Z \tag{6.39}$$

The sequence/signal g^* is the impulse response of the discrete-time transfer function $G^{\varepsilon_v w}$ from the disturbance signal w to the estimation-error signal ε_v. a^* is the impulse response of $\mathcal{F}$ and g^* is the impulse response of $G^{\varepsilon_v w}$.

Substituting Equations (6.33)–(6.35) into (6.39) yields

$$g = B_1^* \xi + D_{21}^* a \tag{6.40}$$

Where the sequence ξ is given by

$$\xi = -X^* \xi_v + X^* * C_2^* a \tag{6.41}$$

The performance index is

$$J = \sum_{\tau=0}^{\infty} g^*(\tau) W g(\tau) \tag{6.42}$$

where W is a non-negative self-adjoint weighting matrix. This J is the square of the W-weighted l_2 norm of the impulse response sequence g^*.

For a given fixed ξ_v and a(0), the problem of minimizing the J in (6.42) is an LQR problem with the proposed tuning method of this research, used for output with the cost function:

$$J_{Wlqr} = \mathbf{S}_0^{\infty}(x^* Q_0 x + u^* R_0 u + y^* W y) \tag{6.43}$$

with correspondences listed in Table 6.1.

Now the optimal filter impulse response sequence, interpreted as a control sequence can be determined by solving an LQR problem of the form described in proposed tuning method with only output itself being penalized.

In the next section we continue the derivation in discrete time where a Kalman predictor and a Kalmna filter can be designed.

Performance index	$J(x(0), u)$	$J_p(\xi(1), a)$
State vector	$x(t)$	$\xi(\tau)$
Control sequence	u	a
Penalized output	y	g
System matrices	(A, B, C, D)	$(A^*, C_2^*, B_1^*, D_{21}^*)$
Weighting matrix	W	W
Weighting matrix	Q_0	0
Weighting matrix	R_0	0

Table 6.1: Correspondences between the estimation problem and the LQR problem

6.4.1 Discrete-Time Kalman Filtering

It follows Equation (6.41) that

$$\xi(\tau + 1) = A^*\xi(\tau) + C_2^* a(\tau), \quad \tau = 1, 2, 3, \ldots \tag{6.44}$$

$$\xi(0) = 0, \quad \xi(1) = -\xi_v + C_2^* a(0) \tag{6.45}$$

The performance index (6.42) can be written as:

$$J = a^*(0) D_{21} W D_{21}^* a(0) + J_p(\xi(1), a) \tag{6.46}$$

Where

$$J_p(\xi(1), a) = \sum_{\tau=1}^{\infty} g^*(\tau) W g(\tau) \tag{6.47}$$

For a given fixed ξ_v and a(0), the problem of choosing the rest of the terms in the sequence a to minimizing the J in (6.46) is an LQR problem that can be solved with proposed tuning method with only output being penalized.

$$R = D_{21} W D_{21}^* \tag{6.48}$$

$$K_0 = R^{-1} D_{21} W B_1^* \tag{6.49}$$

$$Q = (B_1 - K_0^* D_{21}) W (B_1 - K_0^* D_{21})^* \tag{6.50}$$

Now using the discrete time LQR method to calculate the optimal control gain, solving the Riccati equations we have

$$P = Q + [A - K_0^* C_2][P - PC_2^*(R + C_2 PC_2^*)^{-1} C_2 P][A - K_0^* C_2]^* \tag{6.51}$$

$$K_1 = (R + C_2 PC_2^*)^{-1} C_2 P[A - K_0^* C_2]^* \tag{6.52}$$

$$F = (K_0 + K_1)^* \tag{6.53}$$

The control law becomes:

$$a(\tau) = -(K_0 + K_1)\xi(\tau) = -F^*\xi(\tau), \quad \tau = 1, 2, 3, \ldots \tag{6.54}$$

Combining (6.44), (6.45) and (6.54) yields

$$a(\tau) = F^*\big((A - FC_2)^*\big)^{\tau-1}\big(\xi_v - C_2^* a(0)\big), \quad \tau = 1, 2, \ldots \tag{6.55}$$

For a one step Kalman predictor substituting $a(\tau)$ from (6.55) into (6.32) with $a(0) = 0$ in both equations leads to:

$$\hat{v}(t+1|t) = \hat{v}(t+1) = \xi_v^* \sum_{\tau=0}^{t} (A - FC_2)^{(t-\tau)} F y(\tau), \quad t = 0, 1, 2, \ldots \tag{6.56}$$

It is assumed that the system (A, B_1, C_2, D_{21}) is stabilizable and detectable and $D_{21} W D_{21}^* > 0$, in other words matrix R is positive dfefinite. Since only the term ξ_v^* in (6.56) depends on the particular scalar output v being predicted, it can easily be modified to predict an arbitrary output vector v given by

$$v(t) = C_1 x(t), \quad t = 0, 1, 2, \ldots \tag{6.57}$$

And

$$\hat{v}(t+1|t) = C_1 \sum_{\tau=0}^{t} (A - FC_2)^{(t-\tau)} F y(\tau) \tag{6.58}$$

$$= C_1 \hat{x}(t+1|t), \quad t = 0, 1, 2, \ldots \tag{6.59}$$

In state space form

$$\hat{x}(t+1|t) = (A - FC_2)\hat{x}(t|t-1) + Fy(t), \quad t = 0, 1, 2, \ldots \tag{6.60}$$

If there is a known input u to the system, which usually is the control sequence, the system state space equations will be

$$\begin{aligned} x(t+1) &= Ax(t) + B_1 w(t) + B_2 u(t) \\ y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t) \end{aligned} \tag{6.61}$$

Kalman predictor for the states of the system will be

$$\hat{x}(t+1|t) = [A - FC_2]\hat{x}(t|t-1) + Fy(t) + (B_2 - FD_{22})u(t) \tag{6.62}$$

The prediction error sequences for state vector x and output vector v are defined by

$$\begin{aligned} \varepsilon_x(t+1|t) &= \hat{x}(t+1|t) - x(t+1), \\ \varepsilon_v(t+1|t) &= \hat{v}(t+1|t) - v(t+1) \end{aligned} \tag{6.63}$$

The discrete time state space realization of the error transfer function $G^{\varepsilon_v w}$ is

$$\varepsilon_x(t+1|t) = (A - FC_2)\varepsilon_x(t|t-1) + (FD_{21} - B_1)w(t) \tag{6.64}$$

$$\varepsilon_v(t+1|t) = C_1 \varepsilon_x(t+1|t) \tag{6.65}$$

The simulation can be based on (6.64) and (6.65) for the block diagram of Figure 6.1 or could use (6.60) for the filter model.

Thus far we have come up with the Kalman predictor represented in (6.60). Next the Kalman filter is to be derived. The minimum value for $J_p(\xi(1), a)$ is

$$J_{p,min} = \xi^*(1) P \xi(1) = (\xi_v - C_2^* a(0))^* P (\xi_v - C_2^* a(0)) \tag{6.66}$$

The problem of choosing the sequence a to minimize J, as written in (6.46, 6.46), now reduces to the problem of choosing $a(0)$ to minimize

$$\tilde{J}(a(0)) = a^*(0) R a(0) + (\xi_v - C_2^* a(0))^* P (\xi_v - C_2^* a(0)) \tag{6.67}$$

The minimizing $a(0)$ is

$$a(0) = M^* \xi_v \tag{6.68}$$

where

$$M = PC_2^*(R + C_2PC_2^*)^{-1} \tag{6.69}$$

Consequently

$$a(\tau) = F^*\big((A - FC_2)^*\big)^{\tau-1}(I - MC_2)^*\xi_v \tag{6.70}$$

The class of filters being considered is defined as

$$\hat{v} = \sum_{\tau=0}^{t} a^*(t-\tau)y(\tau), \quad t = 0, 1, 2, \ldots \tag{6.71}$$

Kalman filter is to estimate $\hat{v}(t|t)$. Using (6.68, 6.70) in (6.71) yields

$$\hat{v}(0|0) = \xi_v^* My(0) \tag{6.72}$$

and

$$\hat{v}(t|t) = \xi_v^*[My(t) + (I - MC_2)\sum_{\tau=0}^{t-1}[A - FC_2]^{(t-\tau-1)}Fy(\tau)], \quad t = 1, 2, \ldots \tag{6.73}$$

Using (6.58, 6.59) one can write (6.73) as

$$\hat{v}(t|t) = \xi_v^*[My(t) + (I - MC_2)\hat{x}(t|t-1)], \quad t = 1, 2, \ldots \tag{6.74}$$

ξ_v can be replaced by a desired output vector C_1 and C_1 can be the identity matrix to estimate states of the system. If there is a known input sequence u we will have

$$\hat{x}(t|t) = \hat{x}(t|t-1) + M\big(y(t) - C_2\hat{x}(t|t-1) - D_{22}u(t)\big), \quad t = 0, 1, 2, \ldots \tag{6.75}$$

The Kalman filter consists of the one-step Kalman predictor in (6.62) and the update or correction Equation (6.75). Thus for the plant with state space representation of (6.61), the discrete time Kalman filter has the state space realization

$$\hat{x}(t+1|t) = [A - FC_2]\hat{x}(t|t-1) + \begin{bmatrix} F & (B_2 - FD_{22}) \end{bmatrix} \begin{bmatrix} y(t) \\ u(t) \end{bmatrix}, \tag{6.76}$$

$$\hat{x}(t|t) = [I - MC_2]\hat{x}(t|t-1) + \begin{bmatrix} M & -MD_{22} \end{bmatrix} \begin{bmatrix} y(t) \\ u(t) \end{bmatrix} \tag{6.77}$$

Next we use the new tuning method on the estimator. A derivative of g which is the impulse response of disturbance to estimation error transfer function (G^{ε_v}) is going to be included in the cost function and the matrices are to be updated accordingly. Here a causal variation of δ is used for the general derivative. Consequently $\tilde{g} = \begin{bmatrix} g \\ \delta g \end{bmatrix}$ is to replace g in Equation (6.42) where

$$g(\tau) = B_1^* \xi(\tau) + D_{21}^* a(\tau)$$
$$\delta g(\tau) = \frac{g(\tau) - g(\tau - 1)}{\Delta t} \tag{6.78}$$

Taking $\alpha(\tau) = g(\tau - 1)/\Delta t$ as a new additional state, we will have $\tilde{\xi} = \begin{bmatrix} \xi \\ \alpha \end{bmatrix}$ as the extended state of the system.

$$\begin{bmatrix} \xi \\ \alpha \end{bmatrix}(\tau + 1) = \begin{bmatrix} A^* & 0 \\ B_1^* & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \alpha \end{bmatrix}(\tau) + \begin{bmatrix} C_2^* \\ D_{21}^* \end{bmatrix} a(\tau) \tag{6.79}$$

$$\begin{bmatrix} g \\ \delta g \end{bmatrix}(\tau) = \begin{bmatrix} B_1^* & 0 \\ B_1^*/\Delta t & -I \end{bmatrix} \begin{bmatrix} \xi \\ \alpha \end{bmatrix}(\tau) + \begin{bmatrix} D_{21}^* \\ D_{21}^*/\Delta t \end{bmatrix} a(\tau) \tag{6.80}$$

Thus

$$\tilde{\xi}(\tau + 1) = \tilde{A}^* \tilde{\xi}(\tau) + \tilde{C}_2^* a(\tau) \tag{6.81}$$

$$\tilde{g}(\tau) = \tilde{B}_1^* \tilde{\xi}(\tau) + \tilde{D}_{21}^* a(\tau) \tag{6.82}$$

$\tilde{A}, \tilde{B}_1, \tilde{C}_2, \tilde{D}_{21}$ are to replace A, B_1, C_2, D_{21} in the derivation while $\tilde{\xi}$ and $\tilde{g}$ are to replace ξ and g. The new estimator or Kalman filter will have an extra state (or several extra states in case of MIMO systems) compared to the one derived previously. This extra state is the state of the equivalent weighting filter inside the Kalman filter. With this addition, the W matrix will get larger to include the penalization gains for the derivatives of the process and sensor noise. By tuning these gains one can directly affect the shape of the impulse response from disturbance to estimation error making the tuning more intuitive.

When tuning a Kalman filter with this method, we tune the shape of the impulse response of the estimation error from process noise and from sensor noises. We may have a good idea

of what sensor noise may be present but we may not know how process noise affects the states. One reason why we may be interested in the shape of the impulse response from the process noise to estimation error, is that the estimator does not know the states in the first step and having an arbitrary state vector to start the estimation off with is like having an impulse input to the process noise. Examples of using this tuning method will be shown in the next chapter.

CHAPTER 7

Results of Performing New Tuning Method for LQR and Kalman Filter on Example Systems

In this chapter we apply the new tuning method to a few example systems to show its use and effectiveness.

7.1 The LQR Examples

In the next two subsection we demonstrate the new tuning method for the linear quadratic regulator on a continuous time model and a discrete time model.

7.1.1 Continuous Time LQR Example

The example system picked for this presentation is the 2 mass connected by a spring classical system as shown in Figure 7.1. It is assumed that there is no friction.

Figure 7.1: Schematic of the benchmark system

Force is applied to the first mass and we look at the displacement of the second mass as

output. In state space this system is represented as follows.

$$\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-K}{m_1} & 0 & \frac{K}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{K}{m_2} & 0 & \frac{-K}{m_2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u \tag{7.1}$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix}$$

New output, $\tilde{y}$ will be y and its derivatives. It is essential for this method to include at least as many derivatives of the output as to result in non-zero $\tilde{D}$ in new system model as represented by Equations (6.9,6.10) or alternatively, use a lower derivative of the input u as the new control input. Also depending on the model some of the factors in the Equations (6.12) can turn out to be zero so one may choose higher derivatives of u as the new input. For instance in the current example,

$$CB = CAB = CA^2B = 0 \tag{7.2}$$

Deciding to pick $v = \ddot{u}$ and using Equations (6.12) we will make $\tilde{y}$ include y and its first 6 derivatives which will generate a factor of CA^3B for $\ddot{u}$ leading to nonzero $\tilde{D}$. Here we also augment the system with an output integrator and introduce the gain q_i to penalize the integrator state (x_i). Let's assume the gains for penalizing the input u, its first derivative $\dot{u}$ and its second derivative $\ddot{u}$ are R, R_1 and R_2 respectively.

$$X = \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix}, \quad \tilde{X} = \begin{bmatrix} X \\ u \\ \dot{u} \\ x_i \end{bmatrix}, \quad \tilde{y} = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \\ \frac{d^6 y}{dt^6} \end{bmatrix} \tag{7.3}$$

And the matrices for Equations (6.9 and 6.10) will be

$$\tilde{A} = \begin{bmatrix} A & B & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ -C & 0 & 0 & 0 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix} \tag{7.4}$$

$$\tilde{C} = \begin{bmatrix} C & 0 & 0 & 0 \\ CA & CB & 0 & 0 \\ CA^2 & CAB & CB & 0 \\ CA^3 & CA^2B & CAB & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^6 & CA^5B & CA^4B & 0 \end{bmatrix}, \quad \tilde{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ CB \\ CAB \\ CA^2B \\ CA^3B \end{bmatrix}$$

Keeping in mind that some terms involving higher derivatives of u are omitted due to their factors being zero as mentioned in Equation (7.2).

$$\tilde{Q} = \begin{bmatrix} Q & 0 & 0 & 0 \\ 0 & R & 0 & 0 \\ 0 & 0 & R_1 & 0 \\ 0 & 0 & 0 & q_i \end{bmatrix} \tag{7.5}$$

And given $v = \ddot{u}$, the cost function will look like Equation (7.6).

$$J = \int_0^\infty (\tilde{X}^* \tilde{Q} X + v^* R_2 v + \tilde{y}^* W \tilde{y}) dt \tag{7.6}$$

Now applying the steps in Equations (6.18 to 6.20) we will have a standard LQR problem that can be solved through Riccati equations. And through applying Equation (6.23) we get the state feedback gain $\tilde{K}$ for the $\tilde{X}$ states

$$v = -\tilde{K}\tilde{X} \tag{7.7}$$

Where

$$\tilde{K} = \begin{bmatrix} K_x, & [K_0, & K_1], & K_i \end{bmatrix} \tag{7.8}$$

Going from v to u would add some dynamics to the controller as can be seen below. Figure 7.2 shows the block diagram of the closed loop controlled system where the block "dyn" is defined in (7.9).

$$dyn(s) = \frac{1}{s^2 + K_1 s + K_0} \tag{7.9}$$

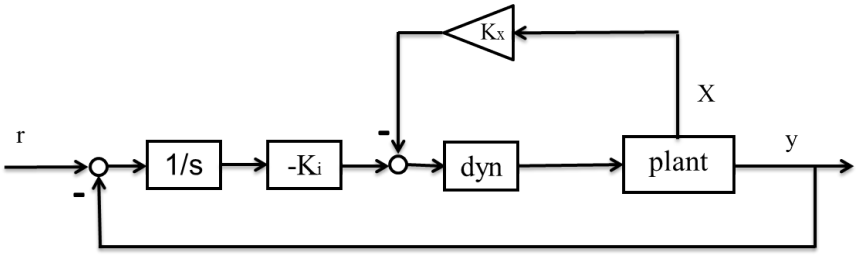


Figure 7.2: Block diagram of the closed loop system

The claim here is that there is intuitive connection between the entries of the diagonal W matrix and the time response of the closed loop system specifically its overshoot, rise time and settling time. Penalizing output itself results in lower overshoot, penalizing the first derivative of the output increases the rise time, penalizing the third and higher derivatives of the output make the response smoother with less ripples which can be translated into faster settling time albeit increasing them too much will slow the response causing higher rise time and consequently settling time. There are two ways to incorporate the proposed method, one can use it to further fine tune the response of an already existing LQR controller, or it can be used to design a controller from scratch. To show the effects described we use the proposed tuning method with the W matrix on top of an already tuned LQI for the two mass system of Figure 7.1 with state space representation 7.1. A very common method of tuning LQR controllers is to use $Q = \beta I$ and a constant R.

Figure 7.3 shows how using the proposed tuning method on top of the existing LQI can reduce overshoot. For that purpose we penalized first entry of the $\tilde{y}$ vector meaning the output itself. Figure 7.4 shows how penalizing the first derivative of output through W_2 can

result in a slower response or higher rise time. Figure 7.5 shows how penalizing the second derivative of the output through W_3 makes the response smoother. Penalizing the third and higher derivatives of the output all have the smoothing effect as well but the higher the derivative the more sensitive the response is to the corresponding penalizing gain W_i. For instance the same effect shown in Figure 7.5 through penalizing gain W_3 can be achieved from penalizing the 4th derivative but with a much lower W_5 as shown in Figure 7.6. Figure 7.7 shows how penalizing high derivatives smoothes an otherwise oscillatory response.

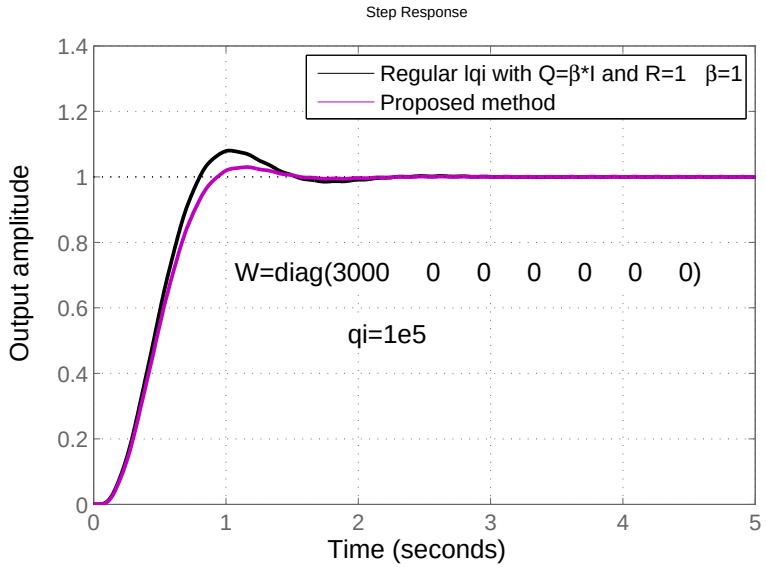


Figure 7.3: Effect of W_1 on the output

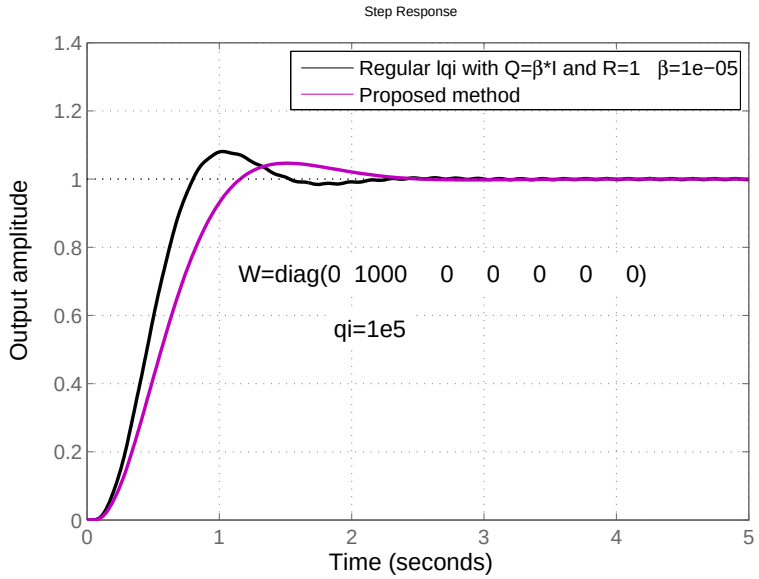


Figure 7.4: Effect of W_2 on the output

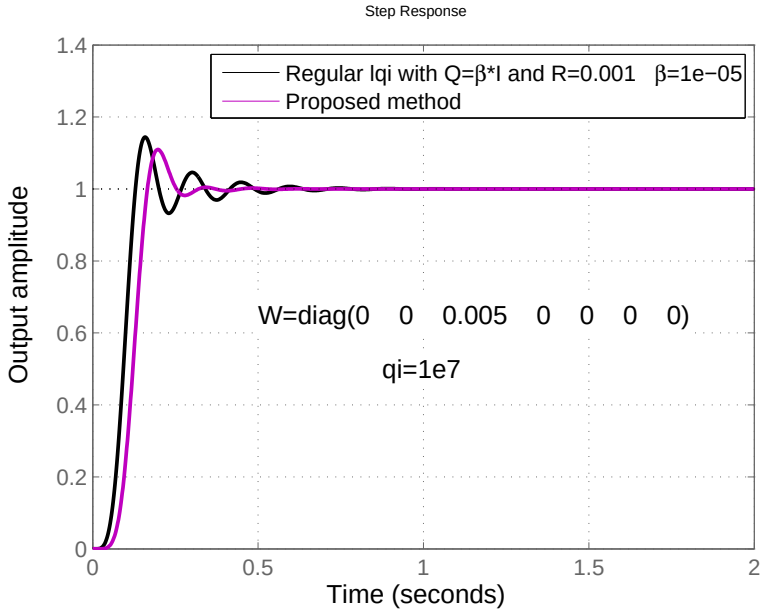


Figure 7.5: Effect of W_3 on the output

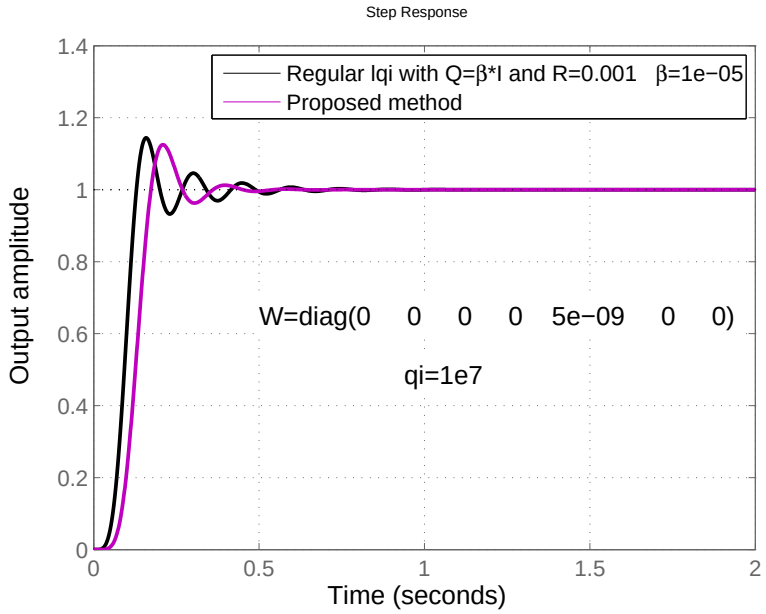


Figure 7.6: Effect of W_5 on the output

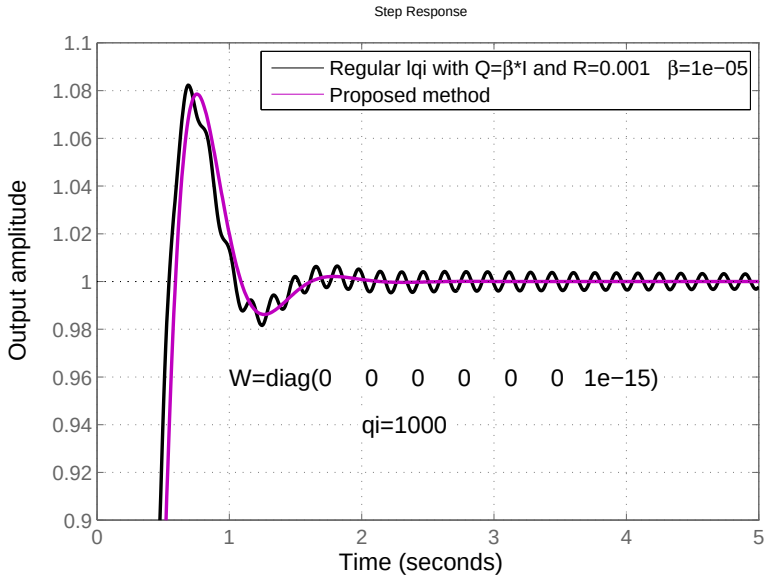


Figure 7.7: Effect of W_7 on the output

Another conventional method for tuning LQR controllers is to pick $Q = \beta C^*C$. In Figure 7.8 that method was tested for control of the benchmark plant. For the decided q_i and R

with $Q = C^*C$ the step response is going to be quite oscillatory as shown in the figure. To reduce the oscillations with this conventional method of tuning people increase the gain β. That would lead to a slower response as shown in Figure 7.8 with dash line. With the proposed method however we directly penalize higher derivatives of the output in the cost function and can smooth out the response and prevent the oscillations without slowing the response down as shown in the figure with the purple line.

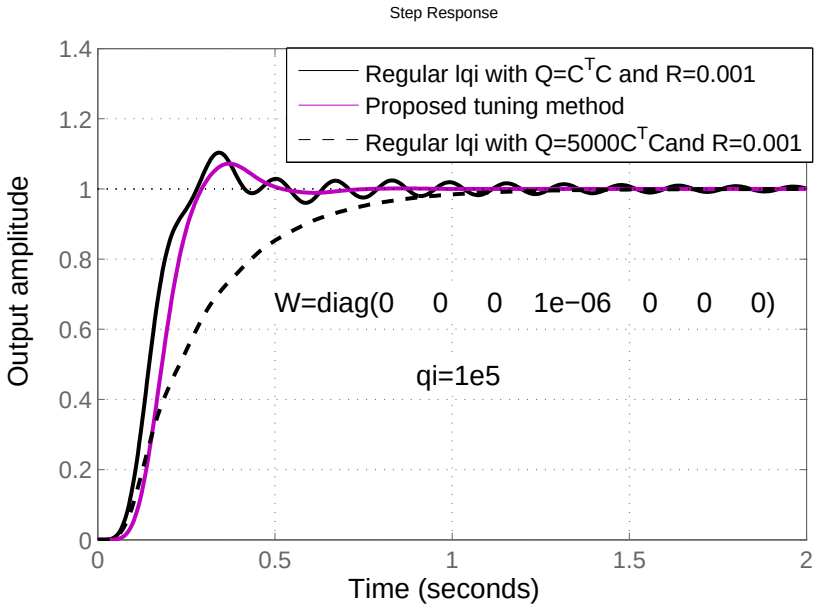


Figure 7.8: The effectiveness of the proposed method compared to method using $Q = C^TC$

7.1.2 Discrete Time LQR Example

In practice to implement a control system it is usually done through digital systems and that means the control system is more practical when designed in discrete time. The example system used for the discrete time version is the magnetic bearing system MBC 500 Turbo developed by LaunchPoint Technologies [PMS96]. A 6th order model for motion in one plane with two inputs and two outputs achieved through system identification is used for the simulation and design of the discrete time controllers. The plant is MIMO and coupled. The controllers are then implemented on the actual physical system and results are presented to

corroborate the relationship we claimed exists between the entries of W matrix and the time response of the closed loop system. Here also we use an integrator to make the results more comparable by eliminating the steady state bias. Results will be shown first in simulation and then from experiment on the magnetically levitated shaft.

7.1.2.1 System Model and Controller Design

A typical discrete time system in state space is represented as below:

$$X_s(k+1) = A_d X_s(k) + B_d u(k) \tag{7.10}$$

$$y(k) = C_d X_s(k) \tag{7.11}$$

The system matrices for the Magnetic bearing's system identified discrete time model are shown in (7.12)

$$A_d = \begin{bmatrix} 0.8594 & 0.09443 & -1.406 & -0.07994 & -1.921 & 0.6 \\ -0.05813 & 0.9384 & 0.03512 & -1.493 & 0.7622 & 2.113 \\ -0.02181 & 0.01143 & 0.9344 & 0.05907 & 2.413 & -0.8436 \\ 0.004357 & -0.007153 & -0.05078 & 0.9571 & -1.061 & -2.649 \\ 0.003451 & -0.003319 & 0.02762 & -0.02151 & 0.6748 & 0.1246 \\ -0.002228 & 0.001817 & -0.02862 & -0.001072 & 0.009399 & 0.6878 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0.1723 & 0.1725 \\ -0.1161 & 0.125 \\ -0.05007 & -0.02145 \\ -0.00438 & -0.006968 \\ -0.005223 & -0.004001 \\ 0.001426 & 0.00692 \end{bmatrix} \tag{7.12}$$

$$C_d = \begin{bmatrix} 0.2348 & -0.2328 & 0.3563 & -0.3668 & -0.4774 & -0.2074 \\ 0.2355 & 0.2396 & 0.3673 & 0.3573 & -0.2221 & 0.4909 \end{bmatrix}, \quad D_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

We include an output feedback integrator in the design to eliminate steady state error. And the augmented system with state space matrices A_a, B_a, C_a, D_a is presented below.

$$\begin{bmatrix} X_s \\ X_i \end{bmatrix}(k+1) = \begin{bmatrix} A_d & 0 \\ -C_d & 1 \end{bmatrix} \begin{bmatrix} X_s \\ X_i \end{bmatrix}(k) + \begin{bmatrix} B_d \\ 0 \end{bmatrix} u(k) \tag{7.13}$$

$$y(t) = \begin{bmatrix} C_d & 0 \end{bmatrix} \begin{bmatrix} X_s \\ X_i \end{bmatrix}(k) \tag{7.14}$$

Where X_i represents the integrator states.

At this point we can take the system from z domain to delta domain where the numerical properties are more stable and we will be able to use the equations described in Section 6.2. In order to take a state space system from z domain to delta domain we use the following transformations [MG90]:

$$A = \frac{A_a - I}{\Delta}, \quad B = \frac{B_a}{\Delta}, \quad C = C_a \tag{7.15}$$

Now this system in delta domain can be represented via Equations(6.6). In this example we chose to go as far as two derivatives of y

$$\tilde{y} = \begin{bmatrix} y \\ \rho y \\ \rho^2 y \end{bmatrix} \tag{7.16}$$

and taking $v = \rho u$.

This is a 2-input 2-output MIMO plant. For this demonstration we took the same W_i gain for both outputs making the W matrix a 6 by 6 diagonal matrix with 3 pairs of W_is to penalize outputs and their first 2 derivatives. As a reminder in delta domain derivative is the delta operator defined as Equation (6.1).

Following the steps in Section 6.2 we extend the output vector adding the derivatives, go through the change of variables and arrive at a regular LQR problem with a standard cost function. The LQR problem can then be solved through algebraic Riccati equations in delta domain. The reader is referred to [MG90] for details. The resulting system vectors and feedback gain would look like the following.

$$\tilde{X} = \begin{bmatrix} X \\ X_i \\ u \end{bmatrix} \tag{7.17}$$

$$\tilde{K} = \begin{bmatrix} K_x & K_i & K_u \end{bmatrix} \tag{7.18}$$

$$v = \rho u = \tilde{K}\tilde{X} \tag{7.19}$$

In order to implement this system, we take notice that in delta domain, $\rho u = \delta u = \frac{u(t+1)-u(t)}{\Delta}$ and this can be implemented through a simple delay. The block diagram of the controlled system looks like Figure 7.9.

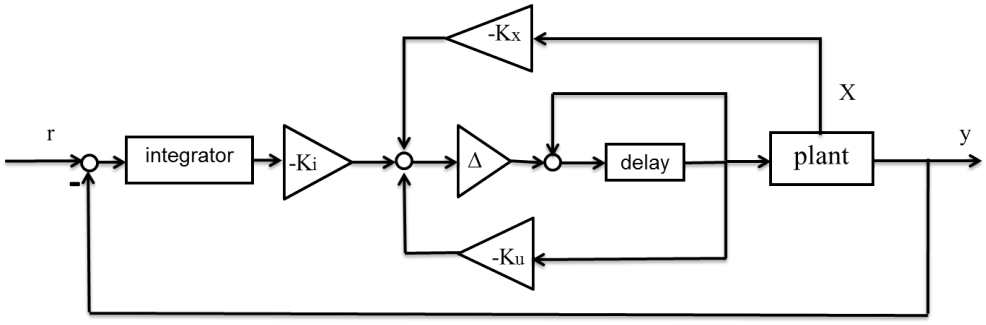


Figure 7.9: Block diagram for implementation in discrete time

7.1.2.2 Comparison with Conventional LQR Tuning

Goal of this section is comparison, and showing the more degrees of freedom available to the control engineer when tuning with the suggested method in this work. For comparison we picked the most popular tuning method used for LQRs which is taking an identity matrix

for R and a scalar β times an identity matrix as Q and using β as the tuning knob. Figure 7.10 shows the closed loop step response evolution as β changes in 2D and 3D. Figure 7.11 shows the closed loop step response evolution as W_1 changes in 2D and 3D. Figure 7.12 shows the closed loop step response evolution as W_3 changes in 2D and 3D. All simulation is for one of the outputs of the magnetically levitated shaft. As can be seen in the figures, for instance with the old method one cannot choose to change overshoot or settling time without affecting the other. However the Ws can affect these two qualities independent of the other.

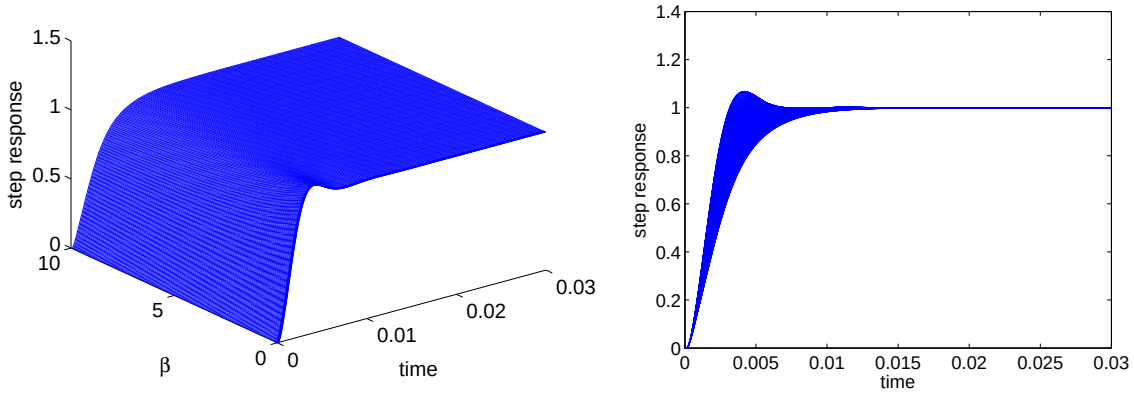


Figure 7.10: Regular LQR tuning with R=I and Q=βI

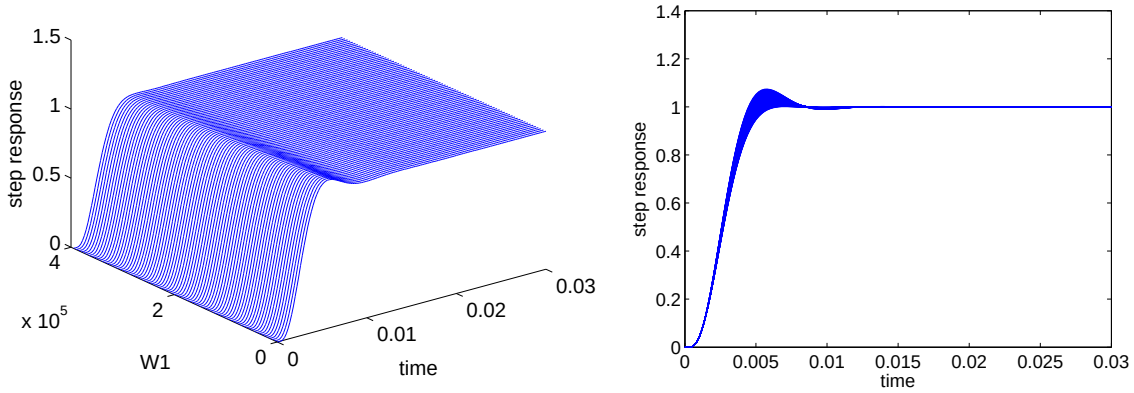


Figure 7.11: LQR tuning with W_1

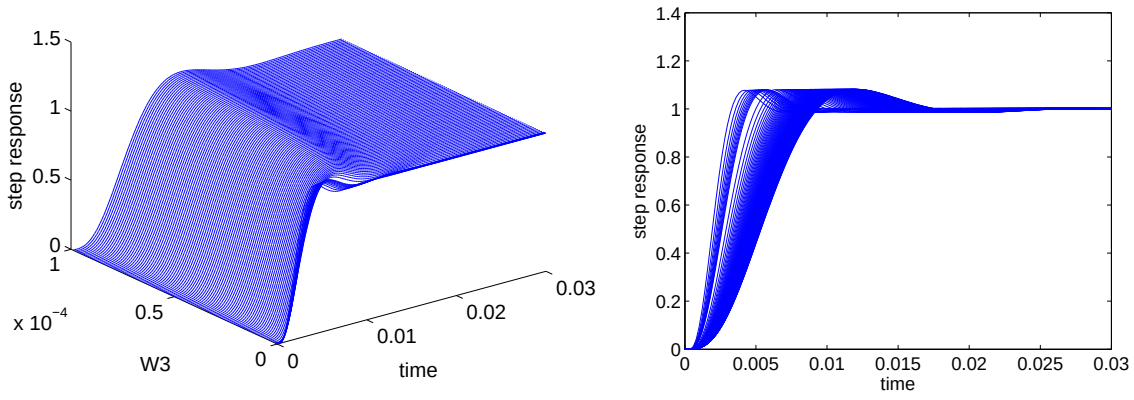


Figure 7.12: LQR tuning with W_3

7.1.2.3 Implementation on the Magnetic Bearing

In the actual implementation of the control, we will not have access to all the states so an observer was used to generate the states of the system. Figures 7.13 and 7.14 show the result of implementing the control designed through this method on the physical magnetic bearing system. For comparison two different values for tuning knob W_1 was chosen and the two controllers were implemented and the step responses plotted for the first of the two outputs. As Figure 7.13 shows increasing the penalizing gain for y reduces the overshoot. Increasing the penalizing gain for higher derivatives slows down the response and makes it smoother as can be seen in Figure 7.14.

The method introduced in Chapter 6 can be used as a more intuitive tuning method for LQR in SISO and MIMO systems.

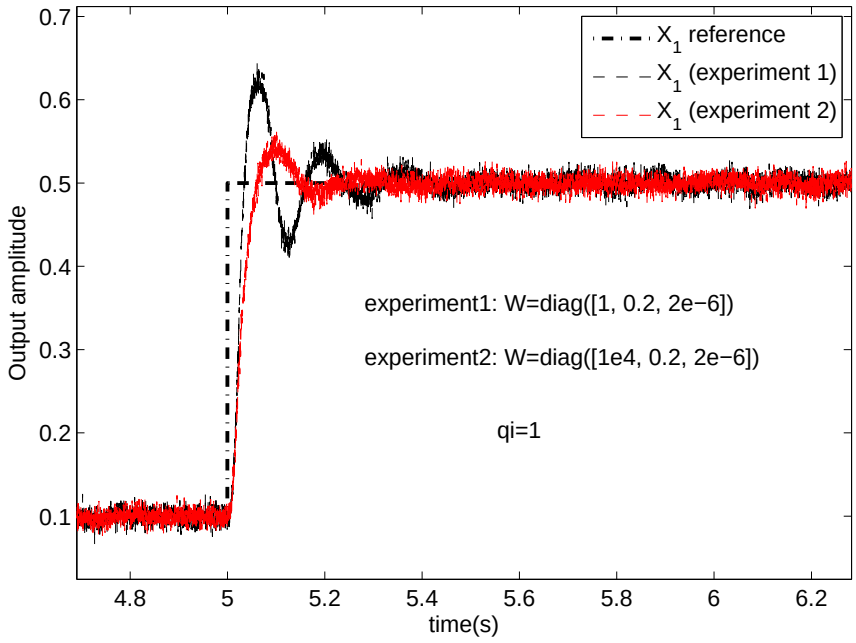


Figure 7.13: Step response of the magnetic bearing with two different tuned controllers (Effect of W_1)

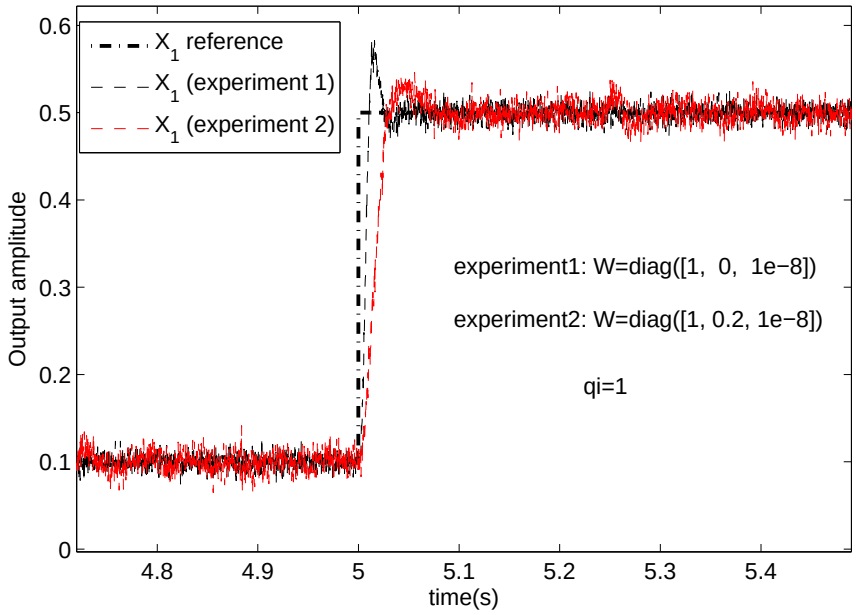


Figure 7.14: Step response of the magnetic bearing with two different tuned controllers (Effect of W_2)

7.2 The Linear Quadratic Estimator Example

We will test the tuning of a Kalman filter on the same magnetic bearing model as used for the discrete time LQR example in the previous section. The model for magnetic bearing has two inputs and two outputs as explained before. Two independent sensor noise sequences are added to the model to act on each sensor. That leads to a 6-by-6 W matrix as below where indices p, s, d stand for process noise, sensor noise and derivative respectively:

$$W = \begin{bmatrix} W_p & 0 & 0 & 0 & 0 & 0 \\ 0 & W_{s1} & 0 & 0 & 0 & 0 \\ 0 & 0 & W_{s2} & 0 & 0 & 0 \\ 0 & 0 & 0 & W_{dp} & 0 & 0 \\ 0 & 0 & 0 & 0 & W_{ds1} & 0 \\ 0 & 0 & 0 & 0 & 0 & W_{ds2} \end{bmatrix} \tag{7.20}$$

In the figures coming up the effect of each of the weighting gains for process noise and sensor noises on the impulse response of the estimation error is presented.

W_p is to penalize the estimation error caused by the process noise. To see the effect of tuning with W_p, an impulse input to the process noise was given to the system and the estimation error for one of the states (here state 6 of the magnetic bearing model) was plotted. Figure 7.15 shows that increasing W_p causes the error to have less overshoot and die out faster. Figure 7.16 shows that penalizing the derivative of the estimation error caused by the process noise, makes the response slower with a longer settling time. Also one can observe that the response is smoother with a more gradual slope because the derivative is being penalized.

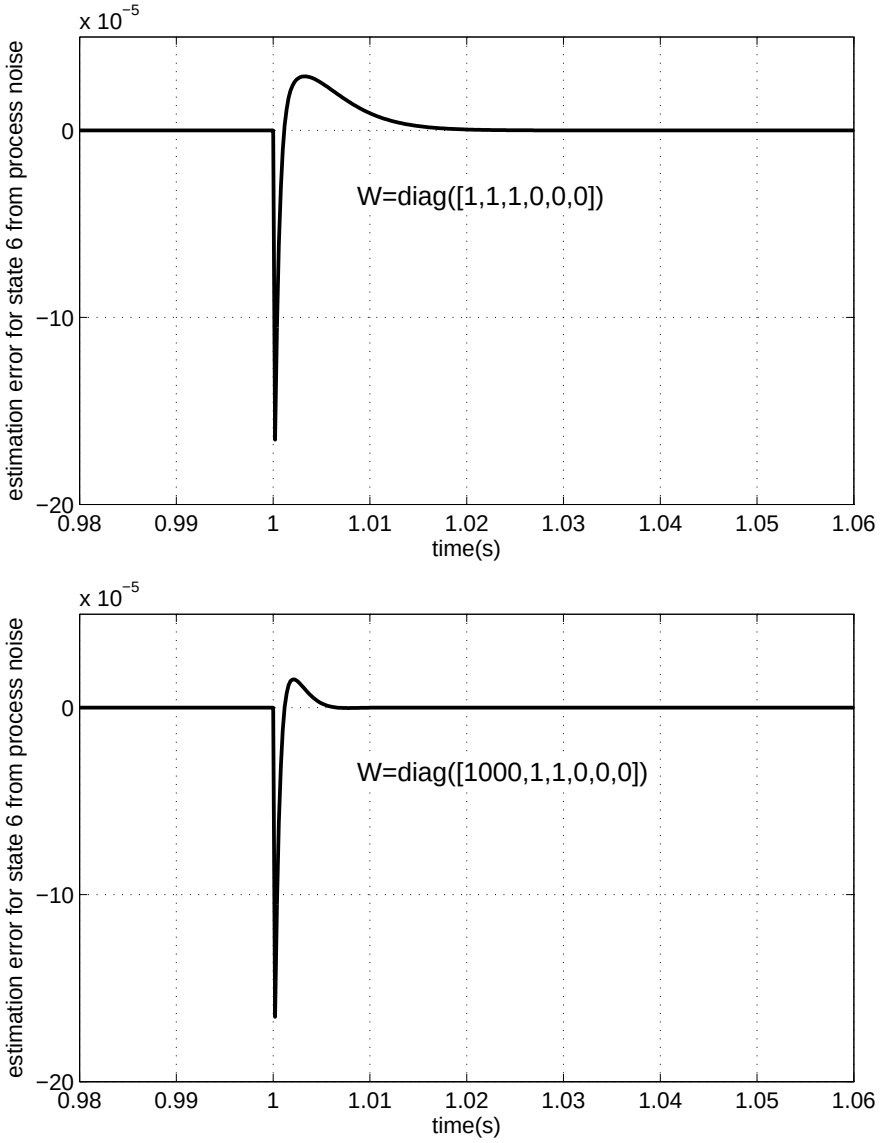


Figure 7.15: Kalman filter tuning with W_p

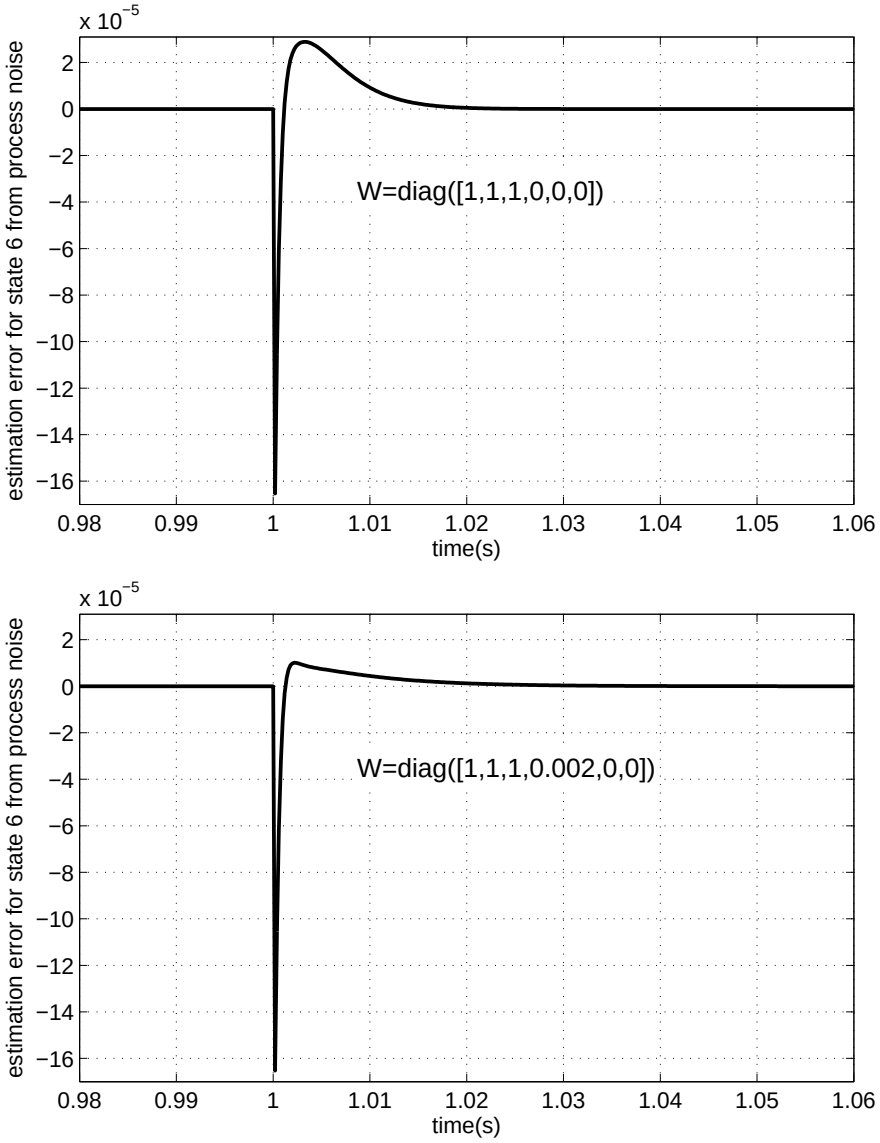


Figure 7.16: Kalman filter tuning with W_{dp}

To see the effect of penalizing the estimation error from the sensor noises and their derivative, the input is going to be two impulses at different times (0.97 sec and 1 sec) to sensor 1 and sensor 2 respectively. The effect of the weightings on estimation error from sensor noises in the W matrix is shown in Figure 7.17. As can be seen increasing the penalty W_{s1} and W_{s2} makes the impulse response of the estimation error have less of an overshoot and undershoot and settle faster. Figure 7.18 shows that increasing the penalty on the derivative

of the estimation error from sensor noises makes the response slower and smoother.

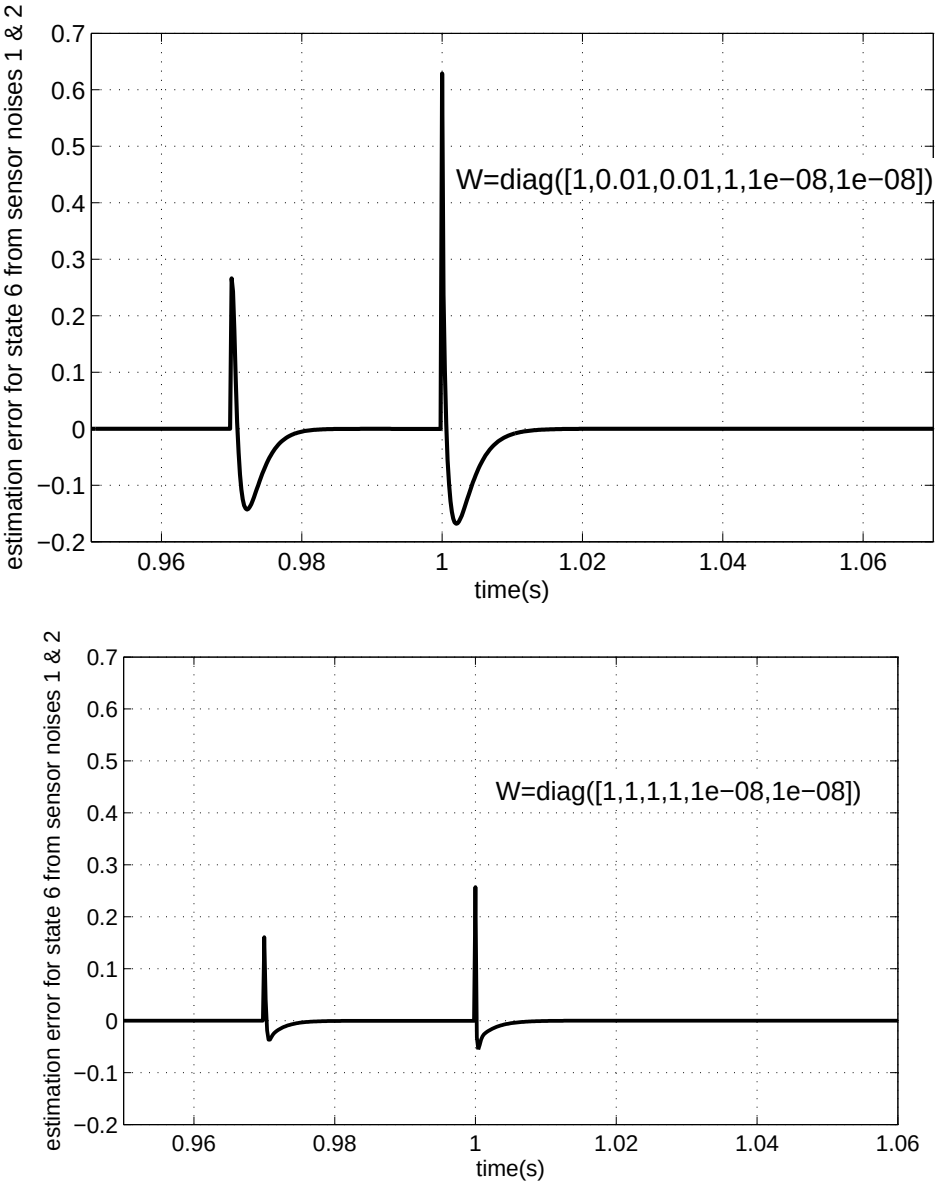


Figure 7.17: Kalman filter tuning with W_{s1} and W_{s2}

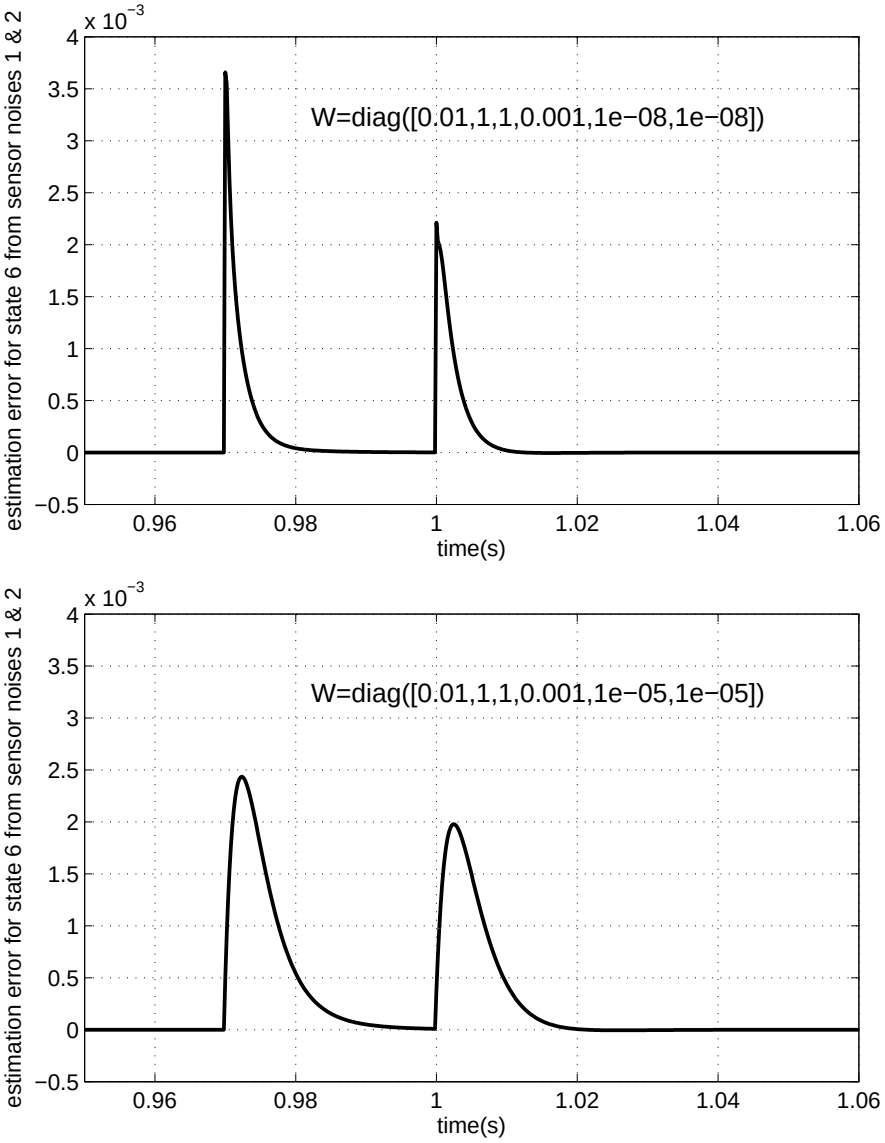


Figure 7.18: Kalman filter tuning with W_{ds1} and W_{ds2}

The example cases in this chapter showed the effectiveness of the proposed tuning method of Chapter 6 for the LQR controllers and estimators.

CHAPTER 8

Conclusion

This dissertation developed and presented a method to model, apply controls and stabilize a time-varying delayed distributed parameter system represented by the machining process of thin-walled structures. We demonstrated the modeling steps to go from the infinite dimensional mathematical model of the workpiece to a finite dimensional one through the use of FEM, reducing the order of the model by taking it to modal domain and performing model reduction, absorbing the delay in the cutting process equations into the states using semi-discretization, and finally creating a linear time invariant model by lifting of the inputs and outputs and performing another round of model reduction. The LTI model created this way was then ready to be used for stability analysis and control design.

In this research we suggested a new method of control to be applied to the machining process. While in the past researchers had explored the control of the cutting tool to avoid chatter in machining, here we recommended applying control to the workpiece through a robotic manipulator. The advantages of this alternative method were discussed among which were the ability to leave the rigidity of the tool uncompromised and the fact that robot manipulators are already available and in use in machining facilities for other tasks such as loading and unloading parts into machines and adding an end effector would be at a small cost with major benefit of increasing productivity and shifting the challenge of creating custom made hardware for work holders to design of controllers for the software enabled active steady rests.

The control applied through the active end effector on a robotic arm is suggested to be applied to the workpiece directly. Two different control strategies were discussed. The first

control method was use of linear quadratic regulator which is an optimal control design. We presented the results of applying the control designed with this method to the cutting process of a thin-walled tube and showed that control applied through the robotic arm to the workpiece while it is being machined, not only can prevent chatter instability and increase the productivity of the machining plant, but also can improve the performance of machining process resulting in better precision and final cut shape and surface finish.

The second control method presented was a robust control method to account for uncertainties in the modeling of the machining process. For an uncertain cutting stiffness which is a number that results from several machining parameters and conditions, we had to introduce a new method of modeling and semi-discretization for the cutting process that would allow for separation of the cutting stiffness from the rest of the model. After presenting this method of modeling, a controller robust to uncertainties in the cutting stiffness number, κ, was designed and we showed that with this control method it is possible to pick a range for the cutting stiffness that could either have an upper and lower bound based on our perception of the uncertainty or could be more conservative and have a range starting from zero and stabilizing up to a desired cutting stiffness.

Future work can focus on the control design and methods that would not necessitate the lifting of inputs and outputs and would rather apply to the time varying system directly and could lead to faster controllers. Less conservative controllers could also be designed if knowledge of the overlapping factor is considered in the design process unlike the assumption of worst case scenario ($\mu = 1$) in this work.

Lastly a new method for tuning LQR controllers and estimators was introduced and derived. The proposed tuning method is based on the desired output behavior in time domain. In the past tuning of LQRs has been a trial and error process with little to no tangible connection between the tuning matrices and the time domain output of the closed loop system. In the suggested method, we penalize not only the output but also its derivatives in the cost function. Penalty on higher derivatives would translate into smoother response while other time domain characteristics such as rise time, settling time, overshoot, etc. could

also be directly affected. The tuning method was also applied to the cutting process of the thin-walled tube as in the LQR control design for that problem fast settling was essential to help increase the productivity of the plant. Results of the tuning method performed on a few other plants were also presented and the proposed tuning method was further verified by implementation on the physical system of magnetic bearing in the lab.

The methodology created in this dissertation to merge FEM analysis results and matrices with either of the two modeling and semi-discretization methods discussed and designing controllers to achieve stability or performance, can be applied to many other distributed parameter systems demonstrating time-delay and time-varying behaviors. And the new tuning method introduced for LQR controllers can help make tuning optimal controllers for high dimensional systems more intuitive.

References

[AA99] V. Adams and A. Askenazi. *Building Better Products with Finite Element Analysis*. Word Press, 1999.

[Aka89] Y. Akashi. "Drawing-in clamping device for chuck of lathe or the like.", June 13 1989. US Patent 4,838,562.

[AT94] DM Alter and Tsu-Chin Tsao. "Stability of turning processes with actively controlled linear motor feed drives." *Transactions-american society of mechanical engineers journal of engineering for industry*, **116**, 1994.

[Bal78] Mark J Balas. "Active control of flexible systems." *Journal of Optimization theory and Applications*, **25**(3):415–436, 1978.

[BL11] Shusheng Bi and Jie Liang. "Robotic drilling system for titanium structures." *The International Journal of Advanced Manufacturing Technology*, **54**(5-8):767–774, 2011.

[Cam89] G.N. Cameron. "Holder for machining thin walled cylinder.", March 14 1989. US Patent 4,811,962.

[CFE14] Arnab Chanda, Achim Fischer, Peter Eberhard, and Santosha Kumar Dwivedy. "Stability analysis of a thin-walled cylinder in turning operation using the semi-discretization method." *Acta Mechanica Sinica*, **30**(2):214–222, 2014.

[CMP02] R. D. Cook, D. Malkus, M. Plesha, and R. Witt. *Concepts and applications of finite element analysis*. Wiley, 4th edition, 2002.

[CXT09] Weifang Chen, Jianbin Xue, Dunbing Tang, Hua Chen, and Shaopeng Qu. "Deformation prediction and error compensation in multilayer milling processes for thin-walled parts." *International Journal of Machine Tools and Manufacture*, **49**(11):859–864, 2009.

[DA67] N.E. D and S.F. A. "Work-piece stabilizing means.", May 16 1967. US Patent 3,319,497.

[DPH13] Saptarshi Das, Indranil Pan, Kaushik Halder, Shantanu Das, and Amitava Gupta. "{LQR} based improved discrete {PID} controller design via optimum selection of weighting matrices using fractional order integral performance index." *Applied Mathematical Modelling*, **37**(6):4253 – 4268, 2013.

[DTS92] T Delio, J Tlusty, and S Smith. "Use of audio signals for chatter detection and control." *Journal of engineering for industry*, **114**(2):146–157, 1992.

[FE11] Achim Fischer and Peter Eberhard. "Simulation-based stability analysis of a thin-walled cylinder during turning with improvements using an adaptronic turning chisel." *Archive of Mechanical Engineering*, **58**(4):367–391, 2011.

[FEA14] Achim Fischer, Peter Eberhard, and Jorge Ambrósio. "Parametric flexible multibody model for material removal during turning." *Journal of Computational and Nonlinear Dynamics*, **9**(1):011007, 2014.

[FM99] A Felix and SN Melkote. "Effect of workpiece flatness and surface finish on the holding force of a magnetic chuck." *Journal of manufacturing science and engineering*, **121**(4):811–814, 1999.

[GA74] B. G and R. A. "Fixture for supporting a workpiece in a machine tool.", October 1 1974. US Patent 3,838,865.

[GMJ16] Yuan-yuan Gao, Jian-wei Ma, Zhen-yuan Jia, Fu-ji Wang, Li-kun Si, and Dening Song. "Tool path planning and machining deformation compensation in high-speed milling for difficult-to-machine material thin-walled parts with curved surface." *The International Journal of Advanced Manufacturing Technology*, **84**(9):1757–1767, 2016.

[GS95] Levent Gven and Krishnaswamy Srinivasan. "Force controller design and evaluation for robot-assisted die and mould polishing." *Mechanical Systems and Signal Processing*, **9**(1):31 – 49, 1995.

[GS97] Levent Gven and Krishnaswamy Srinivasan. "An overview of robot-assisted die and mold polishing with emphasis on process modeling." *Journal of Manufacturing Systems*, **16**(1):48 – 58, 1997.

[HIS06] Ferenc Hartung, Tamás Insperger, Gábor Stépán, and Janos Turi. "Approximate stability charts for milling processes using semi-discretization." *Applied mathematics and computation*, **174**(1):51–73, 2006.

[HK10] U. Heisel and C. Kang. "Model-based form error compensation in the turning of thin-walled cylindrical parts." *Production Engineering*, **5**(2), Dec. 2010.

[HL14] K. Hassani and W. Li. "Optimal Tuning of Linear Quadratic Regulators Using Quantum Particle Swarm Optimization." In *Int. Conference of Control, Dynamic Systems, and Robotics*, May 2014.

[HT74] NH Hanna and SA Tobias. "A theory of nonlinear regenerative chatter." *Journal of Engineering for Industry*, **96**(1):247–255, 1974.

[HT98] RD Hanson and Tsu-Chin Tsao. "Reducing cutting force induced bore cylindricity errors by learning control and variable depth of cut machining." *Journal of manufacturing science and engineering*, **120**(3):547–554, 1998.

[HT00] Reed D Hanson and Tsu-Chin Tsao. "Periodic sampling interval repetitive control and its application to variable spindle speed noncircular turning process." *Journal of Dynamic systems, Measurement, and control*, **122**(3), 2000.

[IS02] Tamás Insperger and Gábor Stépán. "Semi-discretization method for delayed systems." *International Journal for numerical methods in engineering*, **55**(5):503–518, 2002.

[IS04] Tamás Insperger and Gábor Stépán. "Updated semi-discretization method for periodic delay-differential equations with discrete delay." *International Journal for Numerical Methods in Engineering*, **61**(1):117–141, 2004.

[IST07] Tamás Insperger, Gábor Stépán, and Janos Turi. "State-dependent delay in regenerative turning processes." *Nonlinear Dynamics*, **47**(1-3):275–283, 2007.

[KJ13] V. Kumar E and J. Jerome. "LQR based optimal tuning of PID controller for trajectory tracking of Magnetic Levitation System." In *International Conference On DESIGN AND MANUFACTURING*, 2013.

[KLL15] C. Kim, J. Lee, and H. Lee. "A Tuning Algorithm for LQ-PID Controllers using the Combined Time and Frequency-Domain Control Method." *J Electr Eng Technol*, **10**(3), 2015.

[LDK90] S Lin, R DeVor, and S Kapoor. "The effects of variable speed cutting on vibration control in face milling." *Journal of Engineering for Industry*, **112**(1), 1990.

[MB75] R Meyer and CS Burrus. "A unified analysis of multirate and periodically time-varying digital filters." *IEEE Transactions on Circuits and Systems*, **22**(3), 1975.

[Mer65] HE Merritt. "Theory of Self-Excited Machine-Tool Chatter: Contribution to Machine-Tool Chatter Research." *Journal of engineering for industry*, **87**(4):447–454, 1965.

[MG86] R Middleton and G Goodwin. "Improved finite word length characteristics in digital control using delta operators." *IEEE Transactions on Automatic Control*, **31**(11):1015–1021, 1986.

[MG90] R. H. Middleton and G. C. Goodwin. *Digital Control and Estimation A Unified Approach*. Prentice Hall, 1990.

[MHB16] Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. "Automatic LQR tuning based on Gaussian process global optimization." In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 270–277. IEEE, 2016.

[NRR02] J Norberto Pires, John Ramming, Stephen Rauch, and Ricardo Araújo. "Force/torque sensing applied to industrial robotic deburring." *Sensor Review*, **22**(3):232–241, 2002.

[OH98] Nejat Olgac and Martin Hosek. "A new perspective and analysis for regenerative machine tool chatter." *International Journal of Machine Tools and Manufacture*, **38**(7):783–798, 1998.

[OS89] S. Omatu and J. Seinfeld. *Distributed Parameter Systems Theory and Applications*. Oxford Science Publications, 1989.

[PMS96] Brad Paden, Nancy Morse, and Roy Smith. "Magnetic bearing experiment for integrated teaching and research laboratories." In *Control Applications, 1996., Proceedings of the 1996 IEEE International Conference on*, pp. 421–425. IEEE, 1996.

[Ric03] Jean-Pierre Richard. "Time-delay systems: an overview of some recent advances and open problems." *automatica*, **39**(10):1667–1694, 2003.

[RLH04] Svetan Ratchev, Shulong Liu, Wei Huang, and Adib A Becker. "Milling error prediction and compensation in machining of low-rigidity parts." *International Journal of Machine Tools and Manufacture*, **44**(15):1629–1641, 2004.

[RSM05] C. Ramos, J. Sanchis, M. Martinez, and M. Herrero. "LQR AND PREDICTIVE CONTROL TUNED VIA BDU." *Systems Science*, 2005.

[SGY09] J Srinivas, R Giri, and Seung-Han Yang. "Optimization of multi-pass turning using particle swarm intelligence." *The International Journal of Advanced Manufacturing Technology*, **40**(1-2):56–66, 2009.

[SI97] E Soliman and F Ismail. "Chatter suppression by adaptive speed modulation." *International Journal of Machine Tools and Manufacture*, **37**(3):355–369, 1997.

[SMS77] J. Sexton, R. Milne, and B. Stone. "A stability analysis of single-point machining with varying spindle speed." *Appl. Math. Modelling*, **1**, Sep. 1977.

[SN78] K Srinivasan and CL Nachtigal. "Investigation of the cutting process dynamics in turning operations." *Journal of Engineering for Industry*, **100**(3):323–331, 1978.

[ST92] S Smith and J Tlusty. "Stabilizing chatter by automatic spindle speed regulation." *CIRP Annals-Manufacturing Technology*, **41**(1):433–436, 1992.

[Ste01] Gábor Stépán. "Modelling nonlinear regenerative effects in metal cutting." *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, **359**(1781):739–757, 2001.

[TC95] FP Tan and RC Creese. "A generalized multi-pass machining model for machining parameter selection in turning." *The International Journal of Production Research*, **33**(5):1467–1487, 1995.

[Tlu86] J Tlusty. "Dynamics of high-speed milling." *Journal of Engineering for industry*, **108**(2):59–67, 1986.

[TMD14] S. Trimpe, A. Millane, S. Doessegger, and R. DAndrea. "A Self-Tuning LQR Approach Demonstrated on an Inverted Pendulum." In *19th World Congress, The International Federation of Automatic Control*, Aug 2014.

[TMK93] T. C. Tsao, M. McCarthy, and S. Kapoor. "A new approach to stability analysis of variable speed machining systems." *International Journal of Machine Tools and Manufacture*, **33**(6), Dec. 1993.

[TTP01] Tsu-Chin Tsao, Chin-An Tan, Alexander Pesterev, Bingen Yang, and LA Bergman. "Control oriented formulation for structures interacting with moving loads." In *American Control Conference, 2001. Proceedings of the 2001*, volume 1, pp. 441–446. IEEE, 2001.

[YEN02] Alpay Yilmaz, AL-Regib Emad, and Jun Ni. "Machine tool chatter suppression by multi-level random spindle speed variation." *Journal of manufacturing science and engineering*, **124**(2):208–216, 2002.

[ZCX06] Hui Zhang, Heping Chen, Ning Xi, George Zhang, and Jianmin He. "On-line path generation for robotic deburring of cast aluminum wheels." In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2400–2405. IEEE, 2006.