Abstract

Yildizoglu, Suat Ege. Distributed Parameter-Dependent Control of Non-uniform Flexible Structures. (Under the direction of Dr. Fen Wu)

In this thesis, we consider the distributed parameter-dependent modeling and control of non-uniform flexible structures, which are classified under spatially varying distributed systems. A distributed state space model of a non-uniform flexible cantilever beam is developed, in which the spatial variation of the beam parameters is treated as parametric uncertainty, assuming that the system depends on the spatially varying parameters in linear fractional manner. We are particularly interested in the systems discretized in spatial dimension for practical reasons. Spatial discretization is obtained via the central finite difference scheme. We assume that the displacements at each discretized node are measurable in real-time for controller use. Based on the proposed distributed model, sufficient conditions for analysis and synthesis of a distributed output-feedback controller are presented using the induced $\mathcal{L}_2$ norm as the performance criterion. The controller synthesis condition is characterized in terms of linear matrix inequalities, which are convex optimization problems and can be solved efficiently using available software. The distributed controller inherits the same structure as the plant, which results in a localized control architecture and a simple implementation. Each local controller unit processes the available local displacement measurement while sharing information with its adjacent units. We present the main advantage of distributed control; its reliability in the case of malfunctioning actuators or sensors, where many other control techniques would probably fail.

DISTRIBUTED PARAMETER-DEPENDENT CONTROL OF NON-UNIFORM FLEXIBLE STRUCTURES

BY

Suat Ege Yildizoglu

A THESIS SUBMITTED TO THE GRADUATE FACULTY OF

NORTH CAROLINA STATE UNIVERSITY

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

RALEIGH

July 2002

APPROVED BY:

CHAIR OF ADVISORY COMMITTEE

Dedicated to...

my parents *Leman and Savaş*

my sister *Ece*

and

my beloved *Deniz*

Biography

Suat Ege Yildizoglu was born on May 10, 1978 in Ankara, Turkey. He grew up in a small city of Kdz. Ereğli on the Black Sea coast. After qualifying in the top 1% in the university entrance exams, he was admitted to the Middle East Technical University in 1995, and earned his Bachelor of Science degree in Mechanical Engineering in June 2000. He had always planned to pursue a Masters degree outside Turkey to learn more about other cultures while broadening his knowledge in dynamic systems and controls. So he joined the graduate program in Mechanical and Aerospace Engineering at North Carolina State University in August 2000.

Acknowledgements

I would like to express my deepest appreciation to my advisor Dr. Fen Wu for all his support, encouragement, availability and invaluable guidance throughout this research. I cannot thank you enough for your patience and generosity. It has been a great pleasure to work with you.

I would like to thank my committee members, Dr. Larry Silverberg and Dr. Paul Ro, for taking interest in my work and for their valuable time. I would like to take this opportunity to thank the Mechanical and Aerospace Engineering Department for the financial support they provided.

I would like to give my appreciation to my fellow graduate students, Juan Jaramillo and Marco Soto for their kindness and friendship. I would like to extend my special thanks to Marco, who did the painstakingly proof-reading of this thesis. I wish you luck in the rest of your lives. I would also like to thank my roommate Fatih for his patience and support.

I would like to express my heartfelt gratitude to my parents, my aunt and my sister, who have always been there for me. I wouldn't have been here without their invaluable advice and encouragement. Thank you mom and dad, for all the things you have done to provide me with the best possible opportunities in life.

Last but certainly not the least, I would like to thank my lovely fiancee Deniz, without whose constant support and encouragement I wouldn't have survived this experience. She has always been my relief on the other side of the world during the toughest of my times.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The control of flexible structures has been one of the most active fields of research for the last two decades partly due to increased interest in high-speed robotics and space applications. In the construction of large space structures, for example, low-mass density materials need to be used due to the high launch cost. Since these materials are also lightly damped, the increased flexibility of space structures may allow large amplitude vibration, which may cause structural instability. In high-speed industrial robotics applications, however, the highly flexible nature of the links leads to a challenging problem of end-point trajectory control. Therefore, sophisticated controllers that enhance the stiffness and damping properties of these structures are needed to overcome these problems.

There are two main approaches for the vibration control: passive and active. In the passive method, the damping of the structure is increased by using passive dampers or materials with significant viscoelasticity (see for example [21], [39]). Nevertheless, this method can increase the total weight considerably, which is not desirable in many applications. On the other hand, the developments in material science have led to a number of promising materials such as piezoelectric materials, which can transform electrical energy into mechanical energy, and vice versa, so that they can be used as sensor or actuator elements for active control of structural vibration.

Variety of modeling and design approaches have been proposed for the active

vibration control of flexible structures. In this chapter, a brief review of related research in literature is presented.

1.1 Background

Flexible structures are usually described as distributed parameter systems governed by partial differential equations (PDEs). The simplest and most common example of flexible structures studied in the literature is a flexible beam. Derivation of the differential equations of the flexible beam motion follows the classical analytical techniques, namely the Bernoulli-Euler and Timoshenko beam theories. It is generally accepted that for long, thin flexible beams, the Bernoulli-Euler model provides an accurate description of the beam dynamics [40], while the more complicated Timoshenko model works better for short, thick beams and for laminated ones. In most applications, however, Bernoulli-Euler beam theory is preferred.

The control design technique is heavily dependent on the flexible beam modeling approach. So far, the majority of researchers used the conventional modal approach. In this method, the continuous distributed parameter system that has an infinite number of natural frequencies and corresponding mode shapes is approximated with a finite number of dominant modes by truncating the modes predicted by the undamped Bernoulli-Euler beam theory. Therefore, the governing PDE is transformed into a number of ordinary differential equations which are much easier to solve.

Nevertheless, the exact solution of flexible beam is available only for some special cases that are limited to those linear continuous systems with uniform geometry, while the infinite dimensional nature of the problem imposes severe constraints on control design [38]. Thus, for more complicated systems with non-uniform geometries, approximate solutions are often sought through spatial discretization [29]. The finite element method (FEM) is probably the most popular discretization technique in modeling the general flexible beam [20], [6], [41], [25]. In this method, the flexible beam is considered as an assembly of a finite number of elements where prespecified

shape functions are used to determine the natural frequency and mode shape approximations. FEM is especially useful when modeling the non-uniform flexible beams [26].

Modern control techniques are then applied to the resulting state-space model. However, reducing an infinite dimensional distributed system to a finite dimensional system has a well-known disadvantage in the control design, known as *control spillover*, where the controller excites unmodeled higher-order modes, causing unwanted structural vibration [3], [2]. Extremely fine meshes are often required in spatial discretization to avoid such problems. Thus, when a centralized control scheme is employed, control design based on the state-space approximation usually involves an undesirably large number of states.

An alternative to a modal approach is a distributed model, which is based on the original partial differential equation that describe the structure. In this method, continuous measurements are made over the spatial domain of the system and control forces are applied in a similar manner. Spatial discretization of the distributed parameter system is again required to incorporate spatially distributed arrays of sensors and actuators. However, our experience showed that much less elements are needed to provide adequate approximation of the flexible beam. A control design based on a distributed model has the potential to eliminate the effects of control spillover since it is directly based on the mathematical model that represents the actual physical system [28].

Moreover, a distributed model is suitable for decentralized control schemes, also referred to as distributed control, due to the existence of distributed sensing and actuation. Centralized control schemes, however, seem unattractive because high levels of connectivity are required in large arrays of sensors and actuators, resulting in a substantially large computational effort. Furthermore, they are more sensitive to failures, parameter variations and modeling errors than decentralized schemes. On the other hand, in the distributed control, the failure of any unit can be compensated for successfully by its adjacent neighbors. Therefore, in terms of reliability, distributed

control schemes have advantages.

It should be pointed at that, large arrays of sensors and actuators introduce large numbers of inputs and outputs which lead to severe limitations as many of the standard multi-variable control techniques are inadequate in handling systems with high dimensions. This necessitates the development of advanced distributed control design techniques.

Indeed, in the last decade there has been a renewed interest in the distributed control of complex engineering systems that are composed of similar subsystems, which interact with their closest neighbors. Such systems include unmanned aerial vehicles [18] and satellites [22] flying in formation, vehicle platoons [42] and automated highway systems [34] as well as the flexible structures and systems that are also characterized by the same class of partial differential equations.

In [4] and [5], the distributed controllers were designed for spatially-invariant distributed systems as a function of the spatial frequency variables obtained by taking Fourier transforms in the spatial independent variables. It was shown that due to the spatial invariance, only one controller with a single actuator/sensor pair needed to be designed, while all others are obtained by symmetry. It was also reported that resulting optimal controllers have a certain degree of inherent decentralization.

A linear matrix inequality (LMI) based control design method was developed for spatially-invariant distributed systems using multidimensional optimization in [7], [9], [10]. Unlike the *necessary and sufficient* conditions for stability and performance provided by [5], the conditions derived are only sufficient, but may not be necessary and the implementation of the controller is almost decentralized in nature. The application of distributed control synthesis method is demonstrated in [8] on the control of a vibrating elastic cable problem, where the horizontal level of a seismic cable being towed under-water is desired to be kept constant. Comparison of the closed-loop results with centralized $\mathcal{H}_\infty$ controllers is also provided. [27] is another application of the distributed control technique to an adaptive secondary mirror of a telescope, which is a 2D problem in spatial domain. Another example of a similar

method was proposed in [11] for localized control of spatially-invariant distributed systems. Localization is directly imposed in controller design at the expense of some conservatism. The method can be applied to both state and output feedback problems where the performance objective ($\mathcal{H}_2$ or $\mathcal{H}_\infty$) is stated in terms of the search for a suitable Lyapunov matrix over spatial frequency. The sufficient conditions for the existence of a controller with a desired localization and performance were obtained in the form of LMIs over spatial frequency.

Although some systems may be approximated as spatially-invariant, most systems cannot be accurately modeled as such. The conditions derived in [7] for spatially-invariant systems were extended to spatially varying systems in [15] using an operator theoretical approach [16]. The derived conditions are, in general, infinite dimensional. Unfortunately, unless the system is finite extent or periodic in the spatial variables, and periodic and finite duration in the temporal variable, which causes the conditions to become finite dimensional, this approach leads to an infinite dimensional optimization problem. Thus, the problem is computationally involved and hard to solve.

At this point, the distributed control of spatially varying distributed parameter systems is inspired by the gain-scheduling control design methodology [36]. The classical approach to gain scheduling control involves the design of several local linear controllers based on linearized models of a nonlinear system at several different operating conditions. Then, a global nonlinear controller is obtained by interpolating the local controller gains. However, stability and performance guarantees can only be provided for slowly varying parameters [37].

Based on the scaled small gain theorem [12], a systematic gain scheduling design technique was developed in [32] in the framework of linear parameter varying (LPV) systems. It was shown that when the parameter dependency of both the plant and the controller is in linear fractional form, the existence of such a gain scheduling controller is characterized by a set of LMIs. Therefore the synthesis problem becomes a convex optimization problem. This control structure is applicable whenever the

measurements of the parameter is available in real-time and the resulting controller is "scheduled" smoothly by those measurements of the parameter. The same gain-scheduling synthesis problem was cast in one of robust performance with special plant-uncertainty structure in [1]. It has since been improved by [35] to reduce the conservatism associated with full block diagonal scaling.

Motivated by [10] and [15], a distributed gain scheduling control approach was developed in [44] for distributed systems with temporally and spatially varying parameters. The new approach is computationally more attractive when compared to the results of [15]. The localized controller architecture assumes local variations of parameters only, and facilitates the implementation of a control algorithm in a similar manner.

1.2 Thesis Objective

Inspired by [10], [44], [45], the main goal of this thesis is to design a distributed parameter dependent controller using distributed gain scheduling methodology for the vibration suppression problem of non-uniform flexible structures. In order to achieve this goal, an accurate distributed model of the system will be obtained. A flexible cantilever beam with varying thickness is considered as the non-uniform flexible structure. Although the flexible beam is one of the most common structural problems studied in the literature, much of the research has been focused on centralized control of uniform structures. Here we are interested in the development of distributed controller. The main contributions of this research can be summarized as,

- A non-uniform structure is considered resulting in the presence of spatially varying parameters.
- A distributed model is developed based on the differential characterization of the actual system so that the local behavior of the distributed system is emphasized.
- A distributed parameter-dependent control approach is used to obtain a

controller that has a certain degree of decentralization.

- The advantages of distributed controllers such as reduced computational cost and increased reliability has been demonstrated.

1.3 Thesis Outline

This thesis is organized in the following outline. **Chapter 1** is used to introduce the research subject of this thesis, that is, distributed control of flexible structures. A partial list of references has been given to motivate the study.

Chapter 2 introduces the distributed parameter dependent systems framework. We begin with the preliminary definitions of linear fractional transformations (LFT), matrix norms, LMIs, and the scaled bounded real lemma, which are extensively used in the optimal robust control theory. Then following the description of the class of distributed parameter-dependent systems of interest in section 2.2, the stability and performance analysis conditions are discussed. The chapter is concluded by stating the main theory of distributed parameter dependent control synthesis and controller reconstruction results.

The modeling of a non-uniform flexible cantilever beam is presented in **Chapter 3**. This chapter starts with the derivation of the equation of motion of the beam in the transverse direction. The state-space realization of the distributed model of the non-uniform flexible beam is obtained through section 3.2 to section 3.4. Finally, the analysis of the validity of the developed distributed model is provided, where results of a finite element model are compared to those of the distributed model by the analysis of modal responses.

In **Chapter 4**, the distributed parameter-dependent controller synthesis is presented. The performance of the designed controller is tested with various case studies, and numerical simulations of these are provided.

Finally, the discussion of the results of the controller synthesis, concluding remarks and the comments on the related future research are provided in **Chapter 5**.

Chapter 2

Distributed Parameter Systems

The distributed parameter systems are composed of similar interconnected subsystems, which are autonomous in the sense that they have their own sensing and actuation capabilities. Even though each subsystem may have simple dynamics and a similar tractable model, the interaction between the neighboring subsystems leads to a complex dynamics when viewed as a whole. Consider, for example, a large flexible structure integrated with an array of sensors and actuators. It may be modeled as a combination of smaller flexible structures each having a sensor and an actuator, that are strongly attached to each other.

In this chapter, we would like to describe the class of distributed parameter systems in which we are interested, and state the stability and performance analysis conditions. The synthesis conditions for a distributed parameter-dependent controller will then be defined. However, we shall begin with a preliminary section, to define some concepts that are frequently used in optimal robust control theory.

2.1 Preliminary

In this section, some useful notations and important theoretical results will be provided with their proofs omitted.

2.1.1 Linear Fractional Transformations

Linear fractional transformations are matrix functions which have a significant importance in the robust control theory. They provide a framework that generalizes the transfer functions and their state-space realizations to include uncertainties present in the system model.

If M is a complex matrix partitioned in the following form,

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

then, the linear fractional transformations are defined as,

$$\mathcal{F}_l(M, \Delta_l) := M_{11} + M_{12}\Delta_l(I - M_{22}\Delta_l)^{-1}M_{21} \tag{2.1}$$

$$\mathcal{F}_u(M, \Delta_u) := M_{22} + M_{21}\Delta_u(I - M_{11}\Delta_u)^{-1}M_{12} \tag{2.2}$$

Obviously, an LFT is well-defined only if the inverses $(I - M_{22}\Delta_l)^{-1}$ and $(I - M_{11}\Delta_u)^{-1}$ exist, otherwise it is meaningless. The physical meaning of an LFT becomes clear when matrix M is taken as a proper transfer matrix of a feedback system. Hence, M may be the plant to be controlled and Δ may either be the controller or the uncertainty present in the system model. $\mathcal{F}_l(M, \Delta_l)$ is a transformation obtained from closing the loop using *lower* input/output channels, which is usually used for the interconnection of the plant and the controller. Similarly, $\mathcal{F}_u(M, \Delta_u)$ is obtained by closing the loop with *upper* channels, and often used for uncertainty. More detailed information on LFT may be found in [13], [48].

When the spatially interconnected systems are considered, LFT framework possesses another important feature. The interconnection between the subsystems are included in the uncertainty block structure, that is, the temporal and spatial shift operations are defined in the upper LFT form. This will be presented in section 2.2.

2.1.2 Signals, Norms and Operators

In any control system design, the main objective is to achieve certain performance specifications and internal stability. The performance specifications are described in terms of the size of certain signals of interest. The two popular measures in optimal control theory are $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms.

A *Hilbert space* is a complete inner product space with the norm induced by its inner product [19]. A well-known infinite dimensional Hilbert space is $\mathcal{L}_2(j\mathbf{R})$ space, which consists of all complex matrix functions F such that the following integral is bounded,

$$\int_{-\infty}^{\infty} trace\left[F^*(j\omega)F(j\omega)\right] dw < \infty$$

where, the inner product for this space and induced $\mathcal{L}_2$ norm are defined for F, G as,

$$\langle F, G\rangle := \frac{1}{2\pi}\int_{-\infty}^{\infty} trace\left[F^*(j\omega)G(j\omega)\right] dw, \quad and \quad \|F\|_2 := \sqrt{\langle F, F\rangle}$$

$\mathcal{H}_2$ is a closed subspace of $\mathcal{L}_2(j\mathbf{R})$ with matrix functions analytic in the open right half plane. The $\mathcal{L}_2$ spaces defined in frequency domain can be related to the $\mathcal{L}_2$ spaces defined in time domain [48]. Then,

$$\mathcal{L}_2(j\mathbf{R}) \cong \mathcal{L}_2\left(-\infty, \infty\right)$$
$$\mathcal{H}_2 \cong \mathcal{L}_2\left[0, \infty\right)$$

Similarly, $\mathcal{H}_\infty$ is a closed subspace of $\mathcal{L}_\infty$, a Banach space of matrix-valued functions that are essentially bounded on $(j\mathbf{R})$, with functions that are analytic and bounded in the open right-half plane. The $\mathcal{H}_\infty$ norm is defined as,

$$\|F\|_\infty := \sup_w \bar{\sigma}\left(F(j\omega)\right)$$

We will be dealing with signals which are, in general, vector valued functions having $L+1$ independent variables, i.e $(t, s_1, \cdots, s_L)$, where $t \in \mathbf{R}^+$ represents the

temporal dimension and $s_i \in \mathbf{Z}$ denotes spatial dimensions. For this type of signal, the 2-norm is defined by,

$$\|d\|_2^2 = \int_{t=0}^{\infty} \sum_{s_1=-\infty}^{\infty} \cdots \sum_{s_L=-\infty}^{\infty} d^T(s,t)d(s,t)dt \tag{2.3}$$

Therefore, a transfer matrix G operating on $\mathcal{L}_2[0,\infty)$ is bounded if the induced gain $\|G\|_2$ satisfies,

$$\|G\|_2 := \sup_{d\in\mathcal{L}_2, d\neq 0} \frac{\|Gd\|_2}{\|d\|_2} < \infty \tag{2.4}$$

2.1.3 Linear Matrix Inequalities

Linear matrix inequalities play an important role in the postmodern control as Lyapunov and Riccati equations played their part in the modern control theory. In fact, LMIs are direct byproducts of Lyapunov-based criteria.

The general LMI problem [13] can be formulated as to find whether there exists $X \in \mathcal{X}$ such that,

$$A_i^T X A_i - B_i^T X B_i + X C_i + C_i^T X + D_i < 0 \quad \forall i$$

where A_i, B_i, C_i, D_i are a list of matrices and,

$$\mathcal{X} = \left\{ \text{diag}\left[X_1, \cdots, X_S, x_1 I, \cdots, x_F I\right] : X_i \in \mathbf{R}^{r_i \times r_i}, X_i = X_i^T, x_j \in \mathbf{R} \right\}$$

Even though the $\mathcal{H}_\infty$ control problem has an analytical solution in terms of Riccati equations [14], which is all it takes to test for existence of adequate controllers, the LMI approach offers some valuable advantages. First, it is applicable to all plants without restrictions on infinite or pure-imaginary invariant zeros. Secondly, they result in a finite dimensional parametrization of all suboptimal $\mathcal{H}_\infty$ controllers, including the reduced order ones. Finally, finite dimensional LMI problems are convex optimization problems, which can be solved efficiently with available interior point algorithms [30], [17].

2.1.4 Scaled Bounded Real Lemma

The *bounded real lemma*, proposed by Khargonekar and Zhou [24], is used in converting the $\mathcal{H}_\infty$ norm constraint of an LTI system into an equivalent linear matrix inequality condition. For LPV systems with LFT parameter dependency, synthesis of gain scheduled $\mathcal{H}_\infty$ controllers is based on the small gain theorem [12], therefore a scaled bounded real lemma for continuous time case [1] is provided below. In this case, the condition becomes only sufficient.

Lemma 2.1.1 *Consider an uncertain parameter structure Δ, the associated set of positive definite similarity scalings defined by $L_\Delta = \{L > 0 : L\Delta = \Delta L\} \subset \mathbf{R}^{r\times r}$, and a square continuous-time transfer function $T(s) = D + C(sI - A)^{-1}B$. Then the following statements are equivalent,*

1. *A is stable and there exist $L \in L_\Delta$ such that*

$$\|L^{1/2}(D + C(sI - A)^{-1}B)L^{-1/2}\|_\infty < \gamma$$

2. *There exist positive definite solutions X and $L \in L_\Delta$ to the matrix inequality,*

$$\begin{pmatrix} A^TX + XA & XB & C^T \\ B^TX & -\gamma L & D^T \\ C & D & -\gamma L^{-1} \end{pmatrix} < 0.$$

The LMI condition presented later in section 2.4 is based on the scaled bounded real lemma for distributed parameter dependent systems.

2.2 Distributed Parameter Systems

Flexible structures are continuous systems whose dynamics are expressed by partial differential equations. Their material properties are inherently distributed, therefore, they are often referred as *distributed parameter systems*. We are interested in the class of distributed parameter systems with spatially varying parameters. In this section,

the LFT framework is employed to describe parameter varying distributed parameter systems.

Consider the distributed parameter system, $\mathbf{G}$, in figure 2.1. Let the signal $d \in \mathbf{R}^{n_d}$ represent all distributed external inputs including disturbances, sensor noise, and commands, $z \in \mathbf{R}^{n_z}$ represent the error signal, $y \in \mathbf{R}^{n_y}$ be the measured variable, and $u \in \mathbf{R}^{n_u}$ be the control input. Signals $p, q \in \mathbf{R}^{n_p}$ are pseudo input/output channels for uncertainty. Define the state vector $x(s,t) = [x^t \quad x^s]^T$ that has both temporal and spatial components such that, $x^t \in \mathbf{R}^{m_0}$ represent the temporal states and $x^s \in \mathbf{R}^{m_1+m_{-1}+\cdots+m_L+m_{-L}}$ be the spatial states.

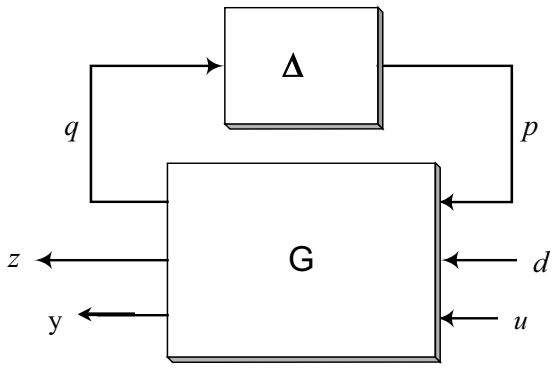


Figure 2.1: Distributed parameter system

It is hard to find exact solutions to the partial differential equations for spatially varying distributed systems. Therefore, approximated solutions are often sought through spatial discretization [29]. In this regard, the system is discretized in its spatial variable $s = [s_1, s_2, \cdots, s_L]^T$. Then, define the spatial shift operator $\mathbf{S}_i$ [10] as,

$$\mathbf{S}_i x(s,t) := x(s_1, \cdots, s_i + 1, \cdots, s_l, t) \quad i = 1, \cdots, L \tag{2.5}$$

In this study, we consider a distributed system with continuous temporal and discrete spatial states. Therefore, differential time operator $\mathbf{T}$ is defined in equation (2.6)

and will be used for the rest of the study. On the other hand, temporal states can be discretized as well. In that case, $\mathbf{T}$ becomes a temporal shift operator and is defined in equation (2.7). In other words,

$$either \quad \mathbf{T}x(s,t) := \frac{\partial x(s,t)}{\partial t} \tag{2.6}$$

$$or \quad \mathbf{T}x(s,k) := x(s,k+1) \tag{2.7}$$

Then, define $w(s,t) = [w^t \quad w^s]^T$ such that,

$$w^t = \Delta_{\mathbf{T}} x^t(s,t), \quad \Delta_{\mathbf{T}} := \mathbf{T} I_{m_0}$$
$$w^s = \Delta_{\mathbf{S}} x^s(s,t), \quad \Delta_{\mathbf{S}} := \mathrm{diag}(\mathbf{S}_1 I_{m_1}, \mathbf{S}_1^{-1} I_{m_{-1}}, \cdots, \mathbf{S}_L I_{m_L}, \mathbf{S}_L^{-1} I_{m_{-L}})$$

The state-space realization of the system $\mathbf{G}$, which is indeed a group of subsystems, above takes the following form:

$$\left[\begin{array}{c} w^t \\ w^s \\ \hline q(s,t) \\ e(s,t) \\ y(s,t) \end{array}\right] = \left[\begin{array}{cc|ccc} A^{tt} & A^{ts} & B_0^t & B_1^t & B_2^t \\ A^{st} & A^{ss} & B_0^s & B_1^s & B_2^s \\ \hline C_0^t & C_0^s & D_{00} & D_{01} & D_{02} \\ C_1^t & C_1^s & D_{10} & D_{11} & D_{12} \\ C_2^t & C_2^s & D_{20} & D_{21} & D_{22} \end{array}\right] \left[\begin{array}{c} x^t \\ x^s \\ \hline p(s,t) \\ d(s,t) \\ u(s,t) \end{array}\right] \tag{2.8}$$

$$\left[\begin{array}{c} x^t \\ x^s \\ \hline p(s,t) \end{array}\right] = \left[\begin{array}{cc|c} \Delta_{\mathbf{T}}^{-1} & & \\ & \Delta_{\mathbf{S}}^{-1} & \\ \hline & & \Delta(s) \end{array}\right] \left[\begin{array}{c} w^t \\ w^s \\ \hline q(s,t) \end{array}\right] \tag{2.9}$$

The integer variable s is indexed over all subsystems. The state space data is in linear fractional dependency [31] of the structured parameter Δ. Parameter δ describes the spatial variation of the distributed system and it is uniquely determined for any given variable s. Δ obeys the following structure,

$$\mathit{\Delta} = \left\{\Delta = \mathrm{diag}\left\{\delta_1 I_{s_1}, \cdots, \delta_g I_{s_g}\right\} : \quad \delta_i : \mathbf{N} \to \mathbf{R}, |\delta_i| \leq 1, i = 1, \cdots, g\right\}$$

For notational convenience in the rest of the presentation, following shorthand

matrices are defined,

$$\mathcal{A} = \begin{bmatrix} A^{tt} & A^{ts} \\ A^{st} & A^{ss} \end{bmatrix}, \quad \mathcal{B}_0 = \begin{bmatrix} B_0^t \\ B_0^s \end{bmatrix}, \quad \mathcal{B}_1 = \begin{bmatrix} B_1^t \\ B_1^s \end{bmatrix} \quad \mathcal{B}_2 = \begin{bmatrix} B_2^t \\ B_2^s \end{bmatrix}, \tag{2.10}$$

$$\mathcal{C}_0 = \begin{bmatrix} C_0^t & C_0^s \end{bmatrix} \quad \mathcal{C}_1 = \begin{bmatrix} C_1^t & C_1^s \end{bmatrix}, \quad \mathcal{C}_2 = \begin{bmatrix} C_2^t & C_2^s \end{bmatrix} \tag{2.11}$$

2.3 System Interconnection

Given the distributed parameter system $\mathbf{G}$ with all the weighting functions included, we would like to find a controller $\mathbf{K}$ which produces control signals u based on the measurement signals y such that the interconnection $\mathbf{M} = \mathcal{F}_l(\mathbf{G}, \mathbf{K})$ is well-posed and the closed loop system $\mathbf{M}$ is exponentially stable and the induced $\mathcal{L}_2$ norm from inputs d to outputs e is minimized.

We are interested in the distributed parameter-dependent controller $\mathbf{K}$ that has the same structure as the distributed plant $\mathbf{G}$. In other words, the controller $\mathbf{K}$ also depends on the structured parameter Δ in linear fractional form. Thus, state-space realization of the controller is given by,

$$\left[\begin{array}{c} w_k^t \\ w_k^s \\ \hline q_k(s,t) \\ u(s,t) \end{array}\right] = \left[\begin{array}{cc|cc} A_k^{tt} & A_k^{ts} & B_{0,k}^t & B_{1,k}^t \\ A_k^{st} & A_k^{ss} & B_{0,k}^s & B_{1,k}^s \\ \hline C_{0,k}^t & C_{0,k}^s & D_{00,k} & D_{01,k} \\ C_{1,k}^t & C_{1,k}^s & D_{10,k} & D_{11,k} \end{array}\right] \left[\begin{array}{c} x_k^t \\ x_k^s \\ \hline p_k(s,t) \\ y(s,t) \end{array}\right] \tag{2.12}$$

$$\left[\begin{array}{c} x_k^t \\ x_k^s \\ \hline p_k(s,t) \end{array}\right] = \left[\begin{array}{cc|c} \Delta_{\mathbf{T}}^{-1} & & \\ & \Delta_{\mathbf{S}}^{-1} & \\ \hline & & \Delta(s) \end{array}\right] \left[\begin{array}{c} w_k^t \\ w_k^s \\ \hline q_k(s,t) \end{array}\right] \tag{2.13}$$

where $x_k = [x_k^t \quad x_k^s]^T \in \mathbf{R}^{n_k}$, $p_k, q_k \in \mathbf{R}^{n_{p_k}}$. The implementation of the distributed control is shown in figure 2.2, where G_s represents subsystem s, and it is controlled by K_s.

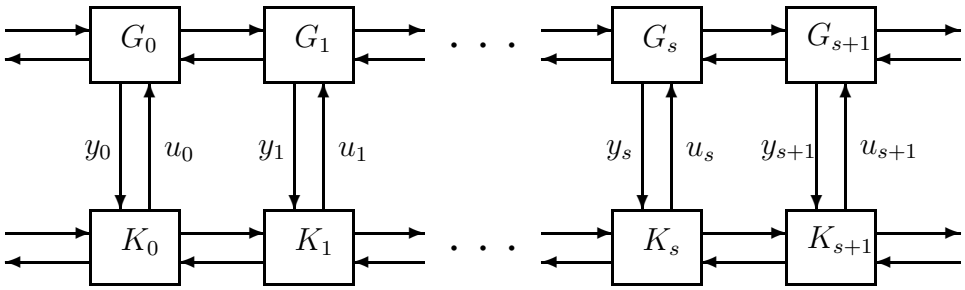


Figure 2.2: Distributed control architecture

If each controller K_s has only the local information y_s, u_s and no interaction with its neighbors, then we have a completely decentralized control. Implementation of completely decentralized controllers is easier because only the local information needs to be processed. However, no computationally tractable algorithms exist for the design of the completely decentralized controllers with performance and stability guarantees. Therefore, in our distributed control approach, the interaction between the neighboring controllers exists, which leads us somewhere in between a completely decentralized and completely centralized control.

Define the controller $\mathbf{K} = \{\mathcal{A}_k, \mathcal{B}_k, \mathcal{C}_k, \mathcal{D}_k\}$ as,

$$\mathcal{A}_k = \begin{bmatrix} A_k^{tt} & A_k^{ts} \\ A_k^{st} & A_k^{ss} \end{bmatrix}, \quad \mathcal{B}_k = \begin{bmatrix} B_{0,k}^t & B_{1,k}^t \\ B_{0,k}^s & B_{1,k}^s \end{bmatrix}$$
$$\mathcal{C}_k = \begin{bmatrix} C_{0,k}^t & C_{0,k}^s \\ C_{1,k}^t & C_{1,k}^s \end{bmatrix}, \quad \mathcal{D}_k = \begin{bmatrix} D_{00,k} & D_{01,k} \\ D_{10,k} & D_{11,k} \end{bmatrix}$$

Then, the closed-loop system $\mathbf{M} = \mathcal{F}_l(\mathbf{G}, \mathbf{K})$ shown in figure 2.3 can be written as,

$$\left[\begin{array}{c} w_{cl}^t \\ \hline w_{cl}^s \\ q_{cl}(s,t) \\ z(s,t) \end{array}\right] = \left[\begin{array}{c|cc} \mathcal{A}_{cl} & \mathcal{B}_{0,cl} & \mathcal{B}_{1,cl} \\ \hline \mathcal{C}_{0,cl} & D_{00,cl} & D_{01,cl} \\ \mathcal{C}_{1,cl} & D_{10,cl} & D_{11,cl} \end{array}\right] \left[\begin{array}{c} x_{cl}^t \\ \hline x_{cl}^s \\ p_{cl}(s,t) \\ d(s,t) \end{array}\right] \tag{2.14}$$

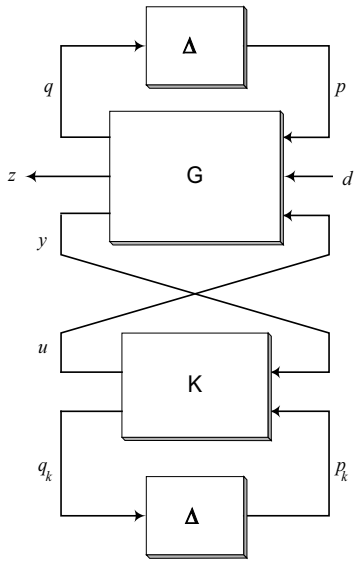


Figure 2.3: Closed-loop LFT Framework

$$\left[\begin{array}{c} x_{cl}^t \\ x_{cl}^s \\ \hline p_{cl}(s,t) \end{array}\right] = \left[\begin{array}{cc|c} \Delta_{\mathbf{T}}^{-1} & & \\ & \Delta_{\mathbf{S}}^{-1} & \\ \hline & & \begin{array}{cc} \Delta(s) & \\ & \Delta(s) \end{array} \end{array}\right] \left[\begin{array}{c} w_{cl}^t \\ w_{cl}^s \\ \hline q_{cl}(s,t) \end{array}\right] \tag{2.15}$$

where, $x_{cl} = \left[\begin{array}{cccc} x^t & x_k^t & x^s & x_k^s \end{array}\right]^T$, $p_{cl} = \left[\begin{array}{cc} p & p_k \end{array}\right]^T$, $q_{cl} = \left[\begin{array}{cc} q & q_k \end{array}\right]^T$, and

$$\left[\begin{array}{c|cc} \mathcal{A}_{cl} & \mathcal{B}_{0,cl} & \mathcal{B}_{1,cl} \\ \hline \mathcal{C}_{0,cl} & D_{00,cl} & D_{01,cl} \\ \mathcal{C}_{1,cl} & D_{10,cl} & D_{11,cl} \end{array}\right] = \left[\begin{array}{cc|cc} A_{cl}^{tt} & A_{cl}^{ts} & B_{0,cl}^t & B_{1,cl}^t \\ A_{cl}^{st} & A_{cl}^{ss} & B_{0,cl}^s & B_{1,cl}^s \\ \hline C_{0,cl}^t & C_{0,cl}^s & D_{00,cl} & D_{01,cl} \\ C_{1,cl}^t & C_{1,cl}^s & D_{10,cl} & D_{11,cl} \end{array}\right]$$

Clearly, the temporal and spatial state variables need to be grouped together. Therefore, a permutation matrix P is needed to convert the closed loop state-space data into the standard system definition. So, define,

$$P := \begin{bmatrix} P^t & 0 & P_k^t & 0 \\ 0 & P^s & 0 & P_k^s \end{bmatrix}$$

where P^t, P^s, P_k^t and P_k^s are permutation matrices compatible with associated temporal and spatial dimensions of the plant and the controller. Thus, the closed-loop system (2.14)-(2.15) can be related to the plant and controller data as,

$$\mathcal{A}_{cl} = P\bar{\mathcal{A}}_{cl}P^T, \tag{2.16}$$

$$\begin{bmatrix}\mathcal{B}_{0,cl} & \mathcal{B}_{1,cl}\end{bmatrix} = P\begin{bmatrix}\bar{\mathcal{B}}_{0,cl} & \bar{\mathcal{B}}_{1,cl}\end{bmatrix} \tag{2.17}$$

$$\begin{bmatrix}\mathcal{C}_{0,cl} \\ \mathcal{C}_{1,cl}\end{bmatrix} = \begin{bmatrix}\bar{\mathcal{C}}_{0,cl} \\ \bar{\mathcal{C}}_{1,cl}\end{bmatrix} P^T, \tag{2.18}$$

$$\begin{bmatrix}D_{00,cl} & D_{01,cl} \\ D_{10,cl} & D_{11,cl}\end{bmatrix} = \begin{bmatrix}\bar{D}_{00,cl} & \bar{D}_{01,cl} \\ \bar{D}_{10,cl} & \bar{D}_{11,cl}\end{bmatrix} \tag{2.19}$$

where,

$$\left[\begin{array}{c|cc} \bar{\mathcal{A}}_{cl} & \bar{\mathcal{B}}_{0,cl} & \bar{\mathcal{B}}_{1,cl} \\ \hline \bar{\mathcal{C}}_{0,cl} & \bar{D}_{00,cl} & \bar{D}_{01,cl} \\ \bar{\mathcal{C}}_{1,cl} & \bar{D}_{10,cl} & \bar{D}_{11,cl}\end{array}\right] = \left[\begin{array}{cc|cc|c} \mathcal{A} & 0 & \mathcal{B}_0 & 0 & \mathcal{B}_1 \\ 0 & 0 & 0 & 0 & 0 \\ \hline \mathcal{C}_0 & 0 & D_{00} & 0 & D_{01} \\ 0 & 0 & 0 & 0 & 0 \\ \hline \mathcal{C}_1 & 0 & D_{10} & 0 & D_{11}\end{array}\right]$$

$$+ \left[\begin{array}{c|cc} 0 & \mathcal{B}_2 & 0 \\ I & 0 & 0 \\ \hline 0 & D_{02} & 0 \\ 0 & 0 & I \\ \hline 0 & D_{12} & 0\end{array}\right] \left[\begin{array}{cc} \mathcal{A}_k & \mathcal{B}_k \\ \mathcal{C}_k & \mathcal{D}_k\end{array}\right] \left[\begin{array}{cc|cc|c} 0 & I & 0 & 0 & 0 \\ \hline \mathcal{C}_2 & 0 & D_{20} & 0 & D_{21} \\ 0 & 0 & 0 & I & 0\end{array}\right]$$

2.4 Stability and Performance Analysis

In this section, the stability and performance conditions of the closed-loop distributed system $\mathbf{M}$ are presented in the form of linear matrix inequalities.

First, we define the block diagonal matrix set $\mathcal{X}$ and scaling matrix set $\mathcal{D}$ as,

$$\mathcal{X} := \left\{X = \text{diag}\left\{X^t, X^s\right\} : \ X^t = X^{tT} > 0, \ X^s = X^{sT}\right\} \tag{2.20}$$

$$\mathcal{D} := \left\{D = \text{diag}\left\{D^{s_1}, \cdots, D^{s_g}\right\} : D = D^T > 0\right\} \tag{2.21}$$

Yet, another matrix transformation is needed before the stability and performance conditions can be tested. All theorems presented in this chapter have been derived for distributed systems that are continuous both in temporal and spatial dimensions. Since the system $\mathbf{M}$ is discretized in spatial dimension, a bilinear transformation is required to obtain its continuous counterpart.

So, let $H = \text{diag}\left\{I_{m_{0,cl}}, I_{m_{1,cl}}, I_{m_{-1,cl}}, \cdots, I_{m_{L,cl}}, I_{m_{-L,cl}}\right\}$, then, the transformation in spatial dimension can be obtained by

$$\begin{bmatrix} \tilde{x}^s \\ w^s \end{bmatrix} = \begin{bmatrix} H & \sqrt{2}I \\ \sqrt{2}H & I \end{bmatrix} \begin{bmatrix} \tilde{w}^s \\ x^s \end{bmatrix} \tag{2.22}$$

Now, the continuous system's state-space realization under such transformation is given by,

$$\tilde{A}^{ss}_{cl} = H\left(A^{ss}_{cl} - I\right)\left(A^{ss}_{cl} + I\right)^{-1}, \tag{2.23}$$

$$\begin{bmatrix} \tilde{A}^{st}_{cl} & \tilde{B}^s_{0,cl} & \tilde{B}^s_{1,cl} \end{bmatrix} = \sqrt{2}H\left(A^{ss}_{cl} + I\right)^{-1} \begin{bmatrix} A^{st}_{cl} & B^s_{0,cl} & B^s_{1,cl} \end{bmatrix} \tag{2.24}$$

$$\begin{bmatrix} \tilde{A}^{ts}_{cl} \\ \tilde{C}^s_{0,cl} \\ \tilde{C}^s_{1,cl} \end{bmatrix} = \sqrt{2} \begin{bmatrix} A^{ts}_{cl} \\ C^s_{0,cl} \\ C^s_{1,cl} \end{bmatrix} \left(A^{ss}_{cl} + I\right)^{-1} \tag{2.25}$$

$$\begin{bmatrix} \tilde{A}^{tt}_{cl} & \tilde{B}^t_{0,cl} & \tilde{B}^t_{1,cl} \\ \tilde{C}^t_{0,cl} & \tilde{D}_{00,cl} & \tilde{D}_{01,cl} \\ \tilde{C}^t_{1,cl} & \tilde{D}_{10,cl} & \tilde{D}_{11,cl} \end{bmatrix} = \begin{bmatrix} A^{tt}_{cl} & B^t_{0,cl} & B^t_{1,cl} \\ C^t_{0,cl} & D_{00,cl} & D_{01,cl} \\ C^t_{1,cl} & D_{10,cl} & D_{11,cl} \end{bmatrix} - \begin{bmatrix} A^{ts}_{cl} \\ C^s_{0,cl} \\ C^s_{1,cl} \end{bmatrix} \left(A^{ss}_{cl} + I\right)^{-1} \begin{bmatrix} A^{st}_{cl} & B^s_{0,cl} & B^s_{1,cl} \end{bmatrix} \tag{2.26}$$

The stability condition of the distributed system can now be stated by in the following theorem [44].

Theorem 1 *The distributed parameter-dependent system (2.14)-(2.15) is exponentially stable if there exists a symmetric matrix $X_{cl} \in \mathcal{X}$ and a scaling matrix $L_{cl} \in \mathcal{D}$, such that*

$$\left[\begin{array}{cc|cc} I & \tilde{\mathcal{A}}_{cl}^T & 0 & \tilde{\mathcal{C}}_{0,cl}^T \\ 0 & \tilde{\mathcal{B}}_{0,cl}^T & I & \tilde{D}_{00,cl}^T \end{array}\right] \left[\begin{array}{cc|cc} 0 & X_{cl} & \multicolumn{2}{c}{\multirow{2}{*}{0}} \\ X_{cl} & 0 & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{0}} & -L_{cl} & 0 \\ & & 0 & L_{cl} \end{array}\right] \left[\begin{array}{cc} I & 0 \\ \tilde{\mathcal{A}}_{cl} & \tilde{\mathcal{B}}_{0,cl} \\ \hline 0 & I \\ \tilde{\mathcal{C}}_{0,cl} & \tilde{D}_{00,cl} \end{array}\right] < 0$$

Proof of theorem 1 can be found in [44]. A distributed Lyapunov function is used to ensure the stability. However, apart from the standard Lyapunov stability result for a single system, only the first block X_{cl}^t of variable X_{cl} is required to be strictly positive in order to guarantee the stability of the distributed system.

Similarly, the performance of the distributed parameter-dependent system can be determined with the results provided by the next theorem. The proof of this theorem can also be found in [44].

Theorem 2 *The distributed parameter-dependent system (2.14)-(2.15) is exponentially stable and has $\|z\|_2 < \gamma\|d\|_2$ for all admissible uncertainties Δ if there exists a symmetric matrix $X_{cl} \in \mathcal{X}$ and a scaling matrix $L_{cl} \in \mathcal{D}$, such that*

$$\left[\begin{array}{c} \star \\ \hline \star \\ \hline \star \end{array}\right]^T \left[\begin{array}{cc|cc|cc} 0 & X_{cl} & \multicolumn{2}{c|}{\multirow{2}{*}{0}} & \multicolumn{2}{c}{\multirow{2}{*}{0}} \\ X_{cl} & 0 & & & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{0}} & -L_{cl} & 0 & \multicolumn{2}{c}{\multirow{2}{*}{0}} \\ & & 0 & L_{cl} & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{0}} & \multicolumn{2}{c|}{\multirow{2}{*}{0}} & -\gamma I & 0 \\ & & & & 0 & \frac{1}{\gamma} I \end{array}\right] \left[\begin{array}{ccc} I & 0 & 0 \\ \tilde{\mathcal{A}}_{cl} & \tilde{\mathcal{B}}_{0,cl} & \tilde{\mathcal{B}}_{1,cl} \\ \hline 0 & I & 0 \\ \tilde{\mathcal{C}}_{0,cl} & \tilde{D}_{00,cl} & \tilde{D}_{01,cl} \\ \hline 0 & 0 & I \\ \tilde{\mathcal{C}}_{1,cl} & \tilde{D}_{10,cl} & \tilde{D}_{11,cl} \end{array}\right] < 0 \tag{2.27}$$

Therefore, the stability and performance analysis conditions stated in theorem 1 and theorem 2, as well as the controller synthesis condition which will be stated shortly, can be obtained for both continuous and discrete systems in their corresponding forms through an appropriate bilinear transformation. Thus, this provides

us a simple way to derive LMI solutions from one system representation form to another.

2.5 Controller Synthesis

Consider the open-loop distributed parameter-dependent system $\mathbf{G}$ of (2.8)-(2.9). The control design objective is to minimize the induced $\mathcal{L}_2$ norm from d to z, where the signal's 2-norm is defined in equation (2.3). Therefore in mathematical terms, we seek the distributed parameter-dependent controller (2.12)-(2.13) to minimize γ such that

$$\|z\|_2 \leq \gamma \|d\|_2$$

The main assumptions on the plant description are,

(A1) $(\mathcal{A}, \mathcal{B}_2, \mathcal{C}_2)$ is stabilizable and detectable by a distributed controller,

(A2) $D_{22} = 0$

The first assumption guarantees the existence of a stabilizing distributed output-feedback controller. Assumption $(A2)$ can be relaxed by redefining the output variable [43].

Although theorem 2 provides us with the results to determine stability and the performance level of a distributed parameter-dependent system, it cannot be used for the design of a stabilizing controller $\mathbf{K}$. Equation (2.27) is an LMI in terms of the decision variable X_{cl}. However, the LMI must be partitioned compatibly to plant and controller states n and n_k such that,

$$X_{cl} = \begin{bmatrix} Y & N \\ N^T & \star \end{bmatrix}, \quad X_{cl}^{-1} = \begin{bmatrix} X & M \\ M^T & \star \end{bmatrix}$$

This time, the bilinear transformation is required for the open-loop system data.

$$\widehat{A}^{ss} = H\left(A^{ss} - I\right)\left(A^{ss} + I\right)^{-1}, \tag{2.28}$$

$$\begin{bmatrix} \widehat{A}^{st} & \widehat{B}_0^s & \widehat{B}_1^s \end{bmatrix} = \sqrt{2}H\left(A^{ss} + I\right)^{-1}\begin{bmatrix} A^{st} & B_0^s & B_1^s \end{bmatrix} \tag{2.29}$$

$$\begin{bmatrix} \widehat{A}^{ts} \\ \widehat{C}_0^s \\ \widehat{C}_1^s \end{bmatrix} = \sqrt{2}\begin{bmatrix} A^{ts} \\ C_0^s \\ C_1^s \end{bmatrix}\left(A^{ss} + I\right)^{-1} \tag{2.30}$$

$$\begin{aligned} \begin{bmatrix} \widehat{A}^{tt} & \widehat{B}_0^t & \widehat{B}_1^t \\ \widehat{C}_0^t & \widehat{D}_{00} & \widehat{D}_{01} \\ \widehat{C}_1^t & \widehat{D}_{10} & \widehat{D}_{11} \end{bmatrix} &= \begin{bmatrix} A^{tt} & B_0^t & B_1^t \\ C_0^t & D_{00} & D_{01} \\ C_1^t & D_{10} & D_{11} \end{bmatrix} \\ &\quad - \begin{bmatrix} A^{ts} \\ C_0^s \\ C_1^s \end{bmatrix}\left(A^{ss} + I\right)^{-1}\begin{bmatrix} A^{st} & B_0^s & B_1^s \end{bmatrix} \end{aligned} \tag{2.31}$$

For the simplicity of the theorem, group the matrices such that,

$$\widehat{\mathcal{A}} = \begin{bmatrix} \widehat{A}^{tt} & \widehat{A}^{ts} \\ \widehat{A}^{st} & \widehat{A}^{ss} \end{bmatrix}, \quad \widehat{\mathcal{B}}_0 = \begin{bmatrix} \widehat{B}_0^t \\ \widehat{B}_0^s \end{bmatrix}, \quad \widehat{\mathcal{B}}_1 = \begin{bmatrix} \widehat{B}_1^t \\ \widehat{B}_1^s \end{bmatrix} \quad \widehat{\mathcal{B}}_2 = \begin{bmatrix} \widehat{B}_2^t \\ \widehat{B}_2^s \end{bmatrix},$$
$$\widehat{\mathcal{C}}_0 = \begin{bmatrix} \widehat{C}_0^t & \widehat{C}_0^s \end{bmatrix} \quad \widehat{\mathcal{C}}_1 = \begin{bmatrix} \widehat{C}_1^t & \widehat{C}_1^s \end{bmatrix}, \quad \widehat{\mathcal{C}}_2 = \begin{bmatrix} \widehat{C}_2^t & \widehat{C}_2^s \end{bmatrix}$$

The alternative version of theorem 2 that leads to the solution of the output-feedback control synthesis problem is expressed in the following theorem.

Theorem 3 *Given the compact set Δ, a performance level $\gamma > 0$, and the open-loop distributed parameter-dependent system $\mathbf{G}$, there exists a distributed parameter dependent controller $\mathbf{K}$ in the form of (2.12)-(2.13) which renders the closed- loop system exponentially stable and $\|z\|_2 \leq \gamma\|d\|_2$ provided that there exist block-diagonal matrices $X, Y \in \mathcal{X}$ and scaling matrices L, J, which are commutable with Δ, such*

that

$$\mathcal{N}_X^T \begin{bmatrix} \star \\ \hline \star \\ \hline \star \end{bmatrix}^T \left[\begin{array}{cc|cc|cc} 0 & X & & & & \\ X & 0 & \multicolumn{2}{c|}{0} & \multicolumn{2}{c}{0} \\ \hline & & -L & 0 & & \\ \multicolumn{2}{c|}{0} & 0 & L & \multicolumn{2}{c}{0} \\ \hline & & & & -\frac{1}{\gamma}I & 0 \\ \multicolumn{2}{c|}{0} & \multicolumn{2}{c|}{0} & 0 & \gamma I \end{array}\right] \left[\begin{array}{ccc} -\widehat{\mathcal{A}}^T & -\widehat{\mathcal{C}}_0^T & -\widehat{\mathcal{C}}_1^T \\ I & 0 & 0 \\ \hline -\widehat{\mathcal{B}}_0^T & -\widehat{D}_{00}^T & -\widehat{D}_{10}^T \\ 0 & I & 0 \\ \hline -\widehat{\mathcal{B}}_1^T & -\widehat{D}_{01}^T & -\widehat{D}_{11}^T \\ 0 & 0 & I \end{array}\right] \mathcal{N}_X > 0 \tag{2.32}$$

$$\mathcal{N}_Y^T \begin{bmatrix} \star \\ \hline \star \\ \hline \star \end{bmatrix}^T \left[\begin{array}{cc|cc|cc} 0 & Y & & & & \\ Y & 0 & \multicolumn{2}{c|}{0} & \multicolumn{2}{c}{0} \\ \hline & & -J & 0 & & \\ \multicolumn{2}{c|}{0} & 0 & J & \multicolumn{2}{c}{0} \\ \hline & & & & -\gamma I & 0 \\ \multicolumn{2}{c|}{0} & \multicolumn{2}{c|}{0} & 0 & \frac{1}{\gamma}I \end{array}\right] \left[\begin{array}{ccc} I & 0 & 0 \\ \widehat{\mathcal{A}} & \widehat{\mathcal{B}}_0 & \widehat{\mathcal{B}}_1 \\ \hline 0 & I & 0 \\ \widehat{\mathcal{C}}_0 & \widehat{D}_{00} & \widehat{D}_{01} \\ \hline 0 & 0 & I \\ \widehat{\mathcal{C}}_1 & \widehat{D}_{10} & \widehat{D}_{11} \end{array}\right] \mathcal{N}_Y < 0 \tag{2.33}$$

$$\begin{bmatrix} Y^t & I \\ I & X^t \end{bmatrix} \geq 0 \tag{2.34}$$

$$\begin{bmatrix} L & I \\ I & J \end{bmatrix} \geq 0 \tag{2.35}$$

where

$$\mathcal{N}_X = \mathrm{Ker} \begin{bmatrix} \widehat{\mathcal{B}}_2^T & \widehat{D}_{02}^T & \widehat{D}_{12}^T \end{bmatrix}, \quad \mathcal{N}_Y = \mathrm{Ker} \begin{bmatrix} \widehat{\mathcal{C}}_2 & \widehat{\mathcal{D}}_{20} & \widehat{D}_{21} \end{bmatrix}$$

Theorem 3 is a generalization of LFT-type linear parameter varying control approach [32] to spatially varying distributed systems. Synthesis conditions are in the form of LMIs, which are convex optimization problems. They can be solved efficiently using interior-point optimization algorithms [17], [30].

X, Y, L and J are calculated as a result of the solution of inequalities (2.32)-(2.35). Then, the distributed parameter-dependent controller can be reconstructed by following the given steps [43], [44], [46]:

- $I - XY$ must be non-singular so perturb X and Y accordingly, if necessary. Let $MN^T = I - XY$ and reconstruct X_{cl} such that

$$X_{cl} = \begin{bmatrix} Y & N \\ N^T & -M^{-1}XN \end{bmatrix}$$

- Next, compute the matrices $\mathcal{F}$ and $\mathcal{G}$, which are defined by,

$$\mathcal{F} := -\left[\begin{bmatrix}\widehat{D}_{02}^T & \widehat{D}_{12}^T\end{bmatrix}\begin{bmatrix}L & 0\\ 0 & \frac{1}{\gamma}I\end{bmatrix}\begin{bmatrix}\widehat{D}_{02}\\ \widehat{D}_{12}\end{bmatrix}\right]^{-1}\left[\widehat{\mathcal{B}}_2^T X^{-1} + \begin{bmatrix}\widehat{D}_{02}^T & \widehat{D}_{12}^T\end{bmatrix}\begin{bmatrix}L & 0\\ 0 & \frac{1}{\gamma}I\end{bmatrix}\begin{bmatrix}\widehat{\mathcal{C}}_0\\ \widehat{\mathcal{C}}_1\end{bmatrix}\right]$$

$$\mathcal{G} := -\left[Y^{-1}\widehat{\mathcal{C}}_2^T + \begin{bmatrix}\widehat{\mathcal{B}}_0 & \widehat{\mathcal{B}}_1\end{bmatrix}\begin{bmatrix}L & 0\\ 0 & \gamma I\end{bmatrix}\begin{bmatrix}\widehat{D}_{20}^T\\ \widehat{D}_{21}^T\end{bmatrix}\right]\left[\begin{bmatrix}\widehat{D}_{20} & \widehat{D}_{21}\end{bmatrix}\begin{bmatrix}L^{-1} & 0\\ 0 & \frac{1}{\gamma}I\end{bmatrix}\begin{bmatrix}\widehat{D}_{20}^T\\ \widehat{D}_{21}^T\end{bmatrix}\right]^{-1}$$

- Compute the state-space matrices of the controller

$$\begin{aligned} A_k =& -N^{-1}\left\{\widehat{\mathcal{A}}^T + Y\left[\widehat{\mathcal{A}} + \widehat{\mathcal{B}}_2\mathcal{F} + \mathcal{G}C_2\right]X\right. \\ &+ Y\left[\left[\widehat{\mathcal{B}}_0 \quad \widehat{\mathcal{B}}_1\right] + \mathcal{G}\begin{bmatrix}\widehat{D}_{20} & \widehat{D}_{21}\end{bmatrix}\right]\begin{bmatrix}L & 0\\ 0 & \frac{1}{\gamma}I\end{bmatrix}\begin{bmatrix}\widehat{\mathcal{B}}_0^T\\ \widehat{\mathcal{B}}_1^T\end{bmatrix} \\ &\left.+ \begin{bmatrix}\widehat{\mathcal{C}}_0^T & \widehat{\mathcal{C}}_1^T\end{bmatrix}\begin{bmatrix}L^{-1} & 0\\ 0 & \frac{1}{\gamma}I\end{bmatrix}\left[\begin{bmatrix}\widehat{\mathcal{C}}_0\\ \widehat{\mathcal{C}}_1\end{bmatrix} + \begin{bmatrix}\widehat{D}_{02}\\ \widehat{D}_{12}\end{bmatrix}\mathcal{F}\right]X\right\}M^{-T} \qquad (2.36) \end{aligned}$$

$$B_k = N^{-1}Y\mathcal{G} \qquad (2.37)$$

$$C_k = \mathcal{F}XM^{-T} \qquad (2.38)$$

$$D_k = 0 \qquad (2.39)$$

It should be noted that the controller data computed from equation (2.36)-(2.39) has continuous temporal and spatial states. Therefore, it should be converted to its continuous temporal, discrete spatial form before it is implemented.

Chapter 3

Modeling of Flexible Structure

In this chapter, the dynamic model of a non-uniform flexible structure is developed. A cantilever beam with a variable thickness is considered as the non-uniform flexible structure. Although analyzing the dynamics of the flexible beam is a well known problem, often the modal approach is used in modeling. Such an approach is not suitable for non-uniform beam due to the lack of analytical solution. On the other end, our interest is in model-based control. Therefore, a distributed state-space model of the non-uniform flexible beam is developed directly using the partial differential equation that governs the beam's motion in transverse direction.

We will start with the derivation of the equation of motion of the non-uniform beam in transverse direction. Next, the modeling of the parameter variations is presented. In section 3.3, the resulting equation is discretized in the spatial dimension, which leads to the distributed state-space model described in section 3.4. Finally, the chapter will be concluded with the modal analysis and verification of the developed distributed model.

3.1 Equation of Motion

In this section, mathematical model of the transverse motion of a non-uniform flexible cantilever beam is derived. Consider the beam shown in figure 3.1. It is fixed from

one end to a solid base, and the other end is free to move in the transverse direction. Let $v(x,t)$ represent the transverse displacement, $f(x,t)$ be the distributed external forces including both the control forces $u(x,t)$, and the disturbance forces $d(x,t)$. It is assumed that the displacement can be measured and control forces can be applied at prespecified locations by distributed sensors and actuators placed on the surface of the beam. However, the dynamics of the beam is assumed to remain unchanged, despite the presence of these sensors and actuators.

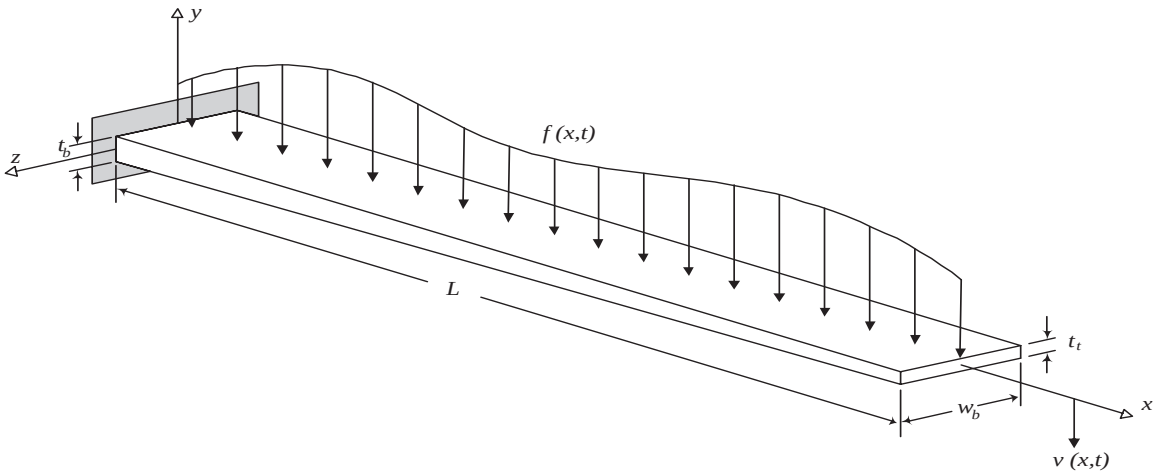


Figure 3.1: Non-uniform flexible cantilever beam

The dynamic equation of the beam can be derived by means of the extended Hamilton's principle as in [29]. It is assumed that the kinetic energy is due to the vertical translation alone, neglecting the shear deformation and rotary inertia. This is a valid assumption for long, thin flexible beams under static deflections [33]. It is shown in [40] that these terms can also be neglected for the frequencies much lower than the shear mode cutoff frequency in the dynamic response. The mechanical properties of the beam studied here are taken from [47], and its shear mode cutoff frequency was calculated there as $w_c = 12.5 \times 10^6 rad/sec$. It was also shown that shear deformation and rotary inertia can be ignored. Hence, Bernoulli-Euler beam theory can be used to model the dynamic response of the beam.

The partial differential equation of the transverse motion of the non-uniform flexible beam is then expressed in undamped Bernoulli-Euler form as:

$$\frac{\partial^2}{\partial x^2}\left[EI(x)\frac{\partial^2 v(x,t)}{\partial x^2}\right] + \frac{\partial^2}{\partial t^2}\left[\rho(x)v(x,t)\right] = f(x,t) \tag{3.1}$$

where $\rho(x)$ is the mass per unit length and $EI(x)$ is the flexural rigidity of the beam. E is the Young's modulus and $I(x)$ is the area moment of inertia about z-axis.

For the cantilever beam case, the displacement and slope must be zero at the fixed end, and the shear force and bending moment must be zero at the free end. Therefore, in mathematical terms these imply the following boundary conditions for equation (3.1):

$$v(x,t) = 0 \qquad x = 0 \tag{3.2}$$

$$\frac{\partial v(x,t)}{\partial x} = 0 \qquad x = 0 \tag{3.3}$$

$$\frac{\partial}{\partial x}\left(EI(x)\frac{\partial^2 v(x,t)}{x^2}\right) = 0 \qquad x = L \tag{3.4}$$

$$EI(x)\frac{\partial^2 v(x,t)}{x^2} = 0 \qquad x = L \tag{3.5}$$

It is assumed that the thickness of the beam changes linearly with x and the beam has a maximum thickness of t_b at the fixed end. Next, we define the thickness ratio c_t as the ratio of the thickness of the beam at the fixed end to thickness at the free one,

$$c_t = \frac{t_t}{t_b}$$

Thickness of the beam at any cross-section $t_c(x)$ is then formulated as:

$$t_c(x) = t_b\left(1 - (1 - c_t)\frac{x}{L}\right) \tag{3.6}$$

The change in the thickness results in the spatial dependence of the parameters $\rho(x)$ and $I(x)$ to the variable x, which is the distance measured from the fixed end. Assuming the beam has a rectangular cross-section these are expressed in the form:

$$\rho(x) = \rho_v w_b t_b\left(1 - (1 - c_t)\frac{x}{L}\right) \tag{3.7}$$

$$I(x) = \frac{1}{12} w_b t_b^3 \left(1 - (1 - c_t)\frac{x}{L}\right)^3 \tag{3.8}$$

where ρ_v is the mass density, w_b is the width and L is the length of the beam. The flexible beam used in this study is made of aluminum and its mechanical and geometric properties are given in table 3.1.

Table 3.1: Non-uniform flexible beam parameters

Description	**Value**	**Units**
Length, L	508.0	mm
Max. thickness, t_b	0.8	mm
Thickness ratio, c_t	0.75	-
Width, w_b	25.4	mm
Mass density, ρ_v	2710.0	$kg.m^{-3}$
Young's modulus, E	72.0x10^9	Pa

Substituting equations (3.7) and (3.8) into (3.1), taking the partial derivatives, simplifying and rearranging the terms, the partial differential equation of motion of the non-uniform flexible beam now becomes,

$$\begin{aligned}\frac{\partial^2 v(x,t)}{\partial t^2} = &- \frac{E t_b^2}{\rho_v}\left[\frac{(1-c_t)^2}{2L^2}\frac{\partial^2 v(x,t)}{\partial x^2} - \frac{(1-c_t)}{2L}\left(1-(1-c_t)\frac{x}{L}\right)\frac{\partial^3 v(x,t)}{\partial x^3}\right. \\ &\left. + \frac{1}{12}\left(1-(1-c_t)\frac{x}{L}\right)^2 \frac{\partial^4 v(x,t)}{\partial x^4}\right] + \frac{d(x,t)+u(x,t)}{\rho_v w_b t_b \left(1-(1-c_t)\frac{x}{L}\right)}\end{aligned} \tag{3.9}$$

3.2 Modeling the Structured Uncertainty

As mentioned in the previous chapter, the parameter dependence on the spatial variable x can be treated as parametric uncertainty. Let δ be the uncertain parameter defined as follows,

$$\delta = 2\frac{x}{L} - 1 \tag{3.10}$$

therefore $|\delta| \leq 1$ for all $x \in [0, L]$. Thus, equation (3.9) becomes

$$\frac{\partial^2 v(x,t)}{\partial t^2} = -\frac{Et_b^2}{\rho_v}\left[\frac{(1-c_t)^2}{2L^2}\frac{\partial^2 v(x,t)}{\partial x^2} - \frac{(1-c_t)}{4L}\left(1+c_t-(1-c_t)\delta\right)\frac{\partial^3 v(x,t)}{\partial x^3}\right.$$
$$\left. +\frac{1}{48}\left(1+c_t-(1-c_t)\delta\right)^2\frac{\partial^4 v(x,t)}{\partial x^4}\right] + \frac{2(d(x,t)+u(x,t))}{\rho_v w_b t_b\left(1+c_t-(1-c_t)\delta\right)} \quad (3.11)$$

We would like to describe the spatial dependence of the parameters on x in a linear fractional form. The LFT structure can be obtained by a simple manipulation, which is often referred as '*pulling out the* δ*'s*' [48]. First, a simple block diagram of equation (3.11) is drawn. Then, pseudo inputs and outputs p_i,q_i are introduced for each δ and all the uncertainties are removed from the diagram.

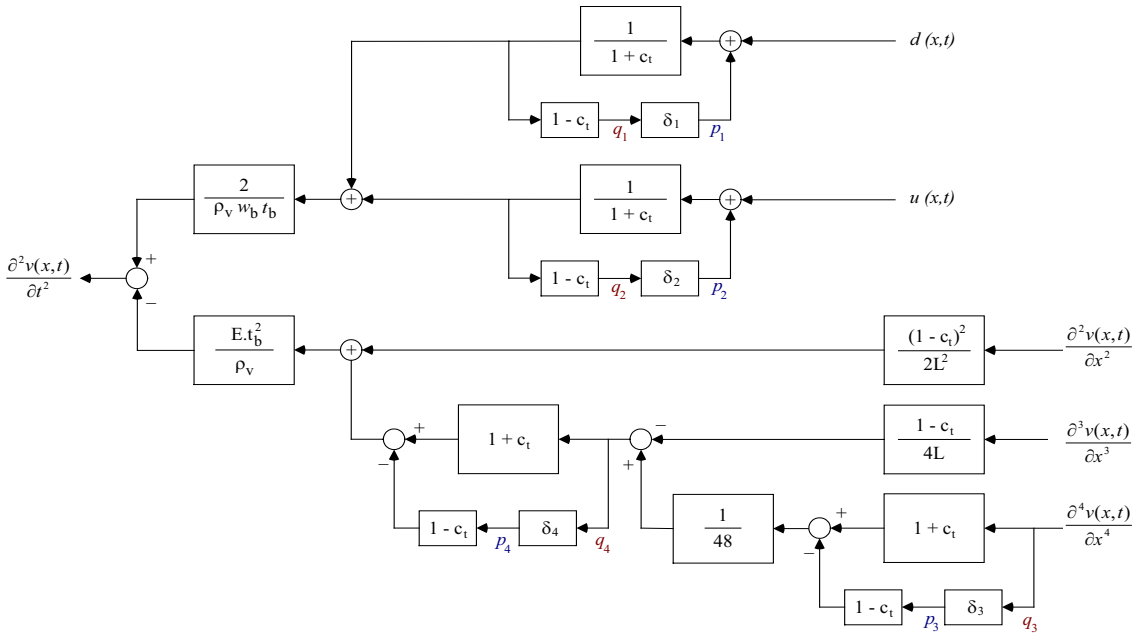


Figure 3.2: Block diagram representation of equation (3.11)

In this non-uniform flexible beam problem, there are four inputs $p = [p_1, p_2, p_3, p_4]^T$ and four outputs $q = [q_1, q_2, q_3, q_4]^T$ associated with the uncertainty block $\Delta = diag\,(\delta, \delta, \delta, \delta)$ as shown in figure 3.2. The signal relation between p_i and q_i can be easily determined as,

$$q_1 = \frac{1}{1+c_t} d(x,t) + \frac{1-c_t}{1+c_t} p_1 \tag{3.12}$$

$$q_2 = \frac{1}{1+c_t} u(x,t) + \frac{1-c_t}{1+c_t} p_2 \tag{3.13}$$

$$q_3 = \frac{\partial^4 v(x,t)}{\partial x^4} \tag{3.14}$$

$$q_4 = \frac{1+c_t}{48} \frac{\partial^4 v(x,t)}{\partial x^4} - \frac{1-c_t}{4L} \frac{\partial^3 v(x,t)}{\partial x^3} - \frac{1-c_t}{48} p_3 \tag{3.15}$$

3.3 Spatial Discretization

Structural elements like beams are continuous systems, and the exact solutions to the partial differential equations of their motion cannot be found except for some special cases. Hence, approximate solutions are often sought through spatial discretization. In this study, partial derivatives in spatial dimension are approximated with the central finite difference scheme, using N grid points separated by a length of h on the beam. Therefore, continuous flexible beam is discretized into $(N-1)$ subsystems (i.e. $L = (N-1)h$). However, in the spatial discretization, while the number of grids may vary, the locations of the displacement measurements and of the applied control forces are kept fixed.

$$\begin{aligned}
\frac{\partial^2 v(x,t)}{\partial t^2} = &-\frac{E t_b^2}{\rho_v} \left[\frac{(1-c_t)^2}{2L^2 h^2} \left(v(x-h,t) - 2v(x,t) + v(x+h,t) \right) \right. \\
&- \frac{(1-c_t^2)}{8Lh^3} \left(-v(x-2h,t) + 2v(x-h,t) - 2v(x+h,t) + v(x+2h,t) \right) \\
&+ \frac{(1+c_t)^2}{48h^4} \left(v(x-2h,t) - 4v(x-h,t) + 6v(x,t) - 4v(x+h,t) \right. \\
&\left. \left. + v(x+2h,t) \right) \right] + \frac{2(1-c_t)}{\rho_v w_b t_b (1+c_t)} (p_1 + p_2) + \frac{E t_b^2 (1-c_t)}{\rho_v} \left(\frac{(1+c_t)}{48} p_3 + p_4 \right) \\
&+ \frac{2}{\rho_v w_b t_b (1+c_t)} \left(d(x,t) + u(x,t) \right)
\end{aligned} \tag{3.16}$$

Next, the temporal differential operator $\mathbf{T}$, and spatial shift operator $\mathbf{S}$ defined in section 2.2 are substituted into equation (3.16). The equation of motion of the

non-uniform flexible beam takes its final form to be used in state-space model by replacing the continuous spatial variable x with its discrete counterpart s,

$$\begin{aligned}\mathbf{T}^2 v(s,t) = -\frac{Et_b^2}{\rho_v}\Bigg[&\frac{(1-c_t)^2}{2L^2h^2}\left(\mathbf{S}^{-1}+\mathbf{S}-2\right) - \frac{(1-c_t^2)}{8Lh^3}\left(-\mathbf{S}^{-2}+2\mathbf{S}^{-1}-2\mathbf{S}+\mathbf{S}^2\right)\\ &+\frac{(1+c_t)^2}{48h^4}\left(\mathbf{S}^{-2}-4\mathbf{S}^{-1}-4\mathbf{S}+\mathbf{S}^2+6\right)\Bigg] v(s,t) + \frac{2(1-c_t)}{\rho_v w_b t_b(1+c_t)}\left(p_1+p_2\right)\\ &+\frac{Et_b^2(1-c_t)}{\rho_v}\left(\frac{(1+c_t)}{48}p_3+p_4\right) + \frac{2\left(d(s,t)+u(s,t)\right)}{\rho_v w_b t_b(1+c_t)} \qquad (3.17)\end{aligned}$$

Finite difference approximation of the spatial derivatives are also applied to the boundary conditions at both ends. They will be used in the MATLAB simulation code.

At the fixed end,

$$\begin{aligned} v(x,t)|_{x=0} = 0 \quad &\Rightarrow \quad v(s,t) = 0, \quad s = 1\\ \left.\frac{\partial v(x,t)}{\partial x}\right|_{x=0} = 0 \quad &\Rightarrow \quad v(s-1,t) = v(s+1,t), \quad s = 1\end{aligned}$$

At the free end,

$$\begin{aligned} \left.EI(x)\frac{\partial^2 v(x,t)}{x^2}\right|_{x=L} = 0 \quad &\Rightarrow \quad v(s+1,t) = 2v(s,t) - v(s-1,t), \quad s = N\\ \left.\frac{\partial}{\partial x}\left(EI(x)\frac{\partial^2 v(x,t)}{x^2}\right)\right|_{x=L} = 0 \quad &\Rightarrow \quad v(s+2,t) = 4v(s,t) - 4v(s-1,t)\\ &\qquad\qquad + v(s-2,t), \quad s = N\end{aligned}$$

3.4 State-Space Model

The state-space formulation of the non-uniform flexible beam in the form of (2.8)-(2.9) will be described in this section. The displacement and velocity of each node are chosen as the temporal states x^t for each subsystem. The interconnection between each subsystem is captured by the spatial states x^s, which are defined as the

displacements of the neighboring nodes. In this regard,

$$x^t = \begin{bmatrix} v(s,t) \\ \mathbf{T}v(s,t) \end{bmatrix}, \quad x^s = \begin{bmatrix} \mathbf{S}^{-2}v(s,t) \\ \mathbf{S}^{-1}v(s,t) \\ \mathbf{S}^{2}v(s,t) \\ \mathbf{S}\ v(s,t) \end{bmatrix} \tag{3.18}$$

The external inputs to the system are noise $n(s,t)$ in the displacement measurements, distributed disturbance forces $d(s,t)$ and the distributed actuator forces $u(s,t)$. Next, the system outputs are defined as the error signals $z(s,t)$ and displacement measurements $y(s,t)$. The error signals are chosen to penalize the control input as $z_1(s,t)$ and the node displacement as $z_2(s,t)$.

Then, state-space formulation of the distributed model of the non-uniform flexible beam becomes,

$$\left[\begin{array}{cc|ccccc} w^t & w^s & q & z_1 & z_2 & y \end{array}\right]^T = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} \left[\begin{array}{cc|ccccc} x^t & x^s & p & n & d & u \end{array}\right]^T \tag{3.19}$$

where

$$\left[\begin{array}{c} x^t \\ x^s \\ \hline p \end{array}\right] = \left[\begin{array}{ccc|c} \mathbf{T}^{-1}I_2 & & & \\ & \mathbf{S}I_2 & & \\ & & \mathbf{S}^{-1}I_2 & \\ \hline & & & \delta I_4 \end{array}\right] \left[\begin{array}{c} w^t \\ w^s \\ \hline q(s,t) \end{array}\right]$$

and

$$\mathcal{A} = \left[\begin{array}{cc|cccc} 0 & 1 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & a_{23} & a_{24} & a_{25} & a_{26} \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right], \qquad \mathcal{B} = \left[\begin{array}{cccc|cc|c} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{21} & b_{22} & b_{23} & b_{24} & 0 & b_{26} & b_{27} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\mathcal{C} = \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ c_{31} & 0 & c_{33} & c_{34} & c_{35} & c_{36} \\ c_{41} & 0 & c_{43} & c_{44} & c_{45} & c_{46} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right], \qquad \mathcal{D} = \left[\begin{array}{cccc|cc|c} d_{11} & 0 & 0 & 0 & 0 & d_{16} & 0 \\ 0 & d_{22} & 0 & 0 & 0 & 0 & d_{27} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_{43} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right]$$

where non-zero coefficients are defined by:

$$\begin{aligned}
a_{21} &= \frac{Et_b^2}{\rho_v}\left(\frac{(1-c_t)^2}{L^2h^2} - \frac{(1+c_t)^2}{8h^4}\right) \\
a_{23} &= -\frac{Et_b^2}{8\rho_v}\left(\frac{(1-c_t^2)}{Lh^3} + \frac{(1+c_t)^2}{6h^4}\right) \\
a_{24} &= \frac{Et_b^2}{2\rho_v}\left(-\frac{(1-c_t)^2}{L^2h^2} + \frac{(1-c_t^2)}{2Lh^3} + \frac{(1+c_t)^2}{6h^4}\right) \\
a_{25} &= \frac{Et_b^2}{8\rho_v}\left(\frac{(1-c_t^2)}{Lh^3} - \frac{(1+c_t)^2}{6h^4}\right) \\
a_{26} &= -\frac{Et_b^2}{2\rho_v}\left(\frac{(1-c_t)^2}{L^2h^2} + \frac{(1-c_t^2)}{2Lh^3} - \frac{(1+c_t)^2}{6h^4}\right) \\
b_{21} &= b_{22} = \frac{2(1-c_t)}{\rho_v w_b t_b (1+c_t)} \\
b_{23} &= \frac{Et_b^2(1-c_t^2)}{48\rho_v} \\
b_{24} &= \frac{Et_b^2(1-c_t)}{\rho_v} \\
b_{26} &= b_{27} = \frac{2}{\rho_v w_b t_b (1+c_t)} \\
c_{31} &= \frac{6}{h^4} \qquad c_{33} = c_{35} = \frac{1}{h^4} \qquad c_{34} = c_{36} = -\frac{4}{h^4} \\
c_{41} &= \frac{(1+c_t)}{8h^4} \\
c_{43} &= \frac{(1+c_t)}{48h^4} + \frac{(1-c_t)}{8Lh^3}
\end{aligned}$$

$$
\begin{aligned}
c_{44} &= -\frac{(1+c_t)}{12h^4} - \frac{(1-c_t)}{4Lh^3} \\
c_{45} &= \frac{(1+c_t)}{48h^4} - \frac{(1-c_t)}{8Lh^3} \\
c_{46} &= -\frac{(1+c_t)}{12h^4} + \frac{(1-c_t)}{4Lh^3} \\
d_{11} &= d_{22} = \frac{(1-c_t)}{(1+c_t)} \\
d_{16} &= d_{27} = \frac{1}{(1+c_t)} \\
d_{43} &= -\frac{(1-c_t)}{48}
\end{aligned}
$$

3.5 Modal Analysis and Beam Model Validation

So far, the state-space formulation of the non-uniform flexible beam has been developed based on the governing partial differential equation of motion. Before proceeding further with the controller design, validity of the developed distributed model will be discussed in this section.

Note that the developed model is only an approximation of the dynamics of the actual system because of the spatial discretization. It is necessary to know how many grid points are sufficient in order to capture the beam dynamics accurately. Therefore, a comparison study between the exact solution and the developed distributed model is needed.

Unfortunately, the exact solution to the equation of motion of a non-uniform flexible beam is not available. Then, another approximate solution, which is known to give an accurate description of the actual system, must be used for comparison purposes. Therefore, a finite element model of the non-uniform beam will be compared with the distributed model.

Now, the question is, how does one conclude that the FEM actually represents the dynamics of the non-uniform flexible beam? In order to verify the FEM, one can analyze the modes of the uniform beam problem through modal analysis approach.

Uniform Beam Case

First, a uniform flexible cantilever beam is considered. Various finite element models were built in ANSYS using different number of 2D rectangular elements, and the first five natural frequencies with their corresponding mode shapes were extracted. The resulting natural frequencies are given in table 3.2 along with the exact ones [23]. It is clearly seen that as the number of elements in FEM increases, the approximated natural frequencies converges to the actual ones, from above.

Table 3.2: First five natural frequencies of a uniform flexible beam

Method	ω_1 (rad/s)	ω_2 (rad/s)	ω_3 (rad/s)	ω_4 (rad/s)	ω_5 (rad/s)
FEM, 100-Element	16.443	101.914	284.997	558.648	924.205
FEM, 500-Element	16.443	101.890	284.804	557.887	922.094
FEM, 1000-Element	16.443	101.889	284.798	557.863	922.028
FEM, 5000-Element	16.441	101.889	284.796	557.856	922.007
Analytical	16.218	101.638	284.589	557.682	922.075

In the distributed model case, by concatenating the distributed model into a single lumped model, one can calculate the natural frequencies of the given uniform beam. If the subsystems are modeled correctly, those should also converge to the exact values qradually. This will be demonstrated from the simulation plots later on.

The free response of the uniform beam is used for modal analysis. Three sets of initial conditions are selected such that each condition will separately excite one of the first three modes of the beam. In each case, the maximum tip deflection was normalized to $0.01m$. All simulations were obtained using MATLAB. Figure 3.3 is the plot showing the beam vibrating in its first mode. The response based on the distributed model closely follows the analytical solution. On the other hand, finite element prediction has slightly higher frequency although the amplitude is accurate. Responses to the second and third mode excitations are plotted in figure 3.4 and figure 3.5, respectively. Again, the distributed model simulation has a good match in frequency with the analytical solution, however, its predicted deflection is slightly

larger. It is observed that as the number of nodes along the beam length increases, the amount of this error decreases and finally converges to the analytical solution. It is also seen that the second and third mode responses from finite element model is inferior to the distributed model solutions even for large number of finite elements are used.

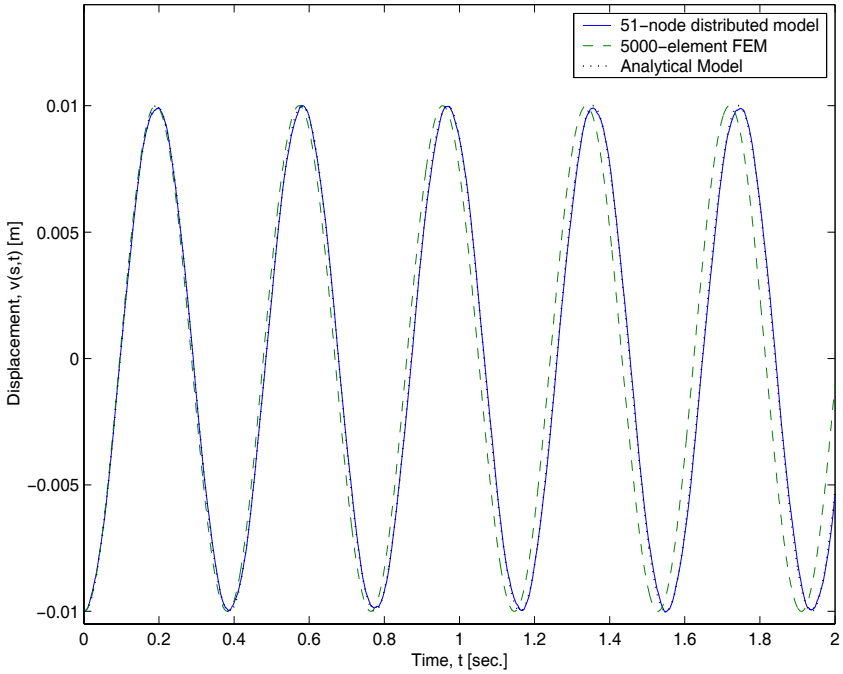


Figure 3.3: Deflection of the free end of the uniform beam, 1^{st} mode excited

Another set of initial conditions is tested to get a combined mode response of the uniform beam. In this study, it is assumed that the uniform beam is initially deflected by a point force applied at the free end to give a tip deflection of $0.01m$. The response of the flexible beam to this set of initial conditions is plotted in figure 3.6. Again, both approximate solutions are very close to the analytical solution.

As a result, it can be concluded that, a 5000-element finite element model is an adequate description of the actual dynamics of the flexible cantilever beam. In addition, it is observed that the distributed model also approximates the behavior of the

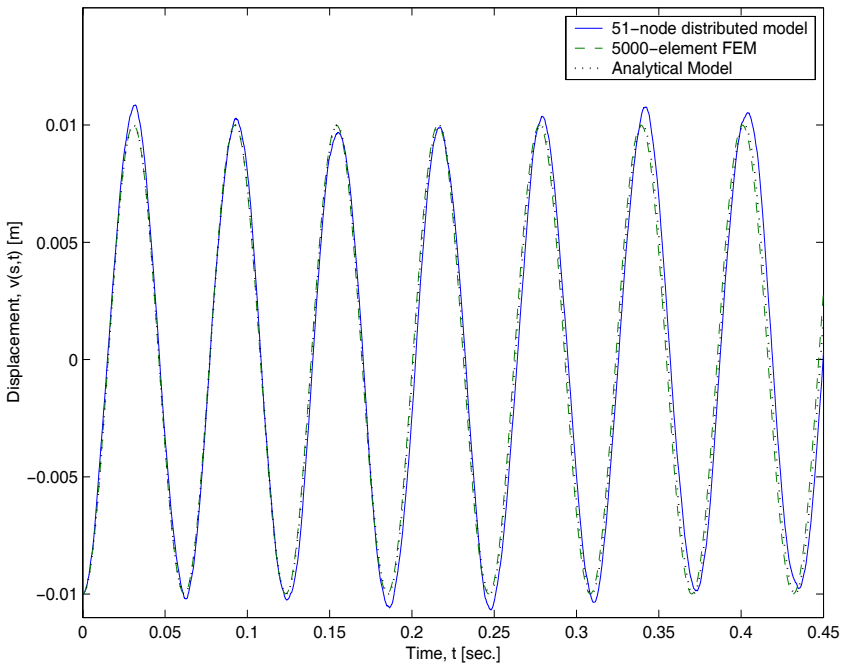


Figure 3.4: Deflection of the free end of the uniform beam, 2^{nd} mode excited

uniform flexible beam accurately while using considerably less number of states. Furthermore, finite element model and distributed model converge to the actual system from opposite sides, i.e., finite element model always predicts the frequency of vibration slightly higher while the frequency of the distributed model is slightly smaller than the actual one.

Non-uniform Beam Case

In the case of the non-uniform beam, however, the finite element model is assumed to be the true model of the dynamic response of the beam due to the lack of the analytical solution. Results of the previous analysis on the uniform beam indicates that this is a reasonable assumption. Therefore, a finite element model composed of 5000 2D quadrilateral elements was built in ANSYS. Similar to the uniform beam analysis, the first five natural frequencies and associated mode shapes were extracted. The resulting natural frequencies of the given non-uniform beam are shown in table 3.3.

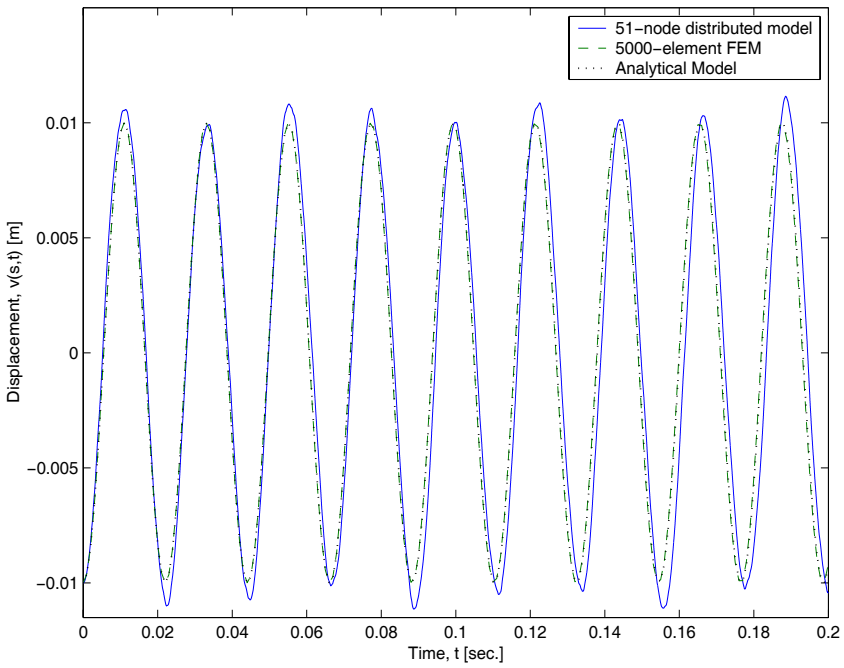


Figure 3.5: Deflection of the free end of the uniform beam, 3^{rd} mode excited

The effect of the non-uniformity of the thickness on the beam dynamics can be seen with the change in the natural frequencies, although the thickness variation is very small.

Table 3.3: First five natural frequencies of a non-uniform flexible beam

Method	ω_1 (rad/s)	ω_2 (rad/s)	ω_3 (rad/s)	ω_4 (rad/s)	ω_5 (rad/s)
FEM, 100-Element	17.015	93.736	253.054	491.320	809.568
FEM, 500-Element	17.015	93.714	252.886	490.658	807.733
FEM, 1000-Element	17.015	93.714	252.880	490.637	807.676
FEM, 5000-Element	17.015	93.714	252.879	490.630	807.658

Time domain simulations of the free response of the non-uniform beam to four other sets of initial conditions are presented below. The first three of these sets excite one mode of vibration only, and the fourth one is the same set that was used in the

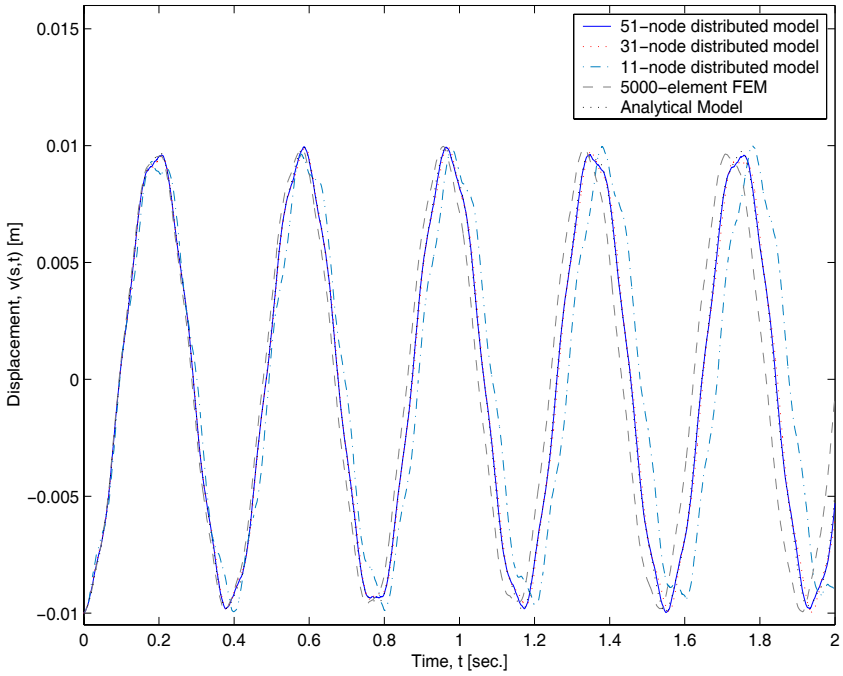


Figure 3.6: Deflection of the free end of the uniform beam, combined modes excited

combined mode response of the uniform beam. Figures 3.7, 3.8 and 3.9 are the plots of the deflection of the free end of the non-uniform beam vibrating in its first, second and third mode of vibration, respectively. From these three figures, it is clearly seen that a 51-node distributed model matches with the finite element model quite well both in terms of frequency and amplitude. Moreover, from the results of the uniform beam analysis, it can be concluded that the actual response of the non-uniform beam would lie in between the distributed model curve and the finite element model curve.

The response to the combined mode excitation is shown in figure 3.10. Distributed models with five different number of nodes are plotted. It is observed that the response of any distributed model over 30 nodes is very close to the response of the finite element model. As a result, it can be concluded that a 31-node distributed model is sufficiently accurate to approximate the dynamics of the non-uniform flexible beam.

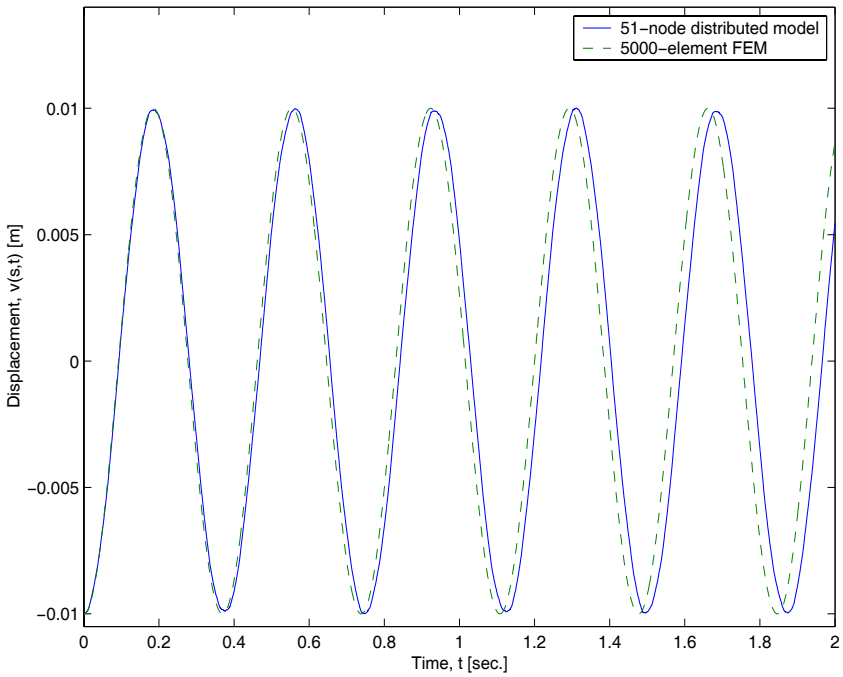


Figure 3.7: Deflection of the free end of the non-uniform beam, 1^{st} mode excited

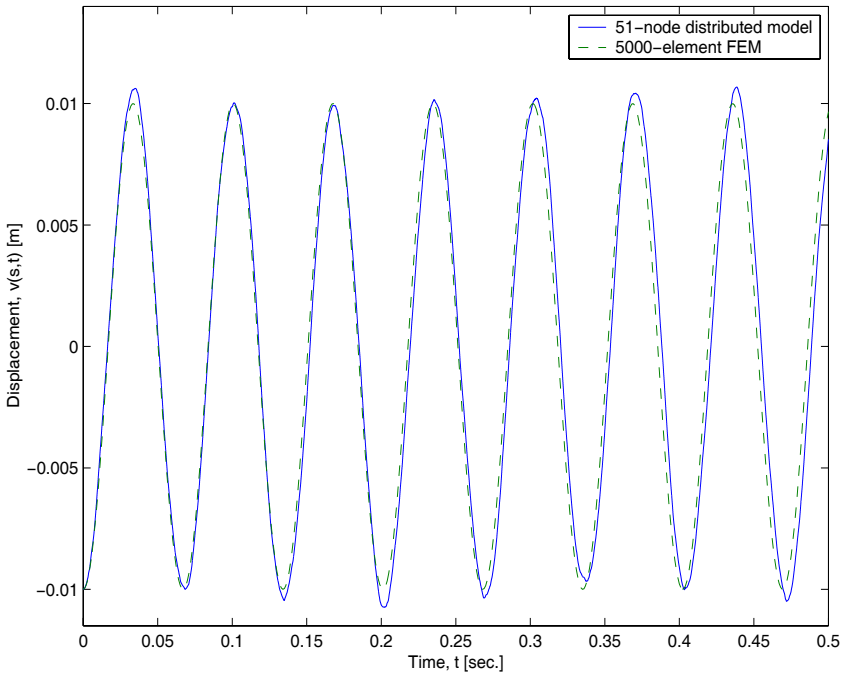


Figure 3.8: Deflection of the free end of the non-uniform beam, 2^{nd} mode excited

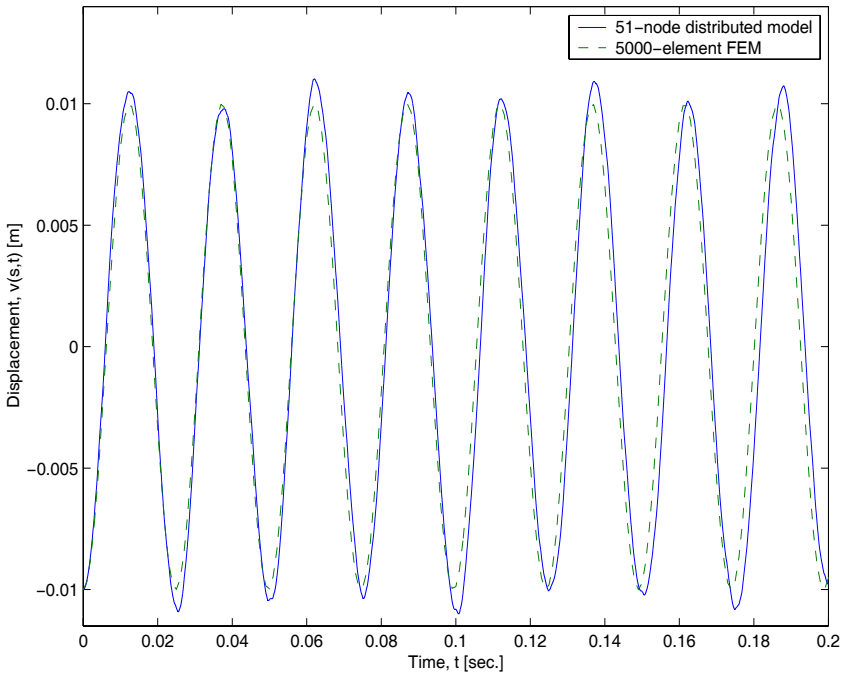


Figure 3.9: Deflection of the free end of the non-uniform beam, 3^{rd} mode excited

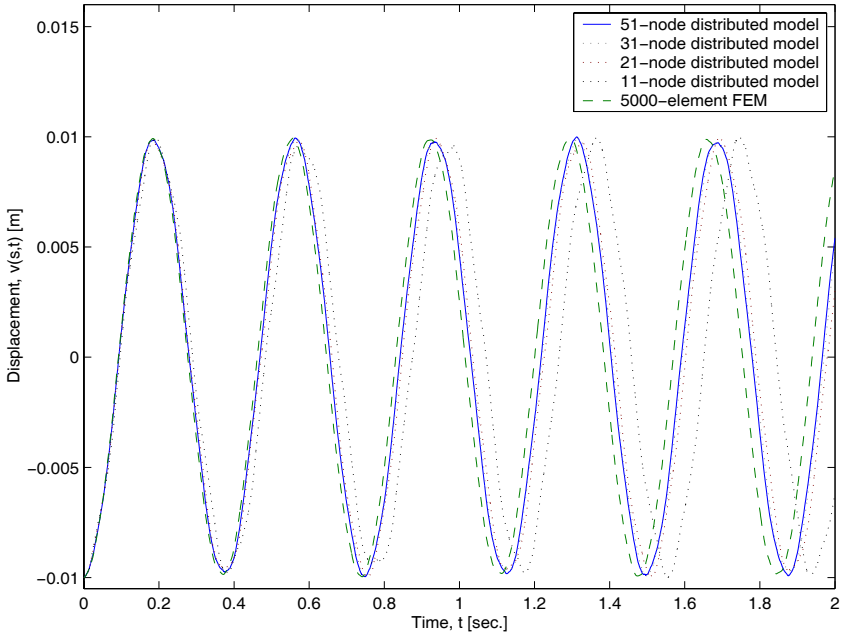


Figure 3.10: Deflection of the free end of the non-uniform beam, combined modes excited

Chapter 4

Controller Design and Numerical Simulations

The distributed state-space model of the non-uniform flexible cantilever beam was developed in the previous chapter. Using this model and the distributed parameter-dependent control design approach introduced in section 2.5, we now focus on the design of a controller for the given flexible structure.

First, the open-loop system will be redefined with the introduction of the weighting functions, and their effects on the controller synthesis will be discussed. Next, the results of the controller synthesis will be presented. The chapter will be concluded with the numerical time domain simulations of the closed-loop system in order to observe the performance of the distributed parameter-dependent controller.

4.1 Weighting Function Specification

Weighting functions are an essential part of any control design, especially in the optimal control theory. They are used to reflect the design objectives and knowledge on the characteristics of sensor noise or disturbances. For instance, the frequency content of the sensor noise or the disturbances may be captured. Moreover, certain closed-loop transfer functions may be shaped according to the design requirements, and restrictions on the control or actuator signals may be implied. So it is clear that

choosing the appropriate weighting functions is a very important step in the controller design process.

On the other hand, selection of the most appropriate weighting functions is not as trivial as it may seem. It is very difficult to express a general formula for the weights to be used. Usually, it involves many iterations until the best compromise between the conflicting objectives is achieved.

In this particular non-uniform flexible beam problem, four weighting functions, namely W_1, W_2, W_3 and W_4, were used to penalize the control effort z_1, disturbance d, node displacement z_2 and sensor noise n, respectively. A block diagram of the open-loop system (3.19) with the inclusion of the weighting functions is shown in figure 4.1.

Let $\mathbf{G}$ represent the system (3.19) partitioned to its individual transfer matrices as,

$$\begin{bmatrix} q \\ z_1 \\ z_2 \\ y \end{bmatrix} = \mathbf{G} \begin{bmatrix} p \\ n \\ d \\ u \end{bmatrix} \tag{4.1}$$

then, the weighted open-loop system $\mathbf{G_W}$ in figure 4.1 becomes,

$$\mathbf{G_W} = \begin{bmatrix} I_4 & & & \\ & W_1 & & \\ & & W_3 & \\ & & & I_1 \end{bmatrix} \mathbf{G} \begin{bmatrix} I_4 & & & \\ & W_4 & & \\ & & W_2 & \\ & & & I_1 \end{bmatrix} \tag{4.2}$$

where individual single-input single-output weighting functions are defined with the following transfer functions. Note that below the variable s stands for the Laplace variable and it should not be confused with the spatial independent variable s which is used to index the subsystems of the distributed model.

$$W_1 = 0.02\frac{5s+1}{s+2}, \qquad W_2 = 0.001$$
$$W_3 = \frac{s+2}{2s+1}, \qquad W_4 = 0.001$$

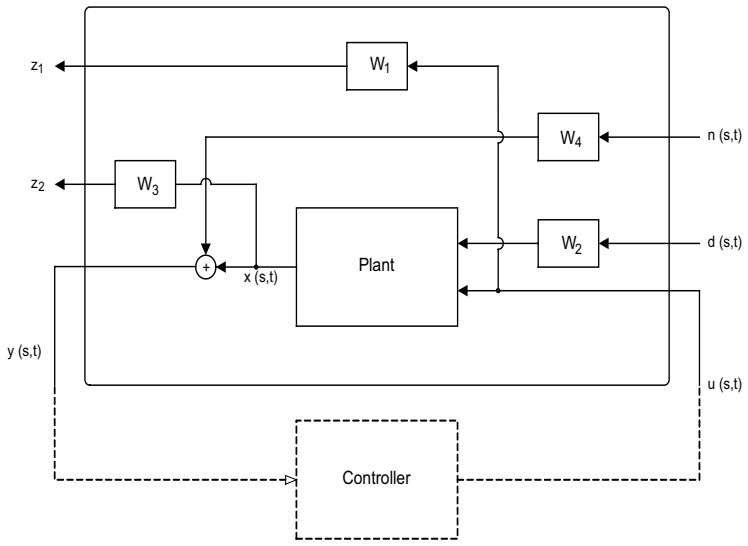


Figure 4.1: Weighted open-loop system for controller synthesis

4.2 Controller Synthesis

The *Multidimensional System Toolbox* (or MD Toolbox) [8] developed for MATLAB 5.x was used for the synthesis of the distributed parameter-dependent controller for the non-uniform flexible beam. It was specifically designed for constructing models, control design and analysis of uncertain and spatially distributed systems. However, most functions were designed for spatially invariant distributed systems. Therefore, some modifications were needed in order to cope with the parameter-dependent nature of the non-uniform flexible beam problem.

It is assumed that 10 actuators and 10 sensors are uniformly placed at the pre-specified locations on the beam. Therefore, an 11-node distributed model was used for the synthesis of the controller.

One of the goals of this study is to evaluate the effect of the beam parameter variation on the performance of the designed controller. For this reason, controller synthesis was performed for various thickness ratios, including $c_t = 1$ for the uniform beam case, using the same weighted system $\mathbf{G_W}$. The resulting performance values,

γ, are given in table 4.1.

Table 4.1: Weighted performance for various uncertainty level

Thickness ratio, c_t	**Upper Bound to** $\mathcal{L}_2$ **gain,** γ	**CPU Time** $[sec.]$
1.00	27.094	10.085
0.99	27.211	4.957
0.90	28.379	4.617
0.75	30.444	6.139
0.50	34.328	4.847
0.25	38.924	4.647

Recall that the uncertainty is directly related to the parameter variation, or change in thickness ratio. It is then clear from the table that as the thickness ratio decreases (i.e. the uncertainty increases), the performance level quantified by $\mathcal{L}_2$ gain γ decreases.

As a result of the control design, the following state-space realization (in the form of equation (2.12)) of a distributed parameter-dependent controller $\mathbf{K}$ for the non-uniform flexible beam with a thickness ratio of $c_t = 0.75$ is obtained.

$$A_k^{tt} = \begin{bmatrix} -3.634 \times 10^{7} & -1.160 \times 10^{4} & -3.247 \times 10^{1} & 6.674 \times 10^{-2} \\ -5.539 \times 10^{10} & -1.769 \times 10^{7} & -4.948 \times 10^{4} & -1.018 \times 10^{2} \\ 8.604 \times 10^{4} & -1.518 \times 10^{2} & -1.882 & -1.061 \times 10^{-3} \\ 6.167 \times 10^{3} & 1.9698 & 5.510 \times 10^{-3} & -4.999 \times 10^{-1} \end{bmatrix}$$

$$A_k^{st} = \begin{bmatrix} -5.834 \times 10^{-1} & 0 & 0 & 0 \\ -9.221 \times 10^{-1} & 0 & 0 & 0 \\ 4.30 \times 10^{-1} & 0 & 0 & 0 \\ 1.039 & 0 & 0 & 0 \end{bmatrix}$$

$$A_k^{ts} = \begin{bmatrix} -3.164 \times 10^6 & -2.678 \times 10^6 & -2.940 \times 10^5 & 4.098 \times 10^6 \\ -4.823 \times 10^9 & -4.082 \times 10^9 & -4.481 \times 10^8 & 6.247 \times 10^9 \\ -1.972 \times 10^3 & -1.684 \times 10^3 & -2.044 \times 10^2 & 2.657 \times 10^3 \\ 5.369 \times 10^2 & 4.545 \times 10^2 & 4.989 \times 10^1 & -6.956 \times 10^2 \end{bmatrix}$$

$$A_k^{ss} = \begin{bmatrix} -4.504 \times 10^{-1} & -6.483 \times 10^{-1} & 0 & 0 \\ 0 & 2.274 \times 10^{-1} & 0 & 0 \\ 0 & 0 & -1.898 \times 10^{-1} & -4.738 \times 10^{-1} \\ 0 & 0 & 0 & -4.278 \times 10^{-1} \end{bmatrix}$$

$$B_{0,k} = \begin{bmatrix} -2.940 & -4.423 \times 10^{-2} & -7.542 \times 10^2 & -9.462 \times 10^2 \\ -4.482 \times 10^3 & -6.742 \times 10^1 & -1.150 \times 10^6 & -1.442 \times 10^6 \\ 2.789 \times 10^{-3} & 0 & -4.043 \times 10^{-1} & -1.124 \times 10^{-1} \\ 0 & 0 & 1.280 \times 10^{-1} & 1.606 \times 10^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_{1,k} = \begin{bmatrix} 3.335 \times 10^7 & 5.083 \times 10^{10} & -1.299 & -5.659 \times 10^3 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_{0,k}^t = \begin{bmatrix} -8.604 \times 10^4 & -1.518 \times 10^2 & 1.175 \times 10^{-1} & -1.061 \times 10^{-3} \\ 3.144 \times 10^{-1} & 1.688 \times 10^{-3} & 0 & 0 \\ 2.725 \times 10^7 & 4.808 \times 10^4 & -3.723 \times 10^1 & -3.361 \times 10^{-1} \\ -7.026 \times 10^2 & 3.058 \times 10^{-1} & 0 & 0 \end{bmatrix}$$

$$C_{0,k}^s = \begin{bmatrix} -1.972 \times 10^3 & -1.684 \times 10^3 & -2.044 \times 10^2 & 2.658 \times 10^3 \\ -2.191 \times 10^{-1} & -1.654 \times 10^{-1} & 1.217 \times 10^{-2} & 1.335 \times 10^{-1} \\ 6.246 \times 10^5 & 5.334 \times 10^5 & 6.475 \times 10^4 & -8.418 \times 10^5 \\ -2.299 \times 10^2 & -1.993 \times 10^2 & -2.849 \times 10^1 & 3.315 \times 10^2 \end{bmatrix}$$

$$C_{1,k}^t = \begin{bmatrix} -7.666 \times 10^2 & 3.929 \times 10^{-1} & 1.037 \times 10^{-3} & 0 \end{bmatrix}$$

$$C_{1,k}^{t} = \begin{bmatrix} -2.728 \times 10^{2} & -2.315 \times 10^{2} & 2.599 \times 10^{1} & 3.570 \times 10^{2} \end{bmatrix}$$

$$\mathcal{D}_k = \begin{bmatrix} 0 & 2.789 \times 10^{1} & 0 & 4.044 \times 10^{1} & -1.124 \times 10^{1} \\ 0 & 1.429 \times 10^{1} & 0 & 0 & 0 \\ 0 & -8.834 \times 10^{1} & 1.296 \times 10^{1} & 1.281 \times 10^{1} & 3.560 \times 10^{1} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -7.841 \times 10^{1} & -2.179 \times 10^{1} \end{bmatrix}$$

Now, the advantages of the LFT framework, with which the spatial parameter variations of the non-uniform flexible beam was modeled, becomes obvious. Parameter δ is uniquely defined for each node, therefore, the individual controller gains can easily be computed with the lower LFT interconnection of $\mathbf{K}$ and Δ. In other words, by designing a single parameter-dependent controller, a desired number of local controllers can be obtained depending on the level of spatial discretization. For this particular non-uniform flexible beam problem, the grid used in the controller synthesis was 11.

4.3 Numerical Simulations

Having designed the distributed parameter-dependent controller, the closed-loop system $\mathbf{M}$ can now be obtained by,

$$\mathbf{M} = \mathcal{F}_l\left(\mathcal{F}_u\left(\mathbf{G}, \Delta\right), \mathcal{F}_l\left(\mathbf{K}, \Delta_k\right)\right)$$

The results of the modal analysis in section 3.5 showed that an 11-node distributed model cannot be used to approximate the dynamics of the flexible beam accurately. However, a 31-node plant model can serve as a very good approximation. Therefore, each subsystem that has actuation and sensing capabilities is divided into three other subsystems leading to the control architecture shown in figure 4.2 below.

In this section, the time domain simulations of the closed-loop system are provided for four different cases. All the simulations were performed using MATLAB MD Toolbox functions, and the code is provided in the appendix.

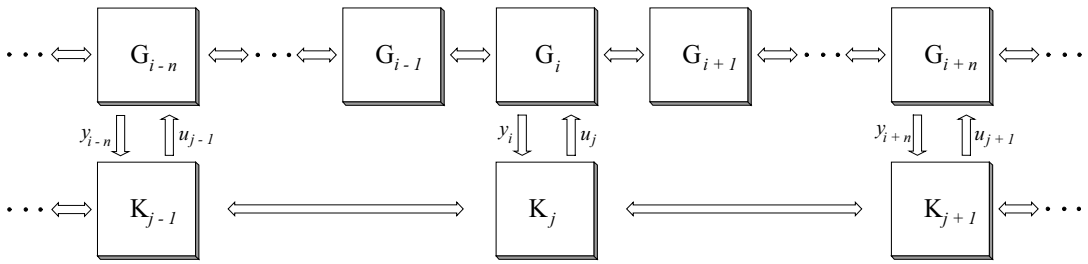


Figure 4.2: Modified distributed control architecture

Case 1:

To begin with, it was assumed that the non-uniform flexible beam was initially deflected by a point load applied at the free end to cause a tip deflection of $0.01m$. Then, the force was assumed to be removed instantly at $t = 0sec$, releasing the beam from rest. At this point no disturbance forces were acting on the beam, neither did there exist any sensor noise.

The plot of the open-loop response has already been given in figure 3.10. Figures 4.3 and 4.4 show the controlled response of the displacement of the free end of the beam and the control input at the corresponding node, respectively. The closed-loop response is fast and non-oscillatory. The initial jump in the control effort may seem unrealistic due to the fact that actuators cannot deliver that much of force in such a short period of time. However, the actuator dynamics was not included in the model, which might have eliminated such fast dynamics. The displacement of the whole beam is plotted for the entire time interval in figure 4.5.

Case 2:

For the second case, same initial conditions were used, however, in this simulation distributed disturbance forces were acting on the beam and sensor noise was added to the displacement measurements. Both disturbance forces and sensor noise were stepwise constant random functions of time. Figures 4.6.a and 4.6.b show the disturbance

input and noise at the free end of the beam.

The deflection of the free end is plotted in figure 4.7. The effect of the noise and disturbance is visible in the plot. Nevertheless, the maximum amplitude of vibration is kept less than $1mm$. The sensor noise plays an important role on this results. The amplitude of the noise exceeds $0.2mm$ at some point, which corresponds to a 20% error in measurement. Therefore, we can conclude that the distributed controller works satisfactorily while rejecting disturbance and noise.

Case 3:

One of the main advantages of the distributed control is its reliability in case of a failure of an actuator or a sensor. The centralized control schemes are more sensitive to the actuator/sensor failures, because a failure may lead to the loss of observability or controllability of the whole system, and may cause devastating results. In the distributed control, however, a possible failure of the actuators/sensors at a location can be compensated by the others owing to the inherent decentralization. Thus, an actuator failure case is presented next.

It was assumed that, the actuator at an arbitrary location, say node 19 (corresponding the actuator-7 in 31-node plant model), had failed. That is, the control input $u(s,t)$ at that location is equal to zero. To show the effect of the failure only, the disturbances and sensor noise were neglected, so that the results could be compared to those of case 1. The displacement of the free end of the beam is shown in figure 4.11, and the plot of the displacement of the node, at which the failure occurred, is also given in figure 4.10. It is clear that, even though the failure affected the performance of the distributed controller negatively, the system is still stable and vibration is damped out in a reasonable period of time.

Case 4:

On the other hand, the case of a sensor failure is a different problem. Since the entire control design is based on the availability of the distributed output measurements, a sensor failure can be considered as a more dangerous situation. In case of an

actuator failure, where one of the actuators cannot provide the required control input to the system, the displacement measurements are still available for the others to compensate for the lost one. Unfortunately, a wrong sensor measurement may cause the sharing of wrong information.

To simulate the sensor failure case, the sensor at the same node was assumed malfunctioning while all actuators were problem-free. The simulation results are shown in figures 4.13 and 4.14.

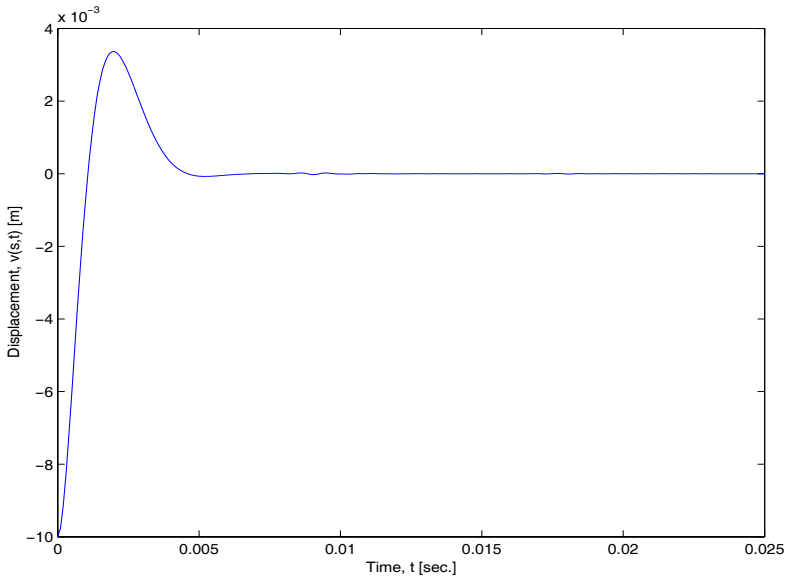


Figure 4.3: **Case 1**, Displacement of the free end, no disturbance

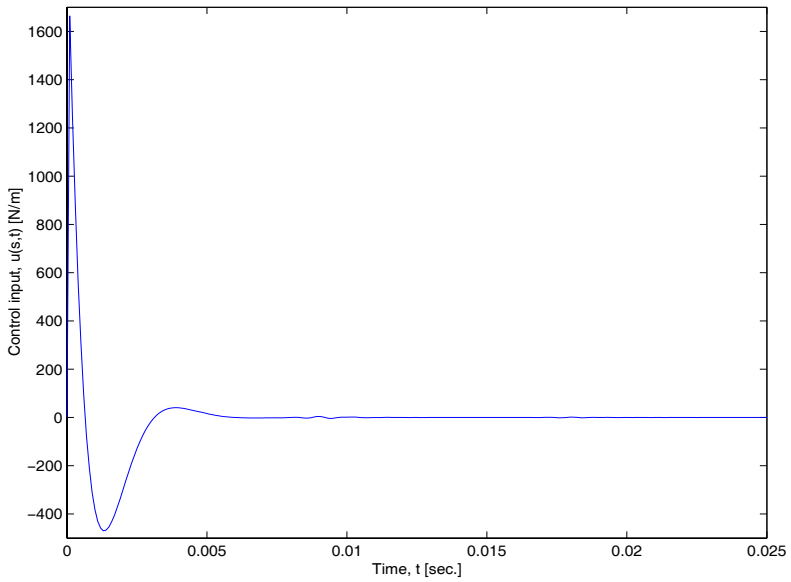


Figure 4.4: **Case 1**, Control input at the free end, no disturbance

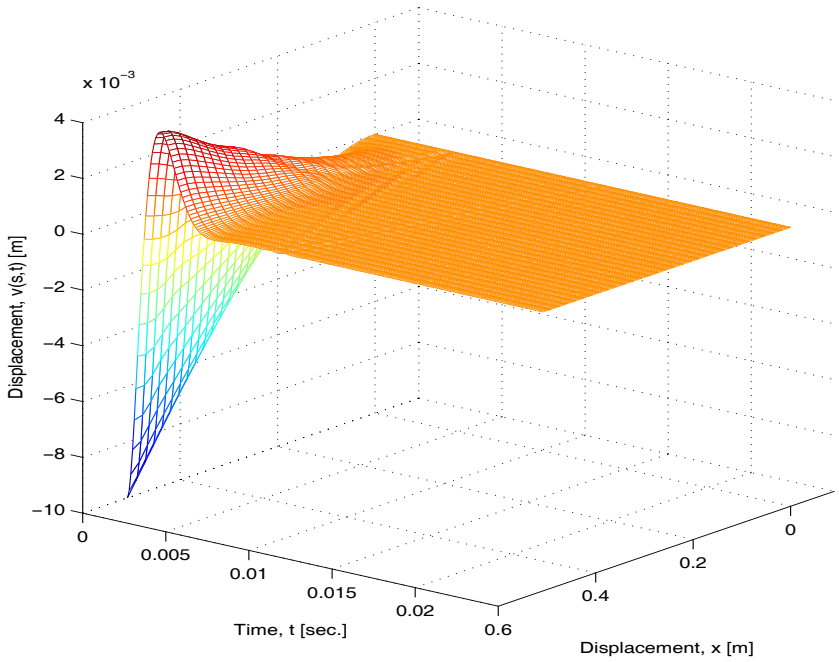
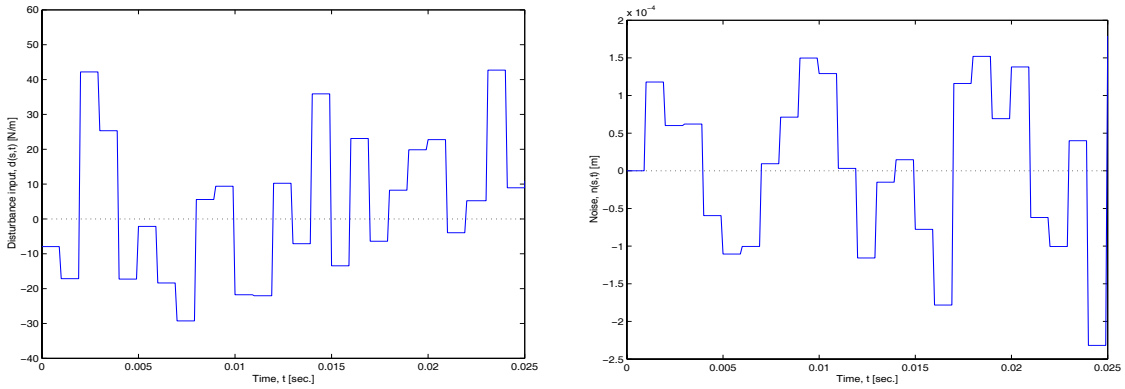


Figure 4.5: **Case 1**, Displacement of the non-uniform beam, no disturbance



(a). Disturbance (b). Noise

Figure 4.6: Disturbance Input and Noise at the free end

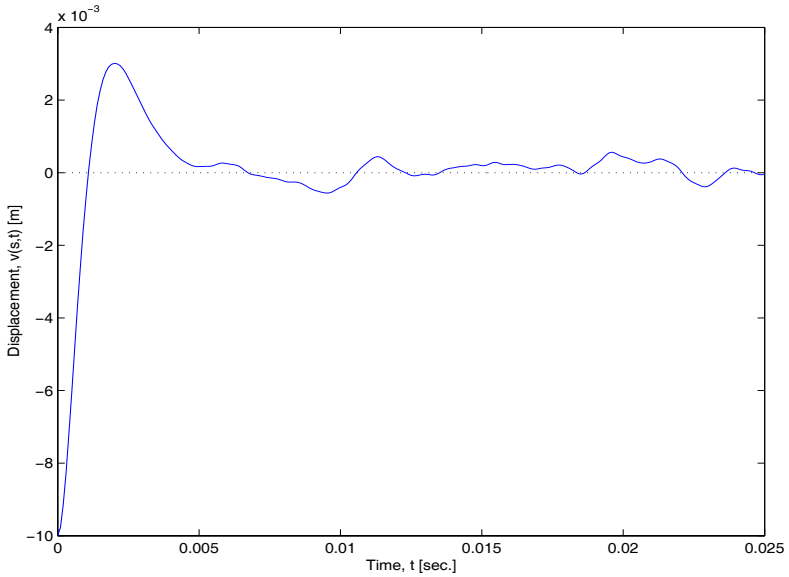


Figure 4.7: **Case 2**, Displacement of the free end, disturbance rejection

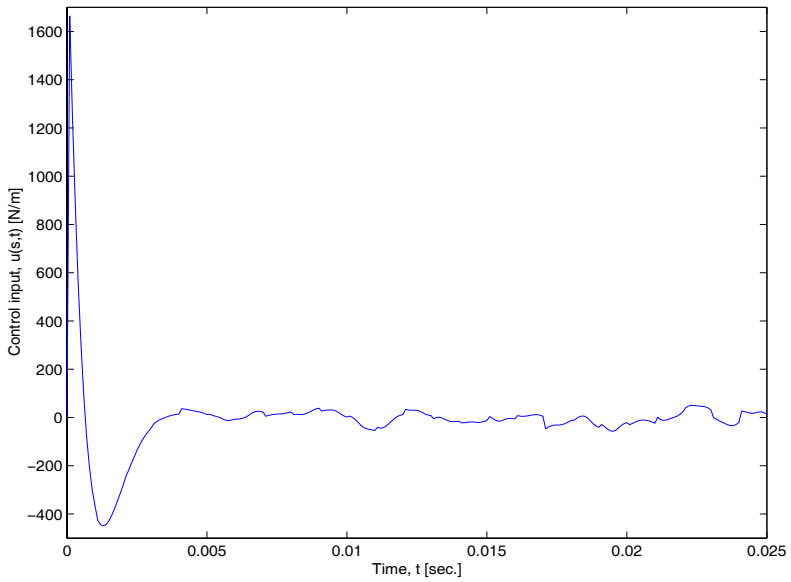


Figure 4.8: **Case 2**, Control input at the free end, disturbance rejection

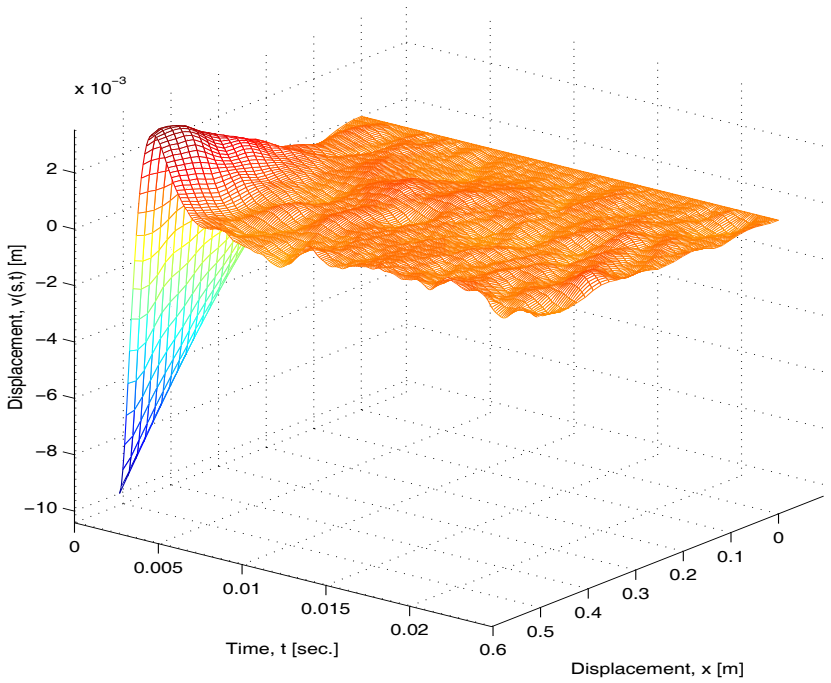


Figure 4.9: **Case 2**, Displacement of the non-uniform beam, disturbance rejection

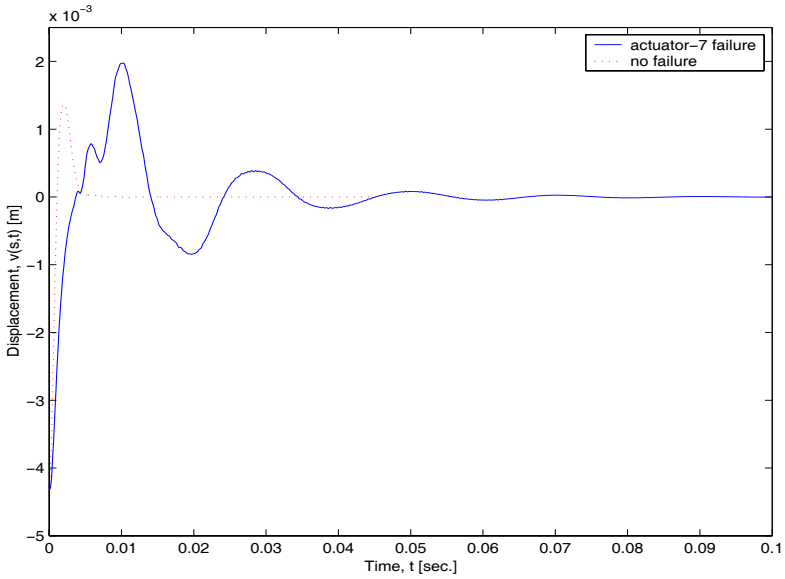


Figure 4.10: **Case 3**, Displacement of the beam at malfunctioning actuator location

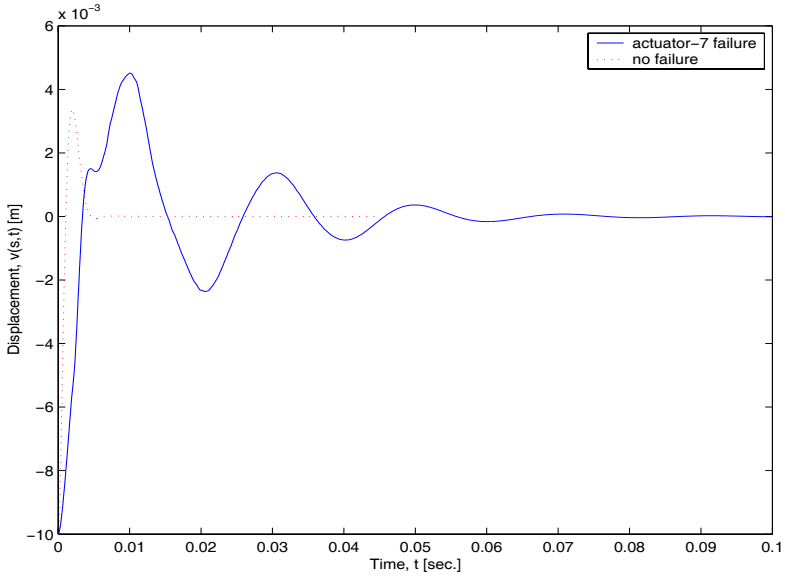


Figure 4.11: **Case 3**, Displacement of the free end

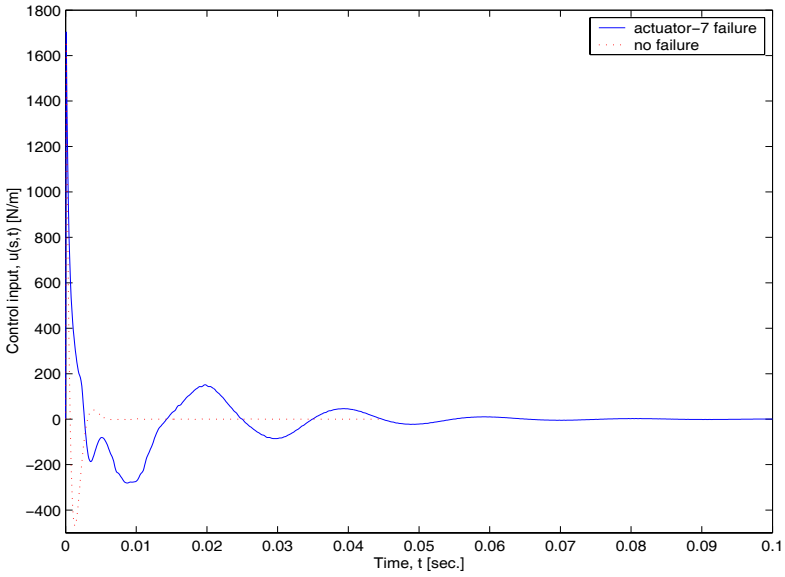


Figure 4.12: **Case 3**, Control input at the free end

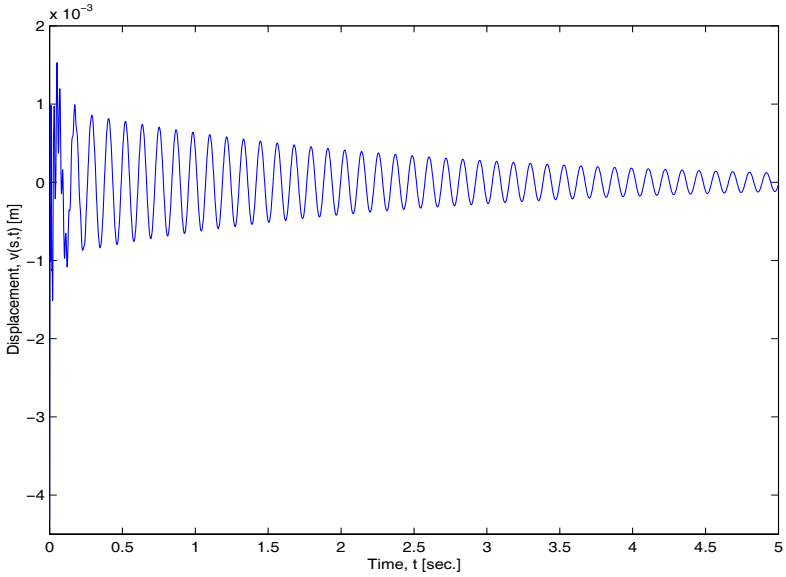


Figure 4.13: **Case 4**, Displacement of the beam at of malfunctioning sensor location

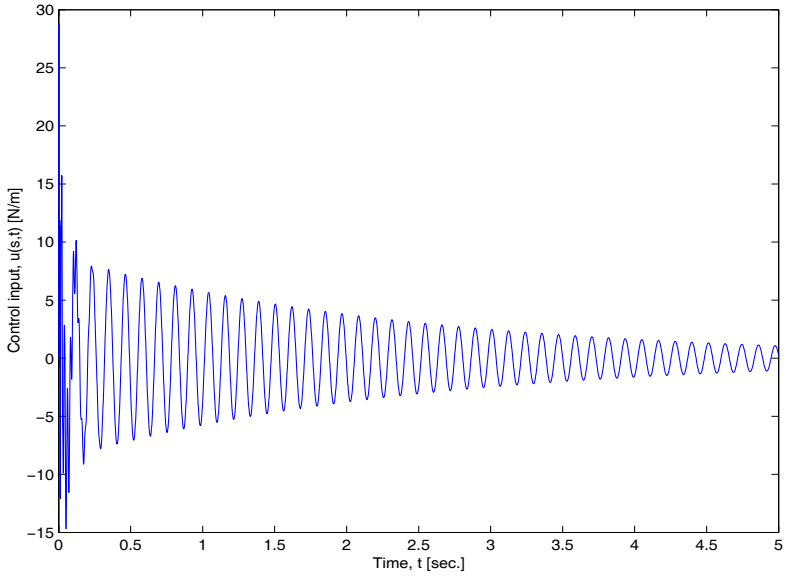


Figure 4.14: **Case 4**, Control input at malfunctioning sensor location

Chapter 5

Conclusions

In this thesis, distributed parameter-dependent control of spatially varying distributed parameter systems was studied with the application to the active vibration suppression of non-uniform flexible structures. The main goal of this research was to develop a distributed model of a flexible structure, and based on that distributed model, design a distributed control system with an array of actuators and sensors to damp out any vibration of the flexible structure. These objectives have been successfully achieved.

A distributed state-space model of a non-uniform flexible beam was developed based on the undamped Bernoulli-Euler beam model. Due to the presence of the non-uniform geometry of the particular flexible beam, the usual assumptions of constant coefficients could not be made. Then, the parameter dependence of the coefficients on the spatial independent variable was modeled as parametric uncertainty so that the desired LFT framework for LPV systems could be employed. The partial differential equation of the beam's transverse motion was discretized in the spatial dimension by approximating the partial derivatives in that dimension using a central finite difference scheme. Thus, the flexible beam was divided into smaller subsystems. However, due to the high orders of the partial derivatives in spatial dimension, the interconnection between these subsystems turned out to be very strong.

A distributed parameter-dependent output-feedback controller was designed, assuming that the displacement measurements at prespecified locations are available in real-time. The resulting distributed controller inherits the distributed structure of the system, and possesses a certain degree of decentralization depending on the designers choice. Therefore, each control unit processes local information as it gets the local sensor measurement while sharing information with its closest neighbors causing a coordinated control action. However, it should be noted that the possible delays and distortion in the transfer of this information were neglected. The controller architecture can be implemented easily and the gains for each local controller are obtained by the LFT framework used to model the parameter dependence of the controller.

The synthesis condition for the distributed parameter-dependent control approach was formulated as finite-dimensional LMI optimization problem. Thus, the solutions can be obtained using available LMI optimization algorithms, and are computationally efficient. Moreover, the distributed analysis and synthesis conditions can be obtained for both discrete and continuous systems using bilinear transformations.

Modifications to the MD toolbox, which is designed for the analysis and synthesis of the spatially-invariant distributed systems, were made so that the tools developed there could be used for the non-uniform flexible beam problem. Details of the modifications are provided in the appendix.

The time domain simulations were performed for different cases in order to test the response of the resulting distributed parameter system. From the analysis of the simulation results, the distributed parameter dependent control of the flexible beam proved to be successful in disturbance rejection in the existence of large sensor noise. Also from the actuator failure case simulations, the reliability of the distributed control was demonstrated.

The results of this research might have been improved, if the actuator and sensor dynamics had been included in the dynamic equations. Then, the resulting dynamic equations would have been more complicated. This is left for future research to explore the effects of actuator and sensor dynamics on the control problem. It might

also be interesting to compare the developed distributed parameter dependent control approach with some centralized control approaches in terms of the computational efficiency and reliability.

List of References

[1] P. Apkarian and P. Gahinet. "A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers". *IEEE Transactions on Automatic Control*, 40(5):853–865, 1995.

[2] M.J. Balas. "Active control of flexible systems". *Journal of Optimization Theory and Aplications*, 25(3):415–436, 1978.

[3] M.J. Balas. "Trends in large space structure control theory: Fondest hopes, wildest dreams". *IEEE Transactions on Automatic Control*, AC-27:522–535, 1982.

[4] B. Bamieh. "The structure of optimal controllers of spatially-invariant distributed parameter systems". *in Proceedings of IEEE Conference on Decision and Control*, pp. 1056–1061, 1997.

[5] B. Bamieh, F. Paganini, and M. Dahleh. "Optimal control of distributed actuator and sensor arrays". *in SPIE Conference on Smart Structures*, pp. 467–478, 1998.

[6] I. Bruant, G. Coffignal, F. Léné, and M. Vergé. "Active control of beam structures with piezoelectric actuators and sensors: Modeling and simulation". *Smart Materials and Structures*, 10:404–408, 2001.

[7] R. D'Andrea. "A linear matrix inequality approach to decentralized control of distributed parameter systems". *in Proceedings of American Control Conference*, pp. 1350–1354, 1998.

[8] R. D'Andrea. "Linear matrix inequalities, multidimensional system optimization, and control of spatially distributed systems: An example". *in Proceedings of American Control Conference*, pp. 2713–2717, 1999.

[9] R. D'Andrea, G. Dullerud, and S. Lall. "Convex ℓ_2 synthesis for multidimensional systems". *in Proceedings of IEEE Conference on Decision and Control*, pp. 1883–1888, 1998.

[10] R. D'Andrea and G.E. Dullerud. "Distributed control of spatially interconnected systems". *IEEE Transactions on Automatic Control*, submitted for publication, 2000.

[11] G.A de Castro and F. Paganini. "Convex synthesis of localized controllers for spatilly invariant systems". *Automatica*, 38:445–456, 2001.

[12] J. Doyle. "Structured uncertainty in control system design". *in Proceedings of the Conference on Decision and Control*, pp. 260–265, 1985.

[13] J. Doyle, A. Packard, and K. Zhou. "Review of LFTs, LMIs, and μ". *in Proceedings of IEEE Conference on Decision and Control*, pp. 1227–1232, 1991.

[14] J. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. "State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems". *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

[15] G.E. Dullerud, R. D'Andrea, and S.G. Lall. "Control of spatially varying distributed systems". *in Proceedings of IEEE Conference on Decision and Control*, pp. 1889–1893, 1998.

[16] G.E. Dullerud and S.G. Lall. "A new approach for analysis and synthesis of time-varying systems". *IEEE Transactions on automatic Control*, AC-44(8):1486–1497, 1999.

[17] P. Gahinet, A. Nemirovsky, A. Laub, and M. Chilali. *LMI Control Toolbox*. Mathworks Inc., Natick, MA, 1995.

[18] F. Giulietti, L. Pollini, and M. Innocenti. "Autonomous formation flight". *IEEE Control Systems Magazine*, 20(6):33–44, 2000.

[19] M. Green and D.J.N. Limebeer. *Linear Robust Control*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1995.

[20] W.S. Hwang and H.C. Park. "Finite element modeling of piezoelectric sensors and actuators". *AIAA Journal*, 31(5):930–937, 1993.

[21] C.D. Johnson and D.A. Kienholz. "Finite element prediction of damping in structures with constrained viscoelastic layer". *AIAA Journal*, 20(9):1284–1290, 1982.

[22] V. Kapila, A.G. Sparks, J. Buffington, and Q. Yan. "Spacecraft formation flying: Dynamics and control". *in Proceedings of American Control Conference*, pp. 4137–4141, 1999.

[23] S.G. Kelly. *Fundamentals of Mechanical Vibrations*. McGraw-Hill, 1993.

[24] P. Khargonekar and K. Zhou. "An algebraic Riccati equation to $\mathcal{H}_\infty$ optimization". *Systems and Control Letters*, 11:85–91, 1988.

[25] J. Kim, V.V. Varadan, V.K. Varadan, and X. Bao. "Finite-element modeling of a smart cantilever plate and comparison with experiments". *Smart Materials and Structures*, 5:165–170, 1996.

[26] L. Klein. "Transverse vibrations of non-uniform beams". *Journal of Sound and Vibration*, 37(4):491–505, 1974.

[27] J.E. Kulkarni and R. D'Andrea. "Application of distributed control techniques to an adaptive secondary mirror". *in Proceedings of American Control Conference*, submitted for publication, 2002.

[28] V.D. Lupi, H.M. Chun, and J.D. Turner. "Distributed control and simulation of a Bernoulli-Euler beam". *Journal of Guidance, Control, and Dynamics*, 15(3):727–734, 1992.

[29] L. Meirovitch. *Principles and Techniques of Vibrations*. Prentice-Hall, Upper Saddle River, New Jersey, 1997.

[30] Y. Nesterov and A. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. SIAM, Phiedelphia, PA, 1993.

[31] A. Packard and J. Doyle. "The complex structured singular value". *Automatica*, 29(1):71–109, 1993.

[32] A. Packard. "Gain scheduling via linear fractional transformations". *Systems and Control Letters*, 22(2):79–92, 1994.

[33] E.P. Popov. *Mechanics of Materials*. Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

[34] H. Raza and P. Ioannou. "Vehicle following control design for automated highway systems". *Journal of Guidance, Control and Dynamics*, 16(6):43–60, 1996.

[35] C.W. Scherer. "LPV control and full block multipliers". *Automatica*, 37:361–375, 1999.

[36] J.S. Shamma and M. Athans. "Analysis of nonlinear gain scheduled control systems". *IEEE Transactions on Automatic Control*, 35:898–907, 1990.

[37] J.S. Shamma and M. Athans. "Gain scheduling: Potential hazards and possible remedies". *IEEE Control Systems Magazine*, 12(3):101–107, 1992.

[38] J.J. Shiffman. "Lyapunov functions and the control of the Euler-Bernoulli beam". *International Journal of Control*, 57(4):971–990, 1993.

[39] M.L. Soni and F.K. Bogner. "Finite element vibration analysis of damped structures". *AIAA Journal*, 20(5):700–707, 1982.

[40] V.A. Spector and H. Flashner. "Flexible manipulator modeling for control system development". *Journal of Guidance, Control and Dynamics*, 12(6):943–945, 1989.

[41] V.V. Varadan, Y.H. Lim, and V.K Varadan. "Closed-loop finite element modeling of active/passive damping in structural vibration control". *Smart Materials ans Structures*, 5:685–694, 1996.

[42] P. Varaiya. "Smart cars on smart roads: Problems of control". *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.

[43] F. Wu. Control of Linear Parameter Varying Systems. PhD dissertation, University of California at Berkeley, 1995.

[44] F. Wu. "Distributed control for interconnected linear parameter-dependent systems". IEE Proceedings. Submitted for publication, 2001.

[45] F. Wu. "A generalized LPV system analysis and control synthesis framework". *International Journal of Control*, 74(7):745–759, 2001.

[46] F. Wu, X.H. Yang, A. Packard, and G. Becker. "Induced $\mathcal{L}_2$ norm control for LPV systems with bounded parameter variation rates". *International Journal of Robust and Nonlinear Control*, 6(9/10):983–998, 1996.

[47] A. Yousefi-Koma. Active Vibration Control of Smart Structures Using Piezoelements. PhD dissertation, University of Michigan, 1997.

[48] K. Zhou and J.C. Doyle. *Essentials of Robust Control*. Prentice-Hall, Englewood Cliffs, New Jersey, 1998.

Appendix A

MATLAB code

A.1 Modifications on MD Toolbox

Throughout this research, some of the functions of the MD toolbox has been modified in order to fix the problems encountered in dealing with the spatially varying distributed systems. We can classify the modifications in three main groups as follows:

1. **Controller synthesis** target functions $\Rightarrow \begin{cases} hfMD.m \\ cnstMD.m \end{cases}$

   "hfMD.m" is the controller synthesis function for spatially invariant interconnected systems. First, a set of LMIs resulting from the generalized form of continuous time bounded real lemma is solved. Then, depending on the designer's choice, controller is reconstructed either explicitly (option 'e') using the subroutine "*cnstMD*" or by solving another LMI (option 'l'). Since this function can only be used for spatially invariant systems, instead of making the extensive modifications, the following functions were developed,

   - *lftsynMD.m* : LMI synthesis condition. Returns X, Y, L, J along with the upper bound on the $\mathcal{L}_2$ gain for the reconstruction of the distributed parameter dependent controller.
   - *mklftkMD.m* : In this function the results of "*lftsynMD.m*" are used for the construction of the distributed controller. "*lftcnstMD_exp.m*" is called for explicit formulation of the controller data while "*lftcnstMD_lmi.m*" may be used if the controller is to be found by solving another LMI.

2. **Simulation** target functions $\Rightarrow \begin{cases} csimulateMD.m \\ simfunMD.m \\ spatsolveMD.m \end{cases}$

   Since spatially-invariant systems were considered when developing the functions, designing one controller for a subsystem would be enough because all the other controllers would be identical. However, for the cases where spatial invariance assumption is not valid, each controller's data is different and is computed

based on the value of the parameter set Δ, which is a function of the spatial independent variable. Therefore, the target functions were modified accordingly.

Another modification in the simulation functions was the change of the ODE solver. The flexible beam problem required a stiff ODE solver because of the numerical issues caused by the large-valued mechanical properties.

3. **Boundary conditions** target function $\Rightarrow$ *spatsolveMD.m*
   There were two types of boundary conditions available in the original function. Either it was d-type, where the data coming into the region was zero, or n-type, in which the data leaving the region was the same as the data coming into the region. However, neither of these two were suitable for the cantilever beam boundary conditions. At the fixed end, where the displacement is always zero, d-type boundary conditions can be implemented. On the other hand, none of the given boundary conditions are applicable to the free end. Boundary conditions for the free end of this flexible beam problem was developed (dn-type) using the finite difference equations as mentioned in modeling section.

A.2 Open-Loop Simulation

```
% opsimulate.m

close all
clear all
clc

% Definitions
grd = 31;          %[-] number of nodes on the beam
L  = 0.508;        %[m] length of the beam
H  = L/(grd-1);    %[m] length of each segment
ct = 0.75;         %[-] ratio of end thickness to base thickness
st = 'n';          %[-] st : if st = 'u' uniform beam
                   %         if st = 'n' nonuniform beam

% Plant model
sysop = beammdl(grd,ct,st)

% nts : number of temporal states in closed-loop system
% nss : number of spatial states in closed-loop system
nts = sysop.blk(1) + sysK.blk(1);
nss = sum(sysop.blk(2:end)) + sum(sysK.blk(2:end));

% Create uncertainty level at each node
xbeam(1) = 0;      %  vector of displacements of each node from
                   % the fixed base
delta(1) = -1;     %  uncertainty vector

for i=1:(grd-1),
    xbeam(i+1) = i*h;
    delta(i+1) = 2*xbeam(i+1)/L - 1;
end

% System interconnection
II=eye(4);
for i=1:grd,
    if st == 'n',
        dlt=delta(i)*II;
        sopP = pck(sysop.A,sysop.B,sysop.C,sysop.D);
        sopP1 = starp(dlt,sclP);
        [ep.A ep.B ep.C ep.D] = unpck(sopP1);
        ep.blk = sysop.blk;
    else
        ep = sysop;
    end
    syscl{i} = tempcl;
end
```

```
% Noise and disturbance
distr = 'disturbance';

% Initial conditions
tipdef = -0.01;    %[m] initial deflection of the free end
typ = 1;           % type of initial condition
                   %     typ = 0 for zero initial conditions
                   %     typ = 1 for normalized first mode shape
                   %     typ = 2 for normalized second mode shape
                   %     typ = 3 for normalized third mode shape
                   %     typ = 4 for point load case

xo = intcond(grd,L,tipdef,nts,st,typ)

% Simulation parameters
tspan = 0:0.001:2.5;     % start time : time step : end time
N = 2;             %[-] number of iterations to solve spatial states
bdry = 'dn';       % boundary condition, fixed-free

% Simulate continuous time system.
%    csimulateMD is modified to cope with the varying state space
%    matrices
[tvec,z,xt,dist] = csimulateMD_mod(syscl,grd,distr,tspan,N,bdry,xo);

% Separate output vector ---> displ : array of node displacements
%                        ---> effrt : array of control inputs
displ=zeros([length(tvec) grd]);
effrt=zeros([length(tvec) grd]);

for i=1:length(tvec)
  for j=0:grd(1)-1,
      displ(i,j+1)=z(i,3*j+2);
      noise(i,j+1)=dsave{i}{j+1}(1);
      distu(i,j+1)=dsave{i}{j+1}(2);
  end
end

% Plot the results
figure(1)          % Displacement of free end vs. time
    plot(tvec,displ(:,grd))
    xlabel('Time, t [s]')
    ylabel('Displacement, v(s,t) [m]')
```

A.3 Controller Feasibility & Design

```
% contdesign.m

close all
clear all
clc

% Definitions
grd = 11;          %[-] number of nodes on the beam
ct  = 1;           %[-] ratio of end thickness to base thickness
st  = 'n';         %[-] nonuniform system

% Plant model
sysop = beammdl(grd,ct,st);

% Weighting functions
% -----> Input side
W2 = 0.001;        %[-] for disturbance
W4 = 0.001;        %[-] for noise

% Add input weighting functions to system model
%     syswop : weighted open loop system
if st == 'n',
    syswop.B(2,6) = sysop.B(2,6)*W2;
    syswop.D(7,5) = sysop.D(7,5)*W4;
else
    syswop.B(2,2) = sysop.B(2,2)*W2;
    syswop.D(3,1) = sysop.D(3,1)*W4;
end

% -----> Output side
num1 = 0.02*[5 1];
den1 = [1 2];
[aa1,bb1,cc1,dd1] = tf2ss(num1,den1);
W1 = pck(aa1,bb1,cc1,dd1);    % [-] penatly on control effort
blk1 = size(aa1,1);

num3 = [1 2];
den3 = [2 1];
[aa3,bb3,cc3,dd3] = tf2ss(num3,den3);
W3 = pck(aa3,bb3,cc3,dd3);    % [-] penalty on node displacement
blk3 = size(aa3,1);

if st == 'n',
    Cf=daug(eye(4),W1,W3,eye(1));
else
    Cf=daug(W1,W3,eye(1));
end
```

```
[cf.A,cf.B,cf.C,cf.D]=unpck(Cf);
cf.blk = [blk1+blk3 0 0];

% Add output weighting functions to system model
syswop = multMD(syswop,cf);

ublk = [4,0];      %[-] uncertainty block structure


% Design parameters
nm = 1;            %[-] number of measured signals
nc = 1;            %[-] number of control signals
reg = [1 0];       %[-] region of analyticity vector
                   %     temporal defined on positive interval (1)
                   %     spatial defined on bi-infinite interval (0)
wrap = 1;          % scaling factor (no scaling)
meth = 'e';        % explicitly solve for controller data
quant = 0;

% Controller synthesis for weighted open loop system
tictac = cputime; % [s] total solution time

if st == 'n',
    [gam,rr,ss,ll,jj] = lftsynMD(syswop,ublk,nm,nc,reg,wrap,0,quant);
    sysK = mklftkMD(syswop,ublk,nm,nc,gam,rr,ss,ll,jj,meth);
else
    [sysK,gam] = hfMD(syswop,nm,nc,reg,meth,wrap);
end

tictac = cputime - tictac;

% Print controller data on screen
disp('Resulting controller:')
sysK

disp('Eigenvalues of the weighted closed-loop system')
sysclw = starMD(syswop,sysK,1,1);
    eig(sysclw.A(1: sysclw.blk(1),1: sysclw.blk(1)))

disp('Eigenvalues of the closed-loop system')
syscl = starMD(sysop,sysK,1,1);
    eig(temp.A(1:syscl.blk(1),1:syscl.blk(1)))

% Save controller data
save controller.mat sysK gam
```

A.4 Closed-Loop Simulation

```
% clsimulate.m

close all
clear all
clc

% Definitions
grd = 31;          %[-] number of nodes on the beam
mv = 3;
L  = 0.508;        %[m] length of the beam
H  = L/(grd-1);    %[m] length of each segment
ct = 0.75;         %[-] ratio of end thickness to base thickness
st = 'n';          %[-] st : if st = 'u' uniform beam
                   %         if st = 'n' nonuniform beam

% Plant model
sysop = beammdl(grd,ct,st)
ublk = [4,0];      %[-] uncertainty block structure

% Load controller data
load controller.mat

% Create a dummy controller for uncontrolled plant nodes
ntk = sysK.blk(1);      %[-] number of temporal states of K

tempK.A = zeros(size(sysK.A,1),size(sysK.A,2));
tempK.A(ntk+1:end,ntk+1:end) = eye(4);

tempK.B = zeros(size(sysK.A,1),size(sysK.B,2));
tempK.C = zeros(size(sysK.C,1),size(sysK.A,2));
tempK.D = zeros(size(sysK.D,1),size(sysK.D,2));
tempK.blk = sysK.blk;

% nts : number of temporal states in closed-loop system
% nss : number of spatial states in closed-loop system
nts = sysop.blk(1) + sysK.blk(1);
nss = sum(sysop.blk(2:end)) + sum(sysK.blk(2:end));

% Create uncertainty level at each node
xbeam(1) = 0;      %  vector of displacements of each node from
                   % the fixed base
delta(1) = -1;     %  uncertainty vector

for i=1:(grd-1),
    xbeam(i+1) = i*h;
    delta(i+1) = 2*xbeam(i+1)/L - 1;
end
```

```
% Failure case simulation variables
cfc = 1;          % cfc = 0 when controller fails
sfc = 1;          % sfc = 0 when sensor fails
nfc = 19;         %[-] node number at which the failure occurs

% System interconnection
II=eye(4);
for i=1:grd,
    dlt=delta(i)*II;
    % Failure case
    if i == nfc,
        sysop.B(2,7) = cfc*sysop.B(2,7);
        sysop.C(7,1) = sfc*sysop.C(7,1);
    end

    % Fu(P,delta) - upper LFT with uncertainty block
    sclP = pck(sysop.A,sysop.B,sysop.C,sysop.D);
    sclP1 = starp(dlt,sclP);
    [ep.A ep.B ep.C ep.D] = unpck(sclP1);
    ep.blk = sysop.blk;

    % Fl(K,delta) - lower LFT with uncertainty block
    sclK = pck(sysK.A,sysK.B,sysK.C,sysK.D);
    sclK1 = starp(sclK,dlt);
    [ek.A ek.B ek.C ek.D] = unpck(sclK1);
    ek.blk = sysK.blk;

    scltK = pck(tempK.A,tempK.B,tempK.C,tempK.D);
    scltK1 = starp(scltK,dlt);
    [etk.A etk.B etk.C etk.D] = unpck(scltK1);
    etk.blk = tempK.blk;

    % Fl(P,K) - lower LFT with controller
    if mod(i,mv) == 1,
        tempcl = starMD(ep,ek,1,1);
        syscl{i} = tempcl;
    else
        tempcl = starMD(ep,etk,1,1);
        syscl{i} = tempcl;
    end
end

% Noise and disturbance
distr = 'disturbance';

% Initial conditions
tipdef = -0.01;   %[m] initial deflection of the free end
typ = 1;          % type of initial condition
                  %     typ = 0 for zero initial conditions
                  %     typ = 1 for normalized first mode shape
                  %     typ = 2 for normalized second mode shape
                  %     typ = 3 for normalized third mode shape
                  %     typ = 4 for point load case
```

```
xo = intcond(grd,L,tipdef,nts,st,typ)

% Simulation parameters
tspan = 0:0.0001:0.025; % start time : time step : end time
N = 7;             %[-] number of iterations to solve spatial states
bdry = 'dn';       % boundary condition, fixed-free

% Simulate continuous time system.
%    csimulateMD is modified to cope with the varying state space
%    matrices
[tvec,z,xt,dist] = csimulateMD_mod(syscl,grd,distr,tspan,N,bdry,xo);

% Separate output vector ---> displ : array of node displacements
%                        ---> effrt : array of control inputs
displ=zeros([length(tvec) grd]);
effrt=zeros([length(tvec) grd]);

for i=1:length(tvec)
  for j=0:grd(1)-1,
      effrt(i,j+1)=z(i,2*j+1);
      displ(i,j+1)=z(i,2*j+2);
      noise(i,j+1)=dsave{i}{j+1}(1);
      distu(i,j+1)=dsave{i}{j+1}(2);
  end
end

% Plot the results
figure(1)          % Displacement of free end vs. time
    plot(tvec,displ(:,grd))
    xlabel('Time, t [s]')
    ylabel('Displacement, v(s,t) [m]')

figure(2)          % Control input at free end vs. time
    plot(tvec,effrt(:,grd))
    xlabel('Time, t [s]')
    ylabel('Control input, u(s,t) [N/m]')

figure(3)          % Disturbance input at free end vs. time
    plot(tvec,distu(:,grd))
    xlabel('Time, t [s]')
    ylabel('Disturbance input, d(s,t) [m]')

figure(4)          % Noise at free end vs. time
    plot(tvec,noise(:,grd))
    xlabel('Time, t [s]')
    ylabel('Noise, n(s,t) [m]')

figure(5)          % Displacement of free end vs. time
    mesh(xbeam,tvec,displ)
    xlabel('Displacement, x [m]')
    ylabel('Time, t [s]')
    zlabel('Displacement, v(s,t) [m]')
```

A.5 Plant Model Function

```
% beammdl.m

function [sysout] = beammdl(grd,ct,st)

% grd : number of nodes on the beam
% ct  : thickness ratio
% st  : if st = 'u' uniform beam
%       if st = 'n' nonuniform beam
%
% Define constant plant parameters
%
L  = 0.508;             %[m] length of the beam
tb = 0.0008;            %[m] thicknes at the fixed end
wb = 0.0254;            %[m] width
E  = 72e9;              %[N/m^2] modulus of elasticity
ro = 2710;              %[kg/m^3] volume density
h  = L/(grd-1);         %[m] length of each segment

%
% Compute non-zero state space matrix coefficients
%
a21 = E*tb^2/ro*( (1-ct)^2 /(L^2*h^2)  - (1+ct)^2 /(8*h^4));
a23 = E*tb^2/ro*(-(1-ct^2) /(8*L*h^3)  - (1+ct)^2 /(48*h^4));
a24 = E*tb^2/ro*(-(1-ct)^2 /(2*L^2*h^2)+ (1-ct^2) / (4*L*h^3)...
      +(1+ct) ^2 /(12*h^4));
a25 = E*tb^2/ro*( (1-ct^2) /(8*L*h^3)  - (1+ct)^2 /(48*h^4));
a26 = E*tb^2/ro*(-(1-ct)^2 /(2*L^2*h^2)- (1-ct^2) /(4*L*h^3)...
      +(1+ct)^2/(12*h^4));

b21 = 2*(1-ct)/(ro*wb*tb*(1+ct));
b22 = b21;
b23 = E*tb^2*(1-ct^2)/(48*ro);
b24 = E*tb^2*(1-ct)/ro;
b26 = 2/(ro*wb*tb*(1+ct));
b27 = b26;

c31 = 6/h^4;
c33 = 1/h^4;
c34 = -4/h^4;
c35 = c33;
c36 = c34;
c41 =  (1+ct)/(8*h^4);
c43 =  (1+ct)/(48*h^4)+(1-ct)/(8*L*h^3);
c44 = -(1+ct)/(12*h^4)-(1-ct)/(4*L*h^3);
c45 =  (1+ct)/(48*h^4)-(1-ct)/(8*L*h^3);
c46 = -(1+ct)/(12*h^4)+(1-ct)/(4*L*h^3);

d11 = (1-ct)/(1+ct);
d16 = 1/(1+ct);
```

```
d22 = d11;
d27 = d16;
d43 = -(1-ct)/48;

%
% State-Space Data for Plant
%     inputs  : [uncertainty(x4);n(s,t),d(s,t),u(s,t)]
%     outputs : [uncertainty(x4);e1(s,t);e2(s,t);y(s,t)]
sysout.A = [0 1 0 0 0 0;
          a21 0 a23 a24 a25 a26;
            0 0 0 1 0 0;
            1 0 0 0 0 0;
            0 0 0 0 0 1;
            1 0 0 0 0 0];

sysout.B = [0 0 0 0 0 0 0;
          b21 b22 b23 b24 0 b26 b27;
            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];

sysout.C = [0 0 0 0 0 0;
            0 0 0 0 0 0;
          c31 0 c33 c34 c35 c36;
          c41 0 c43 c44 c45 c46;
            0 0 0 0 0 0;
            1 0 0 0 0 0;
            1 0 0 0 0 0];

sysout.D = [d11 0 0 0 0 d16 0;
            0 d22 0 0 0 0 d27;
            0 0 0 0 0 0 0;
            0 0 d43 0 0 0 0;
            0 0 0 0 0 0 1;
            0 0 0 0 0 0 0;
            0 0 0 0 1 0 0];

if un == 'u',
    sysout.B = sysout.B(:,5:7);
    sysout.C = sysout.C(5:7,:);
    sysout.D = sysout.D(5:7,5:7);
end

sysout.blk=[2 2 2];
```

A.6 Disturbance Generation Function

```
% disturbance.m

function d = disturbance(t,grd)

% t   : time variable
% grd : number of nodes on the beam

TSAMPLE = 0.001;  %   The disturbance is step wise constant, with step
                  % size equal to TSAMPLE. When simulating a system,
                  % variables must be a fixed function of time.
                  %   The disturbance generated is random, but yields
                  % the same value as a function of the time variable
                  % 't'.
st = 'cl'         % if st = 'cl' closed loop disturbance
                  % if st = 'op' open loop disturbance and u = 0
randstate = floor(t/TSAMPLE);
randn('state',randstate);

if st == 'cl',
    d = randn([2 grd]);
    ni = d(1,:);  % [m]   noise
    di = d(2,:);  % [N/m] disturbance
else
    d = randn([3 grd]);
    ni = d(1,:);
    di = d(2,:);
    ui = d(3,:);
end

nscl = 0.0001;    % scaling constant for noise
dscl = 25;        % scaling constant for disturbance

ni = nscl*ni;
di = dscl*di;
ui = zeros(1,length(ui));  % control input is always zero for open-loop
                           % system

if st == 'cl',
    d = [ni;di];
else
    d = [ni;di;ui];
end

d=num2cell(d,1);
d=squeeze(d);
```

A.7 Initial Condition Generation Function

```
% intcond.m

function [xo] = intcond(grd,L,tipdef,nts,st,typ)

% grd    : number of nodes on the beam
% L      : length of the beam
% tipdef : desired tip deflection
% nts    : number of temporal states
% st     : state of the beam geometry
%           'n' for non-uniform
%           'u' for uniform
% typ    : type of initial condition
%           0 for zero initial conditions
%           1 for normalized first mode shape
%           2 for normalized second mode shape
%           3 for normalized third mode shape
%           4 for point load case

x0=zeros(nts,grd);

% Mode shapes as initial conditions
if typ == 1,
    if st == 'u',
        ms = load('umode1.txt');
    else
        ms = load('numode1.txt');
    end
elseif typ == 2,
    if st == 'u',
        ms = load('umode2.txt');
    else
        ms = load('numode2.txt');
    end
elseif typ == 3,
    if st == 'u',
        ms = load('umode3.txt');
    else
        ms = load('numode3.txt');
    end
end

% Normalize the mode shape using tipdef
msy = ms(:,2);
msy = abs(tipdef)/msy(end,1)*msy;
```

```
% Reduce finite element model grid to grd
inx = 0;    % temporary index
incon=zeros(1,grd);
FEMgrd = size(ms,1);

tempvar = (FEMgrd-1)/(grd-1);

for i = 1:FEMgrd,
    if mod(i,tempvar) == 1,
        inx = inx + 1;
        incon(inx) = msy(i);
    end
end

for i = 1:(grd-1),
    x = i*h;

    if typ == 0,
        x0(1,i+1) = 0;
    elseif typ == 4,
        x0(1,i+1) = (3*tipdef/(L^2))*(0.5*x^2 - 1/(6*L)*x^3);
    else
        x0(1,i+1) = incon(i+1);
    end
end

xo = cat(1,x0(:));
```

A.8 Uniform Beam Analytical Model Simulation

```
% ansol.m

clear all
close all
clc

L  = 0.508;        % [m] Length of the beam
tb = 0.0008;       % [m] Thickness of the beam
wb = 0.0254;       % [m] Width of the beam
E  = 72e9;         % [Pa] Young's Modulus
ro = 2710;         % [kg/m^3] Mass density

I=1/12*tb^3*wb;    % [m^4] Moment of inertia

% The general solution:
% v(x,t) = SUM MSn(x)[Aa.cos(wn.t) + Bb.sin(wn.t) n = 1:5
%      wn : [rad/s] natural frequencies
%      MSn : mode shapes
%      Aa,Bb : arbitrary coefficients that depend on initial conditions

% Calculate the natural frequencies
Bl=[1.875104; 4.694091; 7.854757; 10.995541; 14.138599];
for r=1:5,
   wn(r)=Bl(r)^2*sqrt(E*I/(ro*tb*wb*L^4));
end

grd = 501;         % [-] Number of nodes on the beam

% Find the mode shapes
for i = 1:grd,
    x(i) = (i-1)*L/(grd-1);
    for r = 1:5,
        W(r,i) = -(sin(Bl(r)/L*x(i)) - sinh(Bl(r)/L*x(i)))...
            +(sin(Bl(r)) + sinh(Bl(r)))/(cos(Bl(r)) + cosh(Bl(r)))*...
            (cos(Bl(r)/L*x(i)) - cosh(Bl(r)/L*x(i)));
    end
end

% Mode shapes as initial conditions
tipdef = -0.01;    % [m] Transverse deflection of the free end

for r=1:5,
    tempvar(r)=abs(W(r,end)/tipdef);
    incon(r,:)=W(r,:)/tempvar(r);
end
```

```
% Activate this part if the following initial conditions are desired
%
% Assume a point load is applied at the free end causing a tip
% deflection of tipdef. Corresponding initial conditions are:
%incon=zeros(1,grd);
%for i=1:(grd-1),
%    xcc = i*L/(grd-1);
%    incon(i+1) = (3*tipdef/(L)^2)*(0.5*xcc^2 - 1/(6*L)*xcc^3);
%end
%CoefVo=[incon(170);incon(250);incon(340);incon(450);incon(500)];
% Choose r to select which mode to excite, r = 1, 2, 3, 4, 5
CoefVo = [incon(r,170);
          incon(r,250);
          incon(r,340);
          incon(r,450);
          incon(r,500)];

CoefWn = [W(1,170) W(2,170) W(3,170) W(4,170) W(5,170);
          W(1,250) W(2,250) W(3,250) W(4,250) W(5,250);
          W(1,340) W(2,340) W(3,340) W(4,340) W(5,340);
          W(1,450) W(2,450) W(3,450) W(4,450) W(5,450);
          W(1,500) W(2,500) W(3,500) W(4,500) W(5,500)];

Aa = inv(CoefWn)*CoefVo;

tcon = 2.5;       %[s] time to stop simulation
                  % time step is 0.001 seconds
for i = 1:(tcon*1000+1),
    t(i) = (i-1)/1000;
    for j = 1:grd,
        v(i,j) = Aa(1)*W(1,j)*cos(wn(1)*t(i))...
               + Aa(2)*W(2,j)*cos(wn(2)*t(i))...
               + Aa(3)*W(3,j)*cos(wn(3)*t(i))...
               + Aa(4)*W(4,j)*cos(wn(4)*t(i))...
               + Aa(5)*W(5,j)*cos(wn(5)*t(i));
    end
end

figure
plot(t,v(:,grd))
grid
xlabel('Time, t [sec.]')
ylabel('Displacement, v(x,t) [m]')
```

A.9 FEM Model Simulation

```
% FEMmdl.m

close all
clear all
clc

L   = 0.508;       % [m] Length of the beam
st  = 'n';         % [-] st : if st = 'u' uniform beam, ct = 1
                   %          if st = 'n' non-uniform beam, ct = 0.75

% Load mode shapes and natural frequencies
if st == 'u',
    ms1  = load('umode1.txt');
    ms2  = load('umode2.txt');
    ms3  = load('umode3.txt');
    ms4  = load('umode4.txt');
    ms5  = load('umode5.txt');
    fn = [2.61700 16.21632 45.32804 88.79048 146.75582]; %[Hz]
else
    ms1  = load('numode1.txt');
    ms2  = load('numode2.txt');
    ms3  = load('numode3.txt');
    ms4  = load('numode4.txt');
    ms5  = load('numode5.txt');
    fn = [2.70793 14.91494 40.24688 78.08622 128.54269]; %[Hz]
end

ms1x = ms1(:,1)*L/ms1(end,1);
ms1y = ms1(:,2);

ms2x = ms2(:,1)*L/ms2(end,1);
ms2y = ms2(:,2);

ms3x = ms3(:,1)*L/ms3(end,1);
ms3y = ms3(:,2);

ms4x = ms4(:,1)*L/ms4(end,1);
ms4y = ms4(:,2);

ms5x = ms5(:,1)*L/ms5(end,1);
ms5y = ms5(:,2);

%Natural frequencies
wn = fn*2*pi;           % [rad/sec]

% Mode shapes as initial conditions
tipdef = -0.01;         % [m]
grd = size(ms1,1);      % [-] number of nodes on the beam
```

```
% choose mode shape msiy, i = 1, 2, 3, 4, 5
const = ms1y(end,1)/tipdef;
incon = ms1y/const;

%
% Activate this part if the following initial conditions are desired
%
% Assume a point load is applied at the free end causing a tip
% deflection of tipdef. Corresponding initial conditions are:
%incon=zeros(1,grd);
%for i=1:(grd-1),
%    x = i*L/(grd-1);
%    incon(i+1) = (3*tipdef/(L)^2)*(0.5*x^2-1/(6*L)*x^3);
%end

if st == 'u',
    CoefWn = [ms1y(170) ms2y(170) ms3y(170) ms4y(170) ms5y(170);
              ms1y(249) ms2y(249) ms3y(249) ms4y(249) ms5y(249);
              ms1y(340) ms2y(340) ms3y(340) ms4y(340) ms5y(340);
              ms1y(421) ms2y(421) ms3y(421) ms4y(421) ms5y(421);
              ms1y(495) ms2y(495) ms3y(495) ms4y(495) ms5y(495)];

    CoefVo = [incon(170);incon(249);incon(340);incon(421);incon(495)];
else
    CoefWn = [ms1y(287) ms2y(287) ms3y(287) ms4y(287) ms5y(287);
              ms1y(1071) ms2y(1071) ms3y(1071) ms4y(1071) ms5y(1071);
              ms1y(2311) ms2y(2311) ms3y(2311) ms4y(2311) ms5y(2311);
              ms1y(3177) ms2y(3177) ms3y(3177) ms4y(3177) ms5y(3177);
              ms1y(4815) ms2y(4815) ms3y(4815) ms4y(4815) ms5y(4815)];
    CoefVo = [incon(287);
              incon(1071);
              incon(2311);
              incon(3177);
              incon(4815)];
end
Aa = inv(CoefWn)*CoefVo;
tcon = 2.5;       %[s] time to stop simulation
                  % time step is 0.001 seconds
for i = 1:(tcon*1000+1),
    t(i) = (i-1)/1000;
    for j = 1:grd,
        v(i,j)= Aa(1)*ms1y(j)*cos(wn(1)*t(i))...
              + Aa(2)*ms2y(j)*cos(wn(2)*t(i))...
              + Aa(3)*ms3y(j)*cos(wn(3)*t(i))...
              + Aa(4)*ms4y(j)*cos(wn(4)*t(i))...
              + Aa(5)*ms5y(j)*cos(wn(5)*t(i));
    end
end

plot(t,v(:,grd))
xlabel('Displacement, v(x,t) [m]')
ylabel('Time, t [sec.]')
```