LOW-ORDER LINEAR PARAMETER-VARYING MODEL DEVELOPMENT AND ROBUST CONTROL DESIGN FOR AUTOMOTIVE AND MANUFACTURING APPLICATIONS

by

MARK JEFFREY TRUDGEN

(Under the direction of Javad Mohammadpour Velni)

ABSTRACT

This dissertation proposes low-order model development and controller design for various automotive and manufacturing applications. The first task models a single wafer rapid thermal process (RTP) using first-principles modeling. Then, making use of the linear parameter-varying (LPV) state-space representation, we convert the nonlinear RTP model into an affine LPV model. For controller design purposes, we reduce the number of scheduling variables and the order of the model. Using this reduced-order model, we design a gain-scheduled $\mathscr{H}_\infty$ controller for reference tracking. In the second task, we reconfigure the model of a cooperative adaptive cruise control platoon in order to account for the uncertain and time-varying parameters of the system dynamics and communication delay. Using this model, we design a robust controller and reduce its order. The reduced-order controller remains robust to the model. We validate the controller design by performing experiments on the test bed to show the need for robust control. The third task uses model order reduction techniques to design a low-order robust controller in a parabolic convection-diffusion equation application.

INDEX WORDS: Linear Parameter-varying models, Robust control, Cooperative adaptive cruise control

LOW-ORDER LINEAR PARAMETER-VARYING MODEL DEVELOPMENT AND ROBUST CONTROL DESIGN FOR AUTOMOTIVE AND MANUFACTURING APPLICATIONS

by

MARK JEFFREY TRUDGEN

B.S., Oklahoma Christian University, 2011

M.S., Arizona State University, 2012

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2016

LOW-ORDER LINEAR PARAMETER-VARYING MODEL DEVELOPMENT AND ROBUST CONTROL DESIGN FOR AUTOMOTIVE AND MANUFACTURING APPLICATIONS

by

MARK JEFFREY TRUDGEN

Approved:

Major Professor: Javad Mohammadpour Velni

Committee: Bill Tollner
Lawrence Hornak
Yongqiang Wang

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2016

DEDICATION

I dedicate this dissertation to my Mom and Dad, who instilled in me the drive to pursue higher education. Also to my loving Jennifer, and all my family and friends who have been so instrumental in helping me along the way. We did it! To my Lord and Savior Jesus Christ, thank you for leading and guiding me each step of the way. You are faithful and trustworthy.

ACKNOWLEDGMENTS

I would like to thank Dr. Javad Mohammadpour Velni for giving me the opportunity to join the Complex Systems Control Laboratory. I would like to thank Zeeshan Rizvi, Farshid Abbasi, and Rebecca Miller for their friendships and technical conversations throughout our time together in the lab. I would like to thank my committee members Dr. Bill Tollner, Dr. Lawrence Hornak, and Dr. Yonqiang Wang for their time and guidance. Finally, I'd like to thank the Ph.D.'s in my life that led me to consider graduate school and have helped to guide me along the path: Dr. Gary Byram, Dr. Brett Byram, Dr. Fred Garrett, and Dr. Phil Bartley.

TABLE OF CONTENTS

List of Figures

LIST OF TABLES

CHAPTER 1

INTRODUCTION AND LITERATURE SURVEY

1.1 INTRODUCTION

This dissertation proposes low-order model development and controller design for various automotive and manufacturing applications. Here, we use the framework of linear parameter-varying (LPV) state-space systems. LPV systems are linear dynamic systems whose state-space representation depend on time-varying parameters called scheduling variables. In this framework, nonlinear models are described using linear dynamic relations between the inputs and outputs. This is defined in the discrete time state-space representation as

$$\begin{aligned} x(k+1) &= A(\theta(k))x(k) + B(\theta(k))u(k), \\ y(k) &= C(\theta(k))x(k) + D(\theta(k))u(k), \end{aligned} \tag{1.1}$$

where k is the discrete time instant, x describes the state vector, u is the external input to the system, y is the measurements, and $\theta(k)$ denotes the scheduling parameters ($\theta(k)$ denotes that the scheduling variables are time dependent). The LPV framework introduces a modeling paradigm that finds a middle ground between linear and nonlinear system dynamics [3].

When an LPV model is used for gain-controller design purposes, the controller is designed using the same scheduling variables of the LPV model. Presently, the standard linear matrix inequality (LMI) solutions for gain-scheduled controller synthesis requires solving 2^n vertices of the corresponding polytopic system, where n is the number of scheduling variables [4]. Given this limitation in gain-scheduled controller design, the number of scheduling variables in the original LPV model becomes nontrivial. The data reduction techniques of principle component analysis

can be used to reduce the number of scheduling variables, and proper orthogonal decomposition presents a data-driven method of reducing the model order of a system. Both of these techniques can be used to derive a reduced model that is low in both number of scheduling variables and order. A reduced-order model of this form is more suitable for controller design synthesis as the low number of scheduling variables allows for increased mathematical tractability in controller design using standard $\mathscr{H}_\infty$ loop shaping LMI-based techniques.

Next, in the modeling case that system parameters are known to be time-varying or to vary within a range but are unmeasureable, robust control theory can be applied to design a robust controller. This approach can be conservative, but is suited for applications where the uncertainty cannot be quantified in real-time. There are also several reasons to incorporate robustness into a control design framework as there usually exist several sources of uncertainty within any dynamic system. Fundamentally, there are always parameters that are only approximately known or are modestly in error. Also, linear models may only be adequate for a small operating range, and original measurements for parameter identification have inherent errors despite calibration [5]. For robust controller design we use standard LMI solvers for $\mathscr{H}_\infty$ loop shaping. If the controller order is deemed too high, balanced truncation methods can be used to achieve a low-order controller.

1.2 Literature Survey

We now examine the available literature on the selected topics presented throughout the dissertation.

1.2.1 Rapid Thermal Processing

Thermal processes are very important in the fabrication of semiconductor devices. As such, the semiconductor industry has relied on advancements in control and modeling for these purposes [1], [6]. In rapid thermal processing, the longer a wafer is kept at elevated temperatures, the higher probability it has of defects. As such, minimizing a metric called the *thermal budget* is very important not only for heating cost purposes, but also for purity and defect reasons [6].

Several alternative approaches to modeling and control of single wafer RTP system have been suggested in the literature. Review of an *Steag Inc* RTP system with first principles modeling and genetic *nonlinear model predictive control* (NMPC) was proposed in [1]. An adaptive control model was presented in [7]. The authors in [8] studied the thermal behavior of large silicon wafers. Decentralized control approach in the design of PI controllers was used in [9]. The authors in [10] used *proper orthogonal decomposition* (POD) to reduce the order of an RTP system. A linear quadratic gaussian (LQG) approach to control was taken in [11]. Furthermore, a run to run approach was taken in [12], while [13] used *internal model control* (IMC). Finally, multivariable and multizone control was presented in [14–16]. A survey of RTP processes was presented in [17].

As observed from the aforementioned literature, first principles-based modeling of the RTP system is best represented by a partial differential equation (PDE) with varying coefficients and nonlinear boundary conditions. However, direct control of such nonlinear system is not seen in literature, nor are there modeling frameworks that present the plant in a control-oriented form. In this paper, we propose a *linear parameter-varying* (LPV) modeling approach that directly and systematically copes with the complex nonlinearities seen in the RTP processes. LPV techniques have gained popularity as they have developed into effective tools to control multi-input multi-output (MIMO) nonlinear systems [3]. Furthermore, the application of these methods has not been explored for thermal processes including RTP systems, for which the well known nonlinear material properties can be exploited in the LPV framework of scheduling variables.

1.2.2 Cooperative Adaptive Cruise Control

Connected vehicles are an example of a modern day cyber physical systems (CPS) that through the use of Cooperative Adaptive Cruise Control (CACC) can provide an innovative solution to the traffic congestion problem [18]. Traffic is becoming an increasing problem in today's world as congestion in many urban areas is growing at a much faster rate than the traditional means of traffic alleviation can assuage [19]. CACC is a technology that seeks to reduce traffic congestion by means of achieving higher traffic flow rates using advanced control systems to safely reduce

the allowable *headway* time between vehicles [20]. A widespread advantage of CACC over traditional means of increasing traffic throughput, i.e., road construction, is that CACC has the potential to be implemented on any car without the additional high costs and delays associated with road construction projects [21].

CACC technology is an extension of Adaptive Cruise Control (ACC), which in turn is an extension of conventional cruise control (CCC), a technology traditionally used to regulate a vehicle at a constant highway speed [22]. ACC extends the CCC technology by regulating the so-called *headway* distance between vehicles that are arranged together in a platoon [23]. ACC employs radar (or lidar) sensors to measure the relative velocity and displacement with the preceding vehicle, and a longitudinal control framework is then implemented to space the vehicles to an appropriate *headway* [22] by adjusting the acceleration and deceleration of the vehicle. CACC extends the ACC technology by adding inter-vehicle wireless communication [24]. This extension enables smaller *headway* distances, which is critical for platoon technology to have a noticeable impact on traffic mitigation [21, 25].

According to the *2010 Highway Capacity Manual*, a study observing human drivers showed that the maximum flow rate for a multi-lane highway (at 60 mi/h) equates to 1.1 seconds of *headway* [21]. Herein lies the main drawback of ACC technology; the smallest stable *headway* is larger than the average time-gap that human drivers naturally exhibit [19, 25], thus justifying the need for CACC technology. The vehicles that are virtually connected to each other through CACC technology must ensure an important metric called string stability [2]. This concept was first introduced in [26] and later extended in [27], which led to the development of systems using the nearest neighbor as a measurement. Essentially, string stability is a requirement that all disturbances introduced in the string be attenuated as they propagate in the downstream direction [23,27]. String stability is essential to ensuring the safety and feasibility of the string [2]. Not only do any disturbances in position, velocity or acceleration create increased energy consumption, these disturbances must also be mitigated in order to prevent the so-called *ghost traffic jams* [23], or even in extreme cases, an accident [28]; hence, a control design formulation that can explicitly account

for string stability inherently meets design objectives and exterminates the need for any ad-hoc *a posteriori* tuning to achieve string stability. This notion of string stability has been studied in several aspects such as Lyapunov stability, and input-output stability; however, these methods lack the consideration of a measure of performance as seen in [29, 30], which give a frequency-domain approach for controller synthesis.

Several approaches have been undertaken in designing a controller for a platoon of vehicles. The system model considered to describe the vehicular motion is usually a thrid-order nonlinear model [31, 32], where subsequently the plant is linearized by the use of feedback linearization method. For the control design using the linearized model, several CACC experimental results have been reported, e.g., in [23, 24, 33]. These recent works show the promise in using CACC. Indeed, several aspects of CACC technologies have been studied. The authors in [28] developed a sampled data approach to CACC design in the presence of sensors and actuator failures and [34] studied strategies for worst case sensor failure scenarios. Model predictive control (MPC) has also gained attention as a way to cast the CACC problem in a framework that can directly optimize fuel economy. An MPC based CACC approach was designed for heavy duty vehicles, such as tractor trailer trucks in [35], where smaller *headway* distances can be sacrificed for better fuel economies as traffic throughput may not be the primary objective as is the case with urban rush hour highway demands. CACC can also be viewed in light of the communication as a networked control system where the effects of sampling, hold, and network delays can be taken into account. An $\mathscr{H}_\infty$ formulation of network controlled problems is given in [36]. Still, other works have investigated time-varying communication delays and communication structures beyond the classical architecture as in [37].

1.2.3 2-D Parabolic Convection-Diffusion Equation

Convection-diffusion equations are representative of a class of fluid dynamic systems that accurately describe a particular flow phenomenon, but are mathematically not suited for control design purposes [38–40]. Recent developments in the control system community with respect to model

order reduction techniques have generated interest in applying relevant tools to fluid dynamic systems governed by partial differential equations (PDEs) [41, 42]. Therefore, in order to arrive at lower-order models tractable for control design purposes, there is a need to use a suitable projection method [41]. Convection-diffusion processes are widespread in occurrence in many scientific and engineering fields. A few common applications are: pollutant dispersal, vorticity transport in the incompressible Navier-Stokes equations, atmospheric pollution, semiconductor equations, the Stephan problem on a variable mesh, and viscous compressible flow [38]. The authors in [43, 44] used finite element method to control the convection-diffusion equation; however, their primary focus was on the mesh constraints related to optimality.

A great deal of work developed for model order reduction techniques has come within the controls context, and many methods for model order reduction have been suggested [42]. Proper orthogonal decomposition (POD) has become a well-accepted technique to obtain optimal basis functions that lead to low-order models to accurately represent the original full-order models [39–42, 45–48]. POD was first introduced by Lumley [49] in the context of turbulence, and the method of snapshots was first suggested by Sirovich [50]. POD has the intrinsic property that it is completely data dependent, and the modal decomposition used does not assume prior knowledge of how the data is generated [45]. This property is advantageous as no *apriori* information is needed to choose an ideal set of basis functions.

1.3 Dissertation Aims and Organization

In this dissertation we propose low-order model development and controller design for various automotive and manufacturing applications. As the field of control systems has progressed, researchers have developed advanced techniques for controller synthesis, especially for nonlinear and uncertain systems. Traditionally, Jacobian linearization around an equilibrium point has been used to linearize models for controller design purposes. The resulting linear system, however, only describes the local behavior around that equilibrium point. Controller design from a linearized system cannot guarantee closed-loop stability in the presence of uncertainties and unmodeled

dynamics. Perturbations of the real system that move the dynamics away from the equilibrium point would cause a decrease in closed-loop performance, and could also potentially destabilize the closed-loop system, especially for systems with large uncertainties.

Robust control theory provides a direct methodology to model uncertainties and unmodeled dynamics while presenting an explicit notion of robust stability and robust performance. However, the design of a robust controller could be potentially conservative, especially if uncertain parameters can be measured in real-time. To overcome this potential conservatism, LPV modeling takes into the account the parameter variations of the state-space system dynamics by using scheduling variables. An LPV model can be used in conjunction with a gain-scheduled controller.

This dissertation proposes low-order model development and controller design. Low-order controllers are desired for ease in real-time computational implementation. This desire to develop a low-order controller gives focus to the initial modeling assumptions and controller design methodology. By understanding the type of the uncertainties and nonlinearities within the model, different control design approaches should be made to best achieve a low-order controller. We show this in several tasks throughout the dissertation.

In chapters 2 and 3 we model a single wafer RTP. Here the use of an LPV model to describe the RTP system is justified given the hard nonlinearities and available online temperature measurement for scheduling variables calculations. The nonlinear RTP system is formulated into an affine LPV model, and subsequently the number of scheduling variables and model order are reduced for controller design purposes. We use the reduced-order model to design a gain-scheduled controller and show closed-loop reference tracking. In chapters 4 and 5 we reformulate the model of a cooperative adaptive cruise control platoon in order to account for the uncertain and time-varying parameters of the system dynamics. We then design a robust controller, and validate a low-order robust controller with an experimental laboratory test bed showing the need for robust control. The final work in Chapter 6 uses model order reduction techniques to design a low-order robust controller in a parabolic convection-diffusion equation application. Finally, concluding remarks about the contribution of the dissertation are given in Chapter 7.

CHAPTER 2

LINEAR PARAMETER-VARYING APPROACH FOR MODELING AND CONTROLLING RAPID THERMAL PROCESSES

[1]

[1] Trudgen, Mark and Javad Mohammadpour. "Linear parameter-varying approach for modeling and control of rapid thermal processes." Submitted to the International Journal of Control, Automation and Systems, November 2016.

2.1 ABSTRACT

In the present paper, a new approach is presented to model and control single wafer *rapid thermal processing* (RTP) systems. Within the past decade, RTP has achieved acceptance as the mainstream technology for semiconductor manufacturing. Thermal processing is one of the most efficient ways to control the phase-structure properties; moreover, the time duration of RTP systems reduces the so-called *thermal budget* significantly compared to the traditional methods. RTP implementation is based on the use of light from heating lamps to provide a heat flux. This process is highly nonlinear due to the radiative heat transfer and material properties. By invoking the first principles-based models, we develop in this paper a *linear parameter-varying* (LPV) model to directly account for all the nonlinearities within the system. The model is discretized into a high-order affine LPV system; thereafter, *principal component analysis* (PCA) method is utilized to reduce the number of the LPV model's scheduling variables, followed by the use of *proper orthogonal decomposition* (POD) for model order reduction. From the reduced-order system, we design a gain scheduled controller to satisfy an induced $\mathscr{L}_2$ gain performance for tracking of a temperature profile.

2.2 INTRODUCTION

Embedded deep in the heart of all electrical applications are *integrated circuits* (IC), primarily composed of semiconductor devices made from a sequence of batch processes. With the continual developments in IC technology, we see an increase in the demand for performance improvements in terms of both quality variables and output yield resulting from the use of larger diameter silicon wafers [17]. To achieve these increased yields, precise uniform temperature control of a wafer is of paramount consideration. As such, the semiconductor industry has relied on advancements in control and modeling for these purposes [1], [6].

Thermal processes are very important in the fabrication of semiconductor devices. The longer a wafer is kept at elevated temperatures, the higher probability it has of defects. As such, minimizing a metric called the *thermal budget* is very important not only for heating cost purposes,

but also for purity and defect reasons [6]. The thermal budget is calculated as the integral of the product of the diffusivity and the temperature over time. As wafer dimensions have shrunk down into the micron range, there has been an increase in demand on uniform thermal processing. The push to reduce the thermal budget, combined with the tight quality requirements, has given rise to a new technology called *single wafer processing* (SWP). Traditionally, batch processes were used where wafer holders called "boats" loaded many wafers onto a quartz substrate to be placed inside a furnace. Although furnace construction included insulated walls to improve the isothermal nature of the environment inside, wafer uniformity remains an issue. This issue has led to the development of *rapid thermal processing* (RTP) technologies. Single wafer units are better alternatives to meet temperature uniformity and a lower thermal budget; however, they must be able to heat up and cool down quickly in order to compete with the volume output of batch processing. A typical RTP system undergoes three phases: (1) rapid heating on the order of 50-250° C/s, (2) a processing phase of constant temperature, and (3) a rapid cooling phase. Heating is made possible via high powered lamps. The heating lamps are split into zones, and this allows for control flexibility. Finally, optical pyrometers are used to feedback temperature measurements [51].

Several alternative approaches to modeling and control of single wafer RTP systems have been suggested in the literature. From a modeling standpoint, [52] and [53] studied the feasibility of modeling RTP chambers. Review of a *Steag Inc* RTP system with first principles modeling and genetic *nonlinear model predictive control* (NMPC) was proposed in [1]. An adaptive control model was presented in [7]. The authors in [8] studied the thermal behavior of large silicon wafers. Decentralized control approach in the design of PI controllers was used in [9]. The authors in [10] and [54] used *proper orthogonal decomposition* (POD) to reduce the order of an RTP system. A linear quadratic gaussian (LQG) approach to control was taken in [11] and [55]. Furthermore, a run to run approach was taken in [12], while [13] used *internal model control* (IMC). Finally, multivariable and multizone control was presented in [14–16]. A survey of RTP processes was presented in [17].

As observed from the aforementioned literature, first principles-based modeling of the RTP system is best represented by a partial differential equation (PDE) with varying coefficients and nonlinear boundary conditions. However, direct control of such nonlinear system is not seen in literature, nor are there modeling frameworks that present the plant in a control-oriented form. In this paper, we propose a *linear parameter-varying* (LPV) modeling approach that directly and systematically copes with the complex nonlinearities seen in the RTP processes. LPV techniques have gained popularity as they have developed into effective tools to control multi-input multi-output (MIMO) nonlinear systems [3]. Furthermore, the application of these methods has not been explored for thermal processes including RTP systems, for which the well known nonlinear material properties can be exploited in the LPV framework of scheduling variables.

This paper is organized as follows: Section 2.3 describes the process and the first principles-based model of the generic RTP systems. Section 2.4 reintroduces the system as a high-order discretized state-space model. This model is then converted into an LPV model. In Section 2.5, the number of scheduling variables in the high-order LPV model is first reduced using PCA, and then order of the model is reduced using POD in Section 2.6. Section 2.7 uses the reduced LPV model to design a gain scheduled controller. In Section 2.8 the RTP system is modeled, a gain scheduled controller is designed, and closed-loop simulation results show tracking. Section 2.9 draws conclusions and proposes future work.

2.3 RTP Process Description and Modeling

2.3.1 The Typical RTP Setup

In our modeling of RTP systems, we choose to use a single wafer setup as seen in [1], [6], [7], [9], [11]. For a typical RTP system, a concentric lamp array, usually of halogen lamps, is located above a quartz window. The lamp array is divided into zones, and the zone power percentage can be adjusted independently in each zone in order to aid the uniform processing of large wafers. The heating lamps and chamber are cooled by a cooling flow. The wafer is kept rotating in order to

ensure uniformity. Finally, an optical pyrometer located underneath the wafer provides temperature measurement. The setup is illustrated in Figure 2.1.

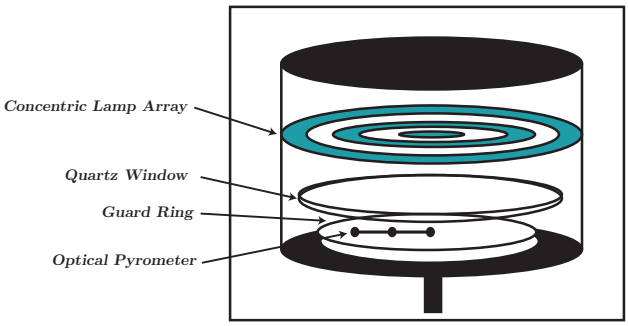

Figure 2.1: Representative single wafer RTP setup.

2.3.2 The First Principles-based Model

The first step in achieving a control-oriented model is to utilize a first principles-based model of the RTP chamber. Energy balance on the wafer in the RTP chamber is given as [1]

$$\rho C(T)\frac{\partial T}{\partial t} = q_k + q_c + q_r, \tag{2.1}$$

where ρ, C, and T are the wafer density, specific heat, and temperature, respectively. Variable t denotes continuous time. The heat transfer rates by conduction, convection, and radiation are denoted by q_k, q_c, and q_r, respectively.

In order to decrease the computational complexity of the model, we first make geometric simplifications. We note that using cylindrical coordinates, the system has rotational symmetry, and hence the full three-dimensional model (r,θ,z) can be reduced to a two-dimensional problem in (r,z). Next, we observe that in order to increase uniformity, the wafer is rotated during the operation, and this allows us to return the problem to Cartesian coordinates by representing the wafer as a radial chord. We use the simplifications and write the energy balance in terms of (x,z) as a *partial differential equation* (PDE) as follows

$$\rho C(T)\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\tilde{k}(T)\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial z}\left(\tilde{k}(T)\frac{\partial T}{\partial z}\right). \tag{2.2}$$

Furthermore, the initial and boundary conditions are given as

$$T(x,z,0) = T_{initial}, \tag{2.3}$$

$$\tilde{k}(T)\frac{\partial T}{\partial x} = 0 \quad \text{at} \quad x = 0, \tag{2.4}$$

$$\tilde{k}(T)\frac{\partial T}{\partial x} = -h_e(T - T_{wall}) \quad \text{at} \quad x = R, \tag{2.5}$$

$$\tilde{k}(T)\frac{\partial T}{\partial z} = F_1\varepsilon_1(T)\sigma(T^4 - T_{cool}^4) + h_w(T - T_{cool}) \text{ at } z = 0, \tag{2.6}$$

$$h_w(x) = h_i + (h_o - h_i)\left(\frac{x}{R}\right)^4, \tag{2.7}$$

$$\tilde{k}(T)\frac{\partial T}{\partial z} = \varepsilon_2(T)Q(x,t) - F_2\varepsilon_2(T)\sigma(T^4 - T_a^4) \text{ at } z = Z, \tag{2.8}$$

where T is the wafer temperature; $T_{initial}$ is the initial wafer temperature; h_w is the overall convective heat transfer coefficient; h_i, h_o, and h_e are the heat transfer coefficients at the center, edge, and wafer edge, respectively [56]; T_{cool} is the temperature of the coolant; T_a is the temperature of the quartz window; T_{wall} is the temperature of side walls; $C(T)$ is the heat capacity; $\tilde{k}(T)$ is the thermal conductivity; σ is the Stefan-Boltzmann constant; ε_1 and ε_2 are the emissivities of the lower and upper wafer surfaces; F_1 and F_2 are the tunable reflective coefficients; x and z are the Cartesian coordinates corresponding to the radial thickness Z, and the radial chord length X; and $Q(x,t)$ is the heat flux as described by $\frac{q(x,t)}{A(x)}$. The heat power $q(x,t)$ is described later in (2.14) and $A(x)$ is the effective wafer area at the chord position.

The initial condition in (2.3) makes the reasonable assumption that the entire wafer starts at a uniform temperature. Next, we assume that the quartz window, the side walls, and cooling temperatures are held constant and equal ($T_a = T_{cool} = T_{wall}$). The boundary condition (2.6) represents the conduction heat losses made with the reactor walls by convection. We use the overall heat transfer coefficient approach as in [56] in order to account for spatial variations. Lastly, the boundary condition at $z = Z$ as described in (2.8) relates the heat transfer in the wafer to the heat generation of the heating lamps and also the heating losses in the quartz window.

Next, we must account for the operation range of the RTP systems. Typical RTP systems range in temperature from 25 to 1200 $^\circ$ C [1]. The material properties of silicon wafers are given in [57]

and the thermal conductivity and heat capacity are given as

$$\tilde{k}(T) = 802.99T^{-1.12}\left[\frac{W}{cmK}\right] \text{ for } T \in [300, 1683]K, \tag{2.9}$$

$$C(T) = 0.641 + 2.473 \times 10^4 \, T \left[\frac{J}{gK}\right] \text{ for } T > 300K. \tag{2.10}$$

Furthermore, the material properties of the emissivity is given by [58]

$$\varepsilon(T) = 0.2662 + 1.8591 \, T^{-0.1996} exp\left[-\frac{1.0359 \, x \, 10^{25}}{T^{8.8328}}\right]. \tag{2.11}$$

For further computational simplicity, we notice that the wafer density can be taken as a constant, $\rho = 2330kg/m^3$, since this density does not strongly depend on temperature. Additionally, this weak temperature dependence allows for a homogeneous energy balance assumption such that (2.2) can be simplified to

$$\rho C(T)\frac{\partial T}{\partial t} = \tilde{k}(T)\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2}\right). \tag{2.12}$$

2.3.3 Modeling Heating Lamp Input Flux

Radiation heat transfer is the main mode heat transfer mechanism that raises the wafer temperature. The lamp array is located directly above the wafer and typically arranged into concentric rings of heating zones. Radiation heat transfer is a complicated heat transfer mode as energy transfer is based on both wavelength and geometry. Therefore, a theoretical model must also account for both diffusive and reflective radiation heat transfer. However, in order to put the model in a form suitable for controller design purposes, we first make the partial simplifying assumption of a diffusive grey body. As seen in (2.11), the emissivity is still a function of temperature, but we relax the condition that it also must be a function of wavelength.

Next, to calculate the heat flux transferred to the wafer, we follow the view factor formula given in [1] that describes the geometric relationship between two areas given as

$$F_{1-2} = \frac{1}{A_1}\int_{A_1}\int_{A_2} \frac{cos(\theta_1)cos(\theta_2)}{\pi S^2} dA_2 dA_1, \tag{2.13}$$

where F_{1-2} is the radiation fraction transmitted from surface 1 to surface 2 and θ_1 and θ_2 are the normal angles at the surfaces while S is the distance between the surfaces, and A_1 and A_2 are the

corresponding surface areas. Following [1], (2.13) is integrated on a differential annular heating ring. We then recast into a generalized form for the multiple zones as

$$q(x,t) = \alpha \cdot \sum_{j=1}^{n} F_{j-x}(x, r_{in}, r_{out}) \cdot q(j), \tag{2.14}$$

where α is a tunable parameter, j represents the ring number, n is the maximum number of zones, r_{in} and r_{out} are the respective radial measurements of the local ring number, and $q(x,t)$ represents the heating ring power.

2.4 Nonlinear Modeling of RTP Systems

The two-dimensional heat equation (2.12) is given on the physical domain $\mathbb{S} = \{x | x \in [0, \chi]\} \cup \{z | z \in [0, \zeta]\}$ and the temporal domain $\mathbb{R} = \{t | t \in [0, \tau]\}$. Now $T: \mathbb{S} \times \mathbb{R} \to \mathbb{T}$ is the space and time dependent temperature. An approximate discrete solution of (2.12) is then represented by

$$T_{i,j}^{k} = T : \hat{\mathbb{S}} \times \hat{\mathbb{R}} \to \mathbb{T}, \tag{2.15}$$

with the finite sets $\hat{\mathbb{S}} = \{s_1, \ldots, s_{mm \times nn}\}$, $\hat{\mathbb{R}} = \{t_1, \ldots, t_K\}$, where $mm \times nn$ is the number of grid points, and K is the number of time samples.

2.4.1 Discretization of the RTP Model

The partial differential equation (PDE) in (2.12) is discretized using a *forward time-center space* (FTCS) discretization method, which gives

$$\rho C(T_{i,j}^{k}) \frac{T_{i,j}^{k+1} - T_{i,j}^{k}}{\Delta t} = \tilde{k}(T_{i,j}^{k}) \left[\frac{T_{i-1,j}^{k} - 2T_{i,j}^{k} + T_{i+1,j}^{k}}{(\Delta x)^2} + \frac{T_{i,j-1}^{k} - 2T_{i,j}^{k} + T_{i,j+1}^{k}}{(\Delta z)^2} \right], \tag{2.16}$$

where Δx and Δz represent the discretization step size in spatial directions, and Δt is the time step; i and j represent the two spatial indices in the x and z dimensions, and k represents the time index. We also discretized the nonlinear boundary conditions subject to (2.5)-(2.8). A simulation result of the discretized system is shown in Figure 2.2 at an arbitrary time instant, where Δx = 1/20, Δz

= 1/4, and $T_{initial}$ = 303 K. The time step Δt is chosen such that it obeys the limits of the FTCS discretization stability restrictions.

$$T^k_{i-1,j} = \frac{\tilde{k}(T^k_{i+1,j})}{\tilde{k}(T^k_{i-1,j})} T^k_{i+1,j} \quad \text{at} \quad x = 0, \tag{2.17}$$

$$T^k_{i+1,j} = \frac{-2\Delta x h_e (T^k_{i,j} - T_{wall}) + \tilde{k}(T^k_{i-1,j}) T^k_{i-1,j}}{\tilde{k}(T^k_{i+1,j})} \quad \text{at} \quad x = R, \tag{2.18}$$

$$T^k_{i,j-1} = \frac{2\Delta z \left[\varepsilon_2 (T^k_{i,j}) \frac{q(x,k)}{A(x)} - F_2 \varepsilon_2 (T^k_{i,j}) \sigma [(T^k_{i,j})^4 - T^4_{cool}] \right]}{\tilde{k}(T^k_{i,j-1})} + \frac{\tilde{k}(T^k_{i,j+1}) T^k_{i,j+1}}{\tilde{k}(T^k_{i,j-1})} \quad \text{at} \quad z = 0, \tag{2.19}$$

$$T^k_{i,j+1} = \frac{-2\Delta z \left[F_1 \varepsilon_1 (T^k_{i,j}) \sigma [(T^k_{i,j})^4 - T^4_{cool}] + h_w(x_i)(T^k_{i,j} - T_{cool}) \right]}{\tilde{k}(T^k_{i,j+1})} + \frac{\tilde{k}(T^k_{i,j-1}) T^k_{i,j-1}}{\tilde{k}(T^k_{i,j+1})} \quad \text{at} \quad z = Z. \tag{2.20}$$

$$\Delta t \leq \frac{1}{2} \frac{\rho C(T) \Delta x^2 \Delta z^2}{k(T)(\Delta x^2 + \Delta z^2)}, \tag{2.21}$$

which is a limit due to the FTCS discretization method. These conditions are chosen to examine the open-loop response to an input signal with typical wafer dimensions [1].

2.4.2 Linear Parameter-varying Model Derivation

It is observed that the system (2.2)-(2.8) is nonlinear, and remains so after discretization. There exists several approaches to transform a nonlinear system represented by

$$x(k+1) = F(x(k), u(k)), \tag{2.22}$$

into a linear model. A well known approach is the Jacobian linearization of (2.22) around an equilibrium trajectory. The resulting linear system will then only describe the local behavior around that trajectory. Another approach is based on rewriting (2.22) into an equivalent form, where the

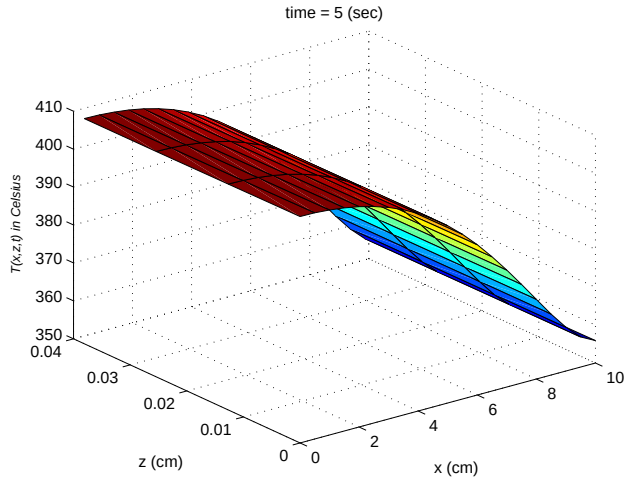

Figure 2.2: Simulation of the open-loop single wafer RTP setup.

nonlinearities can be hidden inside newly defined variables, the so-called *scheduling variables*. Such a model is called a *linear parameter-varying* (LPV) model [3]. A discrete-time LPV model can be represented in state space as

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} A(\theta(k)) & B(\theta(k)) \\ C(\theta(k)) & D(\theta(k)) \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, \tag{2.23}$$

where $y(k)$ represents the control output. We rewrite the nonlinear model into an LPV form (2.23), since this form is suitable for LPV controller synthesis. The state vector $x(k)$ consists of the temperature of the wafer at the discretized locations, with *mm* being the total number of steps in the x direction, and *nn* being the number of steps in the z direction; the state vector is given by

$$x(k) = [x_1(k), \ldots, x_{mm \times nn}(k)]^\top. \tag{2.24}$$

The state vector $x(k)$ is arranged with respect to the spatial coordinates, and thus the structure of the elements of the state vector is as follows

$$x(k) = [T^k_{1,1}, \ldots, T^k_{mm,1}, \ldots, T^k_{1,nn}, \ldots, T^k_{mm,nn}]^\top. \tag{2.25}$$

Next, we define the scheduling variable vector in a similar fashion where $\theta_1(T_{i,j}^k)$-$\theta_4(T_{i,j}^k)$ are derived so that (2.23) is affine in the scheduling variables,

$$\theta(k) = [\theta_1(T_{1,1}^k), \ldots, \theta_4(T_{1,1}^k), \ldots, \theta_1(T_{mm,nn}^k), \ldots, \theta_4(T_{mm,nn}^k)]^\top. \tag{2.26}$$

Remark 1: Each scheduling variable is unique as the scheduling variables are functions of the local temperature at each unique spatial location. We make the definitions θ_1 to θ_4 noting that at each unique spatial location, x_i , z_j, these scheduling variables are unique as they are functions of the local temperature $T_{i,j}^k$.

$$\theta_1(T_{i,j}^k) = \frac{\tilde{k}(T_{i,j}^k)}{C(T_{i,j}^k)}, \tag{2.27}$$

$$\theta_2(T_{i,j}^k) = \frac{\sigma\varepsilon(T_{i,j}^k)(T_{i,j}^k)^3\left(1 - \frac{T_{cool}^4}{(T_{i,j}^k)^4}\right)}{C(T_{i,j}^k)}, \tag{2.28}$$

$$\theta_3(T_{i,j}^k) = \frac{\varepsilon(T_{i,j}^k)}{C(T_{i,j}^k)}, \tag{2.29}$$

$$\theta_4(T_{i,j}^k) = \frac{\left(1 - \frac{T_{cool}}{T_{i,j}^k}\right)}{C(T_{i,j}^k)}. \tag{2.30}$$

Next, to write the model into an affine-in-scheduling-variables high-order LPV state-space model with a large number of scheduling variables, we define the nine distinct zones as shown in Figure 2.3. These zones correspond to the boundary conditions in (2.17)-(2.20), where zones 1, 3, 7 and 9 are a combination of boundary conditions. Each zone temperature is given as:

1	***4***	***7***
2	***5***	***8***
3	***6***	***9***

Figure 2.3: Associated zone numbers of the discretization scheme.

$$
\begin{aligned}
T_1^{k+1} = {} & (1-2\lambda_x\theta_1-2\lambda_z\theta_1-2\lambda_z\Delta z F_2\theta_2)T_{i,j}^k+(2\lambda_x\theta_1)T_{i+1,j}^k \\
& +(2\lambda_z\theta_1)T_{i,j+1}^k+2\lambda_z\Delta_z\theta_3 Q(x,k), && (2.31)\\
T_2^{k+1} = {} & (1-2\lambda_x\theta_1-2\lambda_z\theta_1)T_{i,j}^k+(2\lambda_x\theta_1)T_{i+1,j}^k \\
& +(\lambda_z\theta_1)T_{i,j-1}^k+(\lambda_z\theta_1)T_{i,j+1}^k, && (2.32)\\
T_3^{k+1} = {} & (1-2\lambda_x\theta_1-2\lambda_z\theta_1-2\lambda_z\Delta z h_w(x_i)\theta_4)T_{i,j}^k \\
& +(2\lambda_x\theta_1)T_{i+1,j}^k+(2\lambda_z\theta_1)T_{i,j-1}^k, && (2.33)\\
T_4^{k+1} = {} & (1-2\lambda_x\theta_1-2\lambda_z\theta_1-2\lambda_z\Delta z F_2\theta_2)T_{i,j}^k+(\lambda_x\theta_1)T_{i+1,j}^k \\
& +(\lambda_x\theta_1)T_{i-1,j}^k+(\lambda_z\theta_1)T_{i,j-1}^k+(\lambda_z\theta_1)T_{i,j+1}^k \\
& +2\lambda_z\Delta_z\theta_3 Q(x,k), && (2.34)\\
T_5^{k+1} = {} & (1-2\lambda_x\theta_1-2\lambda_z\theta_1)T_{i+1,j}^k+(\lambda_x\theta_1)T_{i-1,j}^k \\
& +(\lambda_x\theta_1)T_{i+1,j}^k+(\lambda_z\theta_1)T_{i,j-1}^k+(\lambda_z\theta_1)T_{i,j+1}^k, && (2.35)\\
T_6^{k+1} = {} & (1-2\lambda_x\theta_1-2\lambda_z\theta_1-2\lambda_z\Delta z F_2\theta_2-2\lambda_z\Delta z h_w(x_i)\theta_4)T_{i,j}^k \\
& +(\lambda_x\theta_1)T_{i-1,j}^k+(\lambda_x\theta_1)T_{i+1,j}^k+(2\lambda_z\theta_1)T_{i,j-1}^k, && (2.36)\\
T_7^{k+1} = {} & (1-2\lambda_x\theta_1-2\lambda_z\theta_1-2\lambda_x\Delta x h_e\theta_5-2\lambda_z\Delta z F_2\theta_2)T_{i,j}^k \\
& +(2\lambda_x\theta_1)T_{i-1,j}^k+(2\lambda_z\theta_1)T_{i,j+1}^k \\
& +2\lambda_z\Delta_z\theta_3 Q(x,k), && (2.37)\\
T_8^{k+1} = {} & (1-2\lambda_x\theta_1-2\lambda_z\theta_1-2\lambda_x\Delta x h_e\theta_5)T_{i,j}^k \\
& +(2\lambda_x\theta_I)T_{i-1,j}^k+(\lambda_z\theta_1)T_{i,j-1}^k+(\lambda_z\theta_1)T_{i,j+1}^k, && (2.38)\\
T_9^{k+1} = {} & (1-2\lambda_x\theta_1-2\lambda_z\theta_1-2\lambda_x\Delta x h_e\theta_5-2\lambda_z\Delta z F_2\theta_2 \\
& -2\lambda_z\Delta z h_w(x_i)\theta_4)T_{i,j}^k+(2\lambda_x\theta_1)T_{i-1,j}^k \\
& +(2\lambda_z\theta_1)T_{i,j-1}^k, && (2.39)
\end{aligned}
$$

where $T_{\tilde{n}}^k$ represents temperature in zone $\tilde{n}$ at discrete time k with $\tilde{n}=\{1,\ldots,9\}$, $\lambda_x=\frac{\Delta t}{\rho\Delta x^2}$ and $\lambda_z=\frac{\Delta t}{\rho\Delta z^2}$. We use (2.31)-(2.39) to populate the parameter dependent matrices $A(\theta)$, $B(\theta)$, $C(\theta)$,

$D(\theta)$ in (2.23), and we recall that each set of scheduling variables $(\theta_1, \ldots, \theta_4)$ is unique for each discretization point thus giving a large number of scheduling variables. The output matrix $C(\theta)$ is populated with elements corresponding to the spatial locations of the optical pyrometers as seen in Figure 2.1.

In formulating (2.23), the higher the order and the larger the number of scheduling variables in the model, the more accurately the model will represent the original system in (2.2)-(2.8). Hence, a trade-off must be made between model complexity, and the tractability of control design and computational cost. Our objective now becomes to use order reduction techniques in order to achieve a balance between accuracy of the model and the number of scheduling variables.

2.5 LPV Model Reduction Using PCA

The first step in developing a control-oriented LPV model of the RTP system is to reduce the number of scheduling variables through the use of principal component analysis (PCA) [59]. To apply PCA to the LPV scheduling variables data, one first needs to generate and collect data by means of measurements or simulations [60], such that the data covers all regions within the operating range. Given the LPV model (2.23) and assuming that the measurable signals have been sampled at time instants $k \in \{1, 2, \ldots, K\}$, scheduling variables $\theta(k) \in \mathbb{R}^{\tilde{m}}$ are computed and collected in the following $\tilde{m} \times K$ matrix

$$\Theta = \begin{bmatrix} \theta(1) & \cdots & \theta(K) \end{bmatrix} = \begin{bmatrix} f(T_{i,j}^1) & \cdots & f(T_{i,j}^K) \end{bmatrix},$$

where $\tilde{m}$ represents the actual number of scheduling variables and K denotes the number of data samples, with $K \geq \tilde{m}$. PCA is then applied by solving an eigenvalue problem for the covariance matrix. The covariance matrix is given by

$$\bar{C} = \frac{1}{K} \Theta_c \Theta_c^\top,$$

where $\Theta_c = \mathscr{C}(\Theta) = \Theta - \theta_{\text{mean}}$ is the data matrix Θ normalized such that each row of Θ has zero mean. We then solve an eigenvalue problem for the covariance matrix $\bar{C}$, such that $\bar{C} v_i = \tilde{\lambda}_i v_i$, where

$\tilde{\lambda}_i$ and v_i are the i^{th} eigenvalue and eigenvector, respectively. The eigenvectors are then sorted in descending order of their corresponding non-zero eigenvalues, and the m principal components for any test point $\theta(k)$, at a given time sample k, are extracted using

$$\rho(k) = g(T_{i,j}^k) = V_m^\top f(T_{i,j}^k) = V_m^\top \theta(k),$$

where V_m denotes an $\tilde{m} \times \tilde{l}$ matrix whose columns contain the $\tilde{l}$ eigenvectors associated with the first $\tilde{l}$ significant eigenvalues. The approximation of the actual variable $\hat{\theta}(k)$, corresponding to $\rho(k)$, can be easily computed as

$$\hat{\theta}(k) = \mathscr{C}^{-1}(V_m \rho(k)), \tag{2.40}$$

where $\mathscr{C}^{-1}(V_m\rho(k)) = V_m\rho(k) + \theta_{\text{mean}}$. Henceforth, we also drop the time index k for better readability and denote $\rho(k)$ and $\theta(k)$ simply as ρ and θ, respectively. The PCA-based reduced LPV model can be represented as

$$\begin{aligned} x(k+1) &= \hat{A}(\rho)x(k) + \hat{B}(\rho)u(k), \\ y(k) &= \hat{C}(\rho)x(k) + \hat{D}(\rho)u(k). \end{aligned} \tag{2.41}$$

If m equals the number of non-zero eigenvalues, the mapping matrices $\hat{A}(\cdot)$, $\hat{B}(\cdot)$, $\hat{C}(\cdot)$, and $\hat{D}(\cdot)$ are related to the reconstructed scheduling variable $\hat{\theta}$ by [61]

$$\hat{Q}(\rho) = \begin{bmatrix} \hat{A}(\rho) & \hat{B}(\rho) \\ \hat{C}(\rho) & \hat{D}(\rho) \end{bmatrix} = \begin{bmatrix} A(\hat{\theta}) & B(\hat{\theta}) \\ C(\hat{\theta}) & D(\hat{\theta}) \end{bmatrix} = Q(\hat{\theta}). \tag{2.42}$$

We take m to be the number of significant eigenvalues, in which case, $\hat{Q}(\rho)$ will be an approximation of $Q(\hat{\theta})$; what constitutes significance is a user's choice. We write the following:

$$\begin{aligned}
\hat{Q}(\rho) = Q(\hat{\theta}) &= Q_0 + \sum_{i=1}^{\tilde{m}} Q_i \hat{\theta}^i \\
&= Q_0 + \sum_{i=1}^{\tilde{m}} Q_i \left(V_m \rho + \theta_{\text{mean}}\right)^i \\
&= Q_0 + \sum_{i=1}^{\tilde{m}} Q_i \theta_{\text{mean}}^i + \sum_{i=1}^{l} Q_i \left(V_m \rho\right)^i \\
&= \underbrace{Q_0 + \sum_{i=1}^{\tilde{m}} Q_i \theta_{\text{mean}}^i}_{\hat{Q}_0} + \sum_{j=1}^{\tilde{l}} \underbrace{\sum_{i=1}^{\tilde{m}} Q_i [V_m]_{i,j}}_{\hat{Q}_j} \rho^j \\
&= \hat{Q}_0 + \sum_{j=1}^{\tilde{l}} \hat{Q}_j \rho^j,
\end{aligned} \tag{2.43}$$

where θ^i denotes the i^{th} element of the vector θ, and $[V_m]_{i,j}$ denotes the $\{i, j\}$ entry of the matrix V_m. Equation (2.43) is a reduced model which is also affine in the reduced scheduling variables ρ.

2.6 Proper Orthogonal Decomposition

The next step is to reduce the order of the derived LPV model using the *proper orthogonal decomposition* (POD) method. POD delivers a basis for model decomposition in order to extract dominant trends and features [39]. Essentially, POD extracts a set of *orthonormal basis functions* (OBFs) [62], usually with a few modes [39]. To approximate the function of interest over a domain, we write the ensemble into coefficients to be determined,

$$T(x,z,t) \approx \widehat{T}(x,z,t) = \sum_{j=1}^{M} \alpha_j(x,z)\varphi_j(t), \tag{2.44}$$

where α_j's define the set of OBFs, and φ_j's denote the time-dependent coefficients. We employ the method of snapshots [50], which solves an eigenvalue problem and only requires an ensemble of appropriately organized data points [62]. Here we define $\tilde{D} = mm \times nn$. The data needed is captured as $T_{snap} \in \mathbb{R}^{\tilde{D} \times K}$,

$$T_{snap} = \begin{bmatrix} T_1^1 & \cdots & T_1^K \\ \vdots & \ddots & \vdots \\ T_{\tilde{D}}^1 & \cdots & T_{\tilde{D}}^K \end{bmatrix}, \tag{2.45}$$

where $\tilde{D}$ corresponds to the number of discretization points and K corresponds to the number of snapshots. In the finite-dimensional case, POD reduces to a singular value decomposition (SVD) problem as

$$T_{snap} = \Phi \Sigma V^\top = \begin{bmatrix} \Phi_r & \Phi_s \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 & 0 \\ 0 & \Sigma_s & 0 \end{bmatrix} \begin{bmatrix} V_r^\top \\ V_s^\top \end{bmatrix}. \tag{2.46}$$

The columns of Φ from the SVD form the set of basis functions $\{\alpha_1, \cdots, \alpha_{\tilde{D}}\}$. This type of projection captures the most *energy* for the reduced model. In (2.46), $\Phi \in \mathbb{R}^{\tilde{D}\times\tilde{D}}$ and $V \in \mathbb{R}^{K\times K}$ and the sizes of Φ_r, Σ_r, and V_r each correspond to the M dominant singular values chosen. These basis functions, called POD modes, are used to obtain accurate low-order dynamic models via Galerkin projection [62].

Next, we examine the singular values to produce a reduced-order model. A representation of the *energy* that is captured by the reduced-order model is given by the differences in the sum of the squared singular values (2.47). A high percentage of energy preserved is always desired, meaning a larger M, which indicates that the model retains more of the information contained in the original snapshots. The preserved energy percent (PEP) is defined as [60]

$$PEP = 100 \times \frac{\sum_{i=1}^{M} \sigma_i^2}{\sum_{i=1}^{\tilde{D}} \sigma_i^2}, \tag{2.47}$$

where we note that M is the user's choice, and $\tilde{D}$ is the original order of the state-space system. To obtain the reduced-order state-space LPV model, (2.41) is multiplied from both sides by the truncated orthonormal matrix $\Phi_r \in \mathbb{R}^{\tilde{D}\times M}$ as

$$\Phi_r^\top x(k+1) = \Phi_r^\top \hat{A}(\rho) x(k) + \Phi_r^\top \hat{B}(\rho) u(k). \tag{2.48}$$

Recalling that $x(k)$ is the state vector of the original high-order model, the reduced-order state vector becomes

$$x_r(k) = \Phi_r^\top x(k). \tag{2.49}$$

Since each element of $x_r(k)$ is a linear combination of the elements of $x(k)$, substituting (2.49) into (2.48) yields

$$\begin{aligned} x_r(k+1) &= A_r(\rho)x_r(k) + B_r(\rho)u(k) \\ y_r(k) &= C_r(\rho)x_r(k) + D_r(\rho)u(k), \end{aligned} \tag{2.50}$$

with

$$A_r(\rho) = \Phi_r^\top \hat{A}(\rho)\Phi_r, \quad B_r(\rho) = \Phi_r^\top \hat{B}(\rho),$$

$$C_r(\rho) = \hat{C}(\rho)\Phi_r, \qquad D_r(\rho) = \hat{D}(\rho). \tag{2.51}$$

2.7 Linear Parameter-Varying Controller Design

In this section, gain-scheduled controller synthesis using the reduced LPV model is presented.

2.7.1 Polytopic LPV Model

The low-order, low-scheduling variable model in (2.50) can be converted to a polytopic LPV model where the parameter dependent matrices of (2.50) are to be determined and $\rho(k)$ is the scheduling parameter vector. The LPV model can be represented as a linear input-output map

$$P(\rho) = \begin{bmatrix} A_r(\rho) & B_r(\rho) \\ C_r(\rho) & D_r(\rho) \end{bmatrix}. \tag{2.52}$$

Introducing the compact set $\mathscr{P}_\rho \subset \mathscr{R}^{\tilde{l}}$: $\rho(k) \in \mathscr{P}_\rho$, $\forall k > 0$, this set is then the polytope defined by the convex hull

$$\mathscr{P}_\rho := Co\{\rho_{v_1}, \rho_{v_2}, ..., \rho_{v_{n_v}}\}, \tag{2.53}$$

where $n_v = 2^{\tilde{l}}$ is the number of vertices. Next, we note that the system is parameter-affine since the state-space matrices depend affinely on the scheduling parameters as

$$P(\rho) = P_0 + \sum_{i=1}^{\tilde{l}} \rho_i P_i = P_0 + \rho_1 P_1 + \cdots + \rho_{\tilde{l}} P_{\tilde{l}}. \tag{2.54}$$

Since any $\rho(k)$ can be expressed as a convex combination of n_v vertices, this is called a polytopic LPV system where

$$\mathscr{P}(\rho) \in Co\{\rho_{v_1}, \rho_{v_2}, ..., \rho_{v_{n_v}}\} = \sum_{j=1}^{n_v} \xi_j P_j, \tag{2.55}$$

where $\sum_{j=1}^{n_v} \xi_j = 1$, $\xi_j \geq 0$ are the convex coordinates, and P_j's are calculated at corresponding vertex [62]. We note that since (2.50) is the reduced-order model, the low number of scheduling variables allows for tractability in using standard solvers in order to design controllers.

2.7.2 LPV Controller Synthesis

The LPV controller design configuration for the RTP system is shown in Figure 2.4. The polytopic gain-scheduled controller, $K(\rho)$, is designed based on the low-order, low-scheduling variables LPV model $P(\rho)$. In Figure 2.4, r denotes the reference input, u the controller output, T the temperature output of the RTP system, and n the noise introduced into the temperature measurement. We design $K(\rho)$ such that the closed-loop system meets an induced $\mathscr{L}_2$ gain performance with the gain-scheduled controller described as

$$\begin{aligned} K(\rho) : \zeta(k+1) &= A_K(\rho)\zeta(k) + B_K(\rho)e(k), \\ u(k) &= C_K(\rho)\zeta(k) + D_K(\rho)e(k). \end{aligned} \tag{2.56}$$

We see from Figure 2.4 that the temperature measurement is used to calculate the original scheduling variables θ, and then from the PCA analysis, we calculate the reduced-order scheduling variables, which are sent to the controller.

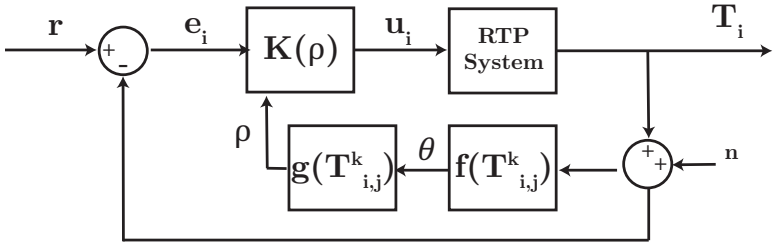

Figure 2.4: Control of the RTP system using an LPV controller $K(\rho)$ designed based on the reduced LPV model.

This control methodology is specific to LPV models with an affine dependence on the scheduling variables ρ that vary within a fixed polytope. Next, we convert Figure 2.4 into the standard lower-fractional transformation (LFT) given in Figure 2.5 where w is the external disturbance, z denotes the controlled outputs, u represents the controller outputs, and y is the measurements.

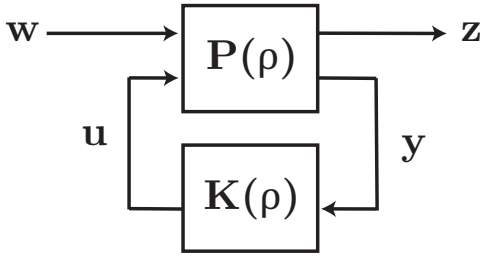

Figure 2.5: Gain-scheduled control design configuration in the lower fractional transformation.

From [4], we define the induced $\mathscr{L}_2$ gain performance as

$$\|T_{wz}\|_{i,2} = \sup_{\rho} \sup_{w \neq 0} \frac{\|z\|_{\mathscr{L}_2}}{\|w\|_{\mathscr{L}_2}}, \tag{2.57}$$

where i denotes the selected induced norm. The gain indicates the worst-case output energy $\|z\|_{\mathscr{L}_2}$ over all bounded energy disturbances $\|w\|_{\mathscr{L}_2}$ for all admissible values of the scheduling variables ρ. The closed-loop LPV system of Figure 2.5 has an induced $\mathscr{L}_2$ gain performance less than γ if

there exists a symmetric positive-definite matrix X such that

$$\begin{bmatrix} A_{cl}^{\top}(\rho)X + XA_{cl}(\rho) & XB_{cl}(\rho) & C_{cl}^{\top}(\rho) \\ \star & -\gamma I & D_{cl}^{\top}(\rho) \\ \star & \star & -\gamma I \end{bmatrix} \prec 0, \tag{2.58}$$

for all admissible trajectories of ρ, where A_{cl}, B_{cl}, C_{cl}, and D_{cl} are the closed-loop state-space system matrices. From (2.58), we see why reducing the number of scheduling variables is critical in LPV applications as the matrix X in the inequality (2.58) is found by solving a finite number of *linear matrix inequalities* (LMIs). We see from the proof in [4] that for polytopic LPV representations with affine dependence on ρ, the inequality condition (2.58) holds for all trajectories of ρ within the polytope, if it holds true at the vertices. This implies that if (2.58) is true, a closed-loop $\mathscr{L}_2$ gain performance, denoted by γ, applies to the entire polytope of scheduling variables.

2.8 Simulation Results and Discussion

We present simulation results starting with the first principle modeling to closed-loop simulations.

2.8.1 Affine LPV Model

First, we followed (2.16)-(2.20) and chose a high discretization of 10 steps in the x direction, and 3 steps in the z direction. We then used (2.31)-(2.39) to write a high-order, high-scheduling variable affine LPV model. The coefficients F_1, F_2, and α were tuned to match the open loop response of the STEAG RTP system in [1]. The input perturbation is given in 2.6 and the response is given in Figure 2.7.

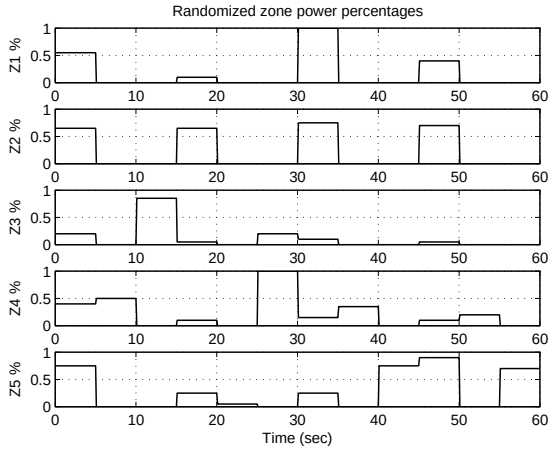

Figure 2.6: The random input trajectory used in [1] to calibrate the tunable model parameters F_1, F_2, and α.

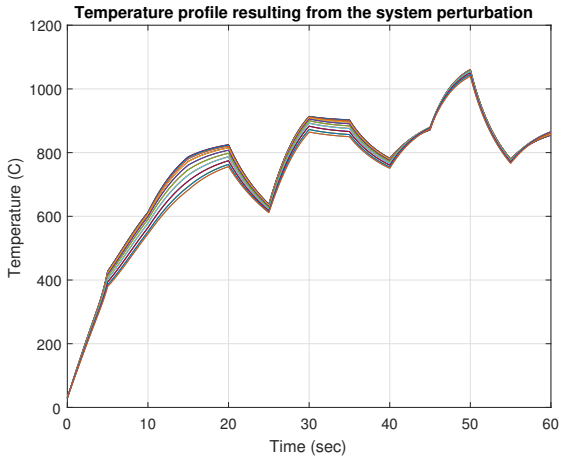

Figure 2.7: The open-loop response of the high-order, high-scheduling variables affine LPV system to the input trajectory of Figure 2.6 tuned to match the Steag CVD RTP System.

2.8.2 SCHEDULING VARIABLE REDUCTION

Next, we proceed with the analysis by generating a low-frequency sinusoidal input with multiple frequency components. This signal was multi-layered to excite the dynamics of the system model. Using this input, we proceeded to follow PCA analysis given in Section 2.5. Figure 2.8 shows the results of the PCA analysis. We choose to map to 3 scheduling variables and retain 95.6% of accuracy.

Remark 2: The LPV framework is a natural framework for RTP modeling since temperature is a readily measureable scheduling variable and the nonlinearities seen are smooth.

Remark 3: All of the scheduling variables share the common thread in that they are all functionals of temperature.

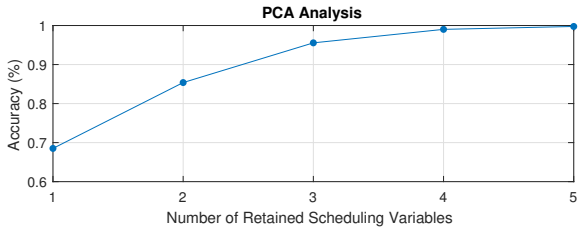

Figure 2.8: Accuracy approximation of (2.40) as a function of the number of scheduling variables.

2.8.3 MODEL ORDER REDUCTION

Next, we used the POD method as described in Section 2.6 to reduce the order of the system. Using (2.47)-(2.51) we reduced the high-order system to a 2^{rd} order system while preserving 98% of the energy. We find good agreement between the reduced-model and the original model.

Remark 4: Since RTP wafer recipes are known *a priori* we can expect to preserve a large amount of energy in the reduced-order system to create a low-order system computationally inexpensive enough to be run in a real time environment.

Figure 2.9 shows tbe comparison between the the low-order, low-scheduling variable system with the nonlinear system. The low-order system was chosen to have 2 POD modes, and 3 scheduling variables. Good agreement is seen around the operating temperatures.

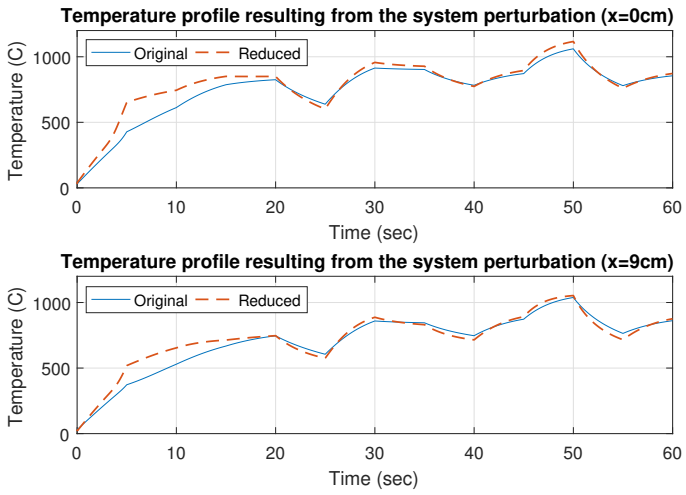

Figure 2.9: Comparison between the low-order, low-scheduling variable system with the nonlinear system. Top: shows a node at the middle of the wafer. Bottom: shows a node at the edge of the wafer.

2.8.4 LPV RTP Controller Design

The design objective to track the reference trajectory given by r. For RTP systems this reference trajectory is a ramp with a large soak time. For controller design purposes, a controller must: have good steady state tracking, with preferably zero steady-state error. Overshoot beyond a few degrees in unacceptable, actuator saturation is plausible, and fast ramp rates are needed. To accomplish this we add loop-shaping filters as in Figure 2.10. We add a first-order low pass filter W_e to impose tracking requirements, and W_u a first-order high pass filter to penalize the control effort on each zone output. Figures 2.11 and 2.12 show the filters respectively. These filters are selected and tuned by trial and error, seeking the minimization of the induced $\mathscr{L}_2$ gain from the external disturbance $w = \begin{bmatrix} r & n \end{bmatrix}^\top$ to the controlled outputs $z = \begin{bmatrix} z_e & z_u \end{bmatrix}^\top$ in order to enforce the performance requirement.

For controller synthesis that is based on polytopic LPV models, the plant input and output matrices, B and C, need to be independent of the scheduling variables [4]. This is not the case for (2.50). Therefore, to put (2.50) in a suitable form for controller synthesis we filter the input with a

low-pass filter of suitable bandwidth. We make note that C and D are not functions of ρ although (2.50) has been formulated in a general sense.

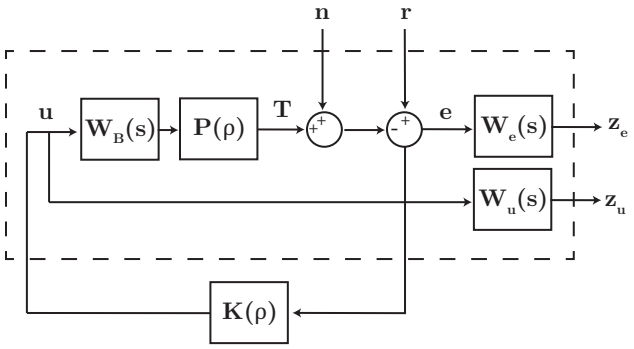

Figure 2.10: Generalized configuration of the closed-loop system composed of the reduced LPV controller and loop-shaping filters.

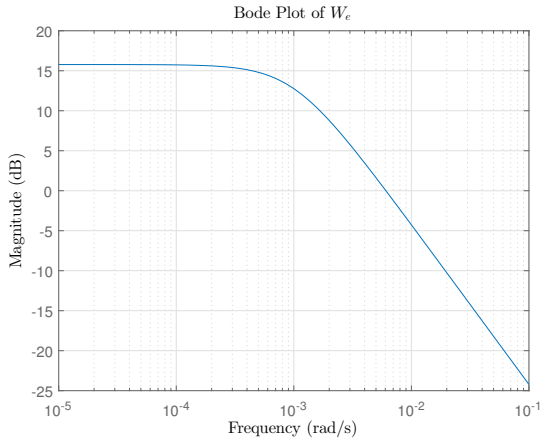

Figure 2.11: Bode plot of the loop shaping filter W_e used to enforce reference tracking.

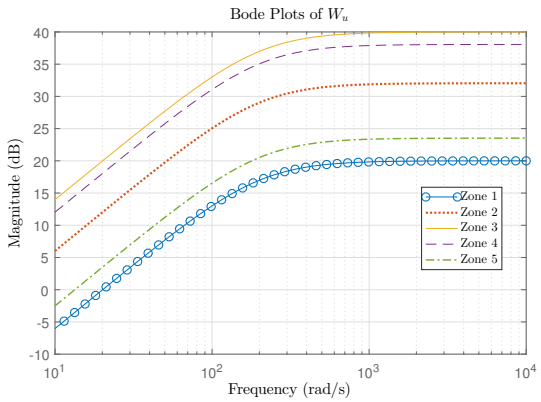

Figure 2.12: Bode plots of the loop shaping filters W_u used to penalize the controller outputs.

2.8.5 Closed Loop Simulation Results

To simulate a realistic RTP process we assume that we only have a single temperature measurement at the middle of the wafer. This temperature can be measured real time, and used to calculate the scheduling variables. From the original scheduling variables we use the PCA analysis to schedule the controller with the low-order scheduling variables in real time. Furthermore, we make the assumption that all scheduling variables can be calculated from the single temperature measurement since the wafer is desired to be at a uniform temperature. Next, we design $K(\rho)$ to track the temperature measurement. The closed-loop simulation results are given in Figure 2.13. For simulation purposes we add noise to the temperature measurement to impose a 15 dB signal-to-noise ratio. As an input reference trajectory we impose a 250 $^\circ$ C ramp with a steady-state of 700 $^\circ$ C. The controller tracks the temperature measurement with only 15 $^\circ$ C of overshoot.

In the closed-loop simulation, saturation blocks were placed after the controller output to simulate that the zone power can only be commanded between 0-100%. Figure 2.14 shows the controller output. Given the saturation of zones 1 and 2, we also design a suitable antiwindup scheme to prevent integrator windup.

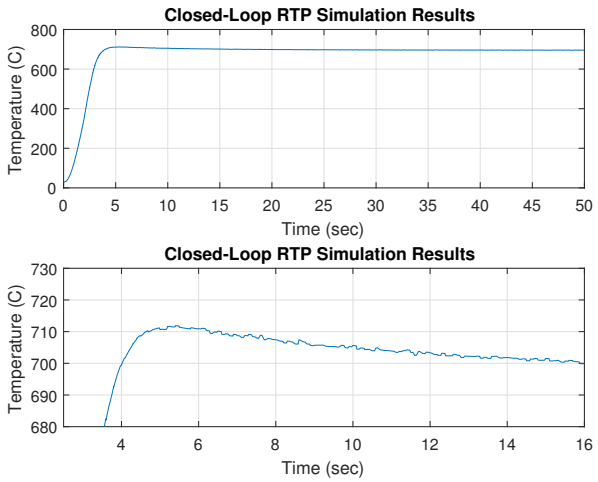

Figure 2.13: Closed-loop simulation results for tracking the pyrometer temperature measurement.

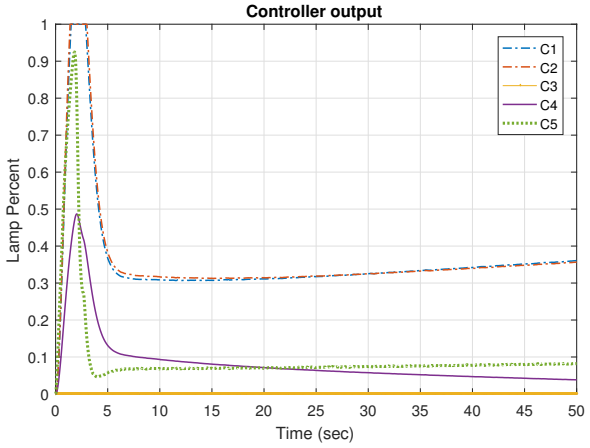

Figure 2.14: Controller output of the gain scheduled controller $K(\rho)$.

Figure 2.15 gives the reduced-order scheduling variables. We note that the reduced-order scheduling variables are synthetic; however, since all of the original variables are temperature dependent, we see the expected result that the scheduling variables settle to a constant value when the temperature in the closed-loop simulation converges.

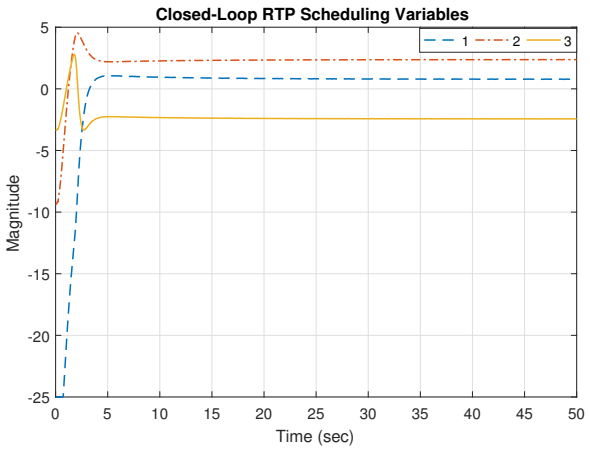

Figure 2.15: Reduced-order scheduling variables from the closed-loop analysis.

Using LPV modeling and gain-scheduled controller design presents a systematic first principles-based procedure. Should the wafer material properties or RTP chamber geometry change, the dynamics of the model can be readily updated, and the controller can be immediately retuned. This is in comparison to purely data driven techniques, e.g., [1], where data is generated from a single wafer setup, and numerical methods are used to fit a model to the data. By comparing the results in [1] which use a nonlinear MPC controller, with the closed-loop simulation shown in Figure 2.13, we see a reduced overshoot and an improved settling time from the gain-scheduled controller. A numerical comparison between the two methods is shown in Table 2.1. We see from the comparison in Table 2.1 that a controller derived from a model that uses the knowledge of the underlying physics provides higher performance.

2.8.6 PID Comparison

Next, for comparison, we design a PID controller to control the reduced-order LPV model. Given that a PID controller is a single input single output (SISO) system, the input to the PID controller is the temperature error and the output is a global lamp power percentage. Since the lamps in the RTP system are divided into different zones, as a solution to better wafer uniformity for SISO

controllers, the literature has suggested different zone power percentages. So ultimately, a lamp's final power percentage is defined to be the product of the global PID controller's output with the local zone power percentage. We use the final zone power percentages presented in [1]. To help properly tune the PID controller, we use MATLAB/Simulink's PID tuning algorithms toolboxes. We also implement an anti-windup scheme given that there exists controller saturation.

First, we allow overshoot, and tune the overshoot to match that of the gain-scheduled controller. A closed-loop response of the system using the PID controller is given in Figure 2.16. A comparison of figures 2.13 and 2.16 show that both controllers achieve a similar overshoot, however, the gain-scheduled controller achieves this in almost half the rise time. We see also that the gain-scheduled controller brings the system to steady-state nearly three times faster. These results justify the extra complexity needed in gain-scheduled controller implementation. Table 2.1 provides a numerical comparison. Since the controller is gain-scheduled, we are able to design a less conservative controller since the dynamics are well modeled by the reduced-order LPV model.

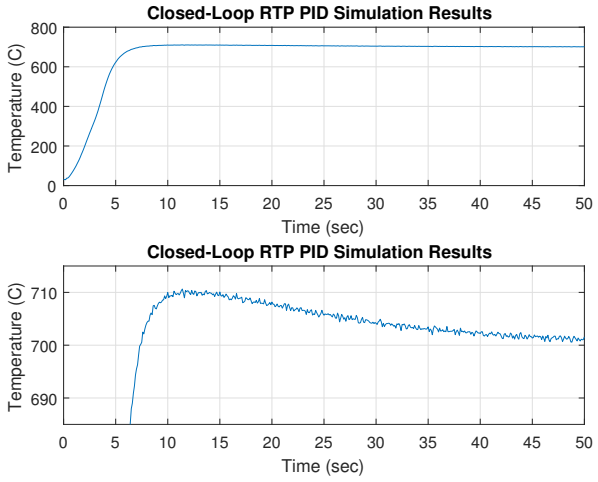

Figure 2.16: Closed-loop response of the RTP system using a PID controller that allows overshoot.

Figure 2.17 shows the PID controller's output for the closed-loop simulation; we see the controller saturates. Since the final lamp powers are a fixed percentages of the PID controller's global output, all zones will follow a scaled response of the controller's output. This is conservative given that each zone is only a percentage of the controller output, and a local zone may not be experiencing

saturation even though the global controller is. Comparing Figure 2.17 with the gain-scheduled controller output in Figure 2.14, the gain-scheduled controller is able to control each zone individually and this results in an improved performance. We see in Figure 2.14 that two of the controller outputs experience saturation, but this does not prevent the gain-scheduled controller from increasing the output in the zones that are not saturating. Figure 2.18 shows the local zone power percentages.

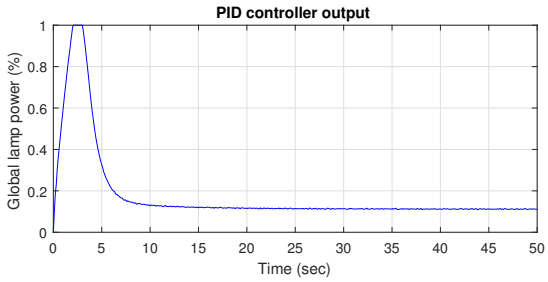

Figure 2.17: Controller output response of the closed-loop RTP system using a PID controller that allows overshoot.

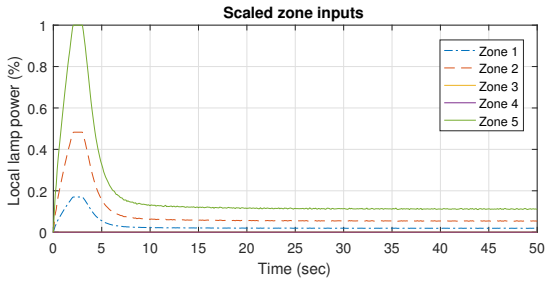

Figure 2.18: Zone inputs of the closed-loop RTP system using a PID controller that allows overshoot.

Due to the slow settling time seen in Figure 2.16, we tune the PID gains to prevent overshoot, which prevents the wafer from remaining above the set point temperature. This response is shown in Figure 2.19.

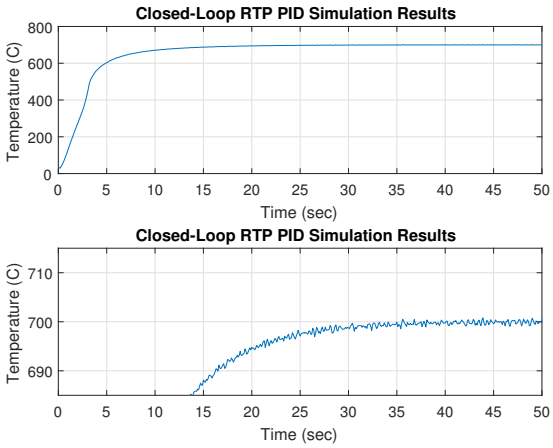

Figure 2.19: Closed-loop response of the RTP system using a PID controller that prevents overshoot.

Overshoot is prevented, but at the cost of a slow rise time and a longer controller saturation period. The corresponding controller output is given in Figure 2.20. Again, we find controller saturation, and remark that since the final lamp powers are fixed percentages of the PID controller's global output, all zones will follow a scaled response of the controller's output. Figure 2.21 shows the local zone power percentages.

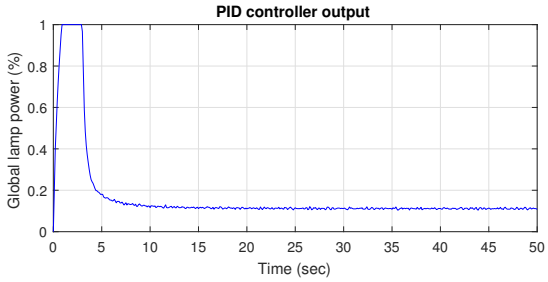

Figure 2.20: Controller output response of the closed-loop RTP system using a PID controller that prevents overshoot.

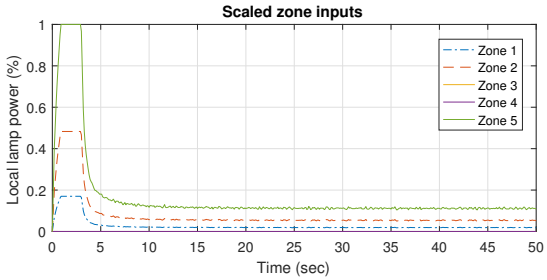

Figure 2.21: Zone inputs of the closed-loop RTP system using a PID controller that allows overshoot.

Table 2.1: Closed-loop performance comparison between the gain-scheduled controller designed using the LPV model, the nonlinear MPC controller designed using the data-driven model of [1], and the well-tuned PID controllers, where PID 1 allows overshoot and PID 2 does not allow.

	Gain-Scheduled	Nonlinear MPC	PID 1	PID 2
Rise Time (sec)	4	$\sim$7	7.35	37.5
Overshoot ($^\circ$ C)	12	$\sim$30	10	0
Settling Time to $\pm$ 1 $^\circ$ C (sec)	15.2	$\sim$18	49.3	30.5

2.9 Concluding Remarks

In this paper, we used a well-established first principles-based modeling approach to develop an affine LPV model for rapid thermal processes. We further employed PCA and POD to reduce the dimensionality of the LPV model into a form tractable for controller design purposes. Reducing the number of scheduling variables is desirable since it affects exponentially the number of *linear matrix inequality* (LMI) constraints required to be solved for LPV control synthesis. Finally, we designed a gain-scheduled controller to track a temperature reference profile. Closed-loop simulation results show that a brief overshoot of less than 12 $^\circ$ C subsides to good steady-state tracking.

CHAPTER 3

LINEAR PARAMETER-VARYING APPROACH FOR MODELING RAPID THERMAL PROCESSES

1

[1]Trudgen, Mark, Syed Z. Rizvi, and Javad Mohammadpour. "Linear parameter-varying approach for modeling rapid thermal processes." In American Control Conference (ACC), 2016, pp. 3243-3248. American Automatic Control Council (AACC), 2016.

ABSTRACT

In the present paper, a new approach is presented to model *rapid thermal processing* (RTP) systems. Within the past decade, RTP has achieved acceptance as the mainstream technology for semiconductor manufacturing. Thermal processing is one of the most efficient ways to control the phase-structure properties; moreover, the time duration of RTP systems reduces the so-called *thermal budget* significantly compared to the traditional methods. RTP implementation is based on the use of light from heating lamps to provide a heat flux. This process is highly nonlinear due to the radiative heat transfer and material properties. By invoking the first principles-based models, we develop in this paper a *linear parameter-varying* (LPV) model to directly account for all the nonlinearities within the system. The model is discretized into a high-order affine LPV system; thereafter, *principal component analysis* (PCA) method is utilized to reduce the number of the LPV model's scheduling variables, followed by the use of *proper orthogonal decomposition* (POD) for model order reduction. Finally, simulations demonstrate that the low-order LPV model, which is in a form suitable for controller design purposes, retains the properties of the original full-order model.

3.1 INTRODUCTION

Embedded deep in the heart of all electrical applications are *integrated circuits* (IC), primarily composed of semiconductor devices made from a sequence of batch processes. With the continual developments in IC technology, we see an increase in the demand for performance improvements in terms of both quality variables and output yield resulting from the use of larger diameter silicon wafers [17]. To achieve these increased yields, precise uniform temperature control of a wafer is of paramount consideration. As such, the semiconductor industry has relied on advancements in control and modeling for these purposes [1], [6].

Thermal processes are very important in the fabrication of semiconductor devices. The longer a wafer is kept at elevated temperatures, the higher probability it has of defects. As such, mini-

mizing a metric called the *thermal budget* is very important not only for heating cost purposes, but also for purity and defect reasons [6]. The thermal budget is calculated as the integral of the product of the diffusivity and the temperature over time. As wafer dimensions have shrunk down into the micron range, there has been an increase in demand on uniform thermal processing. The push to reduce the thermal budget, combined with the tight quality requirements, has given rise to a new technology called *single wafer processing* (SWP). Traditionally, batch processes were used where wafer holders called "boats" loaded many wafers onto a quartz substrate to be placed inside a furnace. Although furnace construction included insulated walls to improve the isothermal nature of the environment inside, wafer uniformity remains an issue. This issue has led to the development of *rapid thermal processing* (RTP) technologies. Single wafer units are better alternatives to meet temperature uniformity and a lower thermal budget; however, they must be able to heat up and cool down quickly in order to compete with the volume output of batch processing. A typical RTP system undergoes three phases: (1) rapid heating on the order of 50-200° C/s, (2) a processing phase of constant temperature, and (3) a rapid cooling phase. Heating is made possible via high powered lamps. The heating lamps are split into zones, and this allows for control flexibility. Finally, optical pyrometers are used to feedback temperature measurements.

Several alternative approaches to modeling and control of single wafer RTP systems have been suggested in the literature. Review of a *Steag Inc* RTP system with first principles modeling and genetic *nonlinear model predictive control* (NMPC) was proposed in [1]. An adaptive control model was presented in [7]. The authors in [8] studied the thermal behavior of large silicon wafers. Decentralized control approach in the design of PI controllers was used in [9]. The authors in [10] used *proper orthogonal decomposition* (POD) to reduce the order of an RTP system. A linear quadratic gaussian (LQG) approach to control was taken in [11]. Furthermore, a run to run approach was taken in [12], while [13] used *internal model control* (IMC). Finally, multivariable and multizone control was presented in [14–16]. A survey of RTP processes was presented in [17].

As observed from the aforementioned literature, first principles-based modeling of the RTP system is best represented by a partial differential equation (PDE) with varying coefficients and

nonlinear boundary conditions. However, direct control of such nonlinear system is not seen in literature, nor are there modeling frameworks that present the plant in a control-oriented form. In this paper, we propose a *linear parameter-varying* (LPV) modeling approach that directly and systematically copes with the complex nonlinearities seen in the RTP processes. LPV techniques have gained popularity as they have developed into effective tools to control multi-input multi-output (MIMO) nonlinear systems [3]. Furthermore, the application of these methods has not been explored for thermal processes including RTP systems, for which the well known nonlinear material properties can be exploited in the LPV framework of scheduling variables.

This paper is organized as follows: Section 3.2 describes the process and the first principles-based model of the generic RTP systems. Section 3.3 reintroduces the system as a high-order discretized state-space model. This model is then converted into an LPV model. In Section 3.4, the number of scheduling variables in the high-order LPV model is first reduced using PCA, and then order of the model is reduced using POD in Section 3.5. Section 3.6 shows simulation results and Section 3.7 draws conclusions.

3.2 RTP Process Description and Modeling

3.2.1 The Typical RTP Setup

In our modeling of RTP systems, we choose to use a single wafer setup as seen in [1], [6], [7], [9], [11]. For a typical RTP system, a concentric lamp array, usually of halogen lamps, is located above a quartz window. The lamp array is divided into zones, and the zone power percentage can be adjusted independently in each zone in order to aid the uniform processing of large wafers. The heating lamps and chamber are cooled by a cooling flow. The wafer is kept rotating in order to ensure uniformity. Finally, an optical pyrometer located underneath the wafer provides temperature measurement. The setup is illustrated in Figure 3.1.

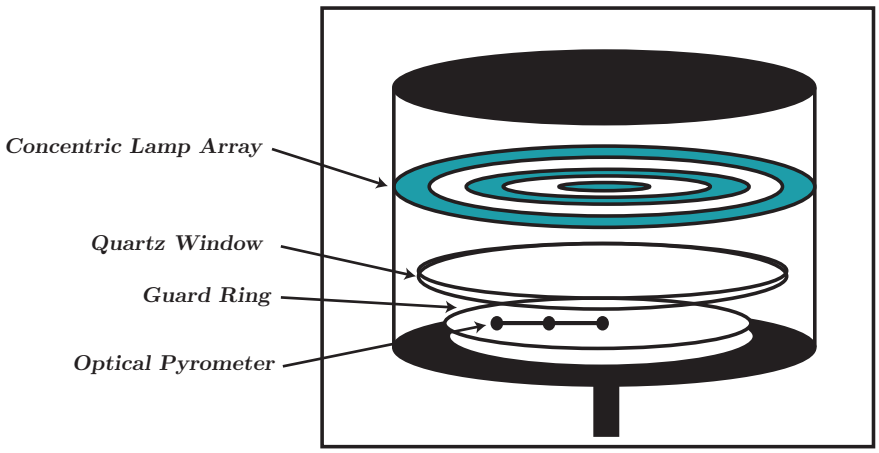

Figure 3.1: Representative single wafer RTP setup.

3.2.2 The First Principles-based Model

The first step in achieving a control-oriented model is to utilize a first principles-based model of the RTP chamber. Energy balance on the wafer in the RTP chamber is given as [1]

$$\rho C(T)\frac{\partial T}{\partial t} = q_k + q_c + q_r, \tag{3.1}$$

where ρ, C, and T are the wafer density, specific heat, and temperature, respectively. Variable t denotes continuous time. The heat transfer rates by conduction, convection, and radiation are denoted by q_k, q_c, and q_r, respectively.

In order to decrease the computational complexity of the model, we first make geometric simplifications. We note that using cylindrical coordinates, the system has rotational symmetry, and hence the full three-dimensional model (r,θ,z) can be reduced to a two-dimensional problem in (r,z). Next, we observe that in order to increase uniformity, the wafer is rotated during the operation, and this allows us to return the problem to Cartesian coordinates by representing the wafer as a radial chord. We use the simplifications and write the energy balance in terms of (x,z) as a *partial differential equation* (PDE) as follows

$$\rho C(T)\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\tilde{k}(T)\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial z}\left(\tilde{k}(T)\frac{\partial T}{\partial z}\right). \tag{3.2}$$

Furthermore, the initial and boundary conditions are given as

$$T(x,z,0) = T_{initial}, \tag{3.3}$$

$$\tilde{k}(T)\frac{\partial T}{\partial x} = 0 \quad \text{at} \quad x = 0, \tag{3.4}$$

$$\tilde{k}(T)\frac{\partial T}{\partial x} = -h_e(T - T_{wall}) \quad \text{at} \quad x = R, \tag{3.5}$$

$$\tilde{k}(T)\frac{\partial T}{\partial z} = F_1\varepsilon_1(T)\sigma(T^4 - T_{cool}^4) + h_w(T - T_{cool}) \text{ at } z = 0, \tag{3.6}$$

$$h_w(x) = h_i + (h_o - h_i)\left(\frac{x}{R}\right)^4, \tag{3.7}$$

$$\tilde{k}(T)\frac{\partial T}{\partial z} = \varepsilon_2(T)Q(x,t) - F_2\varepsilon_2(T)\sigma(T^4 - T_a^4) \text{ at } z = Z, \tag{3.8}$$

where T is the wafer temperature; $T_{initial}$ is the initial wafer temperature; h_w is the overall convective heat transfer coefficient; h_i, h_o, and h_e are the heat transfer coefficients at the center, edge, and wafer edge, respectively [56]; T_{cool} is the temperature of the coolant; T_a is the temperature of the quartz window; T_{wall} is the temperature of side walls; $C(T)$ is the heat capacity; $\tilde{k}(T)$ is the thermal conductivity; σ is the Stefan-Boltzmann constant; ε_1 and ε_2 are the emissivities of the lower and upper wafer surfaces; F_1 and F_2 are the tunable reflective coefficients; x and z are the Cartesian coordinates corresponding to the radial thickness Z, and the radial chord length X; and $Q(x,t)$ is the heat flux as described by $\frac{q(x,t)}{A(x)}$. The heat power $q(x,t)$ is described later in (3.14) and $A(x)$ is the effective wafer area at the chord position.

The initial condition in (3.3) makes the reasonable assumption that the entire wafer starts at a uniform temperature. Next, we assume that the quartz window, the side walls, and cooling temperatures are held constant and equal ($T_a = T_{cool} = T_{wall}$). The boundary condition (3.6) represents the conduction heat losses made with the reactor walls by convection. We use the overall heat transfer

coefficient approach as in [56] in order to account for spatial variations. Lastly, the boundary condition at $z = Z$ as described in (3.8) relates the heat transfer in the wafer to the heat generation of the heating lamps and also the heating losses in the quartz window.

Next, we must account for the operation range of the RTP systems. Typical RTP systems range in temperature from 25 to 1200° C [1]. The material properties of silicon wafers are given in [57] and the thermal conductivity and heat capacity are given as

$$\tilde{k}(T) = 802.99T^{-1.12}\left[\frac{W}{cmK}\right] \text{ for } T \in [300, 1683]K, \tag{3.9}$$

$$C(T) = 0.641 + 2.473 \times 10^4 T \left[\frac{J}{gK}\right] \text{ for } T > 300K. \tag{3.10}$$

Furthermore, the material properties of the emissivity is given by [58]

$$\varepsilon(T) = 0.2662 + 1.8591\ T^{-0.1996} exp\left[-\frac{1.0359\ x\ 10^{25}}{T^{8.8328}}\right]. \tag{3.11}$$

For further computational simplicity, we notice that the wafer density can be taken as a constant, $\rho = 2330kg/m^3$, since this density does not strongly depend on temperature. Additionally, this weak temperature dependence allows for a homogeneous energy balance assumption such that (3.2) can be simplified to

$$\rho C(T)\frac{\partial T}{\partial t} = \tilde{k}(T)\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2}\right). \tag{3.12}$$

3.2.3 Modeling Heating Lamp Input Flux

Radiation heat transfer is the main mode heat transfer mechanism that raises the wafer temperature. The lamp array is located directly above the wafer and typically arranged into concentric rings of heating zones. Radiation heat transfer is a complicated heat transfer mode as energy transfer is based on both wavelength and geometry. Therefore, a theoretical model must also account for both diffusive and reflective radiation heat transfer. However, in order to put the model in a form suitable for controller design purposes, we first make the partial simplifying assumption of a diffusive grey body. As seen in (3.11), the emissivity is still a function of temperature, but we relax the condition that it also must be a function of wavelength.

Next, to calculate the heat flux transferred to the wafer, we follow the view factor formula given in [1] that describes the geometric relationship between two areas given as

$$F_{1-2} = \frac{1}{A_1} \int_{A_1} \int_{A_2} \frac{cos(\theta_1)cos(\theta_2)}{\pi S^2} dA_2 dA_1, \tag{3.13}$$

where F_{1-2} is the radiation fraction transmitted from surface 1 to surface 2 and θ_1 and θ_2 are the normal angles at the surfaces while S is the distance between the surfaces, and A_1 and A_2 are the corresponding surface areas. Following [1], (3.13) is integrated on a differential annular heating ring. We then recast into a generalized form for the multiple zones as

$$q(x,t) = \alpha \cdot \sum_{j=1}^{n} F_{j-x}(x, r_{in}, r_{out}) \cdot q(j), \tag{3.14}$$

where α is a tunable parameter, j represents the ring number, n is the maximum number of zones, r_{in} and r_{out} are the respective radial measurements of the local ring number, and $q(x,t)$ represents the heating ring power.

3.3 Nonlinear Modeling of RTP Systems

The two-dimensional heat equation (3.12) is given on the physical domain $\mathbb{S} = \{x | x \in [0, \chi]\} \cup \{z | z \in [0, \zeta]\}$ and the temporal domain $\mathbb{R} = \{t | t \in [0, \tau]\}$. Now $T: \mathbb{S} \times \mathbb{R} \rightarrow \mathbb{T}$ is the space and time dependent temperature. An approximate discrete solution of (3.12) is then represented by

$$T_{i,j}^{k} = T : \hat{\mathbb{S}} \times \hat{\mathbb{R}} \rightarrow \mathbb{T}, \tag{3.15}$$

with the finite sets $\hat{\mathbb{S}} = \{s_1, \ldots, s_{mm \times nn}\}$, $\hat{\mathbb{R}} = \{t_1, \ldots, t_K\}$, where $mm \times nn$ is the number of grid points, and K is the number of time samples.

3.3.1 DISCRETIZATION OF THE RTP MODEL

The partial differential equation (PDE) in (3.12) is discretized using a *forward time-center space* (FTCS) discretization method, which gives

$$\rho C(T_{i,j}^k)\frac{T_{i,j}^{k+1}-T_{i,j}^k}{\Delta t} = \tilde{k}(T_{i,j}^k)\left[\frac{T_{i-1,j}^k - 2T_{i,j}^k + T_{i+1,j}^k}{(\Delta x)^2} + \frac{T_{i,j-1}^k - 2T_{i,j}^k + T_{i,j+1}^k}{(\Delta z)^2}\right], \tag{3.16}$$

where Δx and Δz represent the discretization step size in spatial directions, and Δt is the time step; i and j represent the two spatial indices in the x and z dimensions, and k represents the time index. We also discretized the nonlinear boundary conditions subject to (3.5)-(3.8). A simulation result of the discretized system is shown in Figure 3.2 at an arbitrary time instant, where Δx = 1/20, Δz = 1/4, and $T_{initial}$ = 303 K. The time step Δt is chosen such that it obeys the limits of the FTCS discretization stability restrictions.

These conditions are chosen to examine the open-loop response to an input signal with typical wafer dimensions [1].

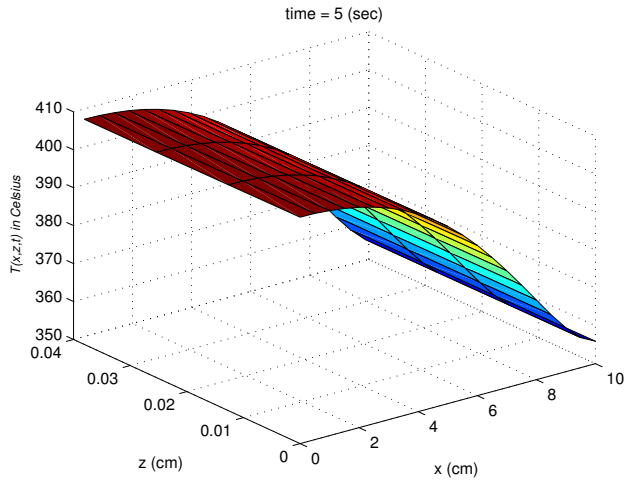

Figure 3.2: Simulation of the open-loop single wafer RTP setup.

3.3.2 Linear Parameter-varying Model Derivation

It is observed that the system (3.2)-(3.8) is nonlinear, and remains so after discretization. There exists several approaches to transform a nonlinear system represented by

$$x(k+1) = F(x(k), u(k)), \tag{3.17}$$

into a linear model. A well known approach is the Jacobian linearization of (3.17) around an equilibrium trajectory. The resulting linear system will then only describe the local behavior around that trajectory. Another approach is based on rewriting (3.17) into an equivalent form, where the nonlinearities can be hidden inside newly defined variables, the so-called *scheduling variables*. Such a model is called a *linear parameter-varying* (LPV) model [3]. A discrete-time LPV model can be represented in state space as

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} A(\theta(k)) & B(\theta(k)) \\ C(\theta(k)) & D(\theta(k)) \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, \tag{3.18}$$

where $y(k)$ represents the control output. We rewrite the nonlinear model into an LPV form (3.18), since this form is suitable for LPV controller synthesis. The state vector $x(k)$ consists of the temperature of the wafer at the discretized locations, with *mm* being the total number of steps in the x direction, and *nn* being the number of steps in the z direction; the state vector is given by

$$x(k) = [x_1(k), \ldots, x_{mm \times nn}(k)]^\top. \tag{3.19}$$

The state vector $x(k)$ is arranged with respect to the spatial coordinates, and thus the structure of the elements of the state vector is as follows

$$x(k) = [T^k_{1,1}, \ldots, T^k_{mm,1}, \ldots, T^k_{1,nn}, \ldots, T^k_{mm,nn}]^\top. \tag{3.20}$$

Next, we define the scheduling variable vector in a similar fashion where $\theta_1(T^k_{i,j})$-$\theta_4(T^k_{i,j})$ are derived so that (3.18) is affine in the scheduling variables,

$$\theta(k) = [\theta_1(T^k_{1,1}), \ldots, \theta_4(T^k_{1,1}), \ldots, \theta_1(T^k_{mm,nn}), \ldots, \theta_4(T^k_{mm,nn})]^\top. \tag{3.21}$$

Remark 1: Each scheduling variable is unique as the scheduling variables are functions of the local temperature at each unique spatial location.

In formulating (3.18), the higher the order and the larger the number of scheduling variables in the model, the more accurately the model will represent the original system in (3.2)-(3.8). Hence, a trade-off must be made between model complexity, and the tractability of control design and computational cost. Our objective now becomes to use order reduction techniques in order to achieve a balance between accuracy of the model and the number of scheduling variables.

3.4 LPV Model Reduction Using PCA

First, we reduce the number of scheduling variables through the use of principal component analysis (PCA) [59]. To apply PCA to the LPV scheduling variables data, one first needs to generate and collect data by means of measurements or simulations [61], such that the data covers all regions of operation within the operating range. Given the LPV model (3.18) and assuming that the measurable signals have been sampled at time instants $k = \{1, 2, \ldots, K\}$, scheduling variables $\theta(k) \in \mathbb{R}^{\tilde{l}}$ with $\tilde{l} = 4 \times mm \times nn$ are computed and collected in the following $\tilde{l} \times K$ matrix

$$\Theta = \begin{bmatrix} \theta(1) & \cdots & \theta(K) \end{bmatrix} = \begin{bmatrix} f(T_{i,j}^{1}) & \cdots & f(T_{i,j}^{K}) \end{bmatrix},$$

where $\tilde{l}$ represents the actual number of scheduling variables and K denotes the number of data samples, with $K \geq \tilde{l}$. PCA is then applied by solving an eigenvalue problem for the covariance matrix $\Theta\Theta^{\top}$. The covariance matrix is given by

$$\bar{C} = \frac{1}{K}\Theta_c\Theta_c^{\top},$$

where $\Theta_c = \mathscr{C}(\Theta) = \Theta - \theta_{\text{mean}}$ is the data matrix Θ normalized such that each row of Θ has zero mean. We then solve an eigenvalue problem for the covariance matrix $\bar{C}$, such that $\bar{C}v_i = \tilde{\lambda}_i v_i$, where $\tilde{\lambda}_i$ and v_i are the i^{th} eigenvalue and eigenvector, respectively. The eigenvectors are then sorted in descending order of their corresponding non-zero eigenvalues, and the m principal components for any test point $\theta(k)$, at a given time sample k, are extracted using

$$\rho(k) = g(T_{i,j}^k) = V_m^\top f(T_{i,j}^k) = V_m^\top \theta(k),$$

where V_m denotes an $\tilde{l} \times m$ matrix whose columns contain the m eigenvectors associated with the first m significant eigenvalues. The approximation of the actual variable $\hat{\theta}(k)$, corresponding to $\rho(k)$, can be easily computed as

$$\hat{\theta}(k) = \mathscr{C}^{-1}(V_m \rho(k)), \tag{3.22}$$

where $\mathscr{C}^{-1}(V_m\rho(k)) = V_m\rho(k) + \theta_{\text{mean}}$. Henceforth, we also drop the time index k for better readability and denote $\rho(k)$ and $\theta(k)$ simply as ρ and θ. The PCA-based reduced model can be represented as

$$\begin{aligned} x(k+1) &= \hat{A}(\rho)x(k) + \hat{B}(\rho)u(k), \\ y(k) &= \hat{C}(\rho)x(k) + \hat{D}(\rho)u(k). \end{aligned} \tag{3.23}$$

If m equals the number of non-zero eigenvalues, the mapping matrices $\hat{A}(\cdot)$, $\hat{B}(\cdot)$, $\hat{C}(\cdot)$, and $\hat{D}(\cdot)$ are related to the reconstructed scheduling variable $\hat{\theta}$ by [61]

$$\hat{Q}(\rho) = \begin{bmatrix} \hat{A}(\rho) & \hat{B}(\rho) \\ \hat{C}(\rho) & \hat{D}(\rho) \end{bmatrix} = \begin{bmatrix} A(\hat{\theta}) & B(\hat{\theta}) \\ C(\hat{\theta}) & D(\hat{\theta}) \end{bmatrix} = Q(\hat{\theta}). \tag{3.24}$$

We take m to be the number of significant eigenvalues, in which case, $\hat{Q}(\rho)$ will be an approximation of $Q(\hat{\theta})$; what constitutes significance is a user's choice. We write the following:

$$\begin{aligned}
\hat{Q}(\rho) = Q(\hat{\theta}) &= Q_0 + \sum_{i=1}^{\tilde{l}} Q_i \hat{\theta}^i \\
&= Q_0 + \sum_{i=1}^{\tilde{l}} Q_i \left(V_m \rho + \theta_{\text{mean}}\right)^i \\
&= Q_0 + \sum_{i=1}^{\tilde{l}} Q_i \theta_{\text{mean}}^i + \sum_{i=1}^{l} Q_i \left(V_m \rho\right)^i \\
&= \underbrace{Q_0 + \sum_{i=1}^{\tilde{l}} Q_i \theta_{\text{mean}}^i}_{\hat{Q}_0} + \sum_{j=1}^{m} \underbrace{\sum_{i=1}^{\tilde{l}} Q_i [V_m]_{i,j}}_{\hat{Q}_j} \rho^j \\
&= \hat{Q}_0 + \sum_{j=1}^{m} \hat{Q}_j \rho^j,
\end{aligned} \tag{3.25}$$

where θ^i denotes the i^{th} element of the vector θ, and $[V_m]_{i,j}$ denotes the $\{i, j\}$ entry of the matrix V_m. (3.25) is a reduced model also affine in the reduced scheduling variables ρ.

3.5 Proper Orthogonal Decomposition

The next step is to reduce the order of the derived LPV model using the *proper orthogonal decomposition* (POD) method. POD delivers a basis for model decomposition in order to extract dominant trends and features [39]. Essentially, POD extracts a set of *orthonormal basis functions* (OBF) [62], usually with a few modes [39]. To approximate the function of interest over a domain, we write the ensemble into coefficients to be determined,

$$T(x,z,t) \approx \widehat{T}(x,z,t) = \sum_{j=1}^{M} \alpha_j(x,z)\varphi_j(t), \tag{3.26}$$

where $\tilde{D}$ corresponds to the number of discretization steps and K corresponds to the number of snapshots. In the finite-dimensional case, POD reduces to an SVD problem. This is done by making use of SVD as

$$T_{snap} = \Phi \Sigma V^\top = \begin{bmatrix} \Phi_r & \Phi_s \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 & 0 \\ 0 & \Sigma_s & 0 \end{bmatrix} \begin{bmatrix} V_r^\top \\ V_s^\top \end{bmatrix}. \tag{3.27}$$

The columns of Φ from the SVD form the set of basis functions $\{\alpha_1, \cdots, \alpha_{\tilde{D}}\}$. This type of projection captures the most *energy* for the reduced model. In (3.27), $\Phi \in \mathbb{R}^{\tilde{D}\times\tilde{D}}$ and $V \in \mathbb{R}^{K\times K}$

and the sizes of Φ_r, Σ_r, and V_r each correspond to the M dominant singular values chosen. These basis functions, called POD modes, are used to obtain accurate low-order dynamic models via Galerkin projection [62].

Next, we examine the singular values to produce a reduced-order model. A representation of the *energy* that is captured by the reduced-order model is given by the differences in the sum of the squared singular values (3.28). A high percentage of energy preserved is always desired, meaning a larger M, which indicates that the model retains more of the information contained in the original snapshots. The preserved energy percent (PEP) is defined as [63]

$$PEP = 100 \times \frac{\sum_{i=1}^{M} \sigma_i^2}{\sum_{i=1}^{N} \sigma_i^2}, \tag{3.28}$$

where we note that M is the user's choice, and N is the original order of the state-space system. To obtain the reduced-order LPV state-space model, (3.18) is multiplied from both sides by the truncated orthonormal matrix $\Phi_r \in \mathbb{R}^{\tilde{D}\times M}$ as

$$\Phi_r^\top x(k+1) = \Phi_r^\top \hat{A}(\rho)x(k) + \Phi_r^\top \hat{B}(\rho)u(k). \tag{3.29}$$

Recalling that $x(k)$ is the state vector of the original high-order approximation, the reduced-order state vector becomes

$$x_r(k) = \Phi_r^\top x(k). \tag{3.30}$$

Since each element of $x_r(k)$ is a linear combination of the elements of $x(k)$, substituting (3.30) into (3.29) yields

$$\begin{aligned} x_r(k+1) &= A_r(\rho)x_r(k) + B_r(\rho)u(k) \\ y_r(k) &= C_r(\rho)x_r(k) + \hat{D}(\rho)u(k), \end{aligned} \tag{3.31}$$

with

$$A_r(\rho) = \Phi_r^\top \hat{A}(\rho)\Phi_r, \quad B_r(\rho) = \Phi_r^\top \hat{B}(\rho), \quad C_r(\rho) = \hat{C}(\rho)\Phi_r. \tag{3.32}$$

3.6 Simulation Results and Discussion

We present simulation results comparing the nonlinear model with the reduced-order LPV model with a low number of scheduling variables.

Remark 2: The LPV framework is a natural framework for RTP modeling since temperature is a readily measureable scheduling variable and the nonlinearities seen are smooth.

Remark 3: All of the scheduling variables share the common thread in that they are all functionals of temperature.

3.6.1 Simulation Results

In our nonlinear simulation we used $\Delta x = \frac{1}{20}$ and $\Delta z = \frac{1}{3}$ and the same geometry setup in [1]. Due to the very high number of initial scheduling variables we averaged the temperature across three zones of the wafer, thus beginning with 12 scheduling variables. Using the PCA analysis described in Section IV we reduced the number of scheduling variables to 3 while retaining 97% of the energy. Figure 3.3 shows a sample projection from the reduced scheduling variables back onto the original high-order space.

Next we used the POD method as described in Section V to reduce the order of the system. Using (3.28)-(3.32) we reduced the high-order system to a 3^{rd} order system while preserving 99% of the energy.

Remark 4: Since RTP wafer recipes are known *a priori* we can expect to preserve a large amount of energy in the reduced-order system to create a low-order system computationally inexpensive enough to be run in a real time environment.

Next, we compare the nonlinear model with the LPV model that has been reduced in order and scheduling variables. Using a random zone power percentage input signal to both models, we see in Figure 3.4 from two sample discretization points that the LPV model is a good representation.

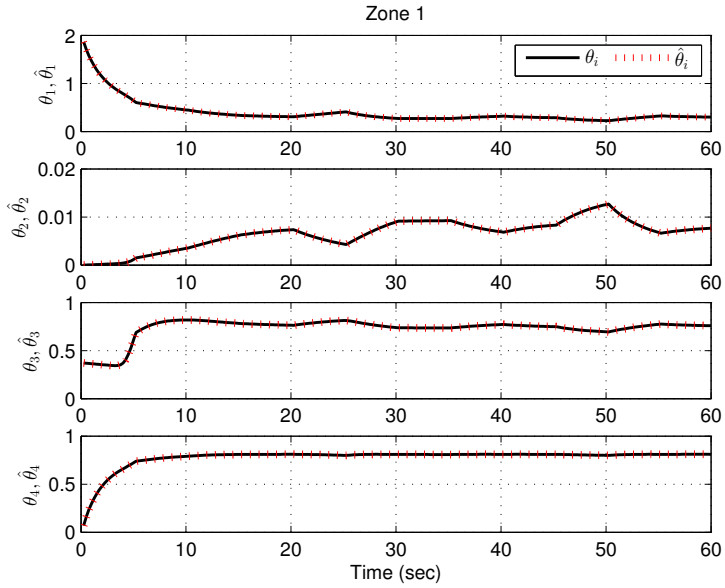

Figure 3.3: Projection of the reduced scheduling variables back onto the original high-order space. The solid line represents the original scheduling data θ, and dotted line represents the projected scheduling data $\hat{\theta}$.

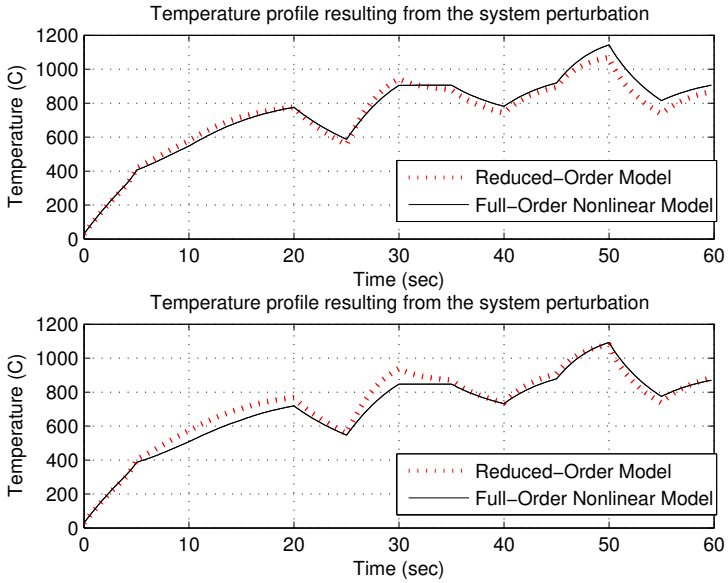

Figure 3.4: Comparison of full-order nonlinear model vs. the low-order LPV model at *two representative node locations*.

3.7 Concluding Remarks

In this paper, we used a first principles-based modeling approach to develop an affine LPV model for rapid thermal processes. PCA and POD were used to reduce the complexity of the LPV model into a form tractable for controller design purposes. Reducing the number of scheduling variables is desirable since it affects exponentially the number of *linear matrix inequality* (LMI) constraints required to be solved for LPV controller synthesis. Finally, using open-loop simulation results, we observed an agreement between the high-order nonlinear model and the reduced-order LPV model.

CHAPTER 4

ROBUST COOPERATIVE ADAPTIVE CRUISE CONTROL DESIGN AND VALIDATION FOR CONNECTED VEHICLES

1

[1]Trudgen, Mark, Miller, Rebecca, and Javad Mohammadpour. "Robust Cooperative Adaptive Cruise Control Design and Validation for Connected Vehicles." Submitted to the Journal of Intelligent Transportation Systems, November 2016.

ABSTRACT

Cooperative adaptive cruise control (CACC) is an emerging technology that takes advantage of onboard sensors and wireless technology working together in order to achieve smaller inter-vehicle following distances than human drivers can provide. The CACC platoon following schemes reported in the literature aim at ensuring string stability, meaning that disturbances are attenuated down the stream of the platoon of vehicles. They also give much lower headway values than adaptive cruise control technology is able to, thus providing a noticeable improvement in road throughput. CACC technology, however, relies on real-time acceleration data from a leading vehicle in which the difficulties and the inexact nature of wireless communication data transfer present various challenges to CACC implementation. The complexity of modern vehicles and their associated onboard computational burdens also grant that certain parameters can only be estimated to be within a range. In order to implement a string stable CACC platoon following scheme while also overcoming the inherent challenges of wireless communication and uncertain internal model parameters, we design an $\mathscr{H}_\infty$ controller that is robust to all aforementioned uncertainties. We implement this controller on a laboratory-scale test bed and we particularly show that the controller design is able to account for communication shortcomings. Inclusive in our design is also an $\mathscr{L}_2$-gain performance based anti-windup compensator to account for saturation limitations in our test bed.

4.1 INTRODUCTION

Connected vehicles are an example of a modern day cyber physical systems (CPS) that through the use of Cooperative Adaptive Cruise Control (CACC) can provide an innovative solution to the traffic congestion problem [18]. Traffic is becoming an increasing problem in today's world as congestion in many urban areas is growing at a much faster rate than the traditional means of traffic alleviation can assuage [19]. CACC is a technology that seeks to reduce traffic congestion by means of achieving higher traffic flow rates using advanced control systems to safely reduce

the allowable *headway* time between vehicles [20]. A widespread advantage of CACC over traditional means of increasing traffic throughput, i.e., road construction, is that CACC has the potential to be implemented on any car without the additional high costs and delays associated with road construction projects [21].

CACC technology is an extension of Adaptive Cruise Control (ACC), which in turn is an extension of conventional cruise control (CCC), a technology traditionally used to regulate a vehicle at a constant highway speed [22]. ACC extends the CCC technology by regulating the so-called *headway* distance between vehicles that are arranged together in a platoon [23]. ACC employs radar (or lidar) sensors to measure the relative velocity and displacement with the preceding vehicle, and a longitudinal control framework is then implemented to space the vehicles to an appropriate *headway* [22] by adjusting the acceleration and deceleration of the vehicle. CACC extends the ACC technology by adding inter-vehicle wireless communication [24]. This extension enables smaller *headway* distances, which is critical for platoon technology to have a noticeable impact on traffic mitigation [21, 25].

According to the *2010 Highway Capacity Manual*, a study observing human drivers showed that the maximum flow rate for a multi-lane highway (at 60 mi/h) equates to 1.1 seconds of *headway* [21]. Herein lies the main drawback of ACC technology, that is the smallest stable *headway* is larger than the average time-gap that human drivers naturally exhibit [19, 25], thus justifying the need for CACC technology. The vehicles that are virtually connected to each other through CACC technology must ensure an important metric called string stability [2]. This concept was first introduced in [26] and later extended in [27], which led to the development of systems using the nearest neighbor as a measurement. Essentially, string stability is a requirement that all disturbances introduced in the string be attenuated as they propagate in the downstream direction [23, 27]. String stability is essential to ensuring the safety and feasibility of the string [2]. Not only do any disturbances in position, velocity or acceleration create increased energy consumption, these disturbances must also be mitigated in order to prevent the so-called *ghost traffic jams* [23], or even in extreme cases, an accident [28]; hence, a control design formulation that can explicitly

account for string stability inherently meets design objectives and exterminates the need for any ad-hoc *a posteriori* tuning to achieve string stability. This notion of string stability has been studied in several aspects such as Lyapunov stability, and input-output stability; however, these methods lack the consideration of a measure of performance as seen in [29, 30], which give a frequency-domain approach for controller synthesis.

Several approaches have been undertaken in designing a controller for a platoon of vehicles. The system model considered to describe the vehicular motion is usually a thrid-order nonlinear model [31, 32], where subsequently the plant is linearized by the use of feedback linearization method. For the control design using the linearized model, several CACC experimental results have been reported, e.g., in [23, 24, 33]. These recent works show the promise in using CACC. Indeed, several aspects of CACC technologies have been studied. The authors in [28] developed a sampled data approach to CACC design in the presence of sensors and actuator failures and [34] studied strategies for worst case sensor failure scenarios. Model predictive control (MPC) has also gained attention as a way to cast the CACC problem in a framework that can directly optimize fuel economy. An MPC based CACC approach was designed for heavy duty vehicles, such as tractor trailer trucks as [35], where smaller *headway* distances can be sacrificed for better fuel economies as traffic throughput may not be the primary objective as is the case with urban rush hour highway demands. CACC can also be viewed in light of the communication as a networked control system where the effects of sampling, hold, and network delays can be taken into account. An $\mathscr{H}_\infty$ formulation of network controlled problems is given in [36]. Still, other works have investigated time-varying communication delays and communication structures beyond the classical architecture as in [37].

This paper offers an $\mathscr{H}_\infty$ controller design framework that is robust to uncertainties in both the acceleration received from the wireless communication and the internal vehicle parameters. We also extend the CACC design problem to include an anti-windup controller that optimizes $\mathscr{L}_2 - gain$ performance. In our formulation, we synthesized the CACC controller using the induced energy-to-energy gain (or $\mathscr{H}_\infty$ norm) due to the presence of uncertainty [5].

This paper is structured as follows: Section 2 describes the platoon following technologies and how the nonlinear model governing vehicles is linearized. We also illustrate simulation results comparing different adaptive cruise control technologies showing why CACC is preferred over ACC. Section 3 explains the design of our robust $\mathscr{H}_\infty$ controller, whose order will then be reduced while still retaining the desired robust properties. Section 4 describes our laboratory-scale test bed that is used to appropriately model the CACC driving scenario. We then present simulation results using the test bed and validate those results with experimental testing. Section 5 draws conclusions and provides insights into further research.

4.2 Cooperative Adaptive Cruise Control Technology

The design of various longitudinal adaptive cruise control strategies has been studied in the literature (see, e.g., [22] and references therein). Figure 4.1 shows a representative view of a typical string of vehicles equipped with cooperative adaptive cruise control (CACC), where the lead car of the string sets a trajectory to follow and communicates its acceleration a_0 only to the following vehicle. Alongside the communicated acceleration, the following vehicle is equipped with onboard sensors to measure the relative distance and velocity. Position sensing is typically done via the use of radar (or lidar) technology [23]. In considering a platoon, the distance between vehicles is broken into three segments: d_i is the desired static distance between vehicles, hv_i is the product of the minimum headway required and the velocity of the i^{th} vehicle, and finally δ_i is an additional spacing parameter. The i^{th} vehicle is said to be in the correct positioning when $\delta_i = 0$. More specifically, δ_i, the spacing policy, is given as [28]

$$\delta_i = q_{i-1} - q_i - L_i - hv_i - d_0, \tag{4.1}$$

where h is the time gap (headway), d_0 is a given minimum distance and L_i is the length of the i^{th}

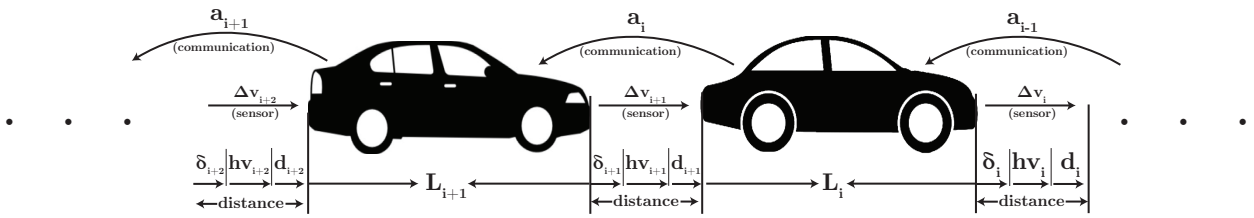

Figure 4.1: A string of vehicles equipped with cooperative adaptive cruise control technology.

vehicle. The system dynamics can be represented as [31, 32]

$$\begin{aligned}\dot{\delta}_i &= v_{i-1} - v_i - h\dot{v}_i,\\ \Delta\dot{v}_i &= a_{i-1} - a_i,\\ \dot{a}_i &= f_i(v_i, a_i) + g_i c_i,\end{aligned} \tag{4.2}$$

where g_i is given as

$$g_i = \frac{1}{\tau_i m_i}, \tag{4.3}$$

where m_i represents the i^{th} vehicle's mass, and τ_i is the engine time-constant of the i^{th} vehicle. The above model is nonlinear due to the nonlinear function $f_i(v_i, a_i)$ described as

$$f_i(v_i, a_i) = -\frac{1}{\tau_i}\left[\dot{v}_i + \frac{\sigma A_i c_{di}}{2m_i} v_i^2 + \frac{d_{mi}}{m_i}\right] - \frac{\sigma A_i c_{di} v_i a_i}{m_i}, \tag{4.4}$$

where $\frac{\tau_i A_i c_{di}}{2m_i}$ is the air resistance, d_{mi} is the mechanical drag, c_{di} is the drag coefficient and σ is the specific mass of the air. To linearize the above nonlinear system dynamics, the following control law is adopted [31, 32]

$$c_i = u_i m_i + \frac{\sigma A_i c_{di} v_i^2}{2} + d_{mi} + \tau_i \sigma A_i c_{di} v_i a_i, \tag{4.5}$$

where u_i is the new control input signal to be designed for the closed-loop system where $c_i < 0$ and $c_i \geq 0$ correspond to brake and throttle actions, respectively. Using (4.5) results in a feedback linearization, which combined with (4.2) gives

$$\dot{a}_i(t) = -\frac{a_i(t)}{\tau_i} + \frac{u_i(t)}{\tau_i}. \tag{4.6}$$

Since $a_{i-1}(t)$ is sent from the preceding vehicle, a communication delay θ_i is induced so the acceleration arriving at the i^{th} vehicle is $a_{i-1}(t-\theta_i)$. Writing the CACC model in the state-space form gives [28]

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_{i1} u_i(t) + B_{i2} w_i(t-\theta_i) \\ y_i(t) &= \left[x_i^T(t), w_i(t)\right]^T, \end{aligned} \tag{4.7}$$

where θ_i is the communication delay, $x_i = [\delta_i, \Delta v_i, a_i]^T$ is the state vector, $w_i(t) = a_{i-1}(t)$, and $y_i(t) = [\delta_i, \Delta v_i, a_i, w_i]^T$ is the output vector, and additionally,

$$A_i = \begin{bmatrix} 0 & 1 & -h \\ 0 & 0 & -1 \\ 0 & 0 & -1/\tau_i \end{bmatrix}, \quad B_{i1} = \begin{bmatrix} 0 \\ 0 \\ 1/\tau_i \end{bmatrix}, \quad B_{i2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{4.8}$$

We follow [23, 24, 64] in assuming a low-level linearizing feedback controller. The system in (4.8) gives the linearization for the i^{th} vehicle, and the overall system is hence a decentralized platoon.

4.2.1 Block Diagram Representation for Connected Vehicles

Assuming the linearized plant dynamics, we can cast the CACC design problem into a block diagram representation. For the i^{th} vehicle, we use the following notation: q_{i-1} denotes the preceding vehicle's position, q_i denotes the local position, e_i is the error signal inputted into the controller $K(s)$ and u_i is the *desired* acceleration (that is used as an input to the linearizing controller, see, e.g., [28]). Finally, d_i denotes an added static following distance, and L_i is the length of the i^{th} vehicle. In addition, $G_i(s)$ represents the system transfer function, and $H(s)$ describes the spacing

policy given as

$$G_i(s) = \frac{q_i(s)}{u_i(s)} = \frac{1}{s^2(\tau_i s + 1)} e^{-\phi_i s}, \tag{4.9}$$

$$H(s) = hs + 1, \tag{4.10}$$

where τ_i is the engine time constant, ϕ_i is the internal time delay, and h is the headway. We note that the linearized model adequately describes the dynamics provided that linearizing accleration controller takes into account parameters given in (4.5) [23].

Next, by introducing a dedicated short range communication (DRSC) protocol between vehicles, the leading vehicle's acceleration can be communicated to the following vehicle. As this signal is transmitted through communication channel, there is a delay represented in the frequency domain as

$$D(s) = e^{-\theta s}, \tag{4.11}$$

where θ is the delay associated with the wireless communication [23, 65]. By adding a stabilizing controller, $K(s)$, and using the communicated acceleration as a feedforward term, the block diagram given between $u_{i-1}(s)$ and $u_i(s)$ is shown in Figure 4.2. Without the loss of generality, L_i = d_i = 0 can be assumed.

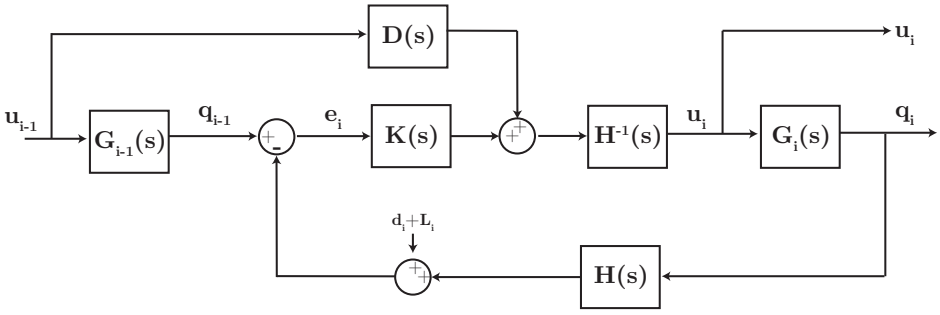

Figure 4.2: CACC block diagram.

4.2.2 String Stability

From the configuration shown in Figure 4.2, we define the transfer function from u_{i-1} to u_i as

$$\Gamma_{CACC}(s) = \frac{1}{H(s)} \frac{D(s) + G_{i-1}(s)K_i(s)}{1 + G_i(s)K_i(s)}. \tag{4.12}$$

However, as in [2] we will assume homogeneity in the string such that $G_{i-1}(s) = G_i(s)$. A complete discussion of string stability is given in [30]. With respect to (4.12) it suffices to limit the discussion of string stability to ensuring that

$$||\Gamma_{CACC}(j\omega)||_{\mathscr{H}_\infty} \leq 1, \tag{4.13}$$

where $||\cdot||_{\mathscr{H}_\infty}$ denotes the system's $\mathscr{H}_\infty$ norm, is met. We recall that the overall goal is to reduce the *headway* while remaining string stable. As noted in [25] for this technology to have a noticeable impact on traffic mitigation, a *headway* significantly smaller than 1.1 sec must be realized as this is the average headway achieved by human drivers [19].

From Figure 4.2, we see that setting $D(s) = 0$ would result in an adaptive cruise control (ACC) scheme. Other smart cruise control schemes exist; the authors in [65] used an onboard estimator to estimate u_{i-1} as a way to gracefully transition between CACC and ACC schemes. To validate the need of CACC versus the previously mentioned ACC and so-called "dCACC" schemes, we designed a proportional-derivative (PD) controller with a headway value of $h = 0.6$ sec and tested the three schemes. Figure 4.3 shows the corresponding Bode plots of the three schemes, again showing the need for CACC. It is seen from Figure 4.3 that the additional complexity needed in CACC implementation is justified given the ability to achieve lower headway values. It is also seen from Figure 4.3 that an onboard estimate of the previous vehicle's acceleration cannot guarantee string stability as effectively as CACC can.

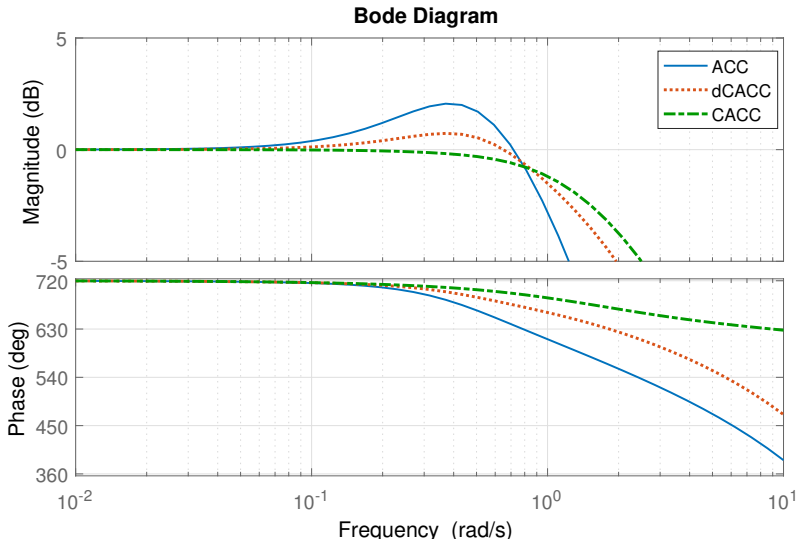

Figure 4.3: Frequency response associated with ACC, dCACC and CACC for a headway of $h = 0.6$ sec.

4.3 Robust CACC Design

In this section, we discuss the design of a robust CACC system in the framework of robust $\mathscr{H}_\infty$ control. To this purpose, we first introduce the sources of uncertainty and describe how to quantify them.

4.3.1 Sources of Uncertainty

There are several reasons to incorporate robustness into a control design framework as there usually exist several sources of uncertainty within any dynamic system. There are always parameters that are only approximately known or are modestly in error. Also, linear models may only be adequate for a small operating range, and original measurements taken to find parameters have inherent errors despite calibration. As we developed our model through system identification methods, at high frequencies the structure of the model can become unknown and uncertainties in parameters always arise. Finally, there might be uncertainties within the controller [5]. There are several

different approaches to model uncertainties, which could be classified under structured or unstructured uncertainties [5].

With respect to our CACC application, we expect uncertainty in all our identified parameters. In our laboratory-scale test bed described in Section 4, τ will vary based on the battery voltage level and the wear on the motor windings. Also, since we consider a lumped parameter model, the exact value of τ for each motor is unknown. We also expect varying time delays due to the computations and the communication packet-handling protocol.

4.3.2 CACC Block Diagram Reformulation

In order to account for the acceleration delay, due to the wireless communication, we reformulate the block diagram shown in Figure 4.2 to Figure 4.4. First, we introduce the same delay, $D(s)$, to the controller output as experienced by the communication delay. While delays are parasitic to controller design and should as a general rule be avoided, the means justify the end as this block diagram reformulation now allows us to account for communication delay in our controller design which is critical to CACC.

From Figure 4.3 we recall that only CACC is able to achieve string stability at low headway values. In typical communication schemes, data loss and an uneven communication time is to be expected. If there is not a dedicated microcontroller for communication, as in our laboratory test bed, this non-uniform delay is even more common. By designing a robust controller we are able to withstand the delays and minor packet losses without having to pull back the CACC scheme into either a dCACC or ACC scheme to maintain string stability with a low headway value. This is crucial as low headway values are needed to realize traffic congestion gains [18, 25]. The issue of implementation where the controller and the communication are synced to share the same delay timing is achieve via microcontroller programming. *In fact, the main novelty in this work is the introduction of robust control to account for varying parameters and a reformulation of Figure 4.2 to account for all uncertain parameters.*

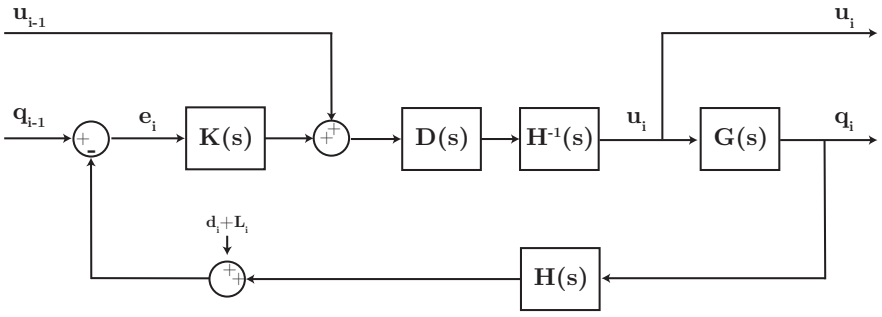

Figure 4.4: CACC block diagram reformulation.

4.3.3 Representing Uncertainty in the CACC Design Framework

For the block diagram shown in Figure 4.4, the parameters ϕ, τ, and θ are assumed to have the nominal values of $\bar{\tau} = 0.14$ sec., $\bar{\phi} = 0.018$ sec., and $\bar{\theta} = 0.02$ sec. for our test bed. We also consider a variation in the range of $\tau \in [0.10, 0.17]$, $\phi \in [0.018, 0.10]$, and $\theta \in [0.02, 0.10]$. To guarantee the closed-loop system stability in the presence of the model uncertainty associated with the above uncertain parameters, we first represent the lumped parameter *multiplicative uncertainty* as shown in Figure 4.5 and equation (4.14).

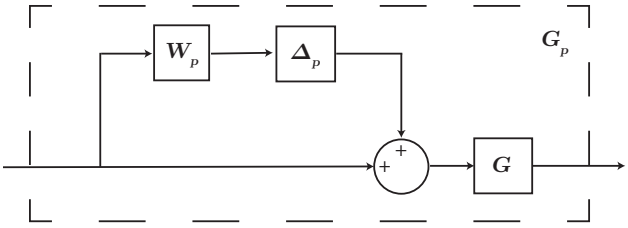

Figure 4.5: Lumped parameter multiplicative uncertainty.

$$G_p(s) = G_0(s)(1 + W_p(s)\Delta_p(s)), \tag{4.14}$$

where $G_p(s)$ represents the perturbed model, $G_0(s)$ represents the nominal model, $\|\Delta_p\|_\infty \leq 1$, and W_p represents the lumped uncertainties transfer function that satisfies [5]

$$\left|\frac{G_p(j\omega)-G_0(j\omega)}{G_0(j\omega)}\right| \leq |W_p(j\omega)|, \tag{4.15}$$

for any frequency ω. We then let τ, ϕ and θ vary over each respective parameter set. Using a fine grid, we plotted the left hand side of (4.15) on a Bode plot shown in Figure 4.6, where in (4.15), $G_p(s)$ is taken as the perturbed plant and $G(s)$ is fixed as the plant designed at $\bar{\tau}$, $\bar{\phi}$, and $\bar{\theta}$. Then, a filter, $W_p(s)$, was fitted to the Bode plot according to (4.15). This results in the following high-pass filter

$$W_p(s) = \frac{2.85s}{s+7}. \tag{4.16}$$

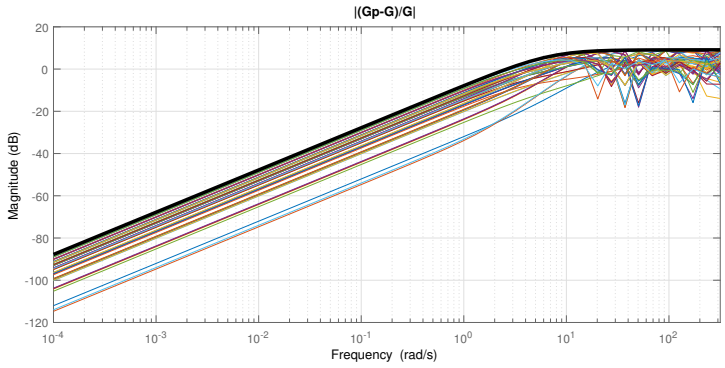

Figure 4.6: Bode plots to find the multiplicative uncertainty weight.

4.3.4 Loop Shaping for $\mathscr{H}_\infty$ Control Design

Next, we use the S/KS loop shaping approach [5] to design a robust controller that can guarantee tracking with zero steady-state error and a low control effort for a headway value of 0.35 sec. The value $h = 0.35$ sec was chosen as it is significantly lower than the human comparison of $h = 1.1$ sec. This headway allows us to achieve both string stability and robustness. The corresponding block diagram in Figure 4.7 depicts how disturbances and noise signals affect the closed-loop system. Using this block diagram setup, the string stability requirement can be directly handled within the $\mathscr{H}_\infty$ framework. In standard loop shaping, weight W_e is tuned to penalize tracking error at low frequencies. The weight W_e is selected to be a low pass filter, tuned to eliminate the steady-state

error. Weight W_v is tuned to penalize controller output, and is selected to be a high pass filter. The filters are tuned as

$$W_e(s) = 5 \cdot \frac{0.075}{s+0.9}, \qquad (4.17) \qquad W_v(s) = 0.4 \cdot \frac{s}{s+50}. \qquad (4.18)$$

Next, we select the *desired* acceleration, u_i, as an exogenous output signal [2]. Writing the transfer function between the exogenous input, i.e., the previous vehicle's acceleration u_{i-1}, and the *desired* acceleration u_i yields

$$T_i(s) = \frac{u_i(s)}{u_{i-1}(s)}. \qquad (4.19)$$

If $||T_i(j\omega)|| \leq 1$ for any ω, we have achieved string stability. The weight W_p is a high pass filter used to model the multiplicative uncertainties as discussed in the previous section. Note that as in [5], this formulation of W_p allows us to capture the variation in delays in our system as uncertainty inside W_p.

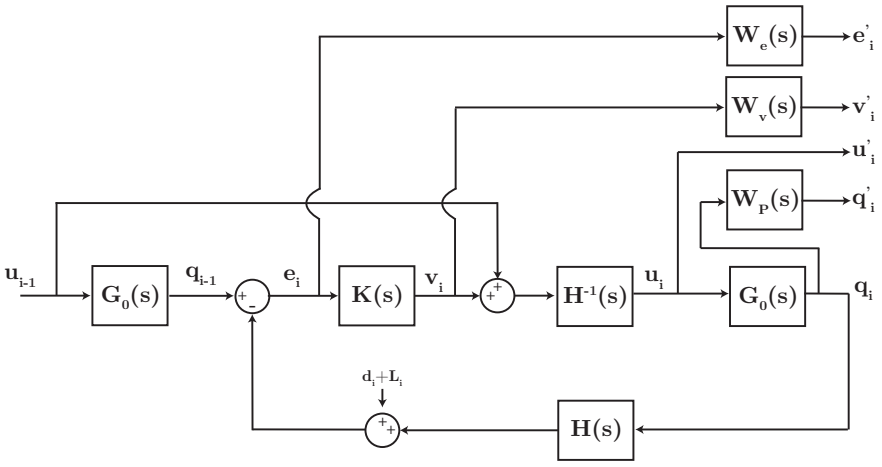

Figure 4.7: Configuration of the closed-loop control system.

4.3.5 H_∞ Robust Controller Design for Connected Vehicles

After selecting the loop shaping weights, we use MATLAB to represent the system interconnection shown in Figure 4.7 into the linear fractional transformation (LFT) form. This is done by using the MATLAB command *sconnect*. Next, we express the closed-loop system as

$$z(s) = N(s)\, w(s), \tag{4.20}$$

where z represents the vector containing controlled output signals, $N(s)$ describes the closed-loop system transfer function matrix and w represents the exogenous input signals [5]. Now, by imposing the following requirement that

$$||N(j\omega)||_\infty \leq 1, \tag{4.21}$$

string stability would be guaranteed. Next, the robust control design problem is solved by invoking the MATLAB command *hinflmi*. The controller obtained matches the model order of the plant, and using model order reduction methods we reduce the order of the controller to 4^{th} order. A comparison of the Bode plots of the reduced-order versus full-order controller shows a good approximation over all frequencies. Next, we analyze the closed-loop properties of the system. Figure 4.8 illustrates that string stability requirement is met. We see from Figure 4.9 that we also achieve closed-loop robust stability. This condition is given as $||W_p(j\omega)T(j\omega)|| \leq 1$ for any ω.

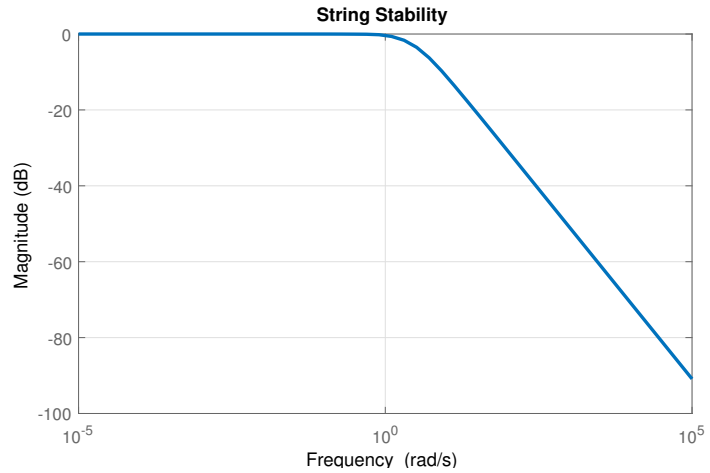

Figure 4.8: String stability of the closed-loop system.

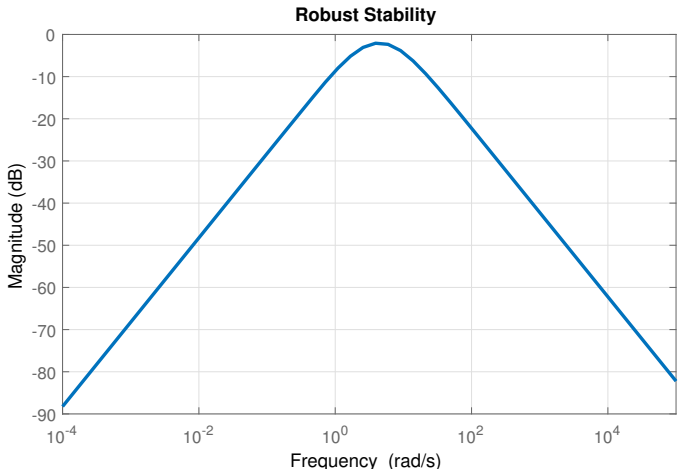

Figure 4.9: Robust stability of the closed-loop system.

Figure 4.10 shows the welcomed result that we have also achieved a level of robust performance in our design.

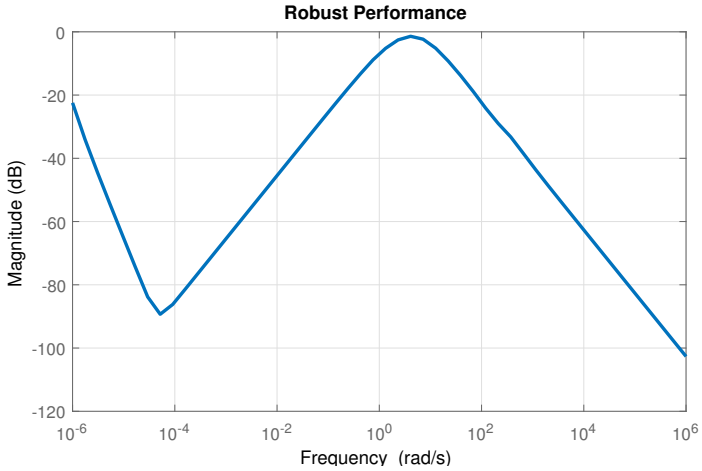

Figure 4.10: Robust performance of the closed-loop system.

Interestingly, we notice from figures 4.2 and 4.4 that the relationship

$$\frac{u_i(s)}{u_{i-1}(s)} = \frac{e_i(s)}{e_{i-1}(s)}, \tag{4.22}$$

holds true. For our reformulated block diagram in Figure 4.4 we rederive

$$\frac{e_i(s)}{e_{i-1}(s)} = \frac{1}{H(s)} \frac{D(s)+D(s)G(s)K(s)}{1+D(s)G(s)K(s)}. \tag{4.23}$$

4.3.6 Anti-Windup Compensator Design

To account for the actuator saturation that exists in our experimental test bed due to the hard limits on actuators, we use the procedure given in [66] to augment our CACC system with an anti-windup controller. First, we consider the CACC block diagram of Figure 4.7 as a general feedback system without saturation as

$$y_{lin} = \mathbf{P} \begin{bmatrix} w \\ \\ u_{lin} \end{bmatrix}, \quad u_{lin} = \mathbf{K} \begin{bmatrix} r \\ \\ y_{lin} \end{bmatrix}, \tag{4.24}$$

where $u_{lin} \in \mathscr{R}^{n_u}, y_{lin} \in \mathscr{R}^{n_y}, r \in \mathscr{R}^{n_r}$, and $w \in \mathscr{R}^{n_w}$. The block diagram of the general feedback system without saturation is shown in Figure 4.11.

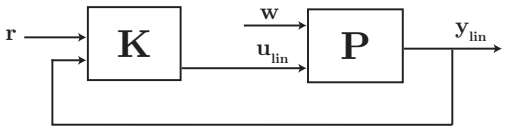

Figure 4.11: General feedback system.

With respect to Figure 4.11 we partition the plant, **P**(s), as

$$\mathbf{P} \triangleq [P_1, P_2] = \left[\begin{array}{c|cc} A_p & B_{p1} & B_{p2} \\ \hline C_p & D_{c1} & 0 \end{array}\right]. \tag{4.25}$$

Likewise, with respect to Figure 4.11 we partition the controller, **K**(s), as

$$\mathbf{K} \triangleq [K_1, K_2] = \left[\begin{array}{c|cc} A_c & B_{c1} & B_{c2} \\ \hline C_c & D_{c1} & D_{c2} \end{array}\right]. \tag{4.26}$$

We choose to use the formulation in [66] because it is assumed that $\mathbf{K}(s)$ has already been designed to guarantee closed-loop stability of (4.24) and to achieve desired performance metrics. Next, we add an anti-windup compensation scheme as

$$y = \mathbf{P}\begin{bmatrix} w \\ \\ u \end{bmatrix}, \quad u = \phi(\hat{u}), \quad \hat{u} = \hat{\mathbf{K}}\begin{bmatrix} r \\ \\ y \\ \\ u-\hat{u} \end{bmatrix}, \tag{4.27}$$

where $u, \hat{u} \in \mathscr{R}^{n_u}, y_{lin} \in \mathscr{R}^{n_y}$. Now, ϕ is given as the saturation function and ψ denotes a deadzone function, both are defined as

$$\phi(v) \triangleq \begin{cases} a \cdot sgn(v), & |v| > a \\ \\ v, & |v| \leq a \end{cases}, \quad \psi \triangleq v - \phi(v), \tag{4.28}$$

where a is used to characterize the saturation. Using (4.24) and (4.28), Figure 4.12 illustrates a general feedback system with saturation.

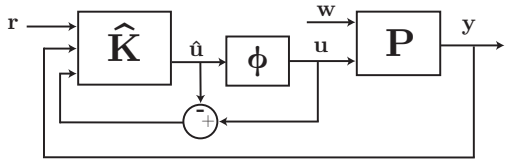

Figure 4.12: General anti-windup system.

Now the controller, $\mathbf{K}(s)$, is augmented with feedback from the saturation block, and the augmented controller, $\hat{\mathbf{K}}(s)$, is defined as

$$\hat{\mathbf{K}} \triangleq [K_1, K_2, K_3] = \left[\begin{array}{c|ccc} A_c & B_{c1} & B_{c2} & \Lambda_1 \\ \hline C_c & D_{c1} & D_{c2} & \Lambda_2 \end{array}\right]. \tag{4.29}$$

The transfer functions of K_1 and K_2 are the same as (4.29) and Λ_1 and Λ_2 are constant matrices introduced for anti-windup compensation. To solve for Λ_1 and Λ_2, the authors in [66] used a linear

matrix inequality (LMI) based formulation to guarantee that

$$||y - y_{in}||_{\mathscr{L}_2} \leq ||\psi(U_{lin})||_{\mathscr{L}_2}, \tag{4.30}$$

where y represents the output of the plant without the saturation, y_{in} represents the output of the plant with the saturation, $\psi(u_{lin})$ represents a system between the linear and nonlinear systems, and $||.||_{\mathscr{L}_2}$ denotes the vector's two-norm. We solve the LMI condition to determine Λ_1 and Λ_2 and augment our CACC system with the anti-windup compensator.

4.4 Our Laboratory-Scale CACC Test Bed

In considering the selection for our test bed, we see from (4.9) that a simple first-order ordinary differential equation can accurately represent the dynamics of the CACC problem. Indeed, [2] verifies that the measured step response of an acceleration controlled test vehicle is well modeled by (4.9). Using this knowledge we built a laboratory test bed model. Figure 4.13 shows a picture of our test bed.

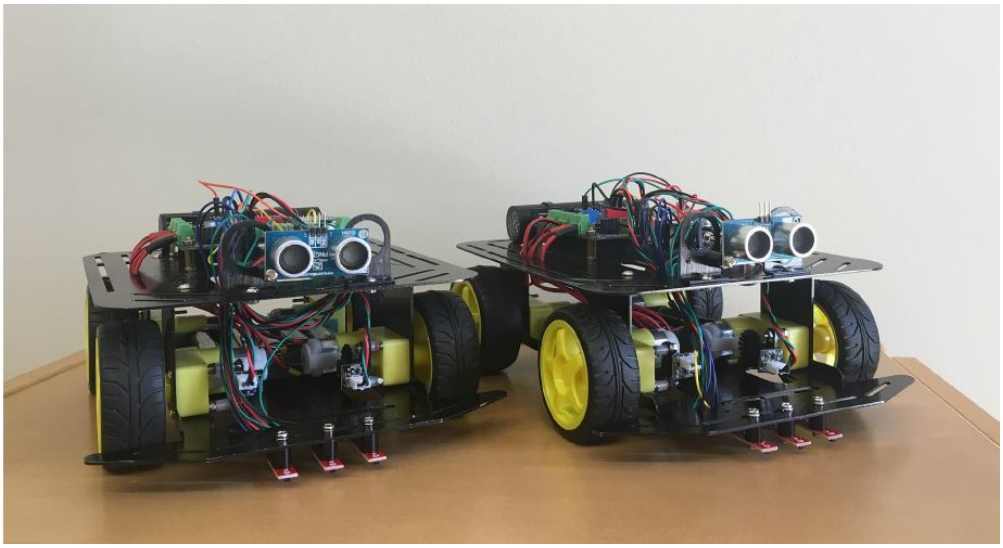

Figure 4.13: CACC test bed.

4.4.1 TEST BED HARDWARE

For our laboratory test bed, we modified a DFRobot Baron 4WD Mobile platform to meet our needs. Various technologies and sensors were added to realize the CACC concepts:

Drivetrain: The onboard DC motors are rated at $6V$ with a no-load current of 71 mA and a stall current of 470 mA. The gear ratio is $1:120$ and the torque is 1.92 $kg \cdot cm$. Through testing, the maximum voltage supplied to the motors from the microcontroller is found to be 4.1 V. The lowest voltage required to turn the motors under the weight of the vehicle is 1.5 V. The velocities corresponding to the maximum and minimum voltages are 0.4 m/s and 0.14 m/s, respectively. Although scaled, these motor nonlinearities pose the same control related conceptual issues as that of a full scale vehicle.

Microcontroller: The Romeo V2.2 (R3) is an expanded microcontroller printed circuit board and it is programmable using the Arduino IDE software.

Communication: We chose to use XBee series 1 (S1) RF modules to communicate with each other. The XBee's implement the IEEE 802.15.4 protocol within the 2.4 GHz frequency band to communicate. The radios transmit data at a rate of 250,000 bits per second (b/s), and testing proved a serial interface data rate of 57,600 b/s to be the fastest reliable speed.

Wheel Encoders: The wheel encoders use a non-contact, optical method to track the rotation of the wheels. We doubled the resolution of the original disk to a final resolution of 0.5 cm.

Proximity Sensor: To detect the distance between the vehicles we used the Paralax PING Ultrasonic proximity sensor. This sensor works by emitting a 40 kilohertz ultrasonic burst and providing an output pulse that corresponds to the time it takes to receive an echo.

Accelerometer: An ADXL335 triple-Axis accelerometer was installed to measure onboard acceleration data.

Infrared Line Following Sensors: To mitigate the problem of longitudinal control, we installed infrared line following sensors.

4.4.2 System Identification

For the purpose of creating a CACC test bed that is modeled by (4.9) and (4.11), we perform system identification to find the internal time delay ϕ, the nominal communication delay θ, and the time constant τ.

Step Response

To identify our test bed parameters we applied a step function input. To do this, we exploit the well known relation that voltage and velocity are related through a constant such that $Velocity = \alpha \cdot Voltage$ where α is given in $\frac{m}{V \cdot s}$. We then command the onboard microcontroller to a maximum step of 4.1 V. Figure 4.14 shows a plot of the corresponding interpolated acceleration data.

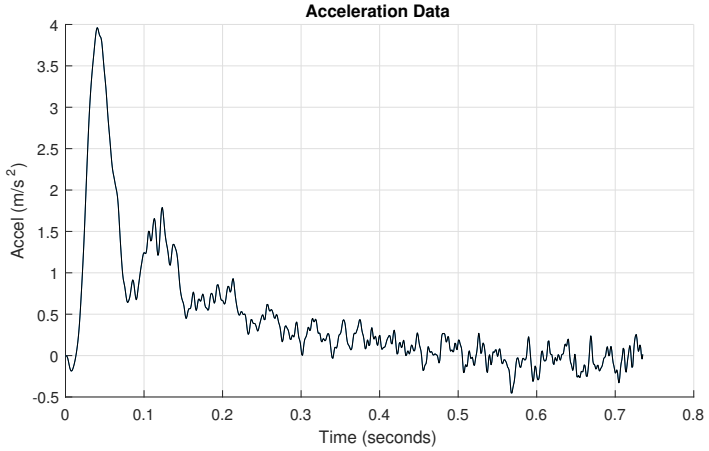

Figure 4.14: CACC acceleration data collected from our test bed.

Figure 4.15 gives the corresponding velocity response. Through experimental testing we find $\alpha = 10.32\ \frac{m}{V \cdot s}$. Next, we fit the model with the unknown parameters to the data. Writing a transfer function from desired acceleration to position as in (4.9) yields

$$G_{sys}(s) = \frac{q_i(s)}{u_i(s)} = \frac{0.0963}{s^2(0.14s+1)} e^{-0.018s}. \tag{4.31}$$

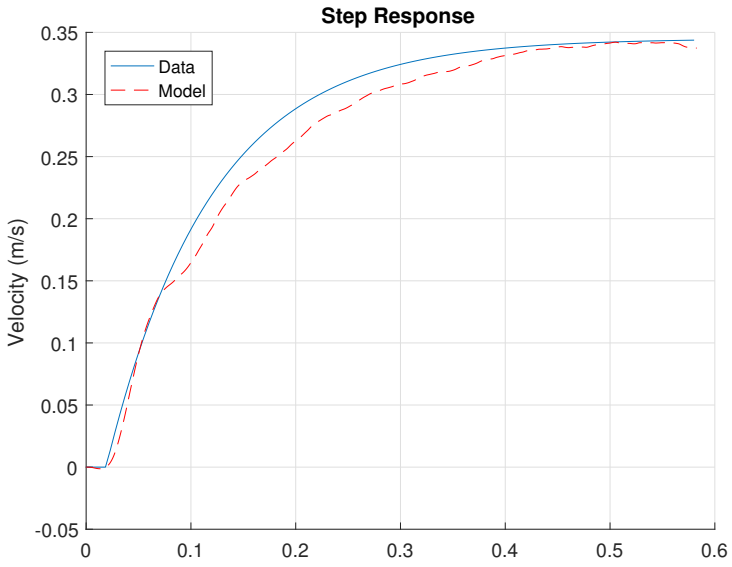

Figure 4.15: CACC test bed step response.

Estimating the Communication Delay

To calculate the communication delay, first we establish the best and worst case scenarios. The propagation delay, i.e., the time the data takes to travel through the air, is on the order of nanoseconds and is negligible in this case. In the best case scenario, the carrier channel is clear to send and the total communication delay is 1.17 ms. In the worst case scenario, we consider packet delivery is successful but takes the longest amount of time for channel assessment prior to transmission. In this scenario, the communication delay is 9.44 ms. Therefore, the communication delay is the inclusive set of $\in$ = [1.17 ms, 9.44 ms].

4.5 Simulation and Experimental Results

We first created a 5-car platoon simulation model using the reduced-order robust controller. We modeled the cars we have in our laboratory test bed in MATLAB/Simulink taking note to also include the corresponding saturations and Coloumb friction characteristics.

4.5.1 Simulation Results

First, we utilized our simulation model to examine the closed-loop performance with the reduced-order robust controller and nominal plant parameters. Figure 4.16 shows the corresponding error responses, and Figure 4.17 shows the corresponding velocity responses.

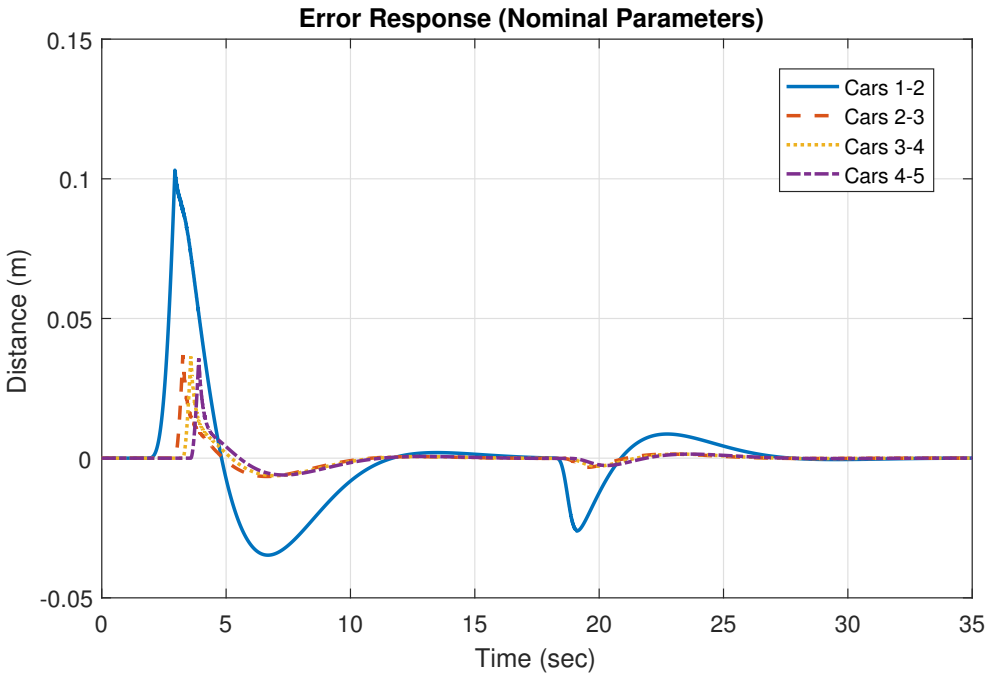

Figure 4.16: Error response of the 5-car simulation model using nominal system parameters.

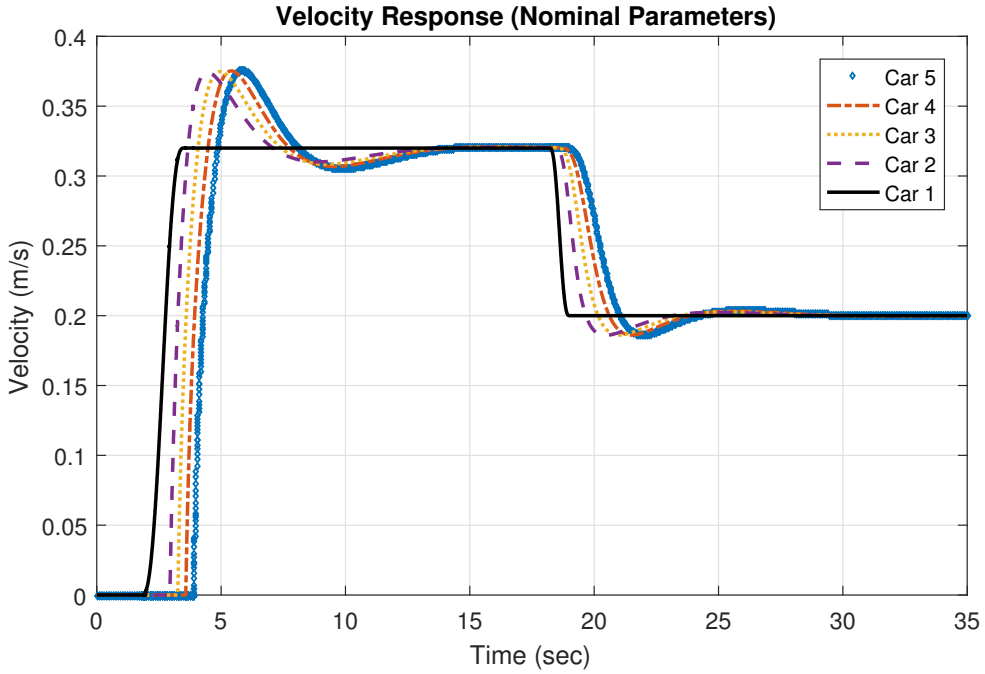

Figure 4.17: Velocity response of the 5-car simulation model using nominal system parameters.

In Figure 4.16, a positive error value indicates conservative behavior, whereas, a negative value indicates that the following car is too close. From Figure 4.17, we observe that all cars start from rest, and the lead car follows a smooth trajectory. The following cars are tuned to allow overshoot in lieu of a faster settling time. After the lead car has reached 0.32 m/s, it undergoes a sharp deceleration. This is reflected by the error going negative, but the controller quickly reestablishes a zero steady-state tracking error (see, Figure 4.16). Next, we perturb both ϕ and θ to their respective maximum values. The resulting simulation outputs are given in figures 4.18 and 4.19.
We see from the comparison of figures 4.16 and 4.17 with figures 4.18 and 4.19 that added delays cause an increase in error, but that the robust controller is still able to provide the steady-state tracking over the region of parameter perturbations.

4.5.2 Experimental Results

Next, using our test bed, we programmed the lead car to follow the smooth trajectory of the 5-car simulation. We discretize the blocks in Figure 4.4 with a 20 *msec* sampling time and implement on the laboratory test bed.

Robust Controller

Figures 4.20 and 4.21 show the test bed results using the robust controller with nominal parameters.

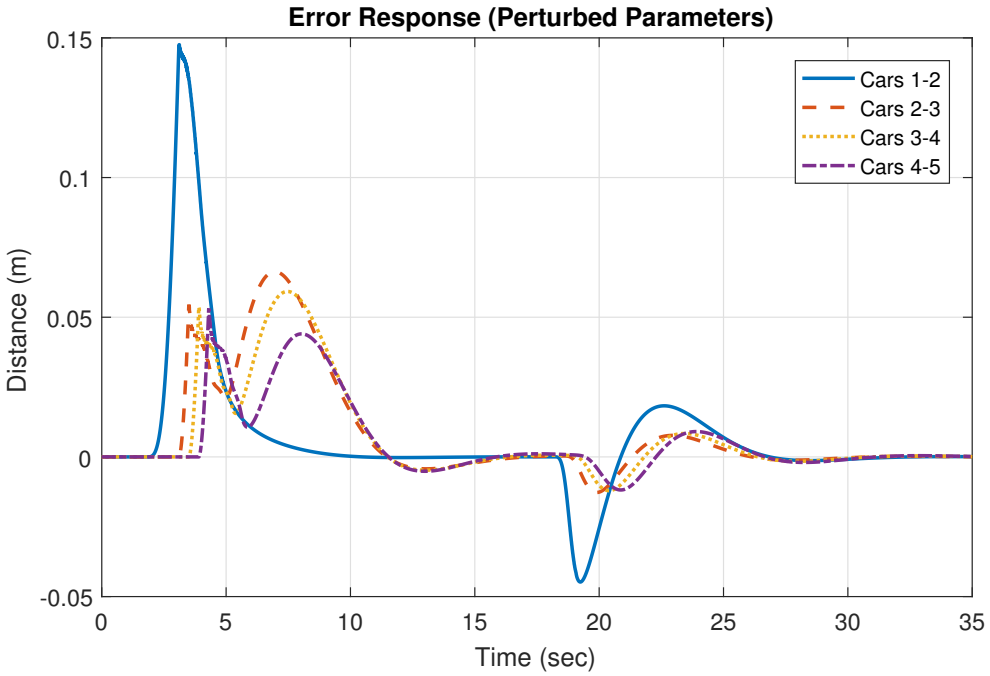

Figure 4.18: Error response for the 5-car simulation model using perturbed system parameters.

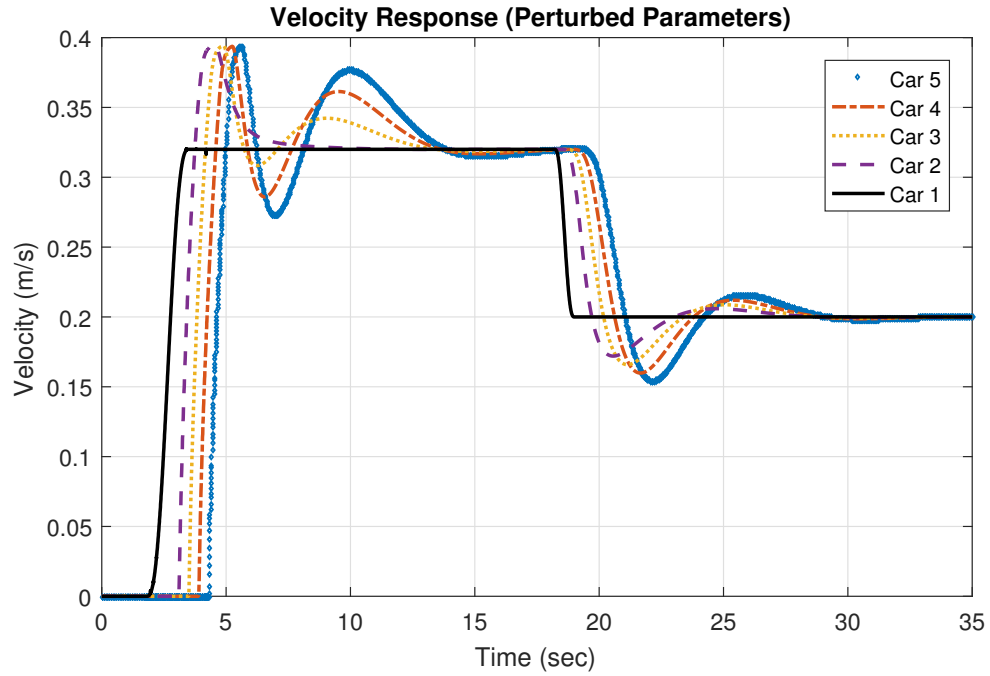

Figure 4.19: Velocity response for the 5-car simulation model using perturbed system parameters.

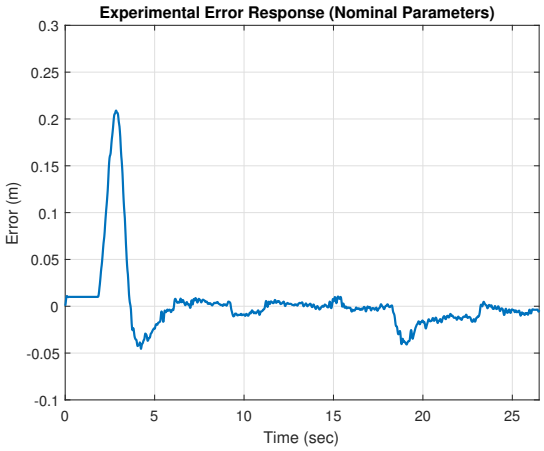

Figure 4.20: Error response of the experimental test bed using nominal parameters.

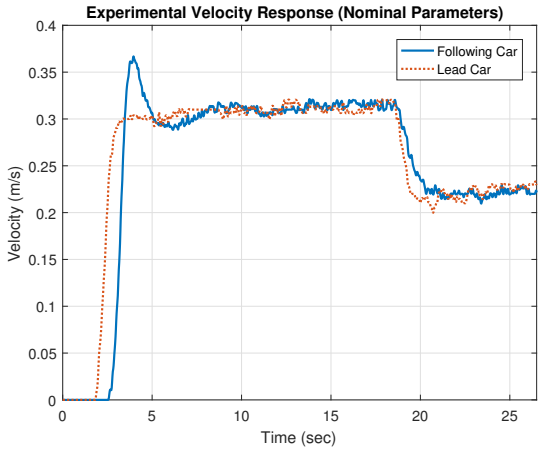

Figure 4.21: Velocity response of the experimental test bed using nominal parameters.

Figures 4.20 and 4.21 show a comparison to figures 4.16 and 4.17. Similarly, we perturbed both ϕ and θ to their respective maximum values and reran the experiments; the results are given in figures 4.22 and 4.23.

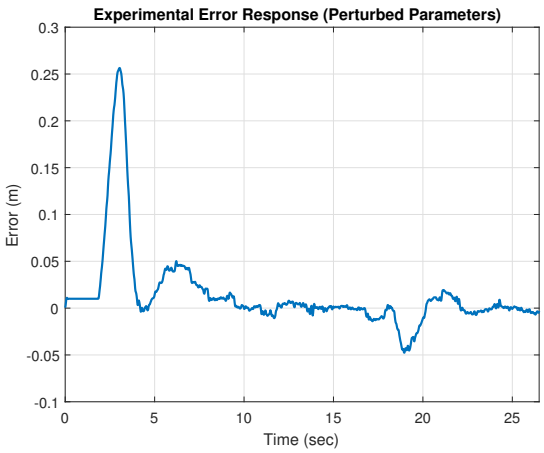

Figure 4.22: Error response of the experimental test bed using perturbed parameters.

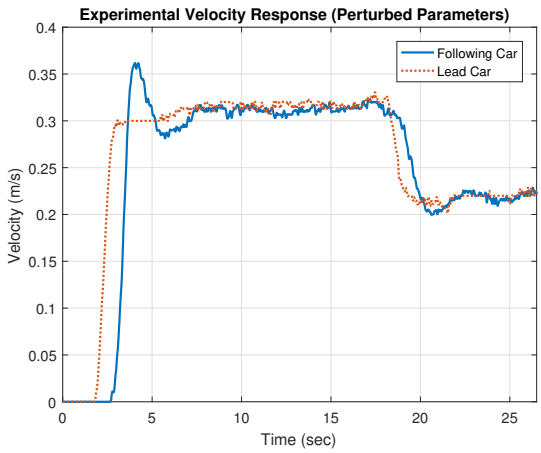

Figure 4.23: Velocity response of the experimental test bed using perturbed parameters.

We observe that figures 4.22 and 4.23 also show a good comparison with the results in figures 4.18 and 4.19 in that the robust controller still maintains a high level of performance despite the parameter variation. This is critical in the CACC application since we see that despite the parameter variations, the robust controller allows for significant headway impact ($h = 0.35$ *sec.* vs $h = 1.1$ *sec.*) while maintaining string stability. We note that all the experimental testings show an increase of

$\sim$0.1 m on the initial acceleration when compared to the simulation results. This consistent discrepancy is due to the initialization needed in the onboard microcontroller.

NOMINAL CONTROLLER

Next, we compare our results to a nominal $\mathscr{H}_\infty$ controller designed around the nominal parameters: $\bar{\tau}$, $\bar{\phi}$, and $\bar{\theta}$. For the experimental case of a nominal controller using nominal parameters we achieve performance gains over the robust controller design shown in figures 4.20 and 4.21. However, when the nominal $\mathscr{H}_\infty$ controller is experimentally tested using the perturbed case, we observe unacceptably large overshoots illustrated in figures 4.24 and 4.25.

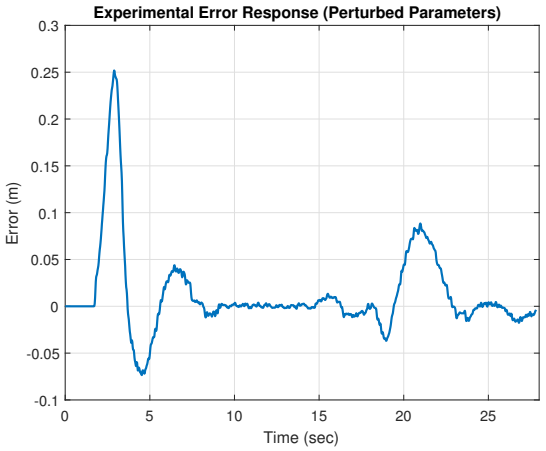

Figure 4.24: Error response of the experimental test bed with the nominal controller and perturbed parameters.

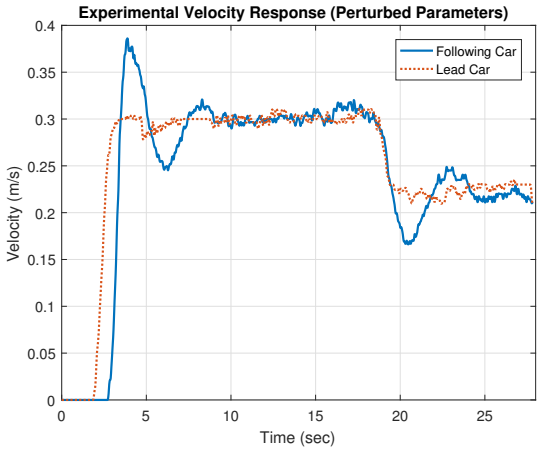

Figure 4.25: Velocity response of the experimental test bed with the nominal controller and perturbed parameters.

Comparison of Figure 4.24 with Figure 4.22 shows that the nominal controller vastly overshoots five times worst than the robust controller, and hence disturbances are magnified instead of attenuated. Table 4.1 gives a comparison of the root mean square (RMS) values of the deceleration event. We see the nominal controller's RMS value of error doubles, but that the robust controller remains small. This validates the need for the (proposed) robust controller design, given the overall need of minimizing headway distance in traffic situations.

Table 4.1: RMS value comparisons for experimental data where *n* represents nominal and *p* represents perturbed.

Robust (n)	**Robust (p)**	**Nominal (n)**	**Nominal (p)**
0.0113	0.0117	0.0104	0.0250

4.6 CONCLUSIONS

In this paper, we have provided experimental results on the design and validation of a robust $\mathscr{H}_\infty$ controller for cooperative adaptive cruise control (CACC) of connected vehicles. The proposed design framework can account for the uncertainties in the vehicle model used for the CACC design to ensure string stability. The control design process includes: (i) quantifying the effect of uncertainties on the plant model, and (ii) employing the mixed-sensitivity, loop shaping-based $\mathscr{H}_\infty$ control design. Simulation and experimental results demonstrate and validate that the robust controller can achieve string stability and tracking performance over the region of parameter perturbations. Experimental tests for a nominal $\mathscr{H}_\infty$ controller are run showing the need for a robust controller design.

CHAPTER 5

ROBUST COOPERATIVE ADAPTIVE CRUISE CONTROL DESIGN FOR CONNECTED VEHICLES

[1]

[1]Trudgen, Mark, and Javad Mohammadpour. "Robust Cooperative Adaptive Cruise Control Design for Connected Vehicles." In ASME 2015 Dynamic Systems and Control Conference, pp. V001T17A004-V001T17A004. American Society of Mechanical Engineers, 2015.

ABSTRACT

In this paper, we design and validate a robust $\mathscr{H}_\infty$ controller for Cooperative Adaptive Cruise Control (CACC) in connected vehicles. CACC systems take advantage of onboard sensors and wireless technologies working together in order to achieve smaller inter-vehicle following distances, with the overall goal of increasing vehicle throughput on busy highways, and hence serving as a viable approach to reduce traffic congestion. A group of connected vehicles equipped with CACC technology must also ensure what is known as string stability. This requirement effectively dictates that disturbances should be attenuated as they propagate along the platoon of following vehicles. In order to guarantee string stability and to cope with the uncertainties seen in the vehicle model used for a model-based CACC, we propose to design and implement a robust $\mathscr{H}_\infty$ controller. Loop shaping design methodology is used in this paper to achieve desired tracking characteristics in the presence of competing string stability, robustness and performance requirements. We then employ model reduction techniques to reduce the order of the controller and finally implement the reduced-order controller on a simulation model demonstrating the robust properties of the closed-loop system.

5.1 INTRODUCTION

Connected vehicles are an example of a modern day cyber physical system (CPS) that through the use of Cooperative Adaptive Cruise Control (CACC) provide an innovative solution to the traffic congestion problem [18]. Traffic is becoming an increasing problem in today's world as congestion in many urban areas is growing at a much faster rate than the traditional means of traffic alleviation can assuage [19]. CACC is a technology that seeks to reduce traffic congestion by means of achieving higher traffic flow rates using advanced control systems to safely reduce the allowable *headway* time between vehicles [20]. A widespread advantage of CACC over traditional means of increasing traffic throughput, i.e., road construction, is that CACC has the potential to

be implemented on any car in highway system without the additional high costs and time delays associated with road construction projects [21].

CACC technology is an extension of Adaptive Cruise Control (ACC), which in turn is an extension of conventional cruise control (CCC), a technology traditionally used to regulate a vehicle at a constant highway speed [22]. ACC extends the CCC technology by regulating the so-called *headway* distance between vehicles that are arranged together in a platoon [23]. ACC employs radar (or lidar) sensors to measure the relative velocity and displacement with the preceding vehicle, and a longitudinal control framework is then implemented to space the vehicles to an appropriate *headway* [22] by adjusting the acceleration and deceleration of the vehicle. CACC extends the ACC technology by adding wireless inter-vehicle communication [24]. This extension enables smaller *headway* distances, which is critical for platoon technology to have a noticeable impact on traffic mitigation [21, 25]. According to the *2010 Highway Capacity Manual*, a study observing human drivers showed that the maximum flow rate for a multilane highway (at 60 mi/h) equates to 1.1 seconds of *headway* [21]. Herein lies the main drawback of ACC technology, that is the smallest stable *headway* is larger than the average time-gap that human drivers naturally exhibit [19, 25], thus justifying the need for CACC technology. The vehicles that are virtually connected to each other through CACC technology must ensure an important metric called string stability [2]. This concept was first introduced in [26] and later extended in [27], which led to the development of systems using the nearest neighbor as a measurement. Essentially, string stability is a requirement that all disturbances introduced in the string be attenuated as they propagate in the upstream direction [23, 27]. String stability is essential to ensuring the safety and feasibility of the string [2]. Not only do any disturbances in position, velocity or acceleration create increased energy consumption, these disturbances must also be mitigated in order to prevent the so-called *ghost traffic jams* [23], or even in extreme cases, an accident [28]; hence, a control design formulation that can explicitly account for string stability inherently meets design objectives and exterminates the need for any ad-hoc *a posteriori* tuning to achieve string stability. This notion of string stability has been studied in several aspects such as Lyapunov stability, and input-output stability;

however, these methods lack the consideration of a measure of performance as seen in [29, 30], which give a frequency-domain approach for controller synthesis.

Several approaches have been undertaken in designing a controller for a platoon of vehicles. The system model considered to describe the vehicular motion is usually a 3^{rd} order nonlinear model [31, 32], where subsequently the plant is linearized by the use of feedback linearization method. For the control design using the linearized model, several CACC experimental results have been reported, e.g., in [23, 24, 33]. These recent works show the promise in using CACC. Indeed, several aspects of CACC technologies have been studied. The authors in [28] developed a sampled data approach to CACC design in the presence of sensors and actuator failures and [34] studied strategies for worst case scenarios. Model predictive control (MPC) has also gained attention as a way to cast the CACC problem in a framework that can directly optimize fuel economy. A CACC MPC approach can be considered very useful for heavy duty vehicles, such as tractor trailer trucks as in [35], where smaller *headway* distances can be sacrificed for better fuel economies as traffic throughput may not be the primary objective as is the case with urban rush hour highway demands. CACC can also be viewed in light of the communication as a networked control problem where the effects of sampling, hold, and network delays can be taken into account. An H_∞ formulation of network controlled problems is given in [36]. Still, other works have investigated communication-based time-varying delays and communication structures beyond the classical architecture as in [37].

To the authors' best knowledge, no previous work has extended the CACC framework to include modeling uncertainties directly arising from the plant using a decentralized framework. Fundamentally, all system models exhibit a level of model uncertainty [5]. Indeed, in the experimental results of [23], it was noted that the parameters of the plant were found using a least squares averaging technique, and it is known that uncertainty comes from the parameters describing the linearized plant. In [64] time constant parameter variations were mentioned, but aside from ensuring LHP stable poles, a robust control design framework was not considered. Similarly, although packet loss and communication delays were considered in [37], no consideration was made with

respect to parameter variations of a linearized plant. We consider in this paper a robust controller design, where an H_∞ controller is sought to be synthesized as the induced energy-to-energy gain (or H_∞ norm) is a natural norm to use in the presence of uncertainty [5], especially considering the literature available on $\mathscr{L}_2$ string stability [2,30]. In our formulation, we choose to model the CACC problem in a decentralized manner similarly to [2]. While other formulations exist for centralized control such as [67], we choose the decentralized formulations as they have strong relevance to every day traffic applications where there is no set leader. A decentralized implementation also gives each driver in the string control over a range of *headway* values, which is desirable considering different driving abilities; however, an investigation of psychological aspects is not considered here, for which the reader is referred to references in [19].

This paper is structured as follows: Section 2 describes the platoon following technologies and how the simple nonlinear model governing vehicles is linearized. We also show simulation results comparing different adaptive cruise control technologies. Section 3 explains the design of a robust $\mathscr{H}_\infty$ controller, whose order will then be reduced while still retaining the desired robust properties. Section 4 illustrates the results of a 5-car simulation, and Section 5 draws conclusions.

5.2 Various Cruise Control Technologies

Design of various longitudinal adaptive cruise control strategies have been studied in the literature (i.e. [22] and references therein). Figure 5.1 shows a representative view of a typical string of vehicles equipped with cooperative adaptive cruise control (CACC), where the lead car of the string sets a trajectory to follow and communicates its acceleration a_0 only to the following vehicle. Alongside the communicated acceleration, the following vehicle is equipped with onboard sensors to measure the relative distance and velocity. This is typically done via the use of radar (or lidar) [23]. In considering a platoon, the distance between vehicles is broken into 3 segments: d_i is the desired static distance between vehicles, hv_i is the product of the minimum headway required and the velocity of the i^{th} vehicle, and finally δ_i is an additional spacing parameter. The i^{th} vehicle is said to be in the correct positioning when $\delta_i = 0$.

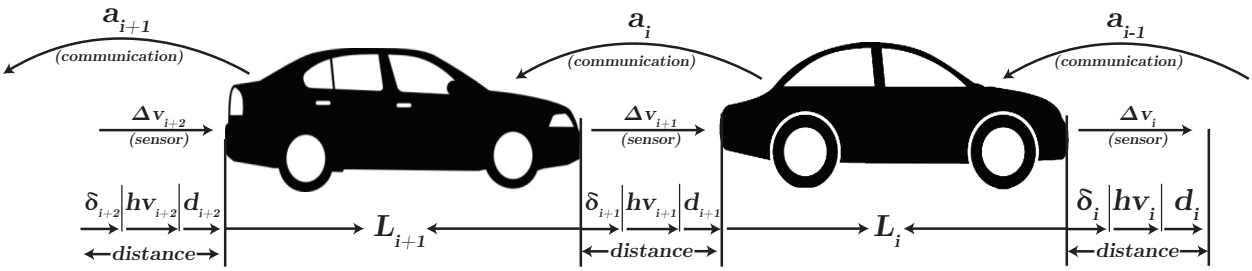

Figure 5.1: A string of vehicles equipped with cooperative adaptive cruise control technology.

More specifically, δ_i, the spacing policy, is given as [28]

$$\delta_i = q_{i-1} - q_i - L_i - hv_i - d_0, \tag{5.1}$$

where h is the time gap (headway), d_0 is a given minimum distance and L_i is the length of the i^{th} vehicle. The system dynamics can be represented as [31, 32]

$$\begin{aligned} \dot{\delta}_i &= v_{i-1} - v_i - h\dot{v}_i \\ \Delta\dot{v}_i &= a_{i-1} - a_i \\ \dot{a}_i &= f_i(v_i, a_i) + g_i(v_i)c_i, \end{aligned} \tag{5.2}$$

where $g_i(v_i)$ is given as

$$g_i(v_i) = \frac{1}{\tau_i m_i}. \tag{5.3}$$

Subsquently, the model is nonlinear due to the nonlinear function $f_i(v_i, a_i)$ which is described as

$$f_i(v_i, a_i) = -\frac{1}{\tau_i}\left[\dot{v}_i + \frac{\sigma A_i c_{di}}{2m_i} v_i^2 + \frac{d_{mi}}{m_i}\right] - \frac{\sigma A_i c_{di} v_i a_i}{m_i}, \tag{5.4}$$

where m_i represents the i^{th} vehicle's mass, τ_i is the engine time-constant of the i^{th} vehicle, $\frac{\tau_i A_i c_{di}}{2m_i}$ is the air resistance, d_{mi} is the mechanical drag, c_{di} is the drag coefficient and σ is the specific mass of the air. To linearize the above nonlinear system dynamics, the following control law is adopted [31, 32]

$$c_i = u_i m_i + \frac{\sigma A_i c_{di} v_i^2}{2} + d_{mi} + \tau_i \sigma A_i c_{di} v_i a_i, \tag{5.5}$$

where u_i is the new control input signal to be designed for the closed-loop system where $c_i < 0$ and $c_i \geq 0$ correspond to brake and throttle actions, respectively. Using (5.5) results in a feedback linearization, which combined with (5.2) gives

$$\dot{a}_i(t) = -\frac{a_i(t)}{\tau_i} + \frac{u_i(t)}{\tau_i}. \tag{5.6}$$

Since $a_{i-1}(t)$ is sent from the preceding vehicle, a communication delay θ_i is introduced so the acceleration arriving at the i^{th} vehicle is $a_{i-1}(t-\theta_i)$. Writing the CACC model in the state-space form gives [28]

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_{i1} u_i(t) + B_{i2} w_i(t-\theta_i) \\ y_i(t) &= \left[x_i^T(t), w_i(t)\right]^T, \end{aligned} \tag{5.7}$$

where θ_i is the communication delay, $x_i = [\delta_i, \Delta v_i, a_i]^T$ is the state vector, $w_i(t) = a_{i-1}(t)$ and $y_i(t) = [\delta_i, \Delta v_i, a_i, w_i]^T$ is the output vector, and additionally,

$$A_i = \begin{bmatrix} 0 & 1 & -h \\ 0 & 0 & -1 \\ 0 & 0 & -1/\tau_i \end{bmatrix}, \quad B_{i1} = \begin{bmatrix} 0 \\ 0 \\ 1/\tau_i \end{bmatrix}, \quad B_{i2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{5.8}$$

We follow [23, 24, 64] in assuming a low-level linearizing feedback controller. The system in (5.8) gives the linearization for the i^{th} vehicle, and the overall system is hence a decentralized platoon.

5.2.1 ADAPTIVE CRUISE CONTROL

By setting a_{i-1} to zero in (5.2), the CACC model reduces to the ACC model, and the same feedback linearizing controller given in (5.5) can be used to achieve a linear model. Next, by using the setup proposed in [23], the corresponding block diagram is given in Figure 5.2.

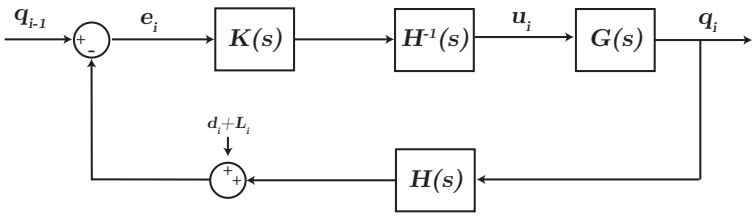

Figure 5.2: Adaptive cruise control block diagram.

For the i^{th} vehicle, we use the following notation: q_{i-1} denotes the preceding vehicle's position, q_i denotes the local position, e_i is the error signal inputted into the controller $K(s)$ and u_i is the so-called *desired* acceleration (that is used as an input to the linearizing controller, see, e.g., [28]). Finally, d_i denotes an added static following distance, and L_i is the length of the i^{th} vehicle. Without the loss of generality, $L_i = d_i = 0$ is assumed. In addition, $G(s)$ represents the system transfer function, and $H(s)$ describes the spacing policy given as

$$G(s) = \frac{q_i(s)}{u_i(s)} = \frac{1}{s^2(\tau_i s + 1)} e^{-\phi_i s} \tag{5.9}$$

$$H(s) = hs + 1, \tag{5.10}$$

where τ_i is the engine time constant and the nominal value is taken as $\bar{\tau} = 0.1$ sec. and $\bar{\phi} = 0.2$ sec is an associated nominal internal delay and h represents the designed *headway* value [23, 65]. We built a simulation model in MATLAB/SIMULINK that was composed of 5 cars using a simple stabilizing controller is given by

$$K(s) = K_D s + K_P, \tag{5.11}$$

where $K_D = 0.7$ and $K_P = 0.2$ [23].

The *headway* time, h, is set to 0.6 sec. Using this headway value, we do not achieve string stability in the ACC case. This headway value is chosen to illustrate that even lower headway values can be achieved with communication, thus justifying additional model complexity required. An inherent goal is to reduce the headway as this correlates to a better traffic mitigation.

5.2.2 Degraded Cooperative Adaptive Cruise Control

As a bridge between ACC and CACC, the authors in [65] propose the use of an onboard observer that uses local measurements to estimate the accerlation of the previous vehicle. This can be used when, e.g., a communication link experiences packet losses and before resorting to an ACC scheme [65]. A block diagram of the degraded Cooperative Adaptive Cruise Control (dCACC) case is shown in Figure 5.3, where $T(s)$ is a Kalman estimator and $T_{aa}(s)$ is a smoothing filter. The boxed section in Figure 5.3 is used to denote the estimation scheme. It is noted that this is an onboard estimation scheme implemented in the i^{th} vehicle. Using (5.11) again, we see that the dCACC has improved damping compared to the ACC case, but still not being able to achieve string stability for low headway values.

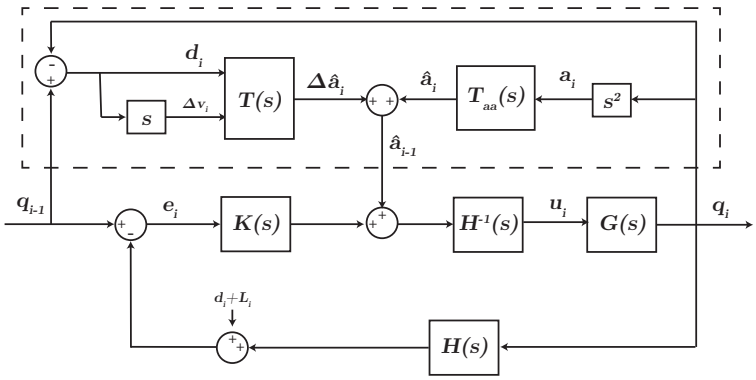

Figure 5.3: dCACC block diagram.

5.2.3 Cooperative Adaptive Cruise Control

Next, by introducing a dedicated short range communication (DRSC) protocol between vehicles, the leading vehicle's acceleration can be communicated to the following vehicle. As this signal is transmitted through communication channel, there is a delay; hence,

$$D(s) = e^{-\theta s}, \tag{5.12}$$

where $\theta = 0.02$ sec. is chosen as in [23, 65]. The implemented model in MATLAB/SIMULINK is modified to now include these communication delays as shown in Figure 5.4.

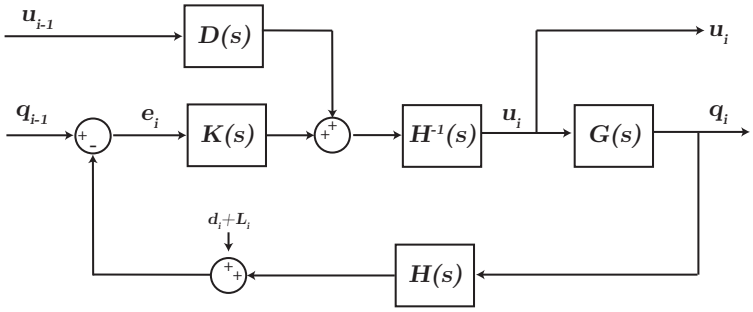

Figure 5.4: CACC block diagram.

In the case of a CACC scheme with a stabilizing controller, the block diagram is shown in Figure 5.4 and the controller $K(s)$ is the same as in (5.11) [23]. The communicated acceleration is used as a feedforward term. Using the same *headway* value used in the previous cases, the CACC scheme does achieve string stability. Although the error is non-zero, it does not increase along the string.

String Stability

We denote the transfer function from q_{i-1} to q_i as $\Gamma_{CACC}(s)$ given by

$$\Gamma_{CACC}(s) = \frac{1}{H(s)} \frac{G(s)K(s) + D(s)}{1 + G(s)K(s)}. \tag{5.13}$$

$D(s)$ represents the delay associated with either the dCACC case or the CACC case. Setting $D(s) = 0$ yields the ACC case. Figure 5.5 shows the Bode plots corresponding to the three platoon control

approaches described before for $h = 0.6$ sec. For string stability $||\Gamma(j\omega)|| < 1$ needs to be achieved for any ω, which physically implies that the position of the vehicle q_i remains behind the preceding vehicle q_{i-1}. From Figure 5.5, it is observed that only the CACC system satisfies this requirement. As noted in [25] for this technology to have a noticeable impact on traffic mitigation, a *headway* significantly smaller than 1.1 sec. already seen in the naturalistic driving must be achieved [19], and the dCACC and ACC cases do not even achieve the naturalistic driving *headway* value.

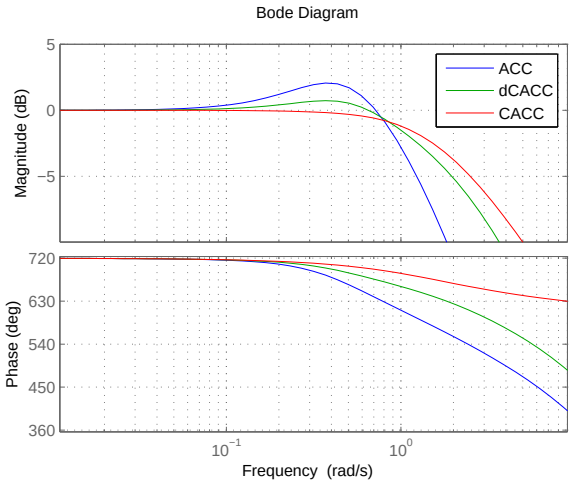

Figure 5.5: Frequency response associated with ACC, dCACC and CACC.

5.3 Robust CACC Design

In this section, we discuss the design of a robust CACC system in the framework of robust $\mathscr{H}_\infty$ control. To this purpose, we first introduce the sources of uncertainty and describe how to quantify them.

5.3.1 Sources of Uncertainty

There are several reasons to incorporate robustness into a control design framework as there usually exist several sources of uncertainty within any dynamic system. There are always parameters that are only approximately known or are modestly in error. Also, linear models may only be adequate for a small operating range, and original measurements taken to find parameters have

inherent errors despite calibration. If the model is obtained through system identification methods, at high frequencies the structure of the model can become unknown and uncertainties in parameters always arise. Finally, there might be uncertainties within the controller [5]. There are several different approaches to model uncertainties, which could be classified under structured or unstructured uncertainties [5].

With respect to CCAC applications, the authors in [23] note that "the parameters were estimated using a least-squares method." Several other authors have noted that alongside parameter variations seen in portion of (5.2) associated with the i^{th} vehicle parameters, the use of radar (or lidar) and the DRSC band gives other sources of uncertainties [22]. Indeed, since all CACC systems run on onboard processors, albeit real-time systems, there is still a non-uniform processing time that adds to the potential time delays resulting in uncertainties in the plant.

5.3.2 Representing Uncertainty

For the plant given in (9), the parameters ϕ and τ are assumed to have the nominal values of $\bar{\phi}$ = 0.2 sec. and $\bar{\tau} = 0.1$ sec., where we consider a variation with $\phi \in [0.05, 0.5]$ and $\tau \in [0.02, 0.2]$. To guarantee the closed-loop system stability in the presence of the model uncertainty associated with ϕ and τ we first represent the lumped parameter *multiplicative uncertainty* as shown in Figure 5.6 and equation (5.14).

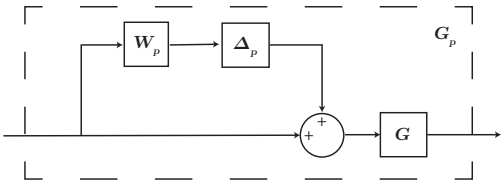

Figure 5.6: Lumped parameter multiplicative uncertainty.

$$G_p(s) = G(s)(1 + W_p(s)\Delta_p(s)), \tag{5.14}$$

where $G_p(s)$ represents the perturbed model, $G(s)$ represents the nominal model, $\|\Delta_p\|_\infty \leq 1$, and W_p represents the lumped uncertainties transfer function that satisfies [5]

$$\left| \frac{G_p(j\omega) - G(j\omega)}{G(j\omega)} \right| \leq |W_p(j\omega)|, \tag{5.15}$$

for any frequency ω. We then let ϕ and τ vary over each respective parameter set. Using a fine grid, we plotted the left hand side of (5.15) on a Bode plot shown in Figure 5.7, where in (5.15) $G_p(s)$ is taken as the perturbed plant and $G(s)$ is fixed as the plant at $\bar{\phi}$ and $\bar{\tau}$. Then, a high pass filter, $W_p(s)$ was fitted to the Bode plot according to (5.15). This results in the following high-pass filter

$$W_p(s) = \frac{6s + 0.003}{s + 14}. \tag{5.16}$$

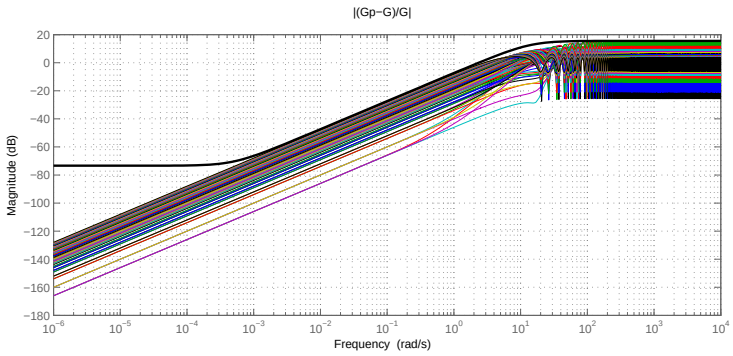

Figure 5.7: Bode plots to find the multiplicative uncertainty weight.

5.3.3 Loop Shaping for $\mathscr{H}_\infty$ Control Design

Next, we use the loop shaping approach [5] to design a controller that can guarantee tracking with zero steady-state error and a low control effort. The corresponding block diagram in Figure 5.8 depicts how disturbances and noise signals affect the closed-loop system. Using this block diagram setup, as in [2], the string stability requirement can be directly handled within the $\mathscr{H}_\infty$ framework. In standard loop shaping, weight W_e shown in Figure 5.8 is tuned to penalize tracking error at low frequencies. The weight W_e is selected to be a low pass filter, tuned to eliminate the steady-state error, as

$$W_e(s) = \frac{0.028}{s+0.02}. \tag{5.17}$$

Next, we select the *desired* acceleration, u_i, as an exogenous output signal [2]. Writing the transfer function between the exogenous input, i.e., the previous vehicle's acceleration u_{i-1}, and the *desired* acceleration u_i yields,

$$T_i(s) = \frac{u_i(s)}{u_{i-1}(s)}. \tag{5.18}$$

If $||T_i(j\omega)|| \leq 1$ for any ω, we have achieved string stability. The weight W_p is a high pass filter used to model the multiplicative uncertainties as discussed in the previous section.

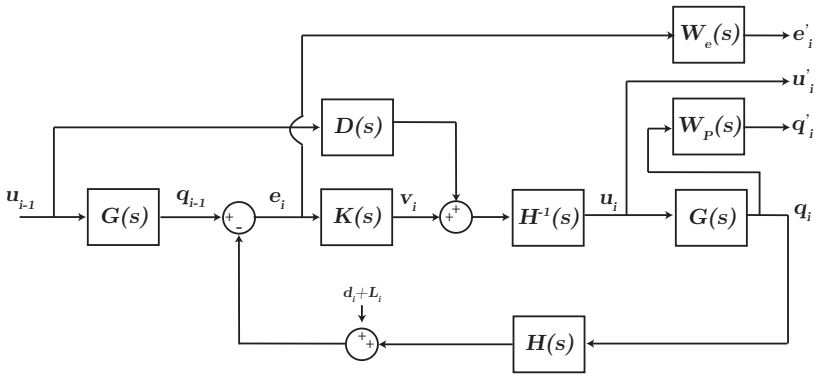

Figure 5.8: Configuration of the closed-loop control system.

5.3.4 H_∞ Robust Control Design for Connected Vehicles

After selecting the loop shaping weights, we use MATLAB to represent the system interconnection shown in Figure 5.8 into the linear fractional transformation (LFT) form. This is done by using the MATLAB command *sconnect*. Next, we express the closed-loop system as

$$z(s) = N(s) * w(s), \tag{5.19}$$

where z represents the vector containing controlled output signals, $N(s)$ describes the closed-loop system transfer function matrix and w represents the exogenous input signals [5]. In formulating

the closed-loop system, the delays associated with (5.9) and (5.12) are approximated by using a 3^{rd} order Padé approximation. Now, by imposing the requirement that,

$$||N(j\omega)||_\infty \leq 1, \tag{5.20}$$

string stability will be achieved. Next, the robust control design problem is solved by invoking the MATLAB command *hinflmi*. A 13^{th} order controller is synthesized to satisfy (5.20). We finally use model order reduction methods to reduce the order of the controller. First, a Gramian-based balancing of state-space realization is performed to isolate states with negligible contribution to the input/output response. This results in an 8^{th} order controller. We further reduce the controller to 6^{th} order by using a balanced truncation model order reduction. Comparing the Bode plot of the 13^{th} order system with the 6^{th} order system shows a good approximation over all frequencies while also satisfying the requirement in (5.20).

5.4 Simulation Results and Discussion

Using the reduced-order controller designed in the previous section, we perform a 5-car simulation with the nominal values of $\bar{\phi}$ = 0.2 sec and $\bar{\tau}$ = 0.1 sec. The results are shown in Figures 5.9 and 5.10 illustrating the string stable behavior, along with the desired tracking performance. For the simulation we follow the same smooth velocity step as in the previous section, where the lead car decreases velocity from 60 kph to 40 kph. Figure 5.10 shows a low value of the error in the response, also demonstrating that after the first following car in the string, the error becomes negligible all together.

Next, by inspecting the block diagram given in Figure 5.8, we write the sensitivity and complementary sensitivity functions as

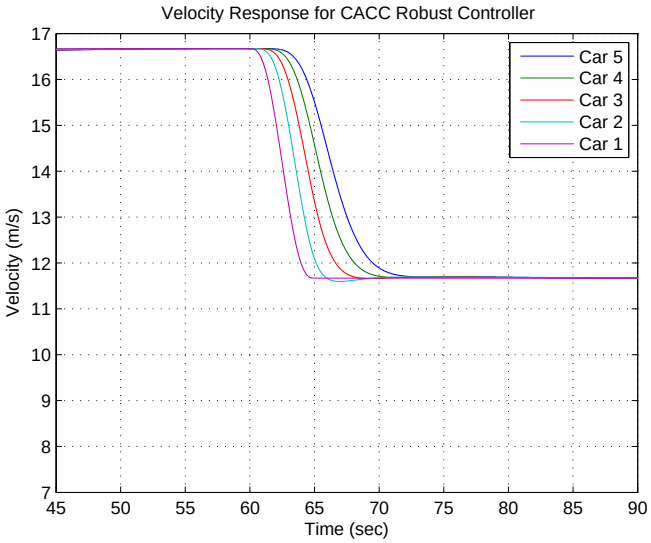

Figure 5.9: Velocity simulation using the designed robust controller.

$$S(s) = \frac{G(s)(1-D(s))}{1+G(s)K(s)}, \tag{5.21}$$

$$T(s) = \frac{H^{-1}(s)(G(s)K(s)+D(s))}{1+G(s)K(s)}. \tag{5.22}$$

Figure 5.11 shows the corresponding Bode plots, which illustrate that string stability is achieved according to (5.18) as the complementary sensitivity transfer function $T(s)$ is always less than 1 at all frequencies. Additionally, Figure 5.12 shows the corresponding robust stability margin illustrating that in the given design, robust stability is achieved. This can be seen from Figure 5.12 since $||W_p(j\omega) * T(j\omega)|| \leq 1$ for any ω.

Next, using the reduced-order robust controller we perform the 5-car simulations for the parameter values of $\phi = 0.5$ sec. and $\tau = 0.2$ sec. We then also perform the same 5-car simulation with the same perturbed parameter values for a standard (non-robust) $\mathscr{H}_\infty$ controller designed in [2]. Figure 5.13 shows the velocity response of the robust controller, demonstrating that the brief undershoot is

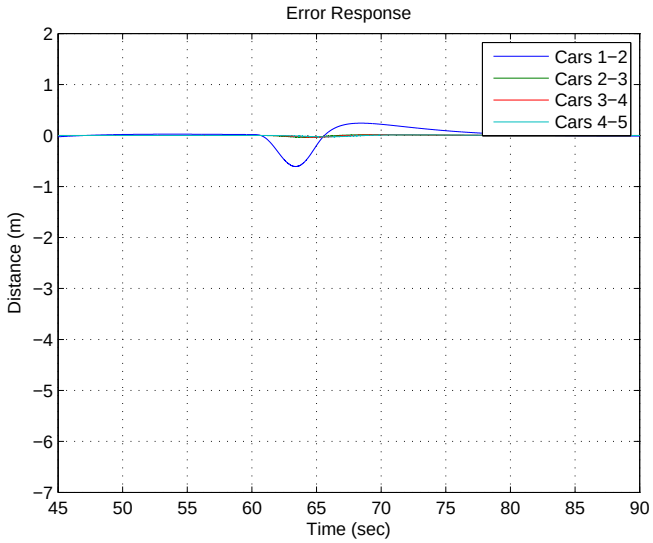

Figure 5.10: Error responses using the designed robust controller.

quickly damped out. Figure 5.14 shows that the non-robust controller experiences several oscillations before reaching steady state. For both controllers, only the response of the first following car is considered non-trivial (similar to Figure 5.10), and a comparison of the error response between the two controllers is given in Figure 5.15. Comparing the two sets of simulations, i.e., the proposed robust design vs. the non-robust one, shows that the robust controller provides a much better performance over the region of parameter perturbation.

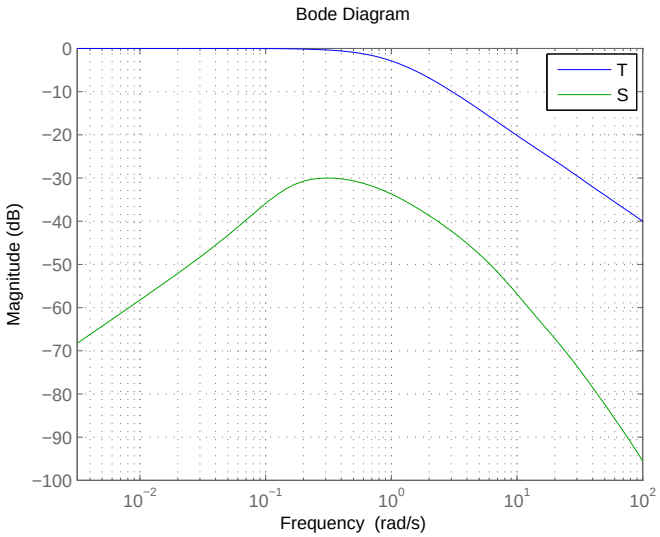

Figure 5.11: Frequency response of sensitivity and complementary sensitivity transfer functions.

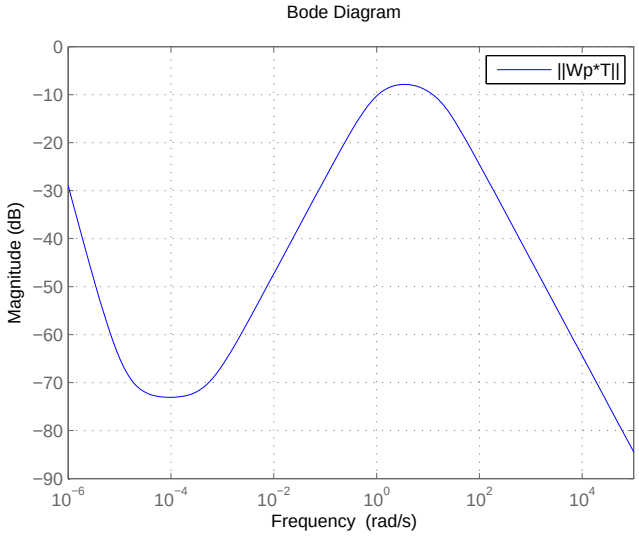

Figure 5.12: Plot showing the robust stability condition.

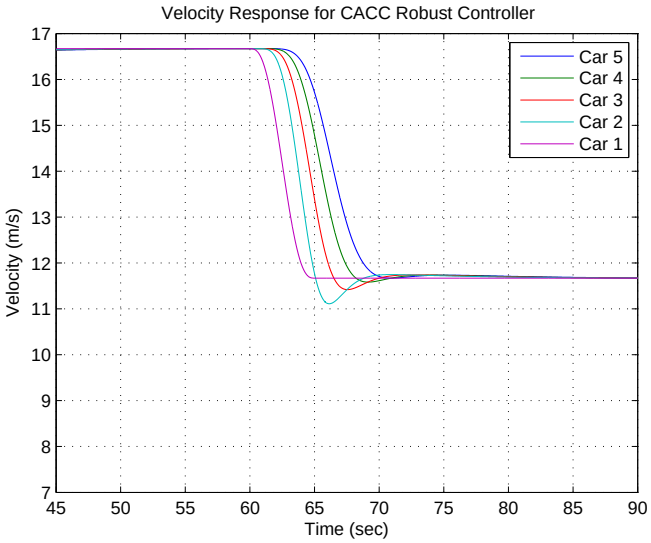

Figure 5.13: Velocity profiles for perturbed 5-car simulations using the proposed robust controller.

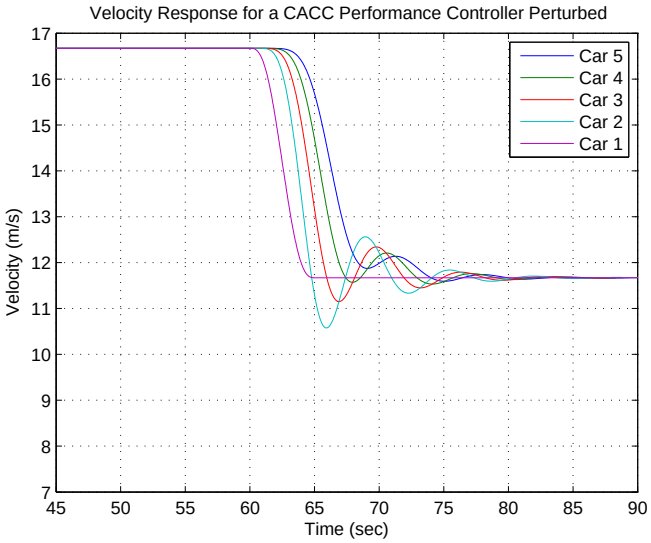

Figure 5.14: Velocity profiles for perturbed 5-car simulations using the (non-robust) controller proposed in [2].

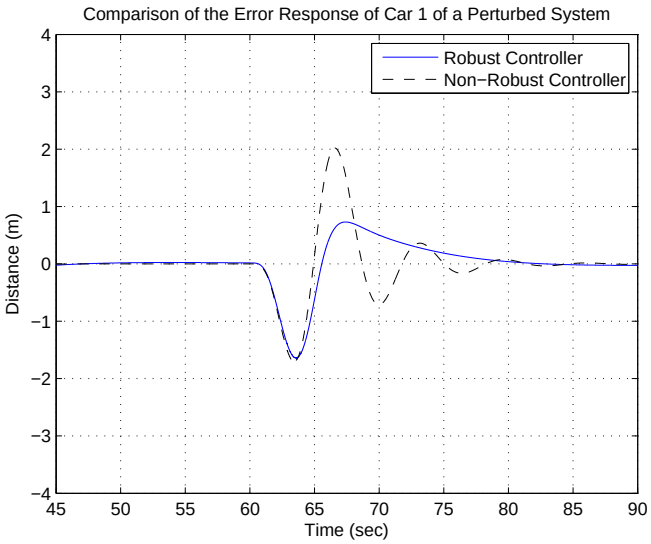

Figure 5.15: Tracking error profiles for perturbed 5-car simulations for the proposed robust controller and the $\mathscr{H}_\infty$ controller designed in [2].

5.5 CONCLUSIONS

In this paper, we have provided some new results on the design and validation of a robust H_∞ controller for cooperative adaptive cruise control (CACC) of connected vehicles. The proposed design framework can account for the uncertainties in the vehicle model used for the CACC design to ensure string stability. The control design process includes: (i) quantifying the effect of uncertainties on the plant model, and (ii) employing the mixed-sensitivity, loop shaping-based $\mathscr{H}_\infty$ control design. Simulation results demonstrate that the robust controller can improve string stability and tracking performance – compared to non-robust designs in the literature – over the region of parameter perturbations.

CHAPTER 6

LUMPED-PARAMETER MODEL DEVELOPMENT AND ROBUST CONTROL OF SYSTEMS GOVERNED BY 2-D PARABOLIC CONVECTION-DIFFUSION EQUATION

[1]

[1] Trudgen, Mark, and Javad Mohammadpour. "Lumped-parameter model development and robust control of systems governed by 2-D parabolic convection-diffusion equation." In 2015 American Control Conference (ACC), pp. 607-612. IEEE, 2015.

6.1 ABSTRACT

In the present paper, proper orthogonal decomposition (POD) method is employed to derive a lumped-parameter model for systems governed by two-dimensional (2-D) parabolic convection-diffusion (PCD) equation. The POD method employs singular value decomposition (SVD) to explore the content of a data set in order to identify the most and least variation to choose lower-order basis functions that provide close approximations of the original data set. In this work, POD is utilized to determine a low-order model that is suitable for control design purposes; using the low-order model, an $\mathscr{H}_\infty$ controller is then designed to ensure closed-loop system stability and reference tracking. This control design framework is chosen since the low-order model presents both parametric uncertainty and unmodeled high frequency dynamics arising from the derivation of the low-order model. A loop-shaping design method is adopted to design a robust $\mathscr{H}_\infty$ controller to achieve desirable tracking and disturbance rejection in the closed-loop system. The simulation results show that the robust controller designed on the basis of the low-order model provides satisfactory reference tracking performance for the system described by the full-order PCD model.

6.2 INTRODUCTION

Recent developments in the control system community with respect to model order reduction techniques have generated interest in applying relevant tools to fluid dynamic systems governed by partial differential equations (PDEs) [41,42]. Often, the fluid flows are described by high-dimensional and/or nonlinear equations, but the fully described fluid dynamical models are not suited for control design purposes [39, 40]. Fundamentally, we find the need to develop low-order models that can capture and accurately represent the system dynamics so that a controller of reasonable order can be designed [40].

Convection-diffusion equations are representative of a class of fluid dynamic systems that accurately describe a particular flow phenomenon, but are mathematically not suited for control design

purposes [38]. Therefore, in order to arrive at lower-order models tractable for control design purposes, there is a need to use a suitable projection method [41]. Convection-diffusion processes are widespread in occurrence in many scientific and engineering fields. A few common applications are: pollutant dispersal, vorticity transport in the incompressible Navier-Stokes equations, atmospheric pollution, semiconductor equations, the Stephan problem on a variable mesh, and viscous compressible flow [38]. The authors in [43, 44] used finite element method to control the convection-diffusion equation; however, their primary focus was on the mesh constraints related to optimality. Here, we extend the control design problem governed by the convection-diffusion equation using a robust $\mathscr{H}_\infty$ controller.

A great deal of work developed for model order reduction techniques has come within the controls context, and many methods for model order reduction have been suggested [42]. Proper orthogonal decomposition (POD) has become a well-accepted technique to obtain optimal basis functions that lead to low-order models to accurately represent the original full-order models [39–42, 45–48]. POD was first introduced by Lumley [49] in the context of turbulence, and the method of snapshots was first suggested by Sirovich [50]. POD has the intrinsic property that it is completely data dependent, and the modal decomposition used does not assume prior knowledge of how the data is generated [45]. This property is advantageous as no *apriori* information is needed to choose an ideal set of basis functions.

In this paper, POD is utilized as a basis for modal decomposition of an ensemble of functions. Given a set of data that lies in a vector space, the problem is to find a subspace of fixed dimension that minimizes the error between the two vector spaces [41]. We describe an infinite-dimensional parabolic convection-diffusion (PCD) equation (see, e.g., [68]) and then represent it in the discrete time domain. Using the method of snapshots to align the data together, we reduce the problem to a singular value decomposition (SVD) analysis. SVD is a valuable tool for identifying dimensions inside data sets that have the most and least variation [69]. We then form a reduced-order model using Galerkin projection [39]. Lastly, we design an $\mathscr{H}_\infty$ controller and show that this low-order controller is a suitable choice to control the high-order PCD model as it is robust to the natural

parametric uncertainties within the flow that the PCD equation describes, and is also robust to the unmodeled high frequency dynamics that arise due to the use of POD for model reduction [5].

This paper is organized as follows: Section II describes the two-dimensional parabolic convection-diffusion equation and the associated discretized state-space model. In Section III, the high-order model is reduced via the application of POD. Section IV describes the associated $\mathscr{H}_\infty$ controller design process and simulation results, and Section V draws conclusions.

6.3 Full-Order PCD Model and the Discretized Model

In this paper, we consider the two-dimensional parabolic convection-diffusion (PCD) equation as [68]

$$\omega_t = \mu(\omega_{xx} + \omega_{yy}) - c_1(x,y)\omega_x - c_2(x,y)\omega_y + b(x,y)u(t), \tag{6.1}$$

over the spatial domain x, $y \in [0,1]$, where ω is the velocity field [38], $u(t)$ is the source, which is treated as the control input, and the convection coefficients are

$$c_1(x,y) = -x * \sin(2\pi x)\sin(\pi y), \tag{6.2a}$$

$$c_2(x,y) = -y * \sin(\pi x)\sin(2\pi y), \tag{6.2b}$$

$$b(x,y) = 5 * \sin(\pi x)\sin(\pi y). \tag{6.2c}$$

The expression for $b(x,y)$ is valid for $x \geq 0.5$, and otherwise, $b(x,y) = 0$. The boundary conditions for (6.1) are given as

$$\omega_x(t,0,y) = 0, \tag{6.3a}$$

$$\omega(t,x,0) = 0, \tag{6.3b}$$

$$\omega(t,x,1) = 0, \tag{6.3c}$$

$$\omega(t,1,y) = 0. \tag{6.3d}$$

Finally, the initial condition is assumed to be

$$\omega(0,x,y) = 0. \tag{6.4}$$

6.3.1 Discretization of the PCD Model

The partial differential equation (PDE) in (6.1)-(6.2) along with the boundary conditions (6.3) is discretized using a forward time-center space (FTCS) discretization method, which gives

$$\begin{aligned}\frac{\omega_{i,j}^{k+1} - \omega_{i,j}^{k}}{\Delta t} = \mu \Bigg[& \frac{\omega_{i+1,j}^{k} - 2\omega_{i,j}^{k} + \omega_{i-1,j}^{k}}{(\Delta x)^2} \\ & + \frac{\omega_{i,j+1}^{k} - 2\omega_{i,j}^{k} + \omega_{i,j-1}^{k}}{(\Delta y)^2} \Bigg] \\ & - c_1(x_i, y_j) \left[\frac{\omega_{i+1,j}^{k} - \omega_{i-1,j}^{k}}{2\Delta x} \right] \\ & - c_2(x_i, y_j) \left[\frac{\omega_{i,j+1}^{k} - \omega_{i,j-1}^{k}}{2\Delta y} \right] \\ & + b(x_i, y_j) u^k, \end{aligned} \tag{6.5}$$

where Δx and Δy represent the discretization step size in spatial directions, and Δt is the time step; i and j represent the two spatial indices in the x and y dimensions, respectively, and k represents the time index. Following the discretization procedure, x_i and y_j, appearing in the convection coefficients and input distribution function, now represent the unique spatial location that corresponds to the current i^{th} or j^{th} step, respectively.

A simulation result of (6.5) at steady-state is shown in Figure 6.1, where $\Delta x = \Delta y = 1/30$, and $\Delta t = 0.0025$ sec. A nominal value of $\mu = 0.1$ is chosen for the diffusion coefficient, and the control input is chosen to be a unit step function. These conditions are chosen to examine the open-loop response to a constant input signal. In Figure 6.1 we see the boundary conditions and convection coefficients represented.

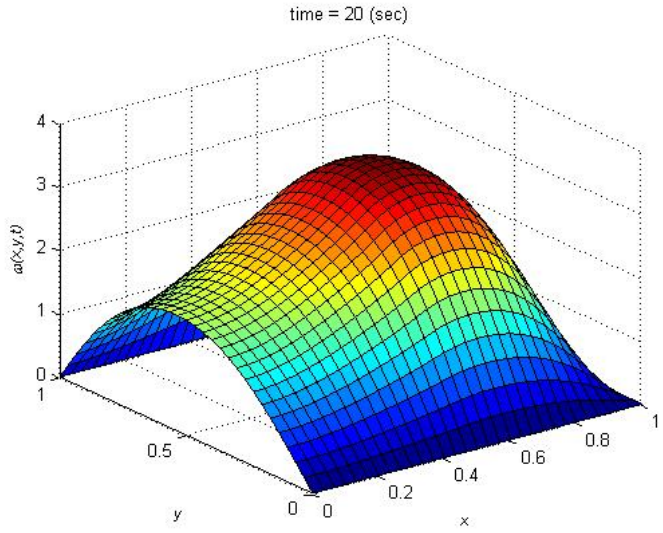

Figure 6.1: Simulation of PCD equation.

$$A = \begin{bmatrix}
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
\bar{\lambda}_y - \alpha_{2,1} & 1-2\bar{\lambda}_x-2\bar{\lambda}_y & \bar{\lambda}_y+\alpha_{2,3} & 0 & \cdots & 2\bar{\lambda}_x & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & \bar{\lambda}_y - \alpha_{3,2} & 1-2\bar{\lambda}_x-2\bar{\lambda}_y & \bar{\lambda}_y+\alpha_{3,4} & \cdots & 0 & 2\bar{\lambda}_x & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & \ddots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & \bar{\lambda}_x+\beta_{n+2,2} & \cdots & \bar{\lambda}_y+\alpha_{n+2,m+1} & 1-2\bar{\lambda}_x-2\bar{\lambda}_y & \bar{\lambda}_y-\alpha_{n+2,m+3} & \cdots & 0 & \bar{\lambda}_x-\beta_{n+2,2m+2} & 0 & \cdots & 0 \\
0 & 0 & \bar{\lambda}_x+\beta_{n+3,3} & \cdots & \bar{\lambda}_y+\alpha_{n+3,m+2} & 1-2\bar{\lambda}_x-2\bar{\lambda}_y & \bar{\lambda}_y-\alpha_{n+3,m+4} & \cdots & 0 & \bar{\lambda}_x-\beta_{n+3,2m+3} & \cdots & 0 \\
0 & 0 & 0 & \ddots & 0 & \ddots & \ddots & \ddots & 0 & 0 & \ddots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0_{n(n-1)+1,1} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0_{n(n-1)+1,m^2} \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
0_{n^2,1} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0_{n^2,m^2}
\end{bmatrix}. \tag{6.6}$$

6.3.2 State-space Model Derivation

Next, using the discretized model (6.5), we derive a discrete time state-space model of the form

$$\begin{aligned} z(k+1) &= Az(k) + Bu(k) \\ y(k) &= Cz(k). \end{aligned} \tag{6.7}$$

The state vector $z(k)$ consists of the discretized PDE variable ω, with n being the total number of steps in the x direction, and m the number of steps in the y direction; the state vector is

$$z(k) = [z_1(k), \ldots, z_{n\times m}(k)]^T. \tag{6.8}$$

The state vector $z(k)$ is arranged with respect to the spatial coordinates, and thus the structure of the elements of the state vector is as follows

$$z(k) = [\omega_{1,1}(k), \ldots, \omega_{1,m}(k), \ldots, \omega_{n,1}(k), \ldots, \omega_{n,m}(k)]^T. \tag{6.9}$$

Additionally, the state vector $z(k)$ is arranged with respect to the given boundary conditions in (6.3). The elements $[\omega^k_{1,1}, \ldots, \omega^k_{1,m}]$ in (6.9) correspond to the Neumann boundary condition in (6.3a). The elements $[\omega^k_{2,1}, \ldots, \omega^k_{n-1,m}]$ correspond to the Dirichlet boundary conditions given in (6.3b)-(6.3c) for the inner elements. Finally, the elements $[\omega^k_{n,1}, \ldots, \omega^k_{n,m}]$ correspond to the Dirichlet boundary condition given in (6.3d) for the outer elements. This same generalized structure, representing the given boundary conditions in (6.3), can be seen in the system matrix A. First, we introduce $\mu = \bar{\mu} + \Delta\mu$, where $\bar{\mu}$ is the nominal value of μ in order to write the A matrix, in which $\bar{\lambda}_x = \frac{\bar{\mu}\Delta t}{(\Delta x)^2}$, $\bar{\lambda}_y = \frac{\bar{\mu}\Delta t}{(\Delta y)^2}$, $\alpha_{i,j} = \frac{\Delta t c_2(x_i,y_j)}{2\Delta y}$, and $\beta_{i,j} = \frac{\Delta t c_1(x_i,y_j)}{2\Delta x}$. Additional discussion on $\bar{\mu}$ is given in Section 4. The corresponding state matrix is given in (6.6).

The input matrix B is prearranged by populating its elements using (6.2c) as

$$B = \begin{bmatrix} 5\sin(\pi x_1)\sin(\pi y_1) \\ \vdots \\ 5\sin(\pi x_1)\sin(\pi y_m) \\ \vdots \\ 5\sin(\pi x_n)\sin(\pi y_1) \\ \vdots \\ 5\sin(\pi x_n)\sin(\pi y_m) \end{bmatrix}. \tag{6.10}$$

The C matrix, representing the system measurements, is formed by assuming that the sensor is located at the middle of the spatial variables range. To verify the state-space representation of the PCD model, i.e., (6.7), we used the same simulation conditions as before, i.e., with $\Delta x = \Delta y = 1/30$ and $\Delta t = 0.0025$ sec., and the corresponding outputs replicate.

For analysis such as the one described above, where a PDE is discretized into a system of ordinary differential equations (ODEs), the higher the order of the model, the more accurate representation of the original system the model becomes. Thus, it is desirable to determine a high-order model to achieve model accuracy. However, lower-order models are advantageous as they are computationally less intensive and more tractable for control design purposes. Hence, a trade-off must be made between model complexity and computational cost. The objective then becomes to use model order reduction techniques to achieve the balance between maintaining the accuracy of the model and obtaining a model of a relatively low order for both time and cost savings.

6.4 Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) method delivers a basis for model decomposition to extract dominant trends and features [39]. POD is a linear procedure and its properties often make it the preferred tool to use in various applications as it is an efficient way to capture the dominant components of even an infinite-dimensional process [42]. Often, this information is captured with only a few modes [39]. POD can be applied to spatio-temporal models, e.g., (6.1), where the goal is to identify a low-dimensional subspace on which to construct a model via projection of the given state-space model in (6.7) [40]. Essentially, POD extracts a set of orthonormal basis functions [62].

First, to approximate any function of interest over a domain of interest, we write the ensemble into coefficients to be determined [41]. For our problem on hand, we start the POD procedure by writing

$$\omega(t,x,y) \approx \widehat{\omega}(t,x,y) = \sum_{k=1}^{M} \alpha_k(t)\varphi_k(x,y), \tag{6.11}$$

where φ_k's define the set of orthonormal basis functions, and α_k's denote the time-dependent coefficients. We employ the method of snapshots - originally suggested by Sirovich [50] - that is a numerical procedure. It solves an eigenvalue problem [39] and only requires an ensemble of appropriately organized data points [62]. The data needed is captured as $\omega_{snap} \in \mathbb{R}^{N\times K}$,

$$\omega_{snap} = \begin{bmatrix} \omega_1^1 & \cdots & \omega_1^K \\ \vdots & \ddots & \vdots \\ \omega_N^1 & \cdots & \omega_N^K \end{bmatrix}, \tag{6.12}$$

where N corresponds to the number of discretization steps and K corresponds to the number of snapshots. In the finite-dimensional case, POD reduces to an SVD problem. To see this connection, we first start with the data matrix and write it as an eigenvalue problem. For a complete discussion of the mentioned problem, see [39]. The SVD is written for ω_{snap} as

$$\omega_{snap} = \Phi\Sigma V^T. \tag{6.13}$$

Using SVD, one can recorrelate data sets to expose various relationships within the data [39]. Also, relationships within the data set are identified and ordered from the most variation to the least [69]. This is done by making use of SVD as

$$\omega_{snap} = \Phi\Sigma V^T = \begin{bmatrix} \Phi_r & \Phi_s \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 & 0 \\ 0 & \Sigma_s & 0 \end{bmatrix} \begin{bmatrix} V_r^T \\ V_s^T \end{bmatrix}. \tag{6.14}$$

The columns of Φ from the SVD form the set of basis functions $\{\varphi_1, \cdots, \varphi_N\}$. This type of projection captures the most *energy* for reduced model [39]. In (6.13), $\Phi \in \mathbb{R}^{N\times N}$ and $V \in \mathbb{R}^{K\times K}$

and the sizes of Φ_r, Σ_r, and V_r each correspond to the M dominant singular values chosen. These basis functions, called POD modes, are used to obtain accurate low-order dynamic models via Galerkin projection [62].

Next, we examine the singular values to produce a reduced-order model. A representation of how much energy is captured by the reduced-order model is given by the differences in the sum of the squared singular values (6.15). A high percentage of energy preserved is always desired, meaning a larger M, which indicates that the model retains more of the information contained in the original snapshots. The preserved energy percent (PEP) is defined as

$$PEP = 100 \times \frac{\sum_{i=1}^{M} \sigma_i^2}{\sum_{i=1}^{N} \sigma_i^2}. \tag{6.15}$$

Furthermore, to proceed with the analysis, we choose an input signal to the system. To gain insight into the system dynamics, we analyzed the Bode plot of (6.7) corresponding to the discretized model with $\Delta x = \Delta y$ = 1/30, and $\Delta t = 0.0025$ sec. This corresponds to a high-order model with 961 states. The Bode plot is shown in Figure 6.2, which represents multiple plots corresponding to $\mu \in [0.06, 0.1]$ as the flow develops.

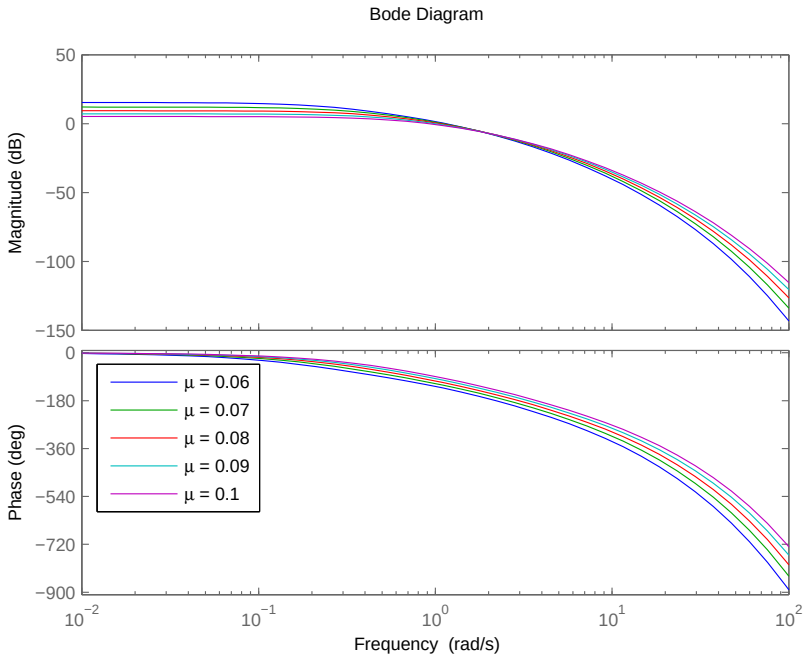

Figure 6.2: Bode plot of the discretized model (6.7).

From the Bode plots depicted in Figure 6.2, it is observed that the system passes through low-frequency inputs, and that it attenuates high-frequency inputs. Therefore, to appropriately excite the system, we use a low-frequency sinusoidal input with multiple frequency components.

The system experiences greater excitation from a low-frequency multi-layered sinusoidal input rather than a single frequency input so we excited the system with the sum of sinusoidal signals with frequencies ranging from 0 to 50Hz with varying amplitudes. The frequency spectrum of the input signal is shown in Figure 6.3.

We used this analysis to determine the amount of captured energy that would be present in the reduced-order models. Figure 6.4 plots the preserved energy percent versus the number of singular values retained (i.e., the order of the reduced-order model or the number of retained POD modes). As observed in Figure 6.4, the first mode retains 78% of the energy, while using the first three

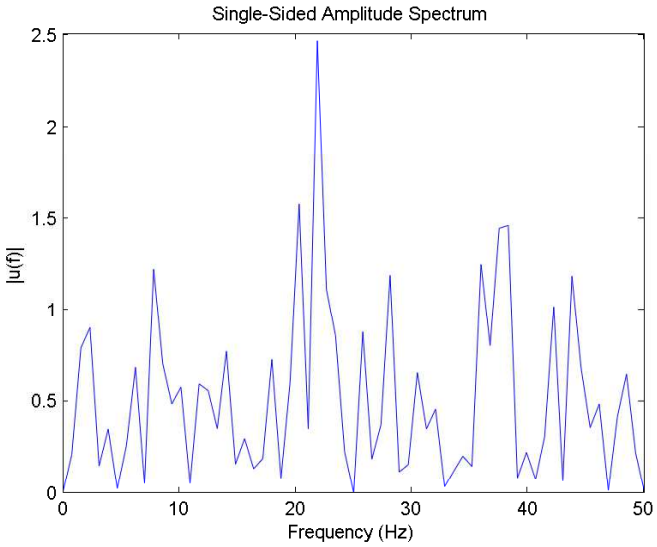

Figure 6.3: Frequency content of the system input.

modes retains approximately 94% of the energy, and the first five modes retain over 98% of the energy of the original high-order model.

To obtain the reduced-order state-space model, (6.7) is multiplied from both sides by the truncated orthonormal matrix $\Phi_r \in \mathbb{R}^{N\times M}$ as

$$\Phi_r^T z(k+1) = \Phi_r^T A z(k) + \Phi_r^T B u(k). \tag{6.16}$$

Recalling that $z(k)$ is the state vector of the original high-order approximation, the reduced-order state vector becomes

$$z_r(k) = \Phi_r^T z(k). \tag{6.17}$$

Since each element of $z_r(k)$ is a linear combination of the elements of $z(k)$, substituting (6.17) into (6.7) yields

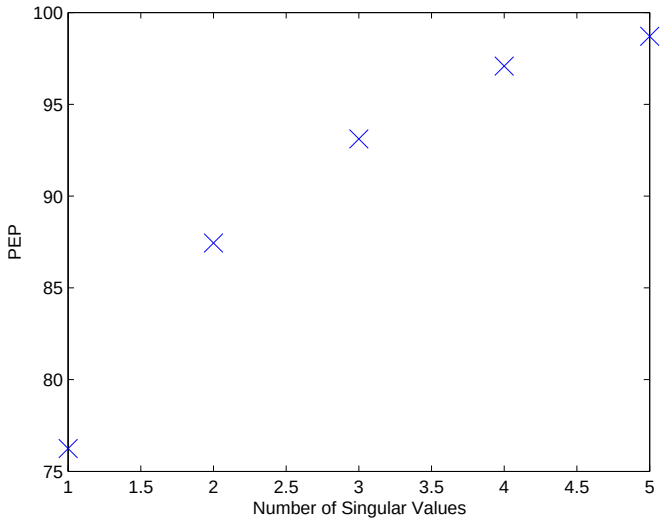

Figure 6.4: Percentage of captured energy.

$$z_r(k+1) = A_r z_r(k) + B_r u(k)$$
$$y(k) = C_r z_r(k), \tag{6.18}$$

with

$$A_r = \Phi_r^T A \Phi_r \qquad B_r = \Phi_r^T B \qquad C_r = C\Phi_r. \tag{6.19}$$

We reduce the full-order model to 5^{th} order, which retains 98% of the *energy* of the high-order model. We note that in the low-order model, A_r, B_r, and C_r can be calculated offline. Next, the reduced-order model is simulated to steady-state with the same conditions (Δx = Δy = 1/30, Δt = 0.0025 sec, and $\mu = 0.1$) used to generate Figure 6.1. Simulation results illustrate a close agreement between the low and high-order models (see Figure 6.5).

6.5 Controller Design and Closed-loop Simulation Results

In this section, we discuss the control design procedure in the robust $\mathscr{H}_\infty$ framework, for which we first introduce the sources of uncertainties and describe how to quantify them.

6.5.1 Representing Uncertainty

An investigation into the natural phenomenon described by the PCD equation revealed that the diffusion term does change as flow develops. We assume the parameter μ to vary in the range $\mu \in [0.06, 0.11]$, with the nominal value of $\bar{\mu} = 0.1$. Indeed, many convection-diffusion equations have time-varying parameters but are instead described with lumped-parameter models for mathematical tractability [38, 70]. However, from a control design perspective, we realize that if we let $\mu = \bar{\mu} + \Delta\mu$, where $\bar{\mu}$ is the nominal value and $\Delta\mu$ represents uncertainty from the steady-state, we can still perform model reduction using $\bar{\mu}$ and later account for the uncertainty $\Delta\mu$. Additionally, the convection coefficients will have a level of uncertainty as (6.1) is derived from incompressible flow assumptions. In light of the competing trade-offs, we choose to derive a model using $\bar{\mu}$, and then use a robust controller design framework to account for the variations in the parameters. All physical systems realistically have a degree of uncertainty due to unmodeled high-frequency dynamics [5], and this is especially true for (6.18) as it is a low-order representation of (6.1). Therefore, we choose to model this as a multiplicative input uncertainty [5].

6.5.2 Robust Stability

To guarantee the closed-loop system stability for $\mu \in [0.06, 0.11]$ and in the presence of the model uncertainty due to the low-order approximation, we first write the lumped parameter multiplicative uncertainty as

$$G_p(s) = G(s)(1 + W_p(s)\Delta_p(s)), \tag{6.20}$$

where $G_p(s)$ represents the high-order model, $G(s)$ represents the low-order model, $\|\Delta_p\|_\infty \leq 1$, and W_p represents the lumped uncertainties [5]. We then simplify to the expression

$$\left|\frac{G_p(j\omega)-G(j\omega)}{G(j\omega)}\right| \leq |W_p(j\omega)|. \tag{6.21}$$

Figure 6.5 shows the Bode plot of the high-order model G_p, and the low-order system G at $\bar{\mu} = 0.1$.

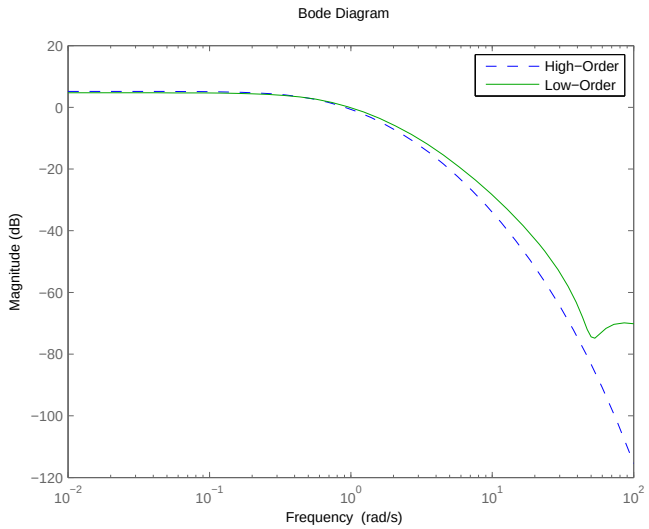

Figure 6.5: Bode plot of the low-order vs high-order model.

Next, we determine an uncertainty weight W_p that satisfies (6.21) for various μ's varying in the interval $[0.06, 0.11]$. This yields the high-pass filter given in (6.22).

$$W_p(s) = \frac{1.85s + 0.1906}{s + 2.097} \tag{6.22}$$

6.5.3 Loop Shaping for Control Design

Next, we use loop shaping techniques [5] to design a controller that guarantees tracking with zero steady-state error and a low control effort. The corresponding block diagram shown in Figure 6.6 depicts how disturbances and noise signals affect the closed-loop system.

In standard loop shaping, three weights W_e, W_p and W_u, shown in Figure 6.6, need to be tuned to penalize the corresponding signals at specific frequency ranges. The weight W_e, selected to be a low pass filter, is tuned to eliminate the steady-state error, W_u to penalize large control inputs to the PCD model, and W_p is a high pass filter used to achieve system robustness with respect to

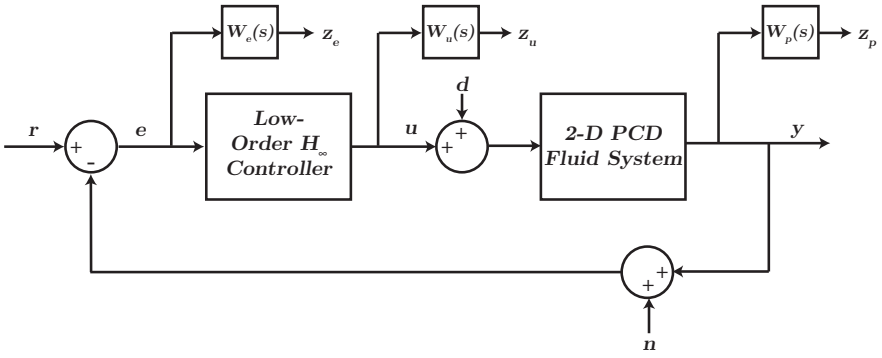

Figure 6.6: Configuration of the closed-loop control system.

uncertainties. W_e is given in (2.11). The weight W_u is chosen to be a high-pass filter to penalize large controller outputs [4]. $W_u(s)$ is given in (6.24). The design of the third weight W_p was discussed earlier and is given in (6.22).

$$W_e(s) = \frac{0.15}{s + 1e^{-5}} \tag{6.23}$$

$$W_u(s) = \frac{s + 0.8}{s + 1.1} \tag{6.24}$$

6.5.4 Simulation Results and Discussion

To design the controller, we used the MATLAB Robust Control toolbox. To this end, we first converted the block diagram shown in Figure 6.6 into the corresponding linear fractional transformation (LFT) form. Next, we used the command *dhinflmi* to design an 8^{th} order controller. The sampling frequency of the simulation was 400Hz, and to simulate the effect of noise n, we corrupted the output signal y with a band-limited white noise with the power $N_P = 0.02$. The closed-loop system shown in Figure 6.6 also accounts for process disturbance corrupting $u(k)$. Finally, the high-order PCD model was discretized by considering $\Delta x = \Delta y = 1/30$, and from the corresponding measurement setup given in (6.7), the C matrix was obtained. We then used a unit

step input, and the simulation result is shown in Figure 6.7 that demonstrates the desired tracking over the range of variation of μ. The designed robust controller guarantees the closed-loop stability for the high-order PCD model with $\mu \in [0.06, 0.11]$.

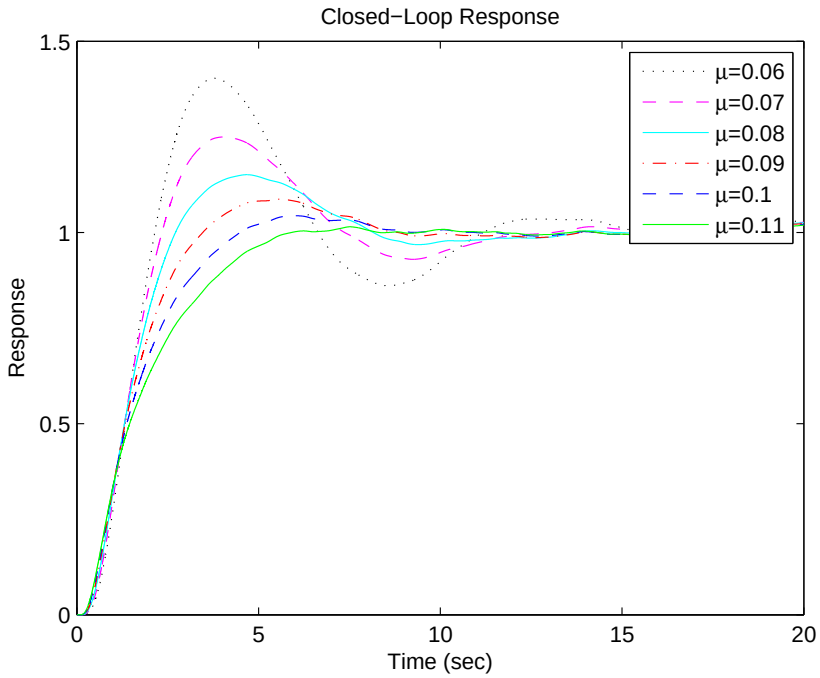

Figure 6.7: Closed-loop response to a reference step input for the robust controller.

From Figure 6.7, it is observed that the robust controller yields a stable closed-loop system. In addition, the H_∞ controller yields an H_∞ norm of $\gamma \approx 1$ for the closed-loop system. A comparison between the nominal and robust controller for the PCD model at $\bar{\mu} = 0.1$ is shown in Figure 6.8. It shows improved performance by the nominal controller which is not surprising since it is designed specifically for $\bar{\mu} = 0.1$. The robust controller is in fact designed for a region, whereas the nominal controller is only designed for a single operating condition [5], and this allows for a better nominal closed-loop performance at the design point.

6.6 Concluding Remarks

In this paper, we employed the proper orthogonal decomposition (POD) method to approximate an infinite-dimensional parabolic convection-diffusion equation with a low-order lumped-parameter

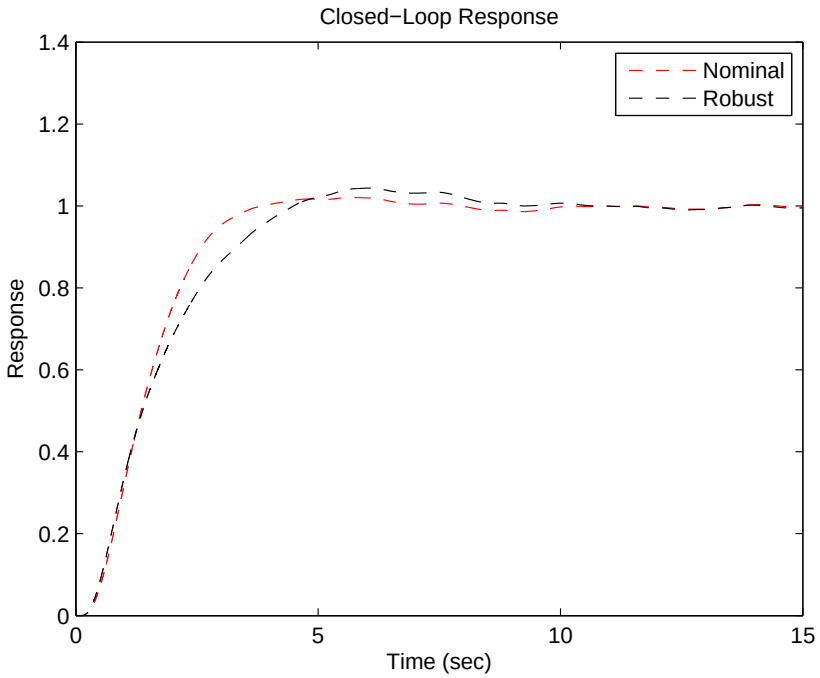

Figure 6.8: Closed-loop response to a reference step input comparing the performance of robust vs nominal controller.

model while preserving a significant portion of the *energy* of the high-order discretized model. We then synthesized an $\mathscr{H}_\infty$ robust controller designed based on the reduced-order model. This was done by implementing loop shaping method to achieve zero steady-state tracking error and robustness against model uncertainties. Simulation results verified that the low-order robust controller could provide the desired tracking for the full-order PCD model with varying coefficients.

CHAPTER 7

CONCLUDING REMARKS

In this dissertation the use of low-order model development and controller design for various automotive and manufacturing applications was presented.

7.1 RAPID THERMAL PROCESSING

In chapters 2 and 3, a new approach was presented for low-order modeling and control of a single wafer RTP system. Using first principles-based models, we developed an LPV model to directly account for all the nonlinearities within the system. The model was then discretized into a high-order affine LPV system; thereafter, the PCA method was utilized to reduce the number of the LPV model's scheduling variables, followed by the use of POD for model order reduction. From the reduced-order system, we designed a gain-scheduled controller to satisfy an induced $\mathscr{L}_2$ gain performance and tracked a temperature reference profile.

We see validation for the necessity of low-order model development. The reduced second order model with three scheduling variables approximated the original nonlinear model well. It retained 95.6% of the scheduling variable information, and 98% of energy of the model order information. This method of modeling nonlinear partial differential equations using the discrete linear parameter-varying state-space allowed for the nonlinear dynamics to be captured. Furthermore, writing into an affine LPV form allowed for powerful model-reduction techniques to be used to obtain low-order controllers. This reduced model was successful in that it allows for feasible gain-scheduled controller design of a model based on first principles and it showed significant improvement over classical PID control. However, a gain-scheduled controller required that sensors provide real-time information to schedule the controller. This adds additional cost and implementation

complexity that might not be available to exiting rapid thermal processing systems. An additional drawback included the lack of closed-form loop shaping weight selection for controller synthesis, which required iterative tuning.

In order to continue development of the control of rapid thermal processing, further research directions can be in the areas of:

- Using the reduced-order LPV model in an adaptive model predictive controller (AMPC) as an AMPC design inherently can impose controller output constraints.
- Modeling RTP systems with other geometries.
- Using kernel based data reduction techniques to achieve a lower number of scheduling variables with comparable accuracy.

7.2 Cooperative Adaptive Cruise Control

We presented a reformulation of the CACC problem. CACC technology relies on real-time acceleration data from a leading vehicle in which the difficulties and the inexact nature of wireless communication data transfer present communication delay, and onboard computational burdens also grant that certain parameters are only estimated within a range. To overcome this, we designed a $\mathscr{H}_\infty$ controller that is robust to all aforementioned uncertainties. The low-order robust controller was then experimentally verified on a laboratory test bed.

As predicted by the reformulated block diagram, the experimental test bed showed the need for robust control. Furthermore, the low-order controller developed from the high-order robust controller also provided a reduced computational burden and footprint on the onboard microprocessor of the test bed which allowed for a faster sampling time while enforcing robust stability and robust performance. The success of this controller is seen in the negligible RMS value change between the nominal and perturbed parameters of the plant when controlled by the robust controller compared with the nominal controller. The strengths of this method include reformulation of the communication delay into the model in order to design a controller that is robust to its variation. The robust

controller design permits a lack of knowledge of uncertain parameters given that they are within a known range. Improvements to the controller design could stem from the knowledge of other system dynamics as the inherent model assumes a linear relationship.

In order to continue development of connected vehicles, future research should include:

- Expanding the test bed for multi-car testing.
- Expanding the longitudinal test bed to incorporate lane changing and maneuvering aspects.

7.3 2-D Parabolic Convection-Diffusion Equation

POD was employed to derive a lumped-parameter model for systems governed by the two-dimensional PCD equation. We derived a low-order model that is suitable for control design purposes; using the low-order model, an $\mathscr{H}_\infty$ controller was then designed via loop shaping to ensure closed-loop system stability and reference tracking.

Partial differential equations, i.e. the PCD equations, are of infinite order. Using discretization and model order reduction techniques, we were able to capture a high portion of the *energy* of the model using a 5^{th} order state-space representation. With the system represented in the state-space, an LMI was solved to synthesize a low-order robust $\mathscr{H}_\infty$ controller to introduce reference tracking into the PDE PCD equation. This technique provides a straightforward methodology to capture the dynamics of a PDE with parameter variation into the state-space where standard LMI solvers can be used for loop-shaping robust controller design. However, because of the discretization approach, the model is inherently simplified, whereas other approaches such as [71] perform controller synthesis without using discretization.

In order to continue development of the control of the PCD equation, future research directions should include:

- Applying robust control techniques to partial differential equations of similar structure.
- Performing experimental tests.

7.4 Final Remarks

Through the tasks undertaken in this dissertation, we show that understanding the type of the uncertainties and nonlinearities within the model directs the choice of low-order control design approach. An LPV model was only justified in one of the three tasks as both scheduling variable reduction and model order reduction are needed for gain-scheduled controller design. However, if parameters are uncertain or time-varying, robust control theory provides adequate modeling given that only model order needs to be reduced. This desire to develop low-order controllers gives focus to the initial modeling assumptions and controller design methodology since low-order controllers are desired for ease in real-time computational implementation.

BIBLIOGRAPHY

[1] E. Dassau, B. Grosman, and D. R. Lewin, "Modeling and temperature control of rapid thermal processing," *Computers & chemical engineering*, vol. 30, no. 4, pp. 686–697, 2006.

[2] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer, "Controller synthesis for string stability of vehicle platoons," vol. 15, no. 2. IEEE, 2014, pp. 854–865.

[3] J. Mohammadpour and C. W. Scherer, Eds., *Control of linear parameter varying systems with applications*. Springer Science & Business Media, 2012.

[4] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H_∞ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.

[5] S. Skogestad and I. Postlethwaite, *Multivariable feedback control*. John Wiley., 2005.

[6] J. Ebert, D. De Roover, L. Porter, V. Lisiewicz, S. Ghosal, R. Kosut, and A. Emami-Naeini, "Model-based control of rapid thermal processing for semiconductor wafers," in *Proc. of American Control Conference*, 2004, pp. 3910–3921.

[7] J. Y. Choi, H. M. Do, and H. S. Choi, "Adaptive control approach of rapid thermal processing," *IEEE Trans. Semiconductor Manufacturing*, vol. 16, no. 4, pp. 621–632, 2003.

[8] W. S. Yoo, T. Fukada, I. Yokoyama, K. Kang, and N. Takahashi, "Thermal behavior of large-diameter silicon wafers during high-temperature rapid thermal processing in single wafer furnace," *Japanese journal of applied physics*, vol. 41, no. 7R, p. 4442, 2002.

[9] C.-A. Lin and Y.-K. Jan, "Control system design for a rapid thermal processing system," *IEEE Trans. Control Systems Technology*, vol. 9, no. 1, pp. 122–129, 2001.

[10] H. Aling, S. Banerjee, A. K. Bangia, V. Cole, J. Ebert, A. Emami-Naeini, K. F. Jensen, I. G. Kevrekidis, and S. Shvartsman, "Nonlinear model reduction for simulation and control of rapid thermal processing," in *Proc. of American Control Conference*, 1997, pp. 2233–2238.

[11] Y. M. Cho and P. Gyugyi, "Control of rapid thermal processing: A system theoretic approach," *IEEE Trans. Control Systems Technology*, vol. 5, no. 6, pp. 644–653, 1997.

[12] E. Zafiriou, R. Adomaitis, G. Gattu *et al.*, "An approach to run-to-run control for rapid thermal processing," in *Proc. of American Control Conference*, 1995, pp. 1286–1288.

[13] C. D. Schaper, M. M. Moslehi, K. C. Saraswat, and T. Kailath, "Modeling, identification, and control of rapid thermal processing systems," *Journal of the Electrochemical Society*, vol. 141, no. 11, pp. 3200–3209, 1994.

[14] C. D. Schaper, "Real-time control of rapid thermal processing semiconductor manufacturing equipment," in *Proc. of American Control Conference*, 1993, pp. 2985–2990.

[15] R. S. Gyurcsik, T. J. Riley, and F. Y. Sorrell, "A model for rapid thermal processing: Achieving uniformity through lamp control," *IEEE Trans. Semiconductor Manufacturing*, vol. 4, no. 1, pp. 9–13, 1991.

[16] P. P. Apte and K. C. Saraswat, "Rapid thermal processing uniformity using multivariable control of a circularly symmetric 3 zone lamp," *IEEE Trans. Semiconductor Manuf.*, vol. 5, no. 3, pp. 180–188, 1992.

[17] T. F. Edgar, S. W. Butler, W. J. Campbell, C. Pfeiffer, C. Bode, S. B. Hwang, K. Balakrishnan, and J. Hahn, "Automatic control in microelectronics manufacturing: Practices, challenges, and possibilities," *Automatica*, vol. 36, no. 11, pp. 1567–1603, 2000.

[18] J. Holdren, E. Lander, and H. Varmus, "Report to the president and congress designing a digital future: Federally funded research and development in networking and information technology executive office of the president," President's Council of Advisors on Science and Technology, Tech. Rep., December 2010.

[19] S. Jones, "Cooperative adaptive cruise control: Human factors analysis," pp. FHWA–HRT–13–045, 2013.

[20] U. S. D. of Transportation, "Focus on congestion relief." Federal Highway Administration, 2012, http://www.fhwa.dot.gov/congestion, Accessed on 2016-07-22.

[21] J. Vander Werf, S. E. Shladover, M. A. Miller, and N. Kourjanskaia, "Effects of adaptive cruise control systems on highway traffic flow capacity," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1800, no. 1, pp. 78–84, 2002.

[22] L. Xiao and F. Gao, "A comprehensive review of the development of adaptive cruise control systems," vol. 48, no. 10, 2010, pp. 1167–1192.

[23] J. Ploeg, B. T. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 260–265.

[24] D. de Bruin, J. Kroon, R. van Klaveren, and M. Nelisse, "Design and test of a cooperative adaptive cruise control system," in *Intelligent Vehicles Symposium*. IEEE, 2004, pp. 392–396.

[25] S. E. Shladover, D. Su, and X.-Y. Lu, "Impacts of cooperative adaptive cruise control on freeway traffic flow," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2324, no. 1, pp. 63–70, 2012.

[26] W. S. Levine and M. Athans, "On the optimal error regulation of a string of moving vehicles," vol. Ac-11, 1966, pp. 355–356.

[27] L. Peppard, "String stability of relative-motion PID vehicle control systems," vol. 19, no. 5. IEEE, 1974, pp. 579–581.

[28] G. Guo and W. Yue, "Sampled-data cooperative adaptive cruise control of vehicles with sensor failures," vol. 15, no. 6. IEEE, 2014, pp. 2404–2418.

[29] G. J. Naus, R. P. Vugts, J. Ploeg, M. Van de Molengraft, and M. Steinbuch, "String-stable CACC design and experimental validation: A frequency-domain approach," vol. 59, no. 9. IEEE, 2010, pp. 4268–4279.

[30] J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Lp string stability of cascaded systems: Application to vehicle platooning," vol. 22, no. 2. IEEE, 2014, pp. 786–793.

[31] D. N. Godbole and J. Lygeros, "Longitudinal control of the lead car of a platoon," vol. 43, no. 4. IEEE, 1994, pp. 1125–1135.

[32] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles," in *American Control Conference, 1990*. IEEE, 1990, pp. 291–296.

[33] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," vol. 15, no. 1. IEEE, 2014, pp. 296–305.

[34] W. van Willigen, M. Schut, and L. Kester, "Evaluating adaptive cruise control strategies in worst-case scenarios," in *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2011, pp. 1910–1915.

[35] V. Turri, B. Besselink, J. Mårtensson, and K. H. Johansson, "Fuel-efficient heavy-duty vehicle platooning by look-ahead control." IEEE, 2014, pp. 654–660.

[36] P. Seiler and R. Sengupta, "An H_∞ approach to networked control," vol. 50, no. 3. IEEE, 2005, pp. 356–364.

[37] U. Montanaro, M. Tufo, G. Fiengo, M. di Bernardo, and S. Santini, "On convergence and robustness of the extended cooperative cruise control." IEEE, 2014, pp. 4083–4088.

[38] K. W. Morton and K. Morton, *Numerical solution of convection-diffusion problems*. Chapman & Hall London, 1996, vol. 12.

[39] P. Holmes, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, 2012.

[40] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *AIAA Journal*, vol. 40, no. 11, pp. 2323–2330, 2002.

[41] C. Rowley, "Model reduction for fluids, using balanced proper orthogonal decomposition," *International Journal of Bifurcation and Chaos*, vol. 15, no. 03, pp. 997–1013, 2005.

[42] A. Antoulas, D. Sorensen, and S. Gugercin, "A survey of model reduction methods for large-scale systems," *Contemporary mathematics*, vol. 280, pp. 193–220, 2001.

[43] A. Valli, G. Carey, and A. Coutinho, "Control strategies for timestep selection in finite element simulation of incompressible flows and coupled reaction–convection–diffusion processes," *International journal for numerical methods in fluids*, vol. 47, no. 3, pp. 201–231, 2005.

[44] R. Becker and B. Vexler, "Optimal control of the convection-diffusion equation using stabilized finite element methods," *Numerische Mathematik*, vol. 106, no. 3, pp. 349–367, 2007.

[45] M. Rathinam and L. R. Petzold, "A new look at proper orthogonal decomposition," *SIAM Journal on Numerical Analysis*, vol. 41, no. 5, pp. 1893–1925, 2003.

[46] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics," *SIAM Journal on Numerical analysis*, vol. 40, no. 2, pp. 492–515, 2002.

[47] S. Volkwein, "Model reduction using proper orthogonal decomposition," *Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz. see http://www. uni-graz. at/imawww/volkwein/POD. pdf*, 2008.

[48] D. A. Lawrence, J. H. Myatt, and R. C. Camphouse, "On model reduction via empirical balanced truncation," in *Proc. of American Control Conference*, pp. 3139–3144.

[49] J. L. Lumley, "The structure of inhomogeneous turbulent flows," *Atmospheric turbulence and radio wave propagation*, pp. 166–178, 1967.

[50] L. Sirovich, "Turbulence and the dynamics of coherent structures," *Quarterly of applied mathematics*, vol. 45, pp. 561–571, 1987.

[51] Y. J. Lee, B. Khuri-Yakub, and K. C. Saraswat, "Temperature measurement in rapid thermal processing using acoustic techniques," *Review of scientific instruments*, vol. 65, no. 4, pp. 974–976, 1994.

[52] J. Liu and Y.-S. Chen, "Simulation of rapid thermal processing in a distributed computing environment," *Numerical Heat Transfer: Part A: Applications*, vol. 38, no. 2, pp. 129–152, 2000.

[53] F. Fasching, S. Halama, and S. Selberherr, *Technology CAD systems*. Springer Science & Business Media, 2012.

[54] A. Theodoropoulou, R. A. Adomaitis, and E. Zafiriou, "Model reduction for optimization of rapid thermal chemical vapor deposition systems," *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 1, pp. 85–98, 1998.

[55] D. De Roover, A. Emami-Naeini, J. L. Ebert, and R. L. Kosut, "Trade-offs in temperature control of fast-ramp rto and rta systems," in *7th Intl Conf. on Advanced Thermal Processing of Semiconds, RTP*, vol. 99. Citeseer, 1999.

[56] H. Lord, "Thermal and stress analysis of semiconductor wafers in a rapid thermal processing oven," *IEEE Trans. Semiconductor Manufacturing*, vol. 1, no. 3, pp. 105–114, 1988.

[57] V. Borisenko and P. J. Hesketh, *Rapid thermal processing of semiconductors*. Springer Science & Business Media, 2013.

[58] A. Virzi, "Computer modelling of heat transfer in czochralski silicon crystal growth," *Journal of crystal growth*, vol. 112, no. 4, pp. 699–722, 1991.

[59] I. Jolliffe, *Principal Component Analysis*, 2nd ed. NY: Springer.

[60] A. Kwiatkowski and H. Werner, "PCA-based parameter set mappings for LPV models with fewer parameters and less overbounding," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 4, pp. 781–788, 2008.

[61] S. Rizvi, J. Mohammadpour, R. Tòth, , and N. Meskin, "A kernel-based PCA approach to model reduction of linear parameter-varying systems," *IEEE Transactions on Control Systems Technology*, pp. 1883–1891, 2016.

[62] S. M. Hashemi and H. Werner, "LPV modelling and control of Burgers equation," in *Proc. 18th IFAC World Congress*, 2011.

[63] M. Trudgen and J. Mohammadpour, "Lumped-parameter model development and robust control of systems governed by 2-d parabolic convection-diffusion equation," in *Proc. of American Control Conference*. IEEE, 2015, pp. 607–612.

[64] K. Lidström, K. Sjoberg, U. Holmberg, J. Andersson, F. Bergh, M. Bjade, and S. Mak, "A modular CACC system integration and design," vol. 13, no. 3. IEEE, 2012, pp. 1050–1061.

[65] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of CACC performance subject to unreliable wireless communication," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). The Hague, The Netherlands*, 2013.

[66] N. Wada and M. Saeki, "Design of a static anti-windup compensator which guarantees robust stability," *Transactions of the Institute of Systems, Control and Instrumentation Engineers*, vol. 12, no. 11, pp. 664–670, 1999.

[67] J. P. Maschuw, G. C. Keßler, and D. Abel, "LMI-based control of vehicle platoons for robust longitudinal guidance," *radio communication*, vol. 11, p. 1, 2008.

[68] J. R. Singler, "Balanced POD for model reduction of linear PDE systems: convergence theory," *Numerische Mathematik*, vol. 121, no. 1, pp. 127–164, 2012.

[69] K. Baker, "Singular value decomposition tutorial," *The Ohio State University*, 2005.

[70] E. L. Cussler, *Diffusion: mass transfer in fluid systems*. Cambridge university press, 2009.

[71] M. Krstic and A. Smyshlyaev, *Boundary control of PDEs: A course on backstepping designs*. Siam, 2008, vol. 16.