The Search for a Reduced Order Controller: Comparison of Balanced Reduction Techniques

Katie A.E. Camp

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mathematics

Belinda B. King, Chair
Jeffrey T. Borggaard
Robert C. Rogers

May 1, 2001
Blacksburg, Virginia

Keywords: Balanced Reduction, Reduced Order Feedback Control, Klein-Gordon Equation, Balanced Truncation, LQG Balancing

The Search for a Reduced Order Controller: A Comparison of Balanced Reduction Techniques

Katie A.E. Camp

(ABSTRACT)

When designing a control for a physical system described by a PDE, it is often necessary to reduce the size of the controller for the PDE system. This is done so that real time control can be achieved. One approach often taken by engineers is to reduce the approximating finite-dimensional system using a balanced reduction method known as balanced truncation and then design a control for the lower order system. The unsettling idea about this method is that it involves discarding information and then designing a control. What if valuable physical information were lost that would have allowed a more effective control to be designed? This paper will explore an alternate balanced reduction method called LQG balancing. This approach allows for the designing of a control on the full order approximating system and then reducing the control. Along the way, the basic ideas of feedback control design will be discussed, including system balancing and model reduction. Following, there will be mention of the linear Klein-Gordon equation and the development of the one-dimensional finite element approximation of the PDE. Finally, simulations and numerical experiments are used to discuss the differences between the two balanced reduction methods.

Acknowledgments

To my advisor, Dr. Belinda King: I thank you for your patience, support, and enthusiasm. You have gotten me excited about doing research again and reminded me what it is like to be excited about mathematics. I am sincerely grateful to have had the opportunity to work with you.

To my committee members, Dr. Jeff Borggaard and Dr. Bob Rogers: I thank you for taking an interest in my project and for your subtle, and sometimes not so subtle, hints that I should consider staying.

To Mom: I thank you for all the sacrifices you have made to make it possible for me to be where I am today. I think Lester would be very proud of both of us.

To Brian: Your help during the last weeks of writing this paper has been priceless. You have convinced me that LaTeX is wonderful! Mission accomplished. I appreciate your immense patience and support. But, most of all, thank you for being my husband.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When designing a control for a physical system described by a PDE, it is often necessary to reduce the size of the controller for the PDE system. This is done so that real time control can be achieved. One approach often taken by engineers is to reduce the approximating finite-dimensional system using a balanced reduction method known as balanced truncation and then design a control for the lower order system.

However, this method may seem a bit troubling in that it involves discarding information and then designing a control. What if valuable physical information were lost that would have allowed a more effective control to be designed? It is for this reason that this paper will also explore the approach of designing a control and then reducing the system. This alternate approach allows for all physical information to be kept while designing the control and then allows for discarding information, based on the form of the controller, may not affect the system or affects it insignificantly. One possible way to do this involves designing a linear quadratic Gaussian (LQG) control and performing LQG balancing to reduce the order of the system.

In this paper the basic ideas of feedback control design will be discussed, including system balancing and model reduction. Following, the PDE of interest in this paper, the linear Klein-Gordon equation, will be introduced. Also discussed will be the development of the 1-D finite element approximation of the PDE. With this groundwork in place, the differences between how the PDE finite-dimensional approximation behaves when both balancing methods are applied will be explored.

Chapter 2

Feedback Control Design

2.1 Full State Feedback Control (LQR)

Much of the discussion in this and the following sections is taken from [3].

Consider the state equation and initial condition

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.1a}$$

$$x(0) = x_0 \tag{2.1b}$$

where $x(t)$ is the state of the system in some state space X; in this work assume X is a Hilbert space. The equation and initial condition

$$\dot{x}(t) = Ax(t) \tag{2.2a}$$

$$x(0) = x_0 \tag{2.2b}$$

describe the uncontrolled system. The operator B describes how the control affects the system and $u(t)$ is the control input from some control space U. According to [5], [14], [17], and [18], PDE control systems can be formulated in this manner.

The idea behind full state feedback control is that the entire state of the system is known at all values of time; then, at each value of time, the control input is determined based upon this complete knowledge of the system. The control is by no means computed in advance.

A classical example of full state feedback control is the linear quadratic regulator design (LQR). LQR design works to find an optimal control by minimizing the cost functional

$$J(u) = \int_0^\infty \{\langle Qx(t), x(t)\rangle_X + \langle Ru(t), u(t)\rangle_U\} dt \tag{2.3}$$

over all controls $u(t) \in U$, where $Q : X \mapsto X$ is a weighting operator for the state and $R : U \mapsto U$ is a weighting operator for the control.

If there exists an optimal control, $u(t)$, it will be given in the form

$$u(t) = -R^{-1}B^*\Pi x(t) = -Kx(t) \tag{2.4}$$

where Π is the solution to the control algebraic Riccati equation

$$A^*\Pi + \Pi A - \Pi B R^{-1} B^* \Pi + Q = 0 \tag{2.5}$$

and $K = R^{-1}B^*\Pi$ is the feedback operator. After determining K, the closed loop system can be formed as

$$\dot{x}(t) = (A - BK)x(t) \tag{2.6a}$$

$$x(0) = x_0. \tag{2.6b}$$

However, there is a problem with implementing full state feedback. Speaking in terms of practicality, it is impossible. For example, let us say we want to design a temperature regulator for a room. In order to implement a full state feedback control for the room, we need to be able to know the temperature at *every* point in the room at *every* instant of time. First of all, a room cannot be filled with thermostats to detect temperature at every point because then there would no longer be a room. Secondly, to collect so much information is not feasible if a real time controller is desired. For these reasons, an alternative approach to full state feedback control is needed.

2.2 State Estimate Feedback Control (LQG)

State estimate feedback control offers a viable option in place of full state feedback control. It is based upon a compensator which provides an estimate of the state. The state estimate is then used in the control law. Just as LQR is a classical example of full state feedback control, linear quadratic Gaussian (LQG) compensator design is a classical example of state estimate feedback control. Both types of control design are discussed in [7] and [12].

To implement LQG, begin as before with the system described in (2.1). Now suppose some state measurement is given by the equation

$$y(t) = Cx(t). \tag{2.7}$$

From the compensator system

$$\dot{x}_c(t) = A_c x_c(t) + Fy(t) \tag{2.8a}$$

$$x_c(0) = x_{c0} \tag{2.8b}$$

a state estimate, $x_c(t)$, can be found. Then the feedback control law is given as

$$u(t) = -Kx_c(t). \tag{2.9}$$

Note that the feedback operator K is the same as previously described for full state feedback control. As a side note, for control problems, the control law can be written in integral form using the Riesz Representation Theorem. That is,

$$u(t) = -Kx_c(t,s) = \textstyle\int_0^L k(s)x_c(t,s)ds, \tag{2.10}$$

where the kernel of the integral, $k(s)$ is called a functional gain. These gains are important for several reasons. First, they can be computed off-line and stored, so that in computation of the control, the gain is multiplied by the state estimate and numerically integrated. In addition, research has been done involving reduced order controllers based on information in the functional gains in [1], [2], [10], [11].

Getting back to the closed loop LQG system, the compensator operators, A_c and F, remain to be computed. Let P be the solution to the filter algebraic Riccati equation

$$PA^* + AP - PC^*N^{-1}CP + M = 0. \tag{2.11}$$

Then define

$$F = PC^*N^{-1} \quad \text{and} \tag{2.12a}$$

$$A_c = A - BK - FC. \tag{2.12b}$$

Based on (2.1), (2.7), (2.8), and (2.9), the closed loop LQG system can be assembled as

$$\begin{pmatrix} \dot{x}(t) \\ \dot{x}_c(t) \end{pmatrix} = \begin{pmatrix} A & -BK \\ FC & A_c \end{pmatrix} \begin{pmatrix} x(t) \\ x_c(t) \end{pmatrix}, \tag{2.13a}$$

$$\begin{pmatrix} x(0) \\ x_c(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ x_{c0} \end{pmatrix}. \tag{2.13b}$$

Note that an implementation problem still exists in that a full order compensator will not offer real time control for most physical problems. For this reason, reduced order compensators need to be considered. However, in order to address this topic, first some attention must be given to the idea of system balancing.

2.3 System Balancing

2.3.1 Balanced Truncation

Before the balancing discussion, first consider the following theory and definitions taken from [13]. For simplicity of discussion, the results presented here correspond to a finite-dimensional system.

Theorem 1 *The linear control process*

$$\dot{x} = Ax + Bu \tag{2.14}$$

where A is a real constant $n \times n$ matrix and B is a real constant $n \times m$ matrix, is controllable if and only if $n \times nm$ matrix

$$[B, AB, A^2B, \ldots, A^{n-1}B] \tag{2.15}$$

has rank n.

Consider the linear control process

$$\dot{x} = Ax + Bu \tag{2.16}$$

where $x \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^m$ is the control vector. Now define an observability equation as

$$y = Cx, \tag{2.17}$$

where C is a real $r \times n$ constant matrix that defines the observable vector $y \in \mathbb{R}^r$ in terms of the state vector x.

Definition 1 *The system formed by (2.16) and (2.17) is called a linear observed process.*

Theorem 2 *The linear observed process in $\mathbb{R}^n$*

$$\dot{x} = Ax + Bu \tag{2.18a}$$

$$y = Cx \tag{2.18b}$$

is observable if and only if

$$rank[C^T \quad A^TC^T \quad [A^T]^2C^T \quad \ldots \quad [A^T]^{n-1}C^T] = n. \tag{2.19}$$

Definition 2 *A matrix A is stable if each eigenvalue λ of A satisfies $Re\lambda < 0$.*

Definition 3 *A system*

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.20a}$$

$$y(t) = Cx(t). \tag{2.20b}$$

is stable if A is a stable matrix.

For the purpose of the following discussion, consider the stable system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.21a}$$

$$y(t) = Cx(t). \tag{2.21b}$$

One way to reduce the order of the compensator associated with the system (2.21) might be to eliminate any states which are not controllable or observable. However, what if a particular state is controllable or observable, but only slightly? That is, is there a degree to which a state is controllable or observable? How can the controllability and observability of a state be measured? To answer such questions, the controllability and observability Grammians are computed. Suppose the pair (A, B) is controllable and the pair (A, C) is observable. Then the controllability Grammian is

$$L_c = \int_0^\infty e^{At} B B^* e^{A^* t} dt \tag{2.22}$$

and the observability Grammian is

$$L_o = \int_0^\infty e^{A^* t} C^* C e^{At} dt. \tag{2.23}$$

The Grammians provide measures of controllability and observability ([16],[5]). Once the most controllable and observable states are identified by the Grammians, they can be used to form a reduced order model. Moore made key contributions to this method in [15].

There are few invariants of the system (2.21) under equivalence transformations, and specifically note that the Grammians are not invariant [16]. This presents serious problems in terms of deciding which states are insignificantly controllable or observable by examining the respective Grammians. For example, the controllability Grammian can be made to appear that very few of the states are important, but this forces the observability Grammian to appear that almost all of the states are important or vice versa. Then, how does one know which states to eliminate? Rather a method of reduction

must be used to find some equilibrium between the number of relevant states from each Grammian.

To answer the need for such a method, Moore [15] shows the existence of an equivalent system where L_cL_0, which is the Hankel operator, is invariant. That is, for any realisation of the system (2.21), L_cL_0 is invariant and contains controllability and observability information about the states of the system. To find such an equivalent system, the idea of balancing is used. The balancing process is described here.

Define a transformation

$$z = Tx \tag{2.24}$$

where T is a nonsingular matrix. Then consider the system

$$\dot{z}(t) = Az(t) + Bu(t) \tag{2.25a}$$

$$y(t) = Cz(t). \tag{2.25b}$$

Now, replace parts of (2.25) by the transformation in (2.24) as appropriate to get

$$T\dot{x}(t) = ATx(t) + Bu(t) \tag{2.26a}$$

$$y(t) = CTx(t). \tag{2.26b}$$

Or to write the system a little differently,

$$\dot{x}(t) = T^{-1}ATx(t) + T^{-1}Bu(t) \tag{2.27a}$$

$$y(t) = CTx(t). \tag{2.27b}$$

Note that this is just another realisation of the system given by (2.21). As mentioned, different realisations have different Grammians and different Grammian eigenvalues. However, $\Lambda(L_cL_o)$ is invariant for any realisation. Thus, to balance, it is necessary to consider the eigenvalues of L_cL_o. Note that when (A, B, C) are transformed to $(balA, balB, balC) = (T^{-1}AT, T^{-1}B, CT)$, the transformed Grammians are $\hat{L}_c = TL_cT^*$ and $\hat{L}_o = (T^{-1})^*L_oT^{-1}$. The columns of T^{-1} are the eigenvectors of the Hankel operator corresponding to the eigenvalues λ_i of the Hankel operator.

In summary, a transformation T is sought such that $\hat{L}_c$ and $\hat{L}_o$ are diagonal and $\hat{L}_c = \Sigma = \hat{L}_o$ where

$$\Sigma = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{pmatrix}. \tag{2.28}$$

and $\Sigma^2 = \Lambda$. A natural ordering arises such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$. These σ_i for $i = 1, 2, \ldots, n$ are called the Hankel singular values. The rate of decay of the Hankel singular values can be observed, and based on this information, particular states may be eliminated that are insignificant in regard to both controllability and observability of the system. Here is the algorithm for finding the necessary transformation T.

Algorithm 1

- *Compute L_c, L_o from A, B, C.*
- *L_c is symmetric positive definite, so the Cholesky decomposition yields $L_c = R^*R$.*
- *R^*L_oR is a way of writing L_cL_o which is symmetric positive definite.*
- *R^*L_oR can be diagonalized by a unitary matrix U such that $R^*L_oR = U\Sigma^2U^*$ where Σ^2 is a diagonal matrix containing the square of the Hankel singular values.*
- *Define $T = \Sigma^{\frac{1}{2}}U^*R$. It follows that $(T^{-1})^*L_oT^{-1} = \Sigma = TL_cT^*$.*

2.3.2 LQG Balancing

Recall that in Section 2.2, the Riccati operators Π and P were identified as the solutions of the control and filter algebraic Riccati equations found respectively in (2.5) and (2.11). Another set of invariants for the system is $\Lambda(\Pi P)$. Instead of balancing around the Grammians as previously described, which would be considered a model balancing, a control type of balancing can be used. That is, the balancing can be based on the Riccati operators. This process allows for the computation of a controller before any physical information is discarded. The balancing process is the same as that previously described. LQG balancing is discussed for the infinite-dimensional PDE system in [4], and the finite-dimensional system is addressed in [9].

2.4 Reduced Order Compensator

To simulate an infinite-dimensional PDE system, a finite-dimensional approximation must be developed. For this paper, the finite element method is applied to the linear Klein Gordon equation. That procedure is discussed in Chapter 3.

For now, assume that the system (2.21) has already been discretized using N basis functions and is represented as

$$\dot{x}^N(t) = A^N x^N(t) + B^N u^N(t), \tag{2.29a}$$

$$x^N(0) = x_0^N, \tag{2.29b}$$

$$y^N(t) = C^N x^N(t). \tag{2.29c}$$

Note the dimensions of the matrices: A is $n \times n$, B is $n \times m$ where m is the number of control inputs, and C is $p \times n$ where p is the number of measurements taken. Using similar notation, the approximations of the full order compensator and feedback law can be written as

$$\dot{x}_c^N(t) = A_c^N x_c^N(t) + F^N y^N(t), \tag{2.30a}$$

$$x_c^N(0) = x_{c0}^N \tag{2.30b}$$

$$u^N(t) = -K^N x_c^N(t). \tag{2.30c}$$

It follows that the approximation to the compensator LQG system is

$$\begin{pmatrix} \dot{x}^N(t) \\ \dot{x}_c^N(t) \end{pmatrix} = \begin{pmatrix} A^N & -B^N K^N \\ F^N C^N & A_c^N \end{pmatrix} \begin{pmatrix} x^N(t) \\ x_c^N(t) \end{pmatrix}, \tag{2.31a}$$

$$\begin{pmatrix} x^N(0) \\ x_c^N(0) \end{pmatrix} = \begin{pmatrix} x_0^N \\ x_{c0}^N \end{pmatrix}. \tag{2.31b}$$

At this point it is appropriate to carry out the balancing process described in Section 2.3, balancing around either the controllability and observability Grammians or the Riccati operators. Through balancing, the matrices A, B, and C in the system (2.21) are transformed into balanced matrices $balA$, $balB$, and $balC$. The resulting balanced matrices depend on the method of balancing used.

Now comes the reduction involved in finding a low order controller. The system to be formed will contain the full order state and a lower order state estimate. To find the matrices needed for the state estimate, the matrices $balA$, $balB$, and $balC$ are truncated to get A^q, B^q, and C^q. This truncation can be based on the rate of decay of the Hankel singular values and Riccati values. Given that the state estimate will contain q states, note the sizes of these reduced matrices: A^q is $q \times q$, B^q is $q \times m$, and C^q is $p \times q$. The lower order state estimate is found by solving the system

$$\dot{x}_c^q(t) = A_c^q x_c^q(t) + F^q y(t) \tag{2.32a}$$

$$x_c^q(0) = x_{c0}^q \tag{2.32b}$$

$$u^q(t) = -K^q x_c^q(t), \tag{2.32c}$$

where A_c^q, F^q, and K^q are found by solving (2.5) and (2.11) with A^q, B^q, and C^q.

Thus, the system to be solved, consisting of the full order state and lower order (q) state estimate, is assembled as

$$\begin{pmatrix} \dot{x}^N(t) \\ \dot{x}_c^q(t) \end{pmatrix} = \begin{pmatrix} A^N & -B^N K^q \\ F^q C^N & A_c^q \end{pmatrix} \begin{pmatrix} x^N(t) \\ x_c^q(t) \end{pmatrix}, \tag{2.33a}$$

$$\begin{pmatrix} x^N(0) \\ x_c^q(0) \end{pmatrix} = \begin{pmatrix} x_0^N \\ x_{c0}^q \end{pmatrix}. \tag{2.33b}$$

For notational purposes, let

$$\begin{pmatrix} A^N & -B^N K^q \\ F^q C^N & A_c^q \end{pmatrix} \tag{2.34}$$

be denoted by A_T in balanced truncation and $ricA_T$ in LQG balancing. Simulations of system (2.33) using both balanced reduction methods will be compared in Chapter 4.

Chapter 3

Klein-Gordon Equation and the Finite Element Approximation

To study the effects of regular and LQG balancing, the linear Klein Gordon equation (KGL) is used as an example. Taking the constants velocity of light in a vacuum and $\hbar$ to be one and adding a damping term, $\gamma\omega_t$, to ensure stabilizability, which is needed to ensure solutions of the algebraic Riccati equations, the KGL equation can be written as

$$\omega_{tt}(t,x) - \omega_{xx}(t,x) + m^2\omega(t,x) + \gamma\omega_t(t,x) = b(x)u(t), \tag{3.1a}$$

$$\omega(0,x) = \omega 1_0(x), \tag{3.1b}$$

$$\dot{\omega}(0,x) = \omega 2_0(x), \tag{3.1c}$$

$$0 \leq x \leq L \qquad t \geq 0, \tag{3.1d}$$

$$\omega(t,0) = 0 = \omega(t,L), \tag{3.1e}$$

where m is the rest mass, L is the length of the spatial domain, t is time, $u(t)$ is the control, and $b(x)$ is the control input function which describes how the control is applied. The KGL equation is a relativistic wave equation which arises in quantum mechanics. It is a momentum equation used when there is a need to describe phenomena at high energies [8].

Using the PDE framework, the KGL equation can be written as a system in the form described in Sections 2.2 and 2.3. Following this, distributed parameter control theory can be used to design a control for the system. To write a finite-dimensional approximation of the PDE as a system, some discretization scheme is introduced. In this research,

the finite element approximation is chosen. To write the finite element approximation to the KGL equation, first write the weak form of the PDE. Considering the weak form of the problem, find an $\omega(t) \in Y \subseteq H_0^1(0, L)$ such that

$$\begin{aligned}\int_0^L \ddot{\omega}(t,x)y(x)dx = &-\int_0^L \omega'(t,x)y'(x)dx - m^2\int_0^L \omega(t,x)y(x)dx \\ &-\gamma\int_0^L \dot{\omega}(t,x)y(x)dx + u(t)\int_0^L b(x)y(x)dx\end{aligned} \tag{3.2}$$

for all $y(x) \in Y \subseteq H_0^1(0, L)$.

In order to develop a finite dimensional problem using the Galerkin finite element approximation, begin by dividing the spatial domain $[0, L]$ into N equidistant subintervals. Then approximate $\omega(t, x)$ by $\omega^N(t, x) = \sum_{i=1}^{N-1} \omega_i^N(t)\varphi_i^N(t)$, where $\varphi_i^N(t)$ are the piecewise linear basis functions depicted in Figure 3.1.

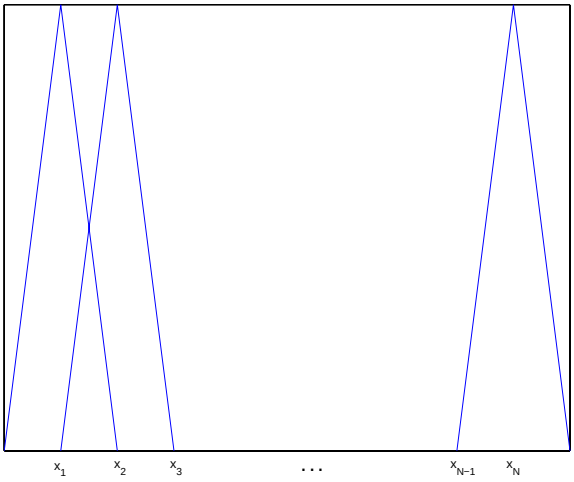

Figure 3.1: Linear Basis Functions

After making the substitution for $\omega(t, x)$ and letting $y(x)$ range over the basis functions, define $v(t, x) = \dot{\omega}(t, x)$. The resulting matrix system corresponding to (2.29)

$$\begin{aligned}\begin{pmatrix} \dot{\omega}^N \\ \dot{v}^N \end{pmatrix} &= \begin{pmatrix} 0 & I \\ -M^{-1}K - m^2 I & -\gamma I \end{pmatrix}\begin{pmatrix} \omega^N \\ v^N \end{pmatrix} \\ &+ \begin{pmatrix} | & | & & | \\ 0 & 0 & \cdots & 0 \\ | & | & & | \\ | & | & & | \\ B_{o1} & B_{o2} & \cdots & B_{om} \\ | & | & & | \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}\end{aligned} \tag{3.3a}$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = C \begin{pmatrix} \omega \\ v \end{pmatrix}, \tag{3.3b}$$

where $M = [\int_0^L \varphi_i(x)\varphi_j(x)dx]_{i,j=1}^{N-1}$ and $K = [\int_0^L \varphi_i'(x)\varphi_j'(x)dx]_{i,j=1}^{N-1}$ are the mass and stiffness matrices, respectively. Note that both M and K are $(N-1)\times(N-1)$ in size. Based upon the finite element basis chosen, M and K were calculated to be given as follows:

$$M = \tfrac{L}{6N} * \begin{pmatrix} 4 & 1 & 0 & \cdots & & 0 \\ 1 & 4 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & 1 & 4 & 1 \\ 0 & & \cdots & 0 & 1 & 4 \end{pmatrix} \tag{3.4a}$$

and

$$K = \tfrac{N}{L} * \begin{pmatrix} 2 & -1 & 0 & \cdots & & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & -1 & 2 & -1 \\ 0 & & \cdots & 0 & -1 & 2 \end{pmatrix}. \tag{3.4b}$$

Since the control input function is piecewise defined in the simulations discussed in this paper, which are described in Chapter 4, note that the control is really

$$b(x)u(t) = \textstyle\sum_{i=1}^{m} b_i(x)u_i(t). \tag{3.5}$$

Then $B_{oi} = \int_0^{x_i} b_i(x)y(x)dx$ for $i = 1, \ldots, m$, where B_{oi} has zeros in all the slots where $b_i(x)$ is zero.

Additionally, for all simulations presented in this paper, four averaging measurements

of both position and velocity are taken and accounted for in the C matrix:

$$C = \begin{pmatrix} \left[\frac{1}{\frac{L}{4}}\int_0^{\frac{L}{4}}\varphi_i(x)dx\right]_{i=1}^{N-1} & 0 \\ \left[\frac{1}{\frac{L}{4}}\int_{\frac{L}{4}}^{\frac{L}{2}}\varphi_i(x)dx\right]_{i=1}^{N-1} & 0 \\ \left[\frac{1}{\frac{L}{4}}\int_{\frac{L}{2}}^{\frac{3L}{4}}\varphi_i(x)dx\right]_{i=1}^{N-1} & 0 \\ \left[\frac{1}{\frac{L}{4}}\int_{\frac{3L}{4}}^{L}\varphi_i(x)dx\right]_{i=1}^{N-1} & 0 \\ 0 & \left[\frac{1}{\frac{L}{4}}\int_0^{\frac{L}{4}}\varphi_i(x)dx\right]_{i=1}^{N-1} \\ 0 & \left[\frac{1}{\frac{L}{4}}\int_{\frac{L}{4}}^{\frac{L}{2}}\varphi_i(x)dx\right]_{i=1}^{N-1} \\ 0 & \left[\frac{1}{\frac{L}{4}}\int_{\frac{L}{2}}^{\frac{3L}{4}}\varphi_i(x)dx\right]_{i=1}^{N-1} \\ 0 & \left[\frac{1}{\frac{L}{4}}\int_{\frac{3L}{4}}^{L}\varphi_i(x)dx\right]_{i=1}^{N-1} \end{pmatrix} \tag{3.6}$$

Based on (3.3), make the following definitions for notational purposes:

$$x^N = \begin{pmatrix} \omega^N \\ v^N \end{pmatrix} \tag{3.7a}$$

$$\dot{x}^N = \begin{pmatrix} \dot{\omega}^N \\ \dot{v}^N \end{pmatrix} \tag{3.7b}$$

$$A = \begin{pmatrix} 0 & I \\ -M^{-1}K - m^2 I & -\gamma I \end{pmatrix} \tag{3.7c}$$

$$B = \begin{pmatrix} 0 \\ M^{-1}\tilde{B} \end{pmatrix}. \tag{3.7d}$$

Then (3.3) can be rewritten as

$$\dot{x}^N = Ax^N + Bu^N \tag{3.8a}$$

$$y^N = Cx^N, \tag{3.8b}$$

which is precisely (2.29) since (3.8) is a finite-dimensional approximation to the KGL equation. Apply the ideas from Section 2.4, to (3.8) to design a reduced order controller.

Chapter 4

Numerical Results

To fit the theory of control design in [4] and [9], Q in the control algebraic Riccati equation (2.5) is taken to be C^*C. Additionally, M in the filter algebraic Riccati equation (2.11) is taken to be BB^*. The control input function $b(x)$ on $[0, L]$, which is part of the B matrix, is taken to be the piecewise defined function $b(x) = e^{-(x-x_i)^2}$, where x_i for $i = 1, 2, \ldots, m$ are the points at the center of m subintervals over the spatial domain of length L. Specifically,

$$b(x) = \begin{cases} e^{-(x-x_1)^2} & 0 \leq x \leq \frac{x_1+x_2}{2} \\ e^{-(x-x_2)^2} & \frac{x_1+x_2}{2} \leq x \leq \frac{x_2+x_3}{2} \\ \vdots & \vdots \\ e^{-(x-x_m)^2} & \frac{x_{m-1}+x_m}{2} \leq x \leq L. \end{cases} \tag{4.1}$$

For now, note the choice of $b(x)$, which is applied on each of the m subintervals. To control the KGL equation amounts to driving this wave type equation to zero. It seems reasonable to choose a $b(x)$ function which will symmetrically disperse the control on each m subinterval. Note that $b(x) = e^{-(x-x_i)^2}$ is symmetric about x_i on each m subinterval. Figure 4.1 depicts $b(x) = e^{-(x-x_i)^2}$, where x_a, x_b, and x_c represent the centers of three subintervals.

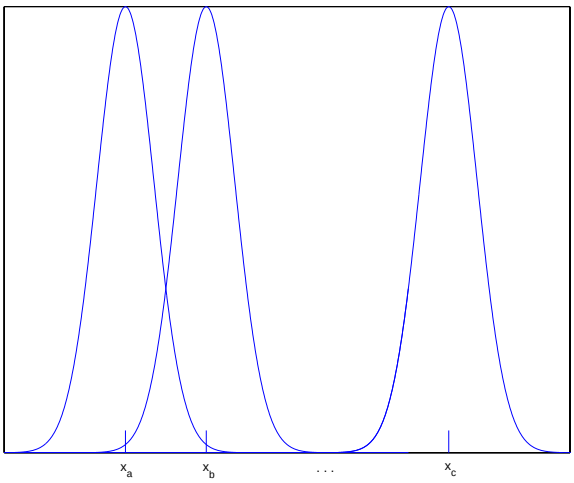

Figure 4.1: Components of Control Input Function b(x)

4.1 Simulations for Initial Condition Set 1

In these particular simulations of the KGL equation, the initial conditions are taken to be

$$\omega(0, x) = \sin(x), \tag{4.2a}$$

$$\dot{\omega}(0, x) = \cos(x). \tag{4.2b}$$

Since the KGL equation is a type of wave equation, one would expect to see some interesting oscillation and wave interaction when using sine and cosine initial conditions.

In the KGL equation (3.1), the mass of the particle is specified to be .01 while the damping constant is set at .1. For the finite element approximation, the number of subintervals, N, is taken to be 100, corresponding to an interval of length 10. The function $b(x)$, as previously described, is applied over the spatial domain in four pieces. That is, $b(x)$ is applied to each quarter of the domain.

4.1.1 Open Loop

Solving the corresponding open loop system amounts to solving

$$\dot{x}^{100}(t) = A^{100}x^{100}(t), \tag{4.3a}$$

$$x^{100}(0) = x_0^{100}. \tag{4.3b}$$

Some of the resulting eigenvalues of the open loop system appear in Figure 4.2.[1]

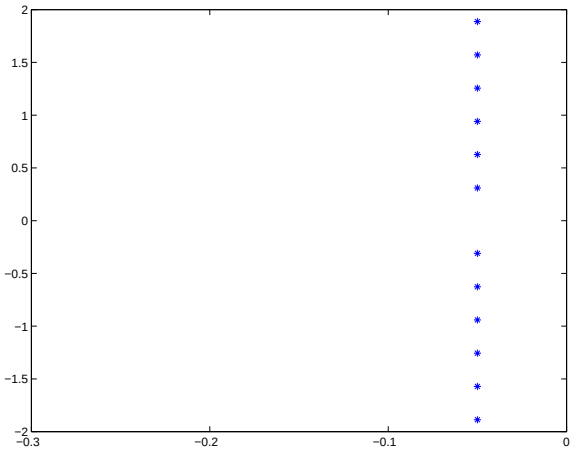

Figure 4.2: Open Loop Eigenvalues

The graph of the simulation is depicted in Figure 4.3.

The solution displays the wave behavior discussed earlier, along with the wave interaction. It should also be noted that the damping term causes the solution to decay, and this is evident in Figure 4.3. To solve the open loop system requires 1.09×10^9 flops.

4.1.2 Full State Feedback Control (LQR)

Recall the system with full state feedback control, given in (2.6). The finite element approximation is given as

$$\dot{x}^{100}(t) = (A^{100} - B^{100}K^{100})x^{100}(t) \tag{4.4a}$$

$$x^{100}(0) = x_0^{100}. \tag{4.4b}$$

Figure 4.4 shows some of the eigenvalues, while Figure 4.5 displays the finite element approximation to the closed loop solution. The graph of the eigenvalues indicates that some of the eigenvalues are being shifted further into the left half plane. This shift provides enhanced stability of the system.

[1]In this paper, the axes of plots containing eigenvalues have been restricted to provide a clearer image of the eigenvalue behavior.

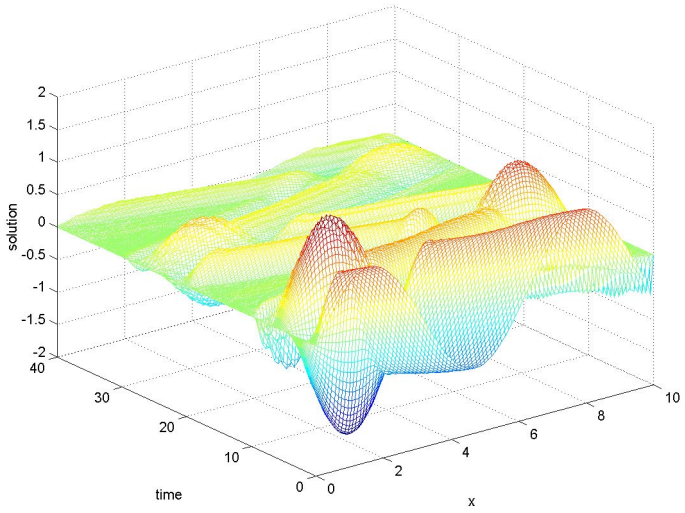

Figure 4.3: Open Loop Solution

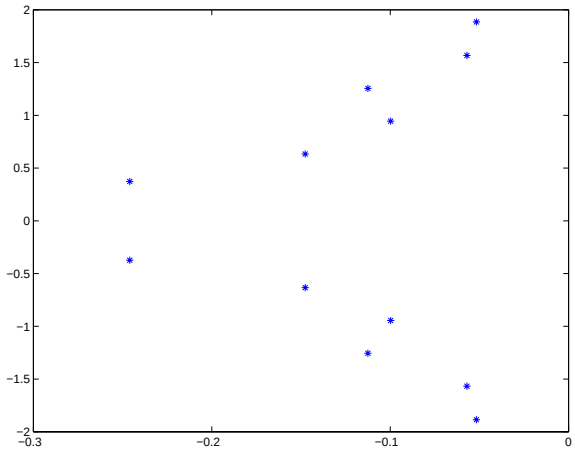

Figure 4.4: Closed Loop Eigenvalues

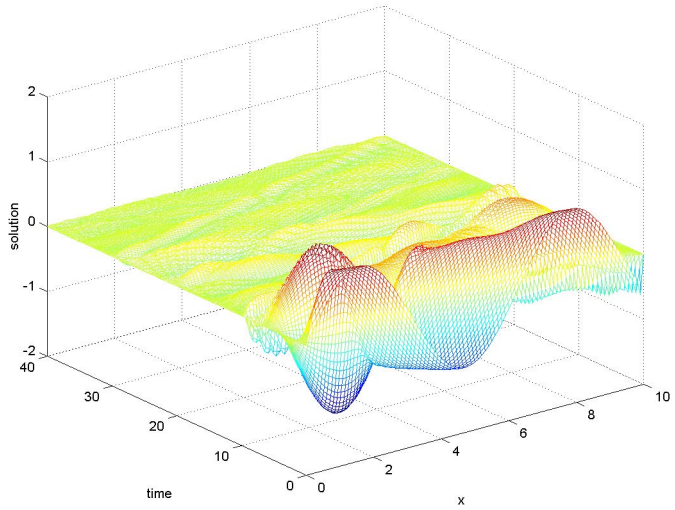

Figure 4.5: Closed Loop Solution

To solve the closed loop system, 1.15×10^9 flops are required. The closed loop solution graph shows the same damping behavior as the open loop solution graph. Additionally, it is evident that the control is driving the solution to zero.

Controller and Functional Gains

A plot of the controller $u(t) = -Kx_c(t)$ is given in Figure 4.6. There are four components, $u_i(t)$, corresponding to each $b_i(x)$.

The full order functional gains for position (left) and velocity (right) are depicted in Figure 4.7.

Recall that the functional gains are independent of initial condition, and the gains depicted here correspond to both the LQR and LQG designs.

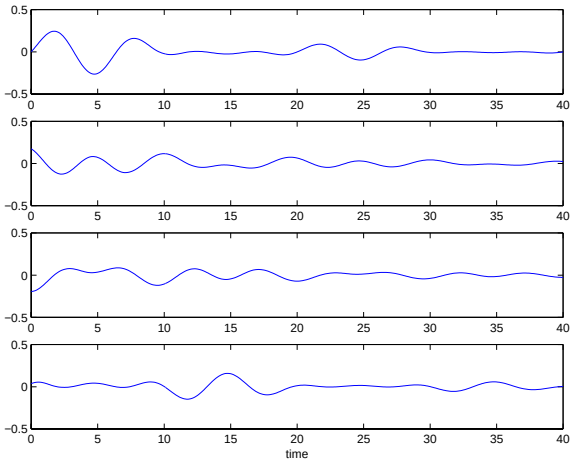

Figure 4.6: Controller for Initial Condition Set 1 (ordered top to bottom)

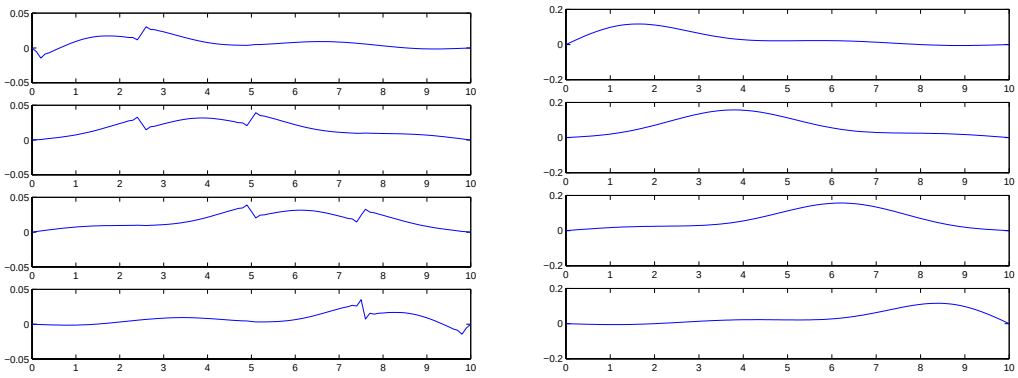

Figure 4.7: Full Order Functional Gains for Position (left) and Velocity (right), (ordered top to bottom within each plot)

4.1.3 Full Order State Estimate Feedback Control (LQG)

For $N = 100$, the full order system with state estimate is given by

$$\begin{pmatrix} \dot{x}^{100}(t) \\ \dot{x}_c^{100}(t) \end{pmatrix} = \begin{pmatrix} A^{100} & -B^{100}K^{100} \\ F^{100}C^{100} & A_c^{100} \end{pmatrix} \begin{pmatrix} x^{100}(t) \\ x_c^{100}(t) \end{pmatrix}, \tag{4.5a}$$

$$\begin{pmatrix} x^{100}(0) \\ x_c^{100}(0) \end{pmatrix} = \begin{pmatrix} x_0^{100} \\ x_{c0}^{100} \end{pmatrix}. \tag{4.5b}$$

To represent an "error" in the initial estimate of the state, $x_c^{100}(0)$ is taken to be $0.75 * x^{100}(0)$.

The full order compensator eigenvalues are displayed in Figure 4.8. Figure 4.9 depicts the full order compensator solution. The flops required to solve this system is 4.45×10^9.

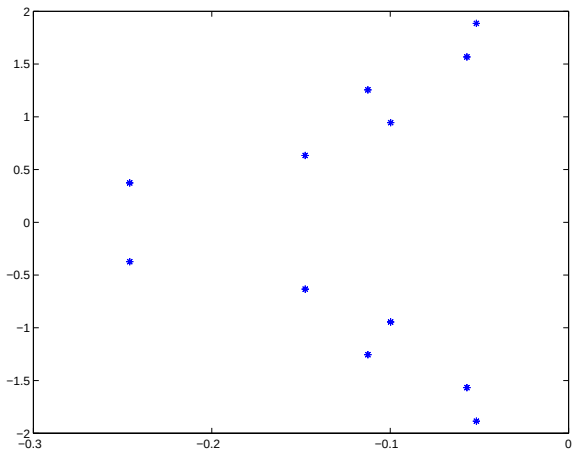

Figure 4.8: Full Order Compensator Eigenvalues

Now see how well the full order compensator does when compared with the closed loop solution. Figure 4.10 shows the difference between these two solutions. Note that the absolute value of the difference is less than 0.4. It is clear from the graph that for $0 \leq t \leq 10$, the LQG model has difficulty capturing the activity of the LQR model.

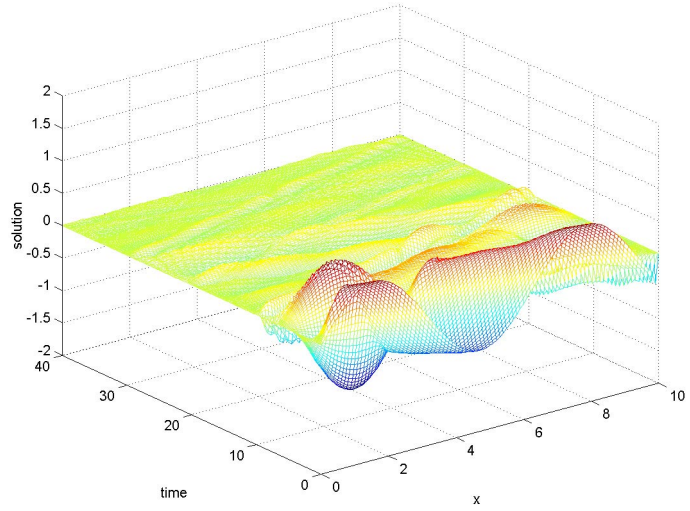

Figure 4.9: Full Order Compensator Solution

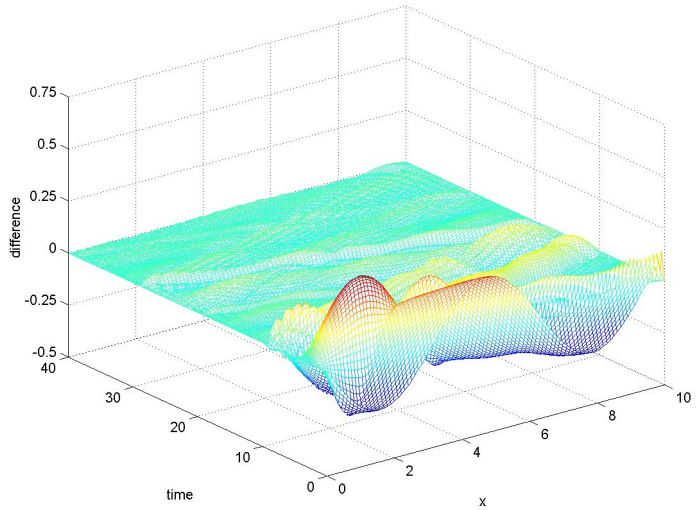

Figure 4.10: Difference Between Closed Loop and Full Order Compensator Solutions

4.1.4 Reduced Order State Estimate Feedback Control via Balancing

As discussed in Section 2.4, the first step involved in finding a reduced order compensator is to balance the system. For the sake of comparison here, the system (2.29) is balanced with respect to both the Grammians and the Riccati operators. Also for the purpose of comparison, for each balancing, there will be simulations involving two state estimates. One simulation will contain a state estimate with ten states, and the other simulation will have fifteen states in the state estimate. The reasoning behind these choices for the sizes of the state estimate will be revealed later.

Suppose the system

$$\dot{x}^{100}(t) = A^{100}x^{100}(t) + B^{100}u^{100}(t), \tag{4.6a}$$

$$x^{100}(0) = x_0^{100}, \tag{4.6b}$$

$$y^{100}(t) = C^{100}x^{100}(t), \tag{4.6c}$$

has been balanced using both methods described in this paper. Figure 4.11 shows the Hankel singular values found through balancing around the Grammians on the left and the Riccati values found through balancing around the Riccati operators on the right.

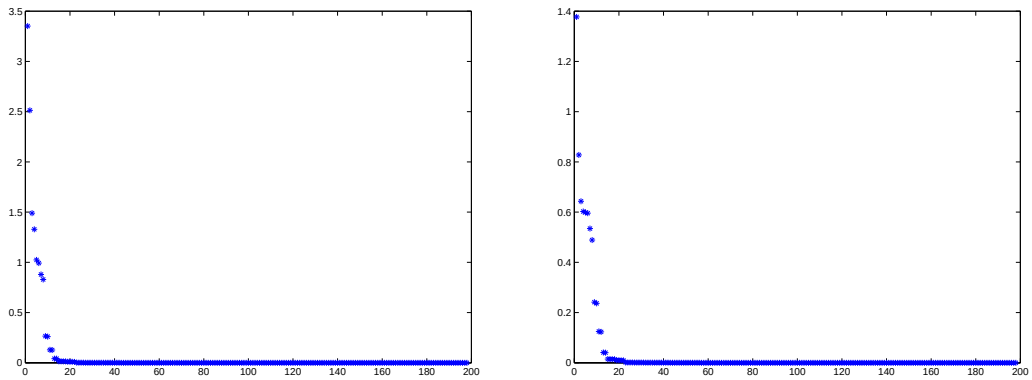

Figure 4.11: Hankel Singular Values (left) and Riccati Values (right)

The purpose of examining Figure 4.11 is to determine the number of Hankel singular values and Riccati values needed to maintain a particular level of significance of information. Table 4.1 summarizes this information.

According to Table 4.1, more Riccati values than Hankel singular values are needed to reach the same level of significance. However, it must be remembered that the balanced

	Values needed for	
	95% significance	99% significance
Hankel singular values	10	14
Riccati values	12	19

Table 4.1: Values needed to achieve 95% and 99% significance.

truncation approach is a model balancing while the LQG balancing method is a control balancing. To compare the number of Hankel and Riccati values needed to reach the same significance is perhaps like comparing apples and oranges. The number of eigenvalues does not necessarily provide an indication of performance. More data is available on this topic, and it can be found in Appendix A.

Functional Gains for Ten States in State Estimate

The functional gains corresponding to the reduced compensator design found through regular balancing can be found in Figure 4.12. The position gains are on the left, and the velocity gains are on the right.

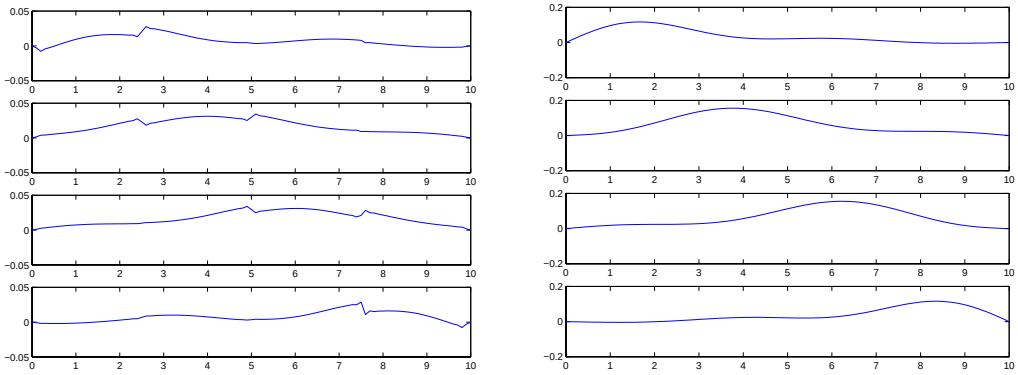

Figure 4.12: Reduced Order Functional Gains for Position (left) and Velocity (right) using Regular Balancing, (ordered top to bottom within each plot)

The functional gains corresponding to the reduced compensator design found through LQG balancing are located Figure 4.13, with the position gains on the left and velocity gains on the right.

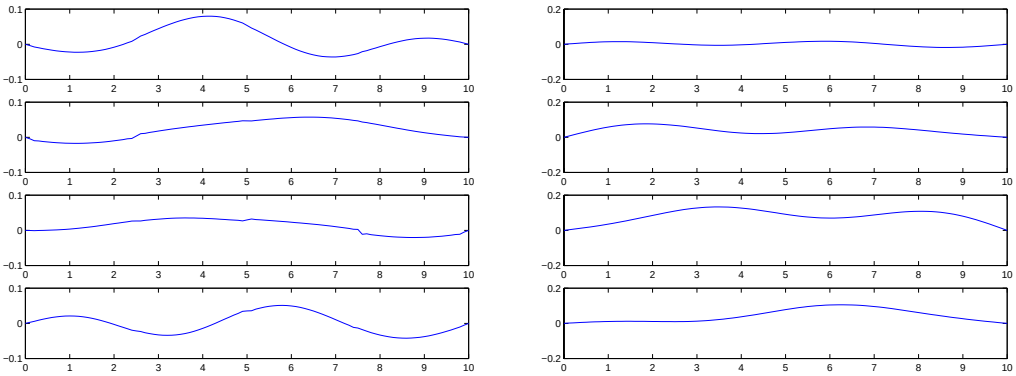

Figure 4.13: Reduced Order Functional Gains for Position (left) and Velocity (right) using LQG Balancing, (ordered top to bottom within each plot)

Simulations with Ten States in State Estimate

Suppose now that truncation on $balA$, $balB$, and $balC$ has been performed. Now form the reduced order compensator system with ten states in the state estimate.

$$\begin{pmatrix} \dot{x}^{100}(t) \\ \dot{x}_c^{10}(t) \end{pmatrix} = \begin{pmatrix} A^{100} & -B^{100}K^{10} \\ F^{10}C^{100} & A_c^{10} \end{pmatrix} \begin{pmatrix} x^{100}(t) \\ x_c^{10}(t) \end{pmatrix}, \tag{4.7a}$$

$$\begin{pmatrix} x^{100}(0) \\ x_c^{10}(0) \end{pmatrix} = \begin{pmatrix} x_0^{100} \\ x_{c0}^{10} \end{pmatrix}. \tag{4.7b}$$

For balanced truncation, the reduced order compensator solution is given in Figure 4.14. This system requires 1.21×10^9.

Comparing Figures 4.9 and 4.14, they have noticeably similar shapes. However, the reduced order compensator contains waves which appear to be more amplified over the entire spatial domain over all time. See the graph in Figure 4.15 to view the difference in the two graphs.

From this graph it can be seen that the absolute value of the difference is less that 0.5, and the lower order model is least accurate for early times. Recall that the full order compensator contains waves of its highest amplitude for $0 \leq t \leq 10$ approximately. It seems that the lower order model has difficulty capturing this behavior.

Also compare the closed loop solution with this reduced order compensator solution. The graph of the difference between the two solutions is found in Figure 4.16. Note

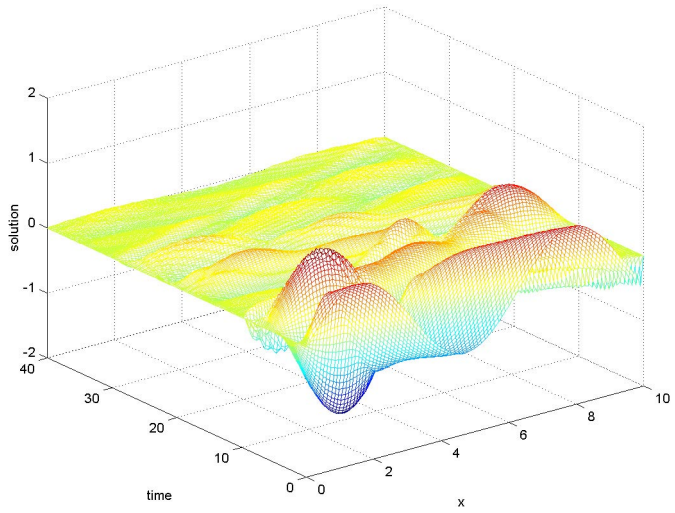

Figure 4.14: Reduced Order Compensator Solution via Balanced Truncation

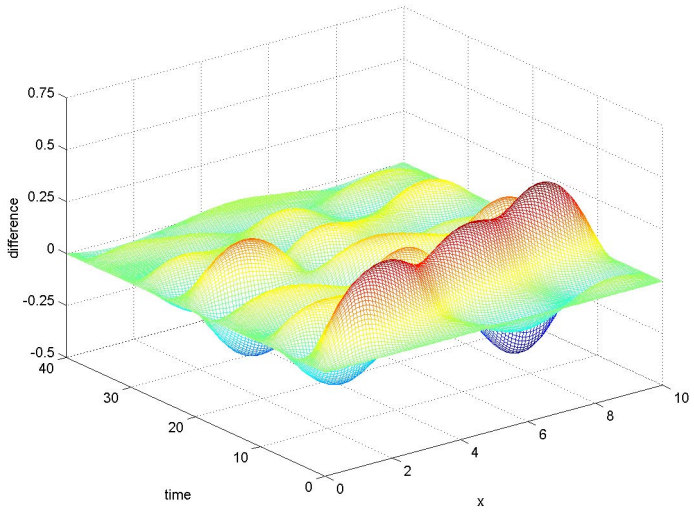

Figure 4.15: Difference Between Full Order Compensator and Reduced Order Compensator via Balanced Truncation

that the absolute value of the difference is less that 0.5, and the lower order model has difficulty capturing the activity of the closed loop solution especially for $0 \leq t \leq 10$.

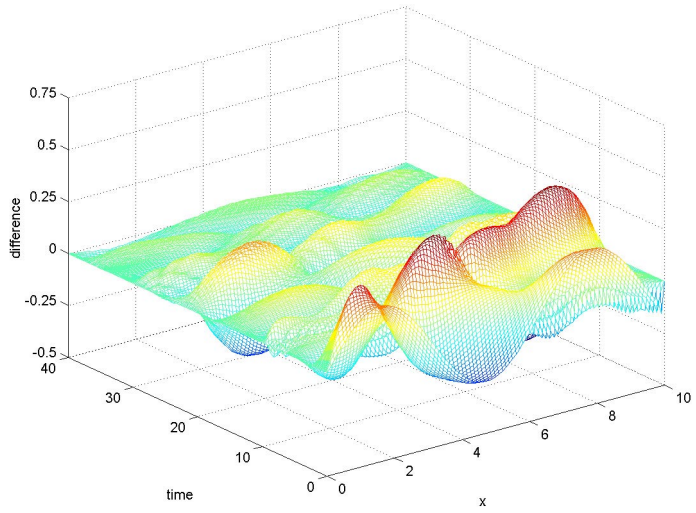

Figure 4.16: Difference Between Closed Loop and Reduced Order Compensator via Balanced Truncation Solutions

Other interesting information is found by comparing the eigenvalues of the full order compensator and the reduced order compensator. For the comparison involving the lower order compensator obtained through balanced truncation, see Figure 4.17.

Notice how several of the eigenvalues from the reduced order compensator shift further into the left half plane. This means that the reduced order system offers greater stability than the full order system.

The reduced order compensator solution obtained using LQG balancing is shown in Figure 4.18.

Again comparing the full order and reduced order solutions in Figures 4.9 and 4.18, it can be said they have similar shapes. The reduced order compensator contains waves which appear to be more amplified, especially for early times. Figure 4.19 depicts the difference in the two graphs.

The absolute value of the difference is higher than before but still less that 0.65, and the lower order model is least accurate for early times. Note that the reduced model obtained

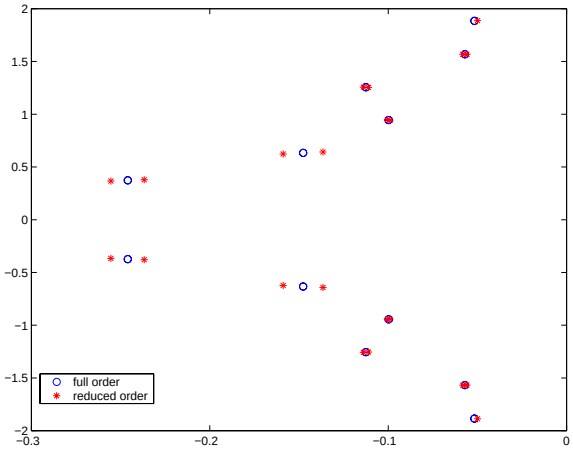

Figure 4.17: Eigenvalues of Full Order Compensator and Reduced Order Compensator via Balanced Truncation

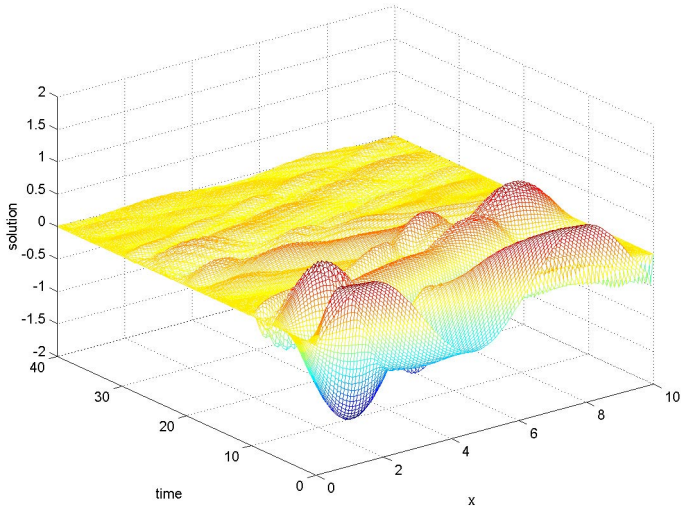

Figure 4.18: Reduced Order Compensator Solution via LQG Balancing

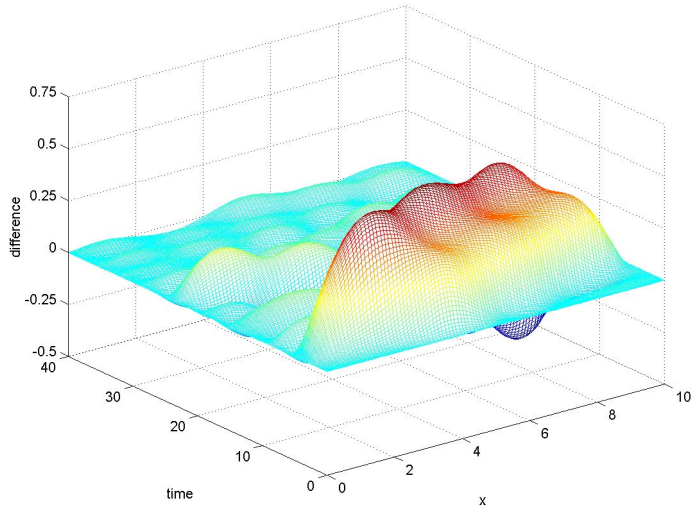

Figure 4.19: Difference Between Full Order Compensator and Reduced Order Compensator via LQG Balancing

through LQG balancing seems to capture the behavior of the full order compensator for later times better than the reduced model obtained through balanced truncation.

Comparing the closed loop and LQG reduced compensator solutions, similar results are found. The graph of the difference in these solutions is depicted in Figure 4.20. For $0 \leq t \leq 10$, the LQG reduced compensator is not clearly representing the behavior of the LQR and full LQG solutions.

As before, now compare the eigenvalues of the full order compensator and the reduced order compensator. Figure 4.21 shows the eigenvalues for the full order compensator and the lower order compensator obtained through LQG balancing.

Again, there is a shift similar to before where several of the eigenvalues from the reduced order compensator shift further into the left half plane. Therefore, this reduced compensator also offers greater stability than the full order compensator.

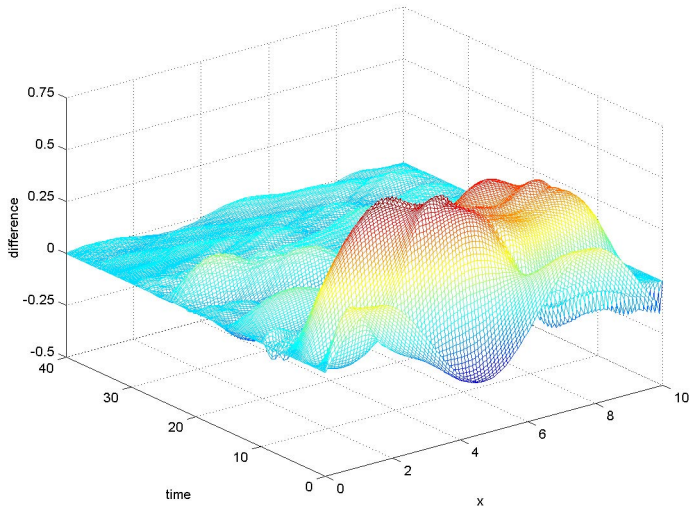

Figure 4.20: Difference Between Closed Loop and Reduced Order Compensator via LQG Balancing Solutions

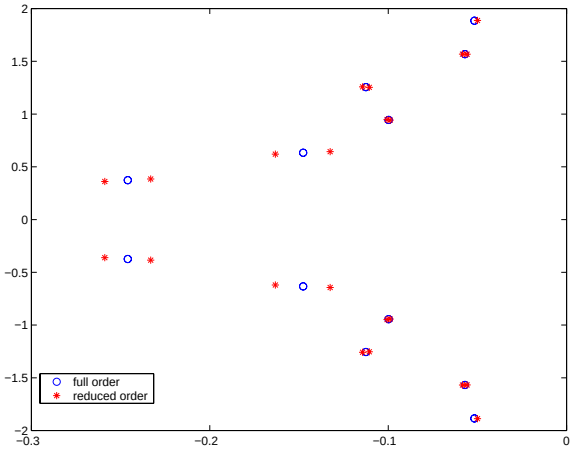

Figure 4.21: Eigenvalues of Full Order Compensator and Reduced Order Compensator via LQG Balancing

Functional Gains for Fifteen States in State Estimate

The functional gains associated with the reduced compensator found using regular balancing can be found in Figure 4.22. The position gains are on the left, and the velocity gains are on the right.

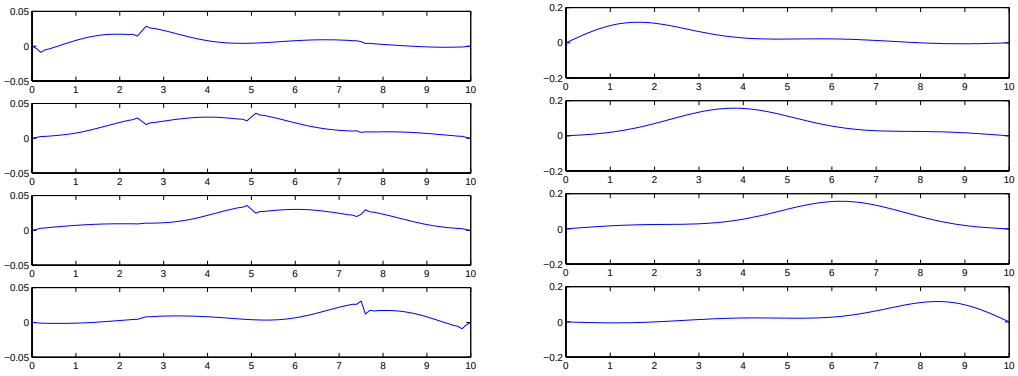

Figure 4.22: Reduced Order Functional Gains for Position (left) and Velocity (right) using Regular Balancing, (ordered top to bottom within each plot)

The functional gains corresponding to the reduced compensator design found using LQG balancing are located Figure 4.23, with the position gains on the left and velocity gains on the right.

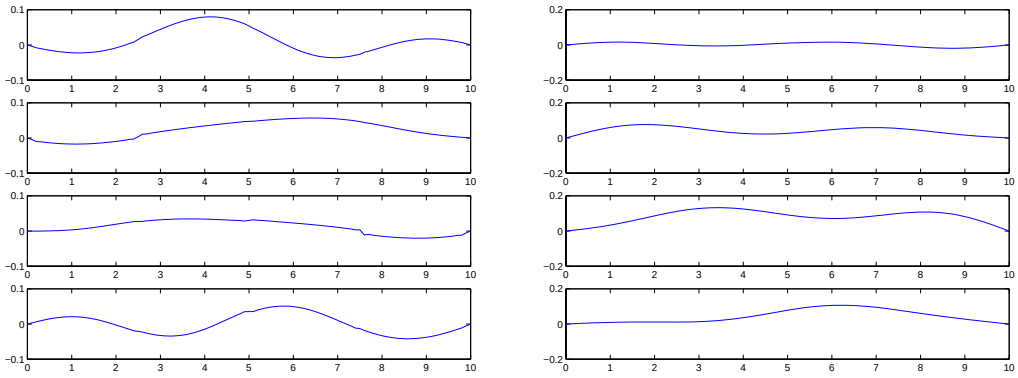

Figure 4.23: Reduced Order Functional Gains for Position (left) and Velocity (right) using LQG Balancing, (ordered top to bottom within each plot)

Simulations with Fifteen States in State Estimate

Suppose once again that truncation on $balA$, $balB$, and $balC$ has been performed. This time form the reduced order compensator system with fifteen states in the state estimate:

$$\begin{pmatrix} \dot{x}^{100}(t) \\ \dot{x}_c^{15}(t) \end{pmatrix} = \begin{pmatrix} A^{100} & -B^{100}K^{15} \\ F^{15}C^{100} & A_c^{15} \end{pmatrix} \begin{pmatrix} x^{100}(t) \\ x_c^{15}(t) \end{pmatrix}, \tag{4.8a}$$

$$\begin{pmatrix} x^{100}(0) \\ x_c^{15}(0) \end{pmatrix} = \begin{pmatrix} x_0^{100} \\ x_{c0}^{15} \end{pmatrix}. \tag{4.8b}$$

Figure 4.24 depicts the reduced order compensator solution obtained through balanced truncation.

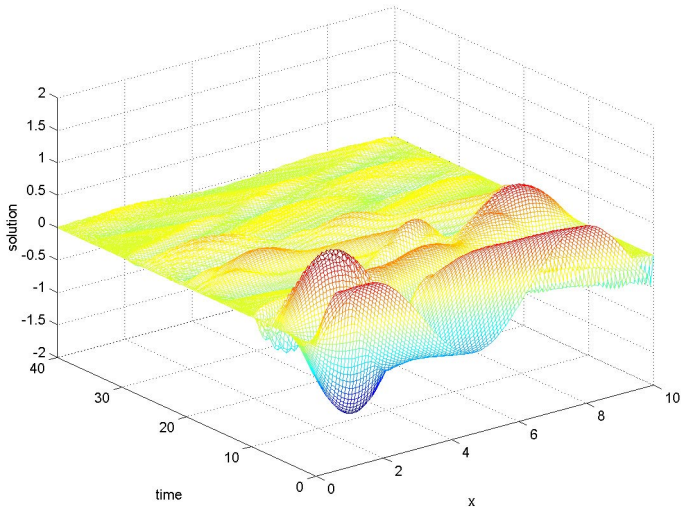

Figure 4.24: Reduced Order Compensator Solution via Balanced Truncation

Figures 4.14 and 4.24 closely resemble each other. The amplitudes of the waves present are quite similar also. It appears that adding five states to the state estimate does not improve accuracy any significant amount. The graph in Figure 4.25 shows the difference between the full order compensator and the reduced compensator.

Again the absolute value of the difference is less than 0.5. However, during early times, the lower order model is amplifying the waves even more so than with ten states in the state estimate. For later times, there is very little difference between Figures 4.15 and

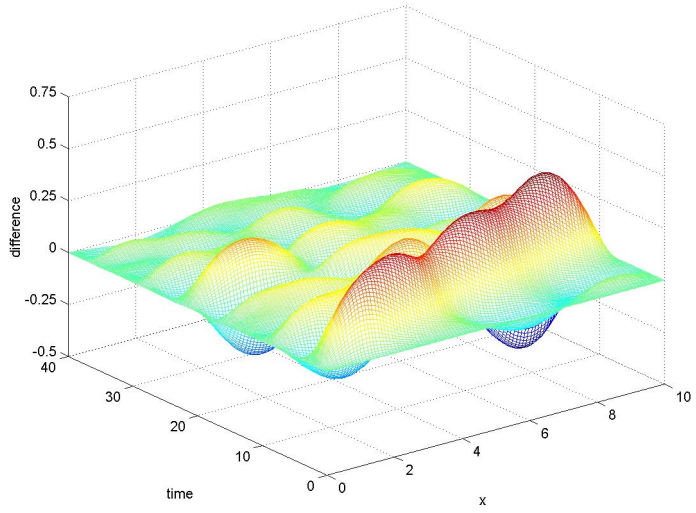

Figure 4.25: Difference Between Full Order Compensator and Reduced Order Compensator via Balanced Truncation

4.25. Looking at the graph of the difference between the closed loop and this reduced compensator solutions, the same bound is found for the difference. This plot is contained in Figure 4.26. In summary, the additional states in the state estimate have not increased the accuracy of the reduced order model.

The reduced order compensator solution obtained using LQG balancing with fifteen states in the state estimate is shown in Figure 4.27.

There appears to be little difference between the reduced order solutions containing ten and fifteen states in the state estimate. The full order and reduced order solutions in Figures 4.9 and 4.27 have similar shapes. The waves present in the reduced order solution are amplified during early times. Figure 4.28 depicts the difference in the two graphs.

The absolute value of the difference is more than with ten states, but still less than 0.75. The lower order model is losing accuracy during early times. For later times, there is very little difference between Figures 4.19 and 4.28. Looking at Figure 4.29, similar comments can be made about the difference between the closed loop and reduced compensator via LQG balancing. Overall, it seems that the additional states in the state estimate have decreased the accuracy in capturing the activity of the full order compensator. However,

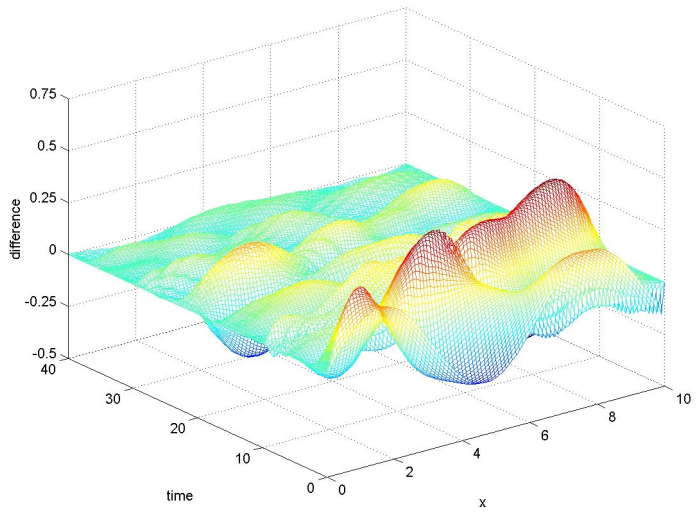

Figure 4.26: Difference Between Closed Loop and Reduced Order Compensator via Balanced Truncation

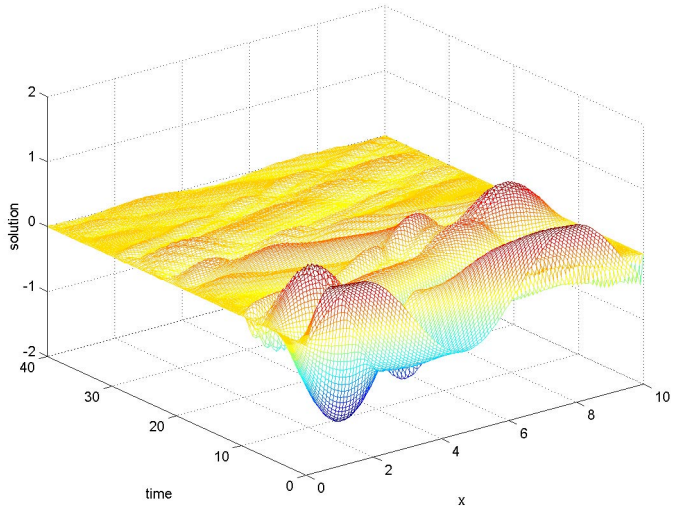

Figure 4.27: Reduced Order Compensator Solution via LQG Balancing

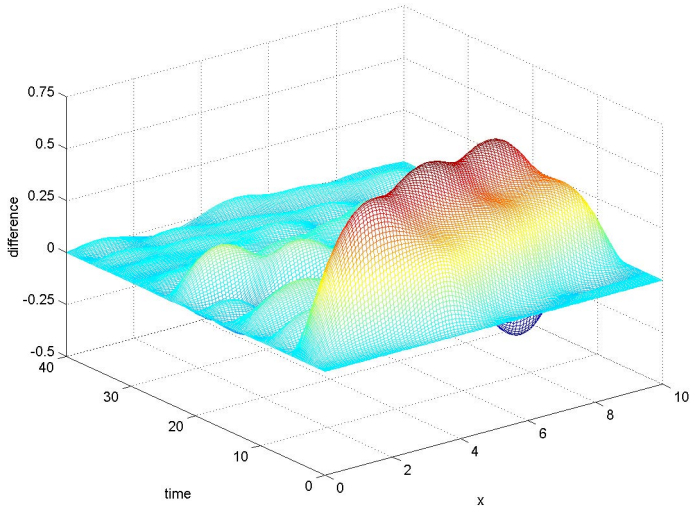

Figure 4.28: Difference Between Full Order Compensator and Reduced Order Compensator via LQG Balancing

other research contained in [2], [1], and [10], has noted similar situations where the lower order state estimate model has performed better.

Stability Radii

Though it is certainly important to compare the behaviors of the full order and reduced order compensators, this visual comparison provides no information about the robustness of the control. That is, for example, if the lower order compensator system is barely stable, i.e. not very resistant to a perturbation or disturbance, then the controller for that system is not very robust. For this reason, the computation of what is known as stability radius is important to analyze the stability of the different compensator systems. To compute the stability radius of the full order compensator, $\mathcal{A}$, for example, find $min\|\Delta\|_2$ such that $eig(\mathcal{A}+\Delta)$ intersects the imaginary axis. For this research, the stability radii are computed using the method discussed in [6].

After computing the stability radii for the full order compensator and the reduced order compensators, the results are summarized in Table 4.2.

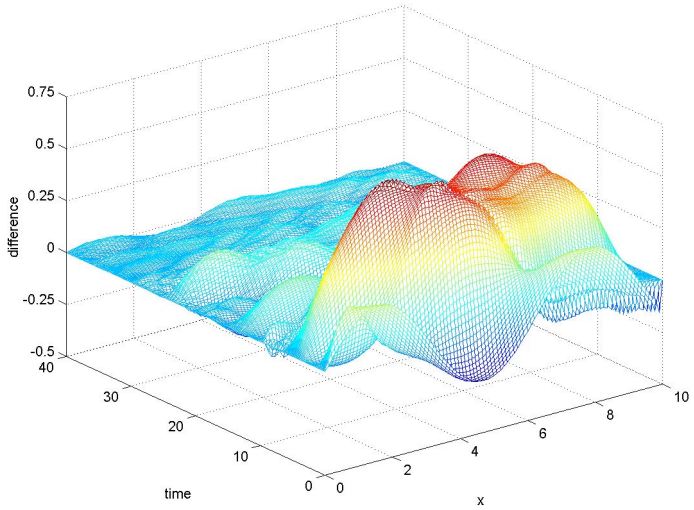

Figure 4.29: Difference Between Closed Loop and Reduced Order Compensator via LQG Balancing

	Stability Radii	
Matrix	x_c with 10 states	x_c with 15 states
$\mathcal{A}$	1.198360×10^{-9}	1.198360×10^{-9}
$\mathcal{A}_T$	4.529261×10^{-7}	5.556982×10^{-7}
$ric\mathcal{A}_T$	7.008909×10^{-5}	1.443033×10^{-6}

Table 4.2: Stability Radii for Simulations with Initial Condition Set 1.

From Table 4.2, it can be noted that the reduced order compensators obtained through LQG balancing are the most robust systems of the five under consideration. This is the conclusion since the table shows that the $ric\mathcal{A}_T$ systems can be perturbed by a matrix of larger norm than the other systems before becoming unstable. (Recall that the full order compensator is not affected by the number of states in the state estimate, so it stands to reason that the stability radius is the same for both state estimate trials.)

4.2 Simulations for Initial Condition Set 2

In the following simulations of the KGL equation, the initial conditions are

$$\omega(0,x) = 4\arctan\left(\operatorname{sech}\left(\tfrac{x-20}{\sqrt{2}}\right)\right), \tag{4.9a}$$

$$\dot{\omega}(0,x) = 0. \tag{4.9b}$$

These conditions are chosen with a purpose. As seen in Section 4.1, the KGL equation with sine and cosine initial conditions generates a great deal of wave interaction. These conditions are chosen so that a "hump" will be generated in the middle of the spatial domain, and the behavior of this hump can be observed over time.

Note that all plots of controllers and functional gains corresponding to simulations in this section are found in Appendix B. Just as with the previous set of initial conditions, the mass of the particle is specified to be 0.01 while the damping constant is set at 0.1. For the finite element approximation, the number of subintervals is set at 100, corresponding to a spatial domain of length 40. The function $b(x)$ is applied over the domain in sixteen pieces, so there are sixteen controllers. Trying different powers of 2, it was found that BB^* is only positive semi-definite until the number of controllers is set to sixteen. From [7], it is known that if P is not positive definite, then BB^* may not be positive definite either. It turns out that P contains negative eigenvalues when the number of controllers is four or eight. As a result, BB^* is only positive semi-definite. Thus, the Cholesky factorization involved in the balancing procedure cannot be applied to P.

4.2.1 Open Loop

The open loop system to be solved is given as

$$\dot{x}^{100}(t) = A^{100}x^{100}(t), \tag{4.10a}$$

$$x^{100}(0) = x_0^{100}. \tag{4.10b}$$

The open loop system eigenvalues are shown in Figure 4.30. The graph of the simulation is depicted in Figure 4.31.

The graph shows the generation of the hump in the middle of the domain described earlier. The hump can be seen traveling over time, but the damping term is certainly causing the system to die out. To solve the open loop system requires 1.04×10^8 flops.

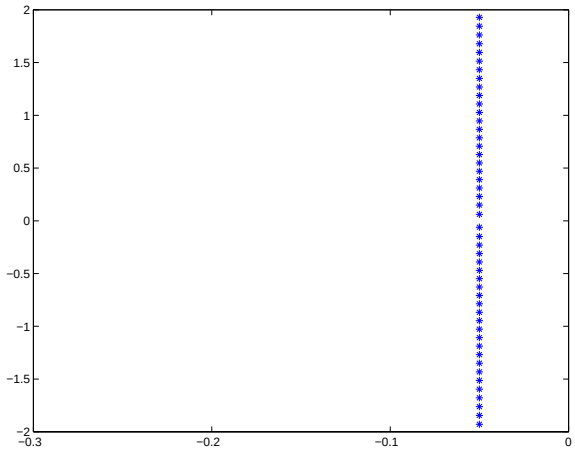

Figure 4.30: Open Loop Eigenvalues

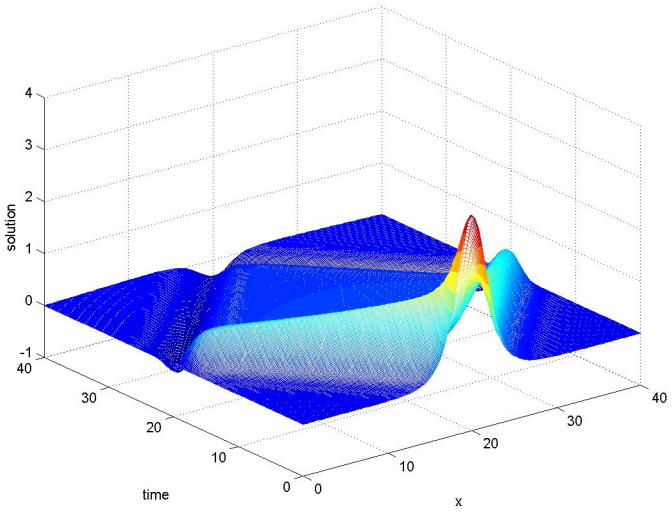

Figure 4.31: Open Loop Solution

4.2.2 Full State Feedback Control (LQR)

The finite element approximate solution to the LQR system is found by solving

$$\dot{x}^{100}(t) = (A^{100} - B^{100}K^{100})x^{100}(t) \tag{4.11a}$$

$$x^{100}(0) = x_0^{100}. \tag{4.11b}$$

The closed loop eigenvalues are displayed in Figure 4.32. The graph indicates that some of the eigenvalues are being shifted further into the left half plane. This shift can be interpreted as the closed loop system offers more stability than the open loop system, which makes sense given that there is no control associated with the open loop system. Figure 4.33 displays the finite element approximation to the closed loop solution.

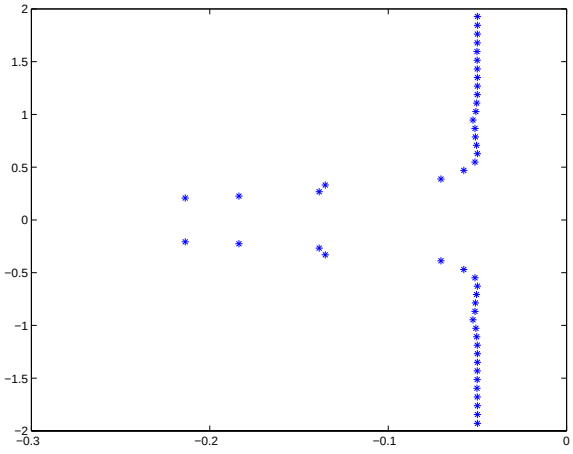

Figure 4.32: Closed Loop Eigenvalues

Solving the closed loop system requires 1.16×10^8 flops. The graph of the closed loop solution shows how the control is trying to drive the solution to zero. This is especially true for early times when the control is acting on the hump generated by the initial conditions. As a result, there is more oscillation in the closed loop solution at later times.

4.2.3 Full Order State Estimate Feedback Control (LQG)

Continuing with $N = 100$, the full order state estimate system is given by

$$\begin{pmatrix} \dot{x}^{100}(t) \\ \dot{x}_c^{100}(t) \end{pmatrix} = \begin{pmatrix} A^{100} & -B^{100}K^{100} \\ F^{100}C^{100} & A_c^{100} \end{pmatrix} \begin{pmatrix} x^{100}(t) \\ x_c^{100}(t) \end{pmatrix}, \tag{4.12a}$$

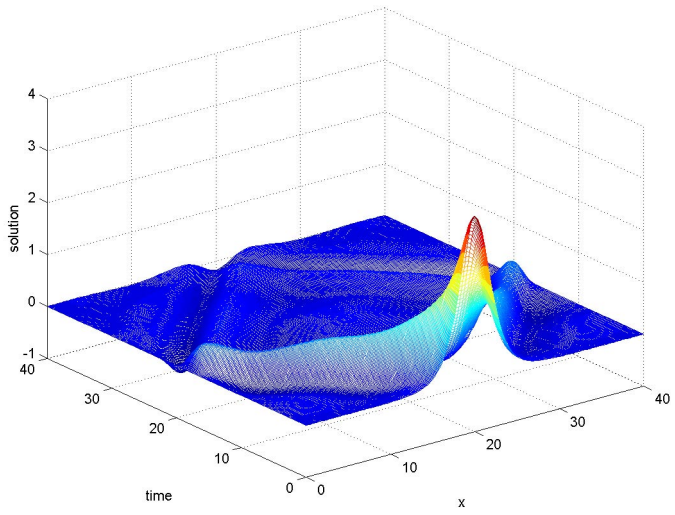

Figure 4.33: Closed Loop Solution

$$\begin{pmatrix} x^{100}(0) \\ x_c^{100}(0) \end{pmatrix} = \begin{pmatrix} x_0^{100} \\ x_{c0}^{100} \end{pmatrix}. \tag{4.12b}$$

Chosen to add some difference between the closed loop and full order compensator solutions, $x_c^{100}(0)$ is taken to be $0.75 * x^{100}(0)$. Figure 4.34 contains the eigenvalues associated with system (4.12).

The solution to the full order compensator is found in Figure 4.35.

The flops required to solve this system is 4.18×10^8. Note that when the amplitude of the hump begins to decrease, it breaks into two waves which continue to decrease in amplitude over time. At time $t = 20$, the waves encounter the boundary and are reflected back across the domain, this time below the $z = 0$ surface. Also note that while the amplitude of the waves is decreasing over time $0 \leq t \leq 20$, the waves dip below the $z = 0$ surface briefly. This will be important later. Now that the LQR and LQG systems have been solved, compare them by viewing the graph of their difference found in Figure 4.36. Note the large difference occurring near $t = 0$. The full order compensator is having significant difficulty capturing the height of the hump generated by the initial conditions.

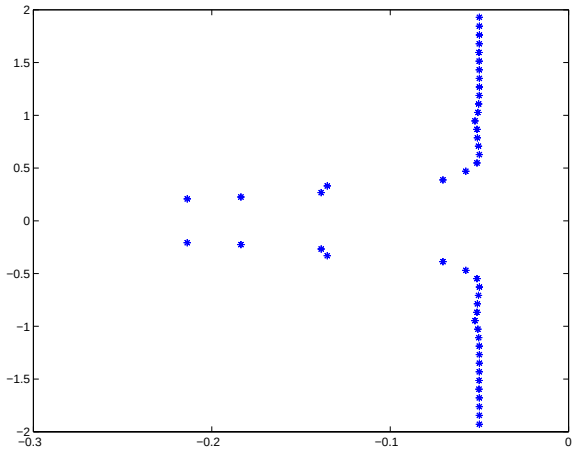

Figure 4.34: Full Order Compensator Eigenvalues

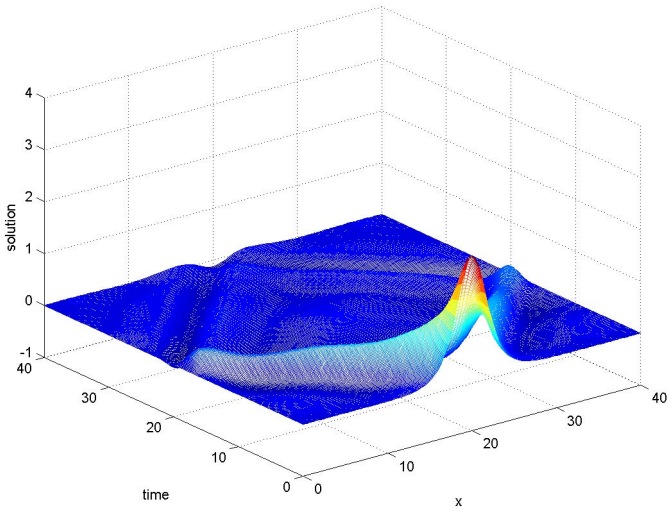

Figure 4.35: Full Order Compensator Solution

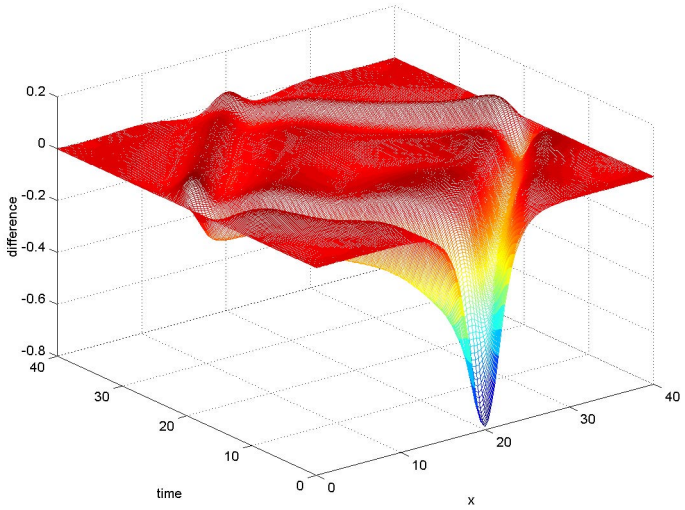

Figure 4.36: Difference Between Closed Loop and Full Order Compensator Solutions

4.2.4 Reduced Order State Estimate Feedback Control via Balancing

As discussed earlier in Subsection 4.1.4, the system (2.29) is balanced with respect to both the Grammians and the Riccati operators, and for each balancing, there will be simulations involving two different size state estimates. Simulations will contain ten and fifteen states in the state estimate, respectively. For these particular simulations, it was found that using less than ten states in the state estimate yields unstable reduced order compensator systems. For the sake of consistency, all simulations reported here reference either ten or fifteen states in the state estimate. It is for these reasons that no simulations in this paper contain fewer than ten states in the state estimate.

Again, suppose the system

$$\dot{x}^{100}(t) = A^{100}x^{100}(t) + B^{100}u^{100}(t), \tag{4.13a}$$

$$x^{100}(0) = x_0^{100}, \tag{4.13b}$$

$$y^{100}(t) = C^{100}x^{100}(t), \tag{4.13c}$$

has been balanced around the Grammians and the Riccati operators. Figure 4.37 shows the Hankel singular values on the left and the Riccati values on the right.

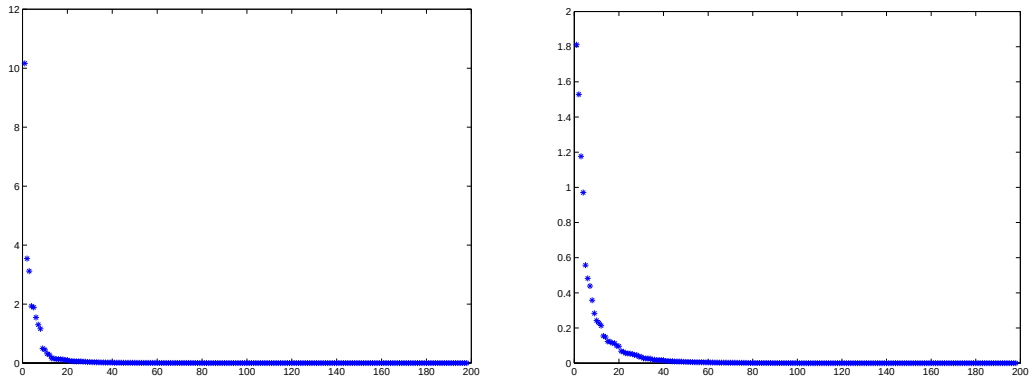

Figure 4.37: Hankel Singular Values (left) and Riccati Values (right)

Table 4.3 describes how many Hankel singular values and Riccati values should be kept in the model to achieve particular levels of significance.

	Values needed for	
	95% significance	99% significance
Hankel singular values	17	37
Riccati values	28	53

Table 4.3: Values needed to achieve 95% and 99% significance.

As before, this information indicates that more Riccati values than Hankel singular values are needed to reach the same level of significance. However, this information does not give a measure of performance for the controls of the corresponding reduced order compensator systems.

Simulations with Ten States in State Estimate

Suppose once again that $balA$, $balB$, and $balC$ have been truncated. The reduced order compensator system with ten states in the state estimate is given by

$$\begin{pmatrix} \dot{x}^{100}(t) \\ \dot{x}_c^{10}(t) \end{pmatrix} = \begin{pmatrix} A^{100} & -B^{100}K^{10} \\ F^{10}C^{100} & A_c^{10} \end{pmatrix} \begin{pmatrix} x^{100}(t) \\ x_c^{10}(t) \end{pmatrix}, \tag{4.14a}$$

$$\begin{pmatrix} x^{100}(0) \\ x_c^{10}(0) \end{pmatrix} = \begin{pmatrix} x_0^{100} \\ x_{c0}^{10} \end{pmatrix}. \tag{4.14b}$$

First considering balanced truncation, Figure 4.38 shows the reduced order compensator solution.

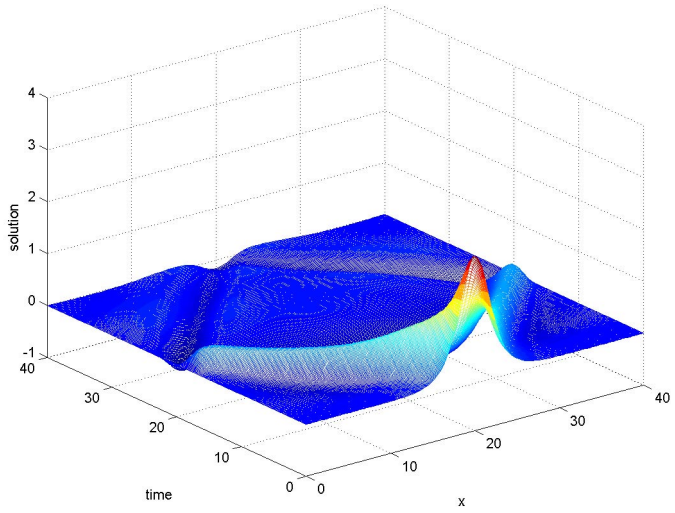

Figure 4.38: Reduced Order Compensator Solution via Balanced Truncation

Comparing the full order compensator with the reduced order compensator, Figures 4.35 and 4.38, there does not appear to be much difference. The amplitude of the wave propagated by the hump generation is slightly larger for the reduced compensator than the full order compensator at some instants of time. Also, for $10 \leq t \leq 20$, the amplitude of the wave below the $z = 0$ plane appears to be enlarged in the reduced compensator. The similarities of the graphs is verified by Figure 4.39, which displays the difference between the full order and reduced order compensators.

The graph shows that the absolute value of the difference is less than 0.2, and the lower order model is least accurate at preserving the full compensator behavior for $10 \leq t \leq 20$. The reduced order compensator seems to be doing a fairly accurate job of capturing the behavior of the full order compensator. However, recall that the full compensator did poorly at preserving the height of the initial hump. Compare the closed loop and reduced compensator designs by considering the difference graph in Figure 4.40. The graph indicates that this reduced compensator also does poorly at capturing the height of the hump.

As mentioned earlier, it is interesting to compare the eigenvalues of the full order compensator and the reduced order compensator. See Figure 4.41 to view the eigenvalues of

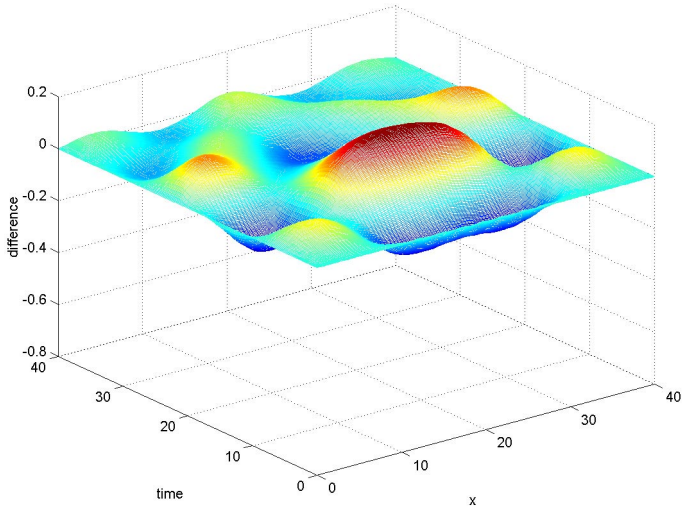

Figure 4.39: Difference Between Full Order Compensator and Reduced Order Compensator via Balanced Truncation

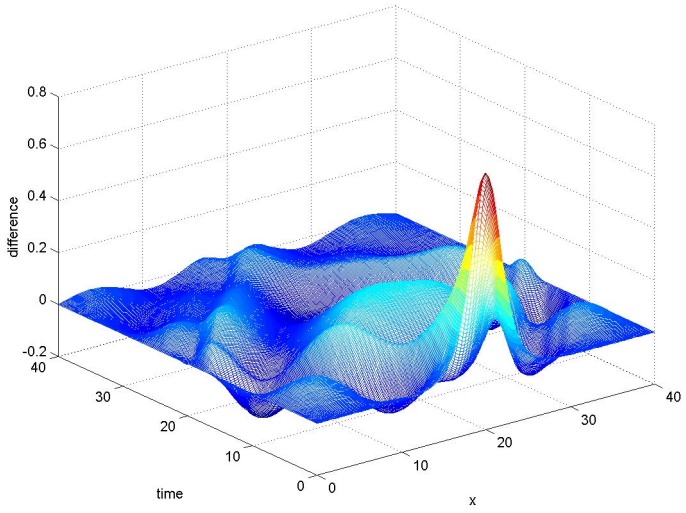

Figure 4.40: Difference Between Full Order Compensator and Reduced Order Compensator via Balanced Truncation

the full order compensator plotted on the same axes with the reduced order compensator obtained through balanced truncation.

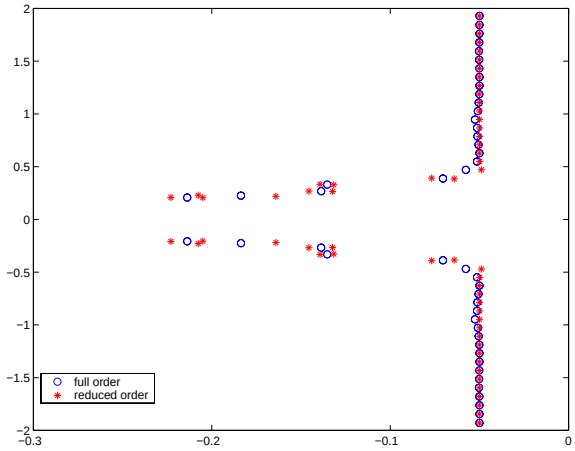

Figure 4.41: Eigenvalues of Full Order Compensator and Reduced Order Compensator via Balanced Truncation

A similar pattern is seen to that obtained during the simulations involving the previous set of initial conditions. Several of the eigenvalues from the reduced order compensator shift further into the left half plane, thereby providing information that the reduced compensator found by balanced truncation offers enhanced stability over the full order compensator.

Now compare the balanced truncation solution with the LQG balancing solution. The reduced order compensator solution obtained using LQG balancing is shown in Figure 4.42.

Just as there is little difference between the full compensator in Figure 4.35 and the reduced compensator through balanced truncation in Figure 4.38, there is little difference between the full compensator and the reduced compensator through LQG balancing in Figure 4.42. In fact, Figures 4.38 and 4.42 bear a striking resemblance to each other. Both seem to cause the hump to decay at a slower rate than the full order compensator, and both seem to amplify the wave traveling below the plane $z = 0$. Based on these similarities, it is expected that the graph of the difference between the full order compensator and reduced order compensator obtained via LQG balancing will be similar to Figure 4.39. See Figure 4.43 to verify.

The absolute value of the difference is the same as obtained using balanced truncation,

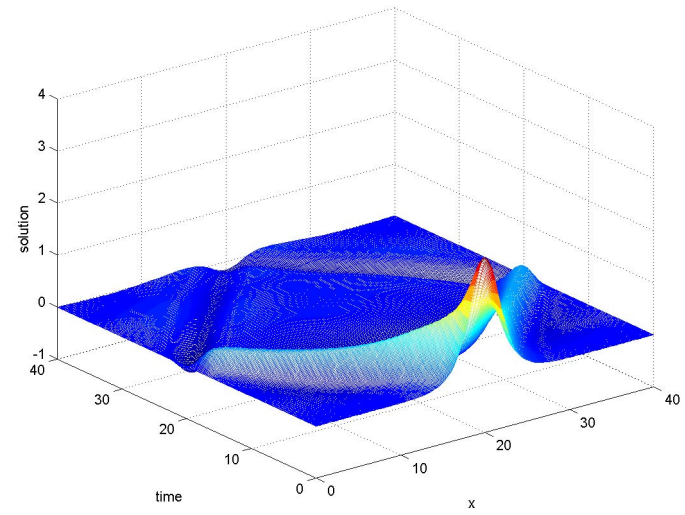

Figure 4.42: Reduced Order Compensator Solution via LQG Balancing

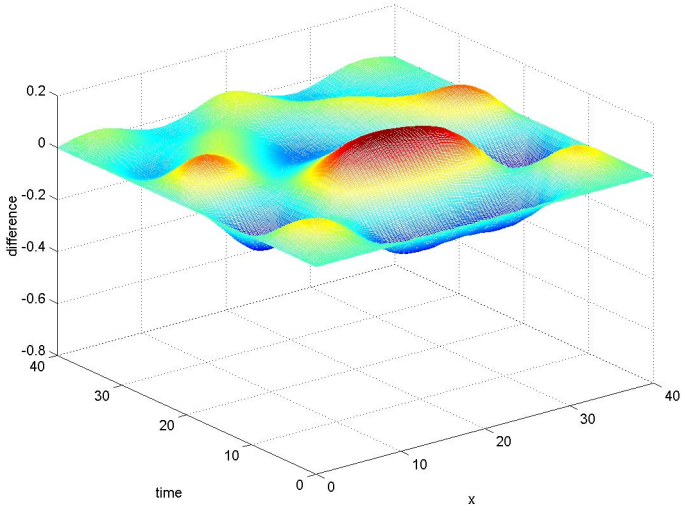

Figure 4.43: Difference Between Full Order Compensator and Reduced Order Compensator via LQG Balancing

less than 0.2. It appears that the LQG balancing also has the most difficulty capturing the activity of the full compensator solution for $10 \leq t \leq 20$. Likewise, it should not seem strange that the LQG balancing also fails to capture the hump height found in the closed loop solution. Figure 4.44 confirms this observation.

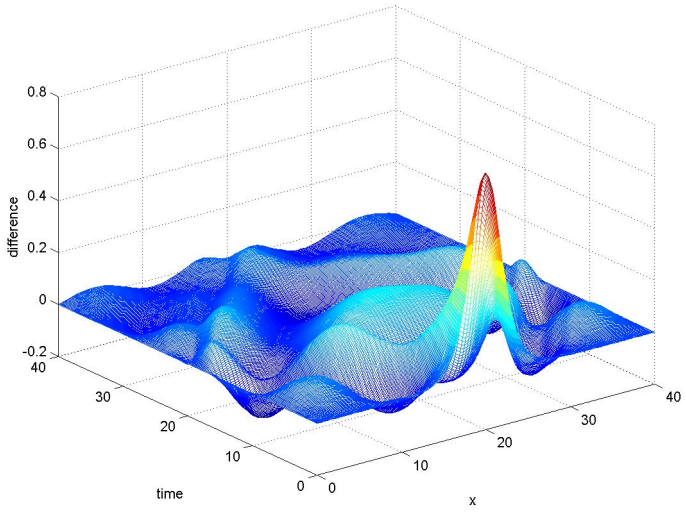

Figure 4.44: Difference Between Closed Loop and Reduced Order Compensator via LQG Balancing

Now, consider the eigenvalues of the full order compensator and the reduced order compensator obtained through LQG balancing. Figure 4.45 shows the eigenvalues for both compensators plotted on the same set of axes.

There is a visible shift of several eigenvalues from the reduced order compensator further into the left half plane. Notice again that this reduced order compensator offers greater stability than the full order compensator.

Simulations with Fifteen States in State Estimate

Again suppose truncation on $balA$, $balB$, and $balC$ has been performed. Now consider the reduced order compensator system with fifteen states in the state estimate.

$$\begin{pmatrix} \dot{x}^{100}(t) \\ \dot{x}_c^{15}(t) \end{pmatrix} = \begin{pmatrix} A^{100} & -B^{100}K^{15} \\ F^{15}C^{100} & A_c^{15} \end{pmatrix} \begin{pmatrix} x^{100}(t) \\ x_c^{15}(t) \end{pmatrix}, \tag{4.15a}$$

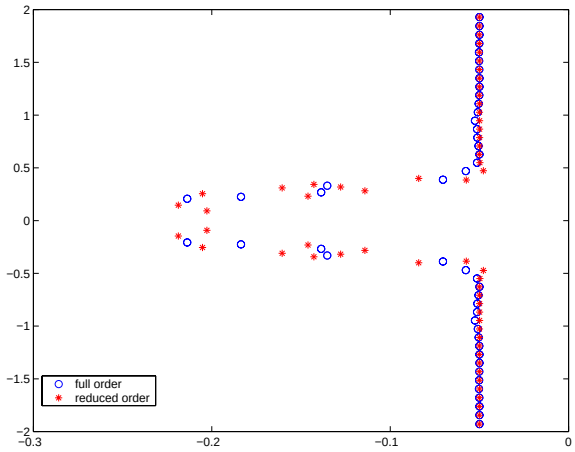

Figure 4.45: Eigenvalues of Full Order Compensator and Reduced Order Compensator via LQG Balancing

$$\begin{pmatrix} x^{100}(0) \\ x_c^{15}(0) \end{pmatrix} = \begin{pmatrix} x_0^{100} \\ x_{c0}^{15} \end{pmatrix}. \tag{4.15b}$$

The reduced order compensator solution obtained through balanced truncation is depicted in Figure 4.46.

Figures 4.38 and 4.46 possess similar shapes, and the amplitudes of the humps, which are both close to a height of 2.5, closely approximate the hump of the full order compensator found in Figure 4.35. However, in the reduced compensator, the wave beneath the $z = 0$ surface is still being amplified. It appears that adding five states to the state estimate does not improve accuracy in this problem area where $10 \leq t \leq 20$. The graph in Figure 4.47 shows the difference between the full and reduced compensators.

The absolute value of the difference is less than 0.2. However, the graph clearly indicates that the reduced compensator still has difficulty tracking the behavior of the full compensator over $10 \leq t \leq 20$. Now consider Figure 4.48 which corresponds to the difference between the closed loop solution and the reduced compensator solution obtained through regular balancing. Note that even though an additional five states are added to the state estimate, the reduced LQG system cannot capture the height of the hump found in the closed loop system.

The reduced order compensator solution obtained using LQG balancing with fifteen states in the state estimate is shown in Figure 4.49.

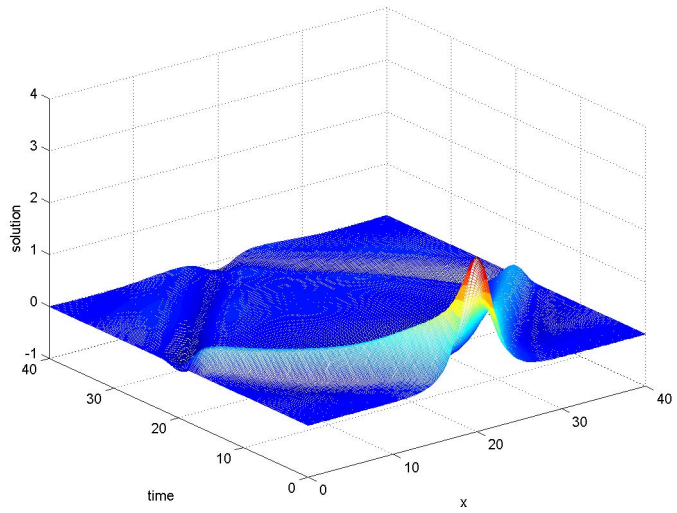

Figure 4.46: Reduced Order Compensator Solution via Balanced Truncation

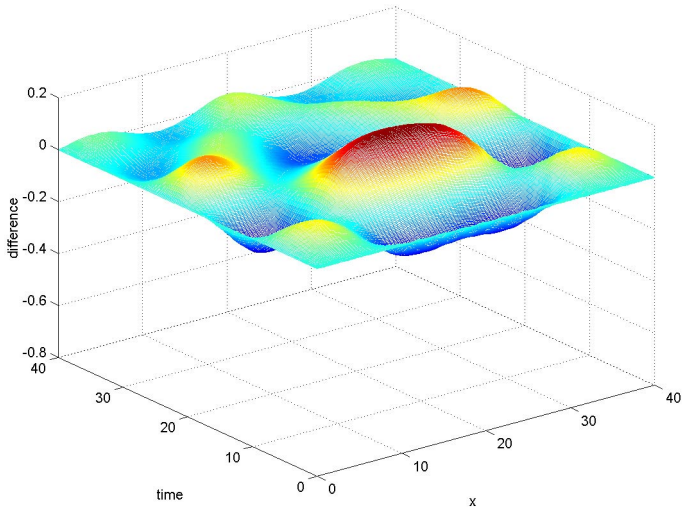

Figure 4.47: Difference Between Full Order Compensator and Reduced Order Compensator via Balanced Truncation

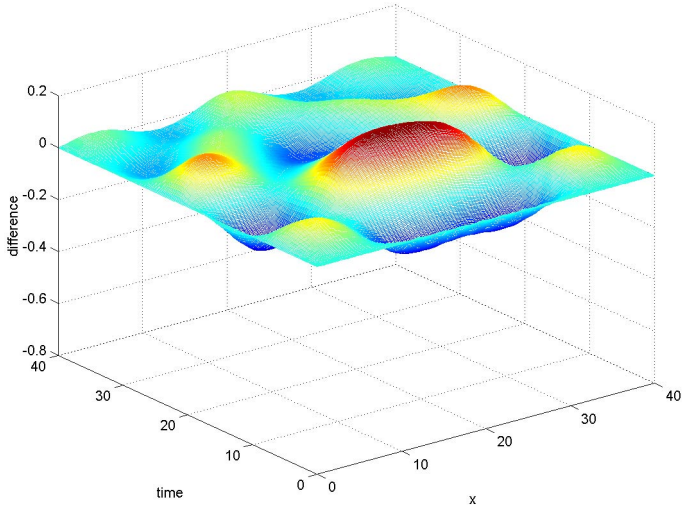

Figure 4.48: Difference Between Closed Loop and Reduced Order Compensator via Balanced Truncation Solutions

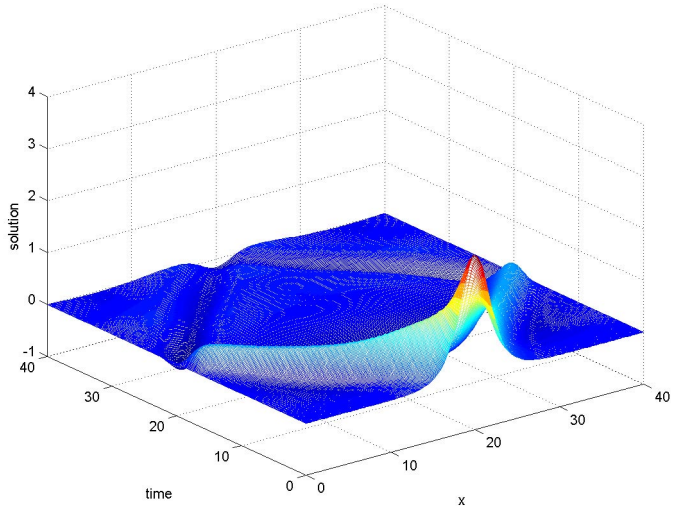

Figure 4.49: Reduced Order Compensator Solution via LQG Balancing

Comparing the reduced order solutions containing ten and fifteen states in the state estimate, it is quite difficult to judge if one is doing a better job than the other since the appearance of the two is very similar. For further investigation, see Figure 4.50 which depicts the difference between the full order compensator and the reduced order compensator obtained through LQG balancing.

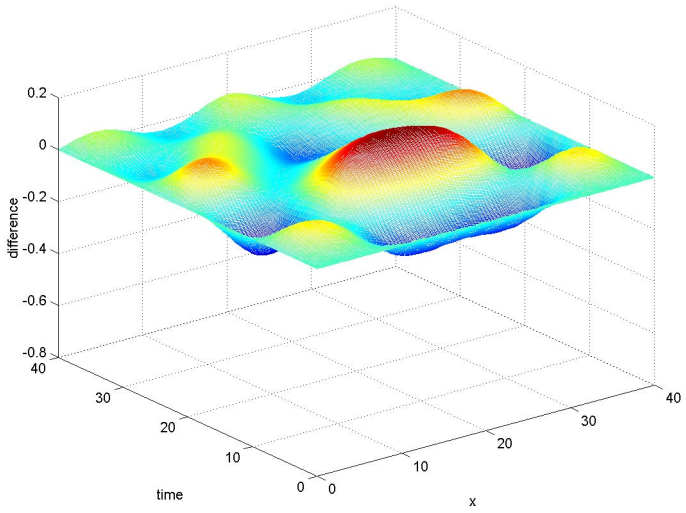

Figure 4.50: Difference Between Full Order Compensator and Reduced Order Compensator via LQG Balancing

The absolute value of the difference is less than 0.2, the same bound obtained for both types of balancing with both ten and fifteen states in the state estimate. Frankly, looking at the difference graphs found in Figures 4.39, 4.43, 4.47, and 4.50, it is difficult to tell them apart because their appearances are so similar. Similarity also arises when viewing the graphs of the differences between the closed loop solution and the reduced compensator solutions. See Figure 4.51 for the graph of the difference between the closed loop and the reduced compensator found through LQG balancing.

Overall, it seems that the additional states in the state estimate have not helped in capturing the activity of the full order compensator. Additionally, there does not seem to be much difference in the simulations using both balancing methods. Likewise, neither of the balancing methods seems effective in preserving the height of the hump from the closed loop solution. It should be noted again that the full order compensator is also deficient in this task.

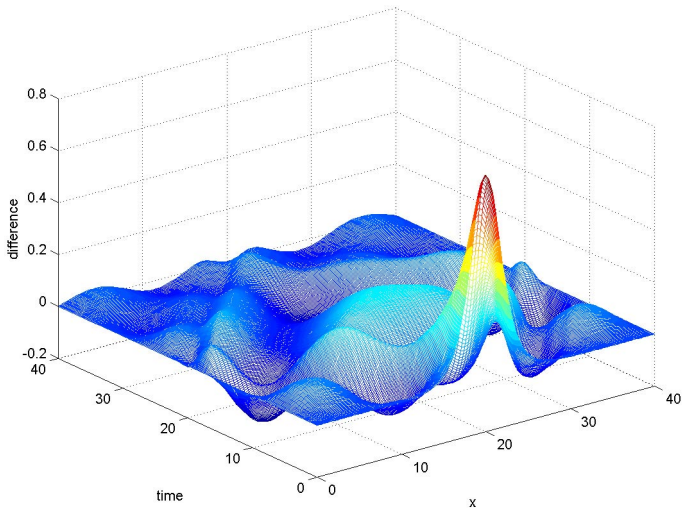

Figure 4.51: Difference Between Closed Loop and Reduced Order Compensator via LQG Balancing Solutions

Stability Radii

Just as with the previous set of simulations, looking at the stability radii of different systems is a meaningful way to compare the sensitivities of the systems to disturbance. The computed stability radii for the full order compensator and both reduced order compensators, are summarized in Table 4.4.

	Stability Radii	
Matrix	x_c with 10 states	x_c with 15 states
$\mathcal{A}$	9.445054×10^{-10}	9.445066×10^{-10}
$\mathcal{A}_T$	1.473063×10^{-7}	6.682086×10^{-6}
$ric\mathcal{A}_T$	6.119489×10^{-7}	2.789615×10^{-7}

Table 4.4: Stability Radii for simulations with Initial Condition Set 2.

Since the full order compensator is not affected by the number of states in the state estimate, it can be assumed that the difference in the first row of Table 4.4 is attributable to numerical error. From the table, it can be noted that the stability radii of both types of reduced order compensators differ very little. In fact, with the exception of the

stability radius for $\mathcal{A}_T$ with 15 states in the state estimate, the stability radii for the reduced compensators are all on the order of 10^{-7}.

Chapter 5

Conclusions and Future Work

The method of balanced reduction has been in use for the past two decades, so this is certainly not a new concept. Due to [4], the theory behind performing LQG balancing in order to reduce a model in finite-dimensional control design is now in place. However, the method has not been tested experimentally, and this paper helps to fill that hole in the literature.

Based on the examples in this paper, it can be observed that a reduced order compensator system obtained through LQG balancing is certainly more stable than the full order compensator and more stable or just as stable as the reduced order compensator found using balanced truncation.

In terms of performance, the results presented here are inconclusive. For the first set of initial conditions, the reduced compensator found using LQG balancing performed slightly worse than the reduced compensator found with balanced truncation. In fact, with the addition of more states in the state estimate, the LQG reduced compensator performed worse again. For the second set of initial conditions, both balancing methods achieved comparable results and seemed to capture the full order compensator well. Based on these outcomes, it seems difficult to speculate on the performance of LQG balancing as an alternative to balanced truncation.

To help answer the question of which type of balancing is preferable, more research needs to be done. Some possible changes which could be made to further explore this area include the following.

- Try other $b(x)$ functions.
- Try different numbers of controllers.
- Try different numbers of measurements.
- Apply both balancing methods to different PDE's, perhaps linear and nonlinear.
- Design a different type of control, e.g. Min-Max or the robust design in [5].

In closing, preliminary results indicate that balancing around the Riccati operators could prove effective in designing a feedback controller. However, there are several untapped directions associated with this topic in which research could be pursued.

Bibliography

[1] J. A. Atwell and B. B. King. Reduced order controllers for spatially distributed systems via proper orthogonal decomposition. *SIAM J. Sci. Comput., accepted*.

[2] J. A. Atwell and B. B. King. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Math. and Comput. Model.*, pages 1–19, Jan 2001.

[3] Jeanne A. Atwell. *Proper Orthogonal Decomposition for Reduced Order Control of Partial Differential Equations*. PhD thesis, Virginia Polytechnic Institute and State University, April 2000.

[4] R. F. Curtain. A new approach to model reduction in finite-dimensional control design for distributed parameter systems. *preprint*, 2001.

[5] R. F. Curtain and H. J. Zwart. *An introduction to infinite-dimensional linear systems theory*. Springer-Verlag, New York, 1995.

[6] J. W. Demmel. On condition numbers and the distance to the nearest ill-posed problem. *Numer. Math.*, 51:251–289, 1987.

[7] P. Dorato, C. Abdallah, and V. Cerone. *Linear-Quadratic Control, An Introduction*. Prentice Hall, Englewood Cliffs, 1995.

[8] Walter Greiner. *Relativistic Quantum Mechanics*. Springer-Verlag, Berlin, 1997.

[9] E. A. Jonckheere and L. M. Silverman. A new set of invariants for linear systems—application to reduced order compensator design. *IEEE Trans. AC*, 28:953–964, 1983.

[10] B. B. King. Nonuniform grids for reduced basis design of low order feedback controllers for nonlinear continuous systems. *Mathematical Models and Methods in Applied Sciences*, 8(7):1223–1241, 1998.

[11] B. B. King and E. W. Sachs. Semidefinite programming techniques for reduced order systems with guaranteed stability margins. *Comput. Opt. and Appl.*, pages 37–59, 2000.

[12] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. John Wiley & Sons, New York, 1972.

[13] E. B. Lee and L. Markus. *Foundations of Optimal Control Theory*. Robert F. Krieger Publishing Company, Malabar, FL, reprint edition, 1986. This book was first published in 1967 by John Wiley and Sons, Inc. The edition referred to here was reprinted with corrections.

[14] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, Berlin, 1971.

[15] Bruce C. Moore. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.

[16] Lars Pernebo and Leonard M. Silverman. Model reduction via balanced state space representations. *IEEE Transactions on Automatic Control*, 27(2):382–387, 1982.

[17] J. L. Walker. *Dynamical Systems and Evolution Equations*. Plenum Press, New York, 1980.

[18] J. Wloka. *Partial Differential Equations*. Cambridge University Press, Cambridge, 1987.

Appendix A

				Hankel		**Riccati**	
			$b(x)$	95%	99%	95%	99%
$Q = C^*C$	1 c	1 b	e^x	10	19	2	19
		4 b's	e^x	11	20	18	47
	4 c's	4 b's	e^x	24	51	104	133
	1 c	1 b	x	3	5	19	19
		4 b's	x	8	12	35	35
	4 c's	4 b's	x	14	37	45	101
	1 c	1 b	x^2	4	6	29	32
		4 b's	x^2	11	14	53	58
	4 c's	4 b's	x^2	14	38	90	130
	1 c	1 b	x^3	6	6	2	33
		4 b's	x^3	13	14	38	57
	4 c's	4 b's	x^3	16	40	99	131
	4 c's	4 b's	$e^{-(x-x_i)^2}$	10	14	12	19
		16 b's	$e^{-(x-x_i)^2}$	17	37	28	53
$Q = \begin{pmatrix} K & 0 \\ 0 & M \end{pmatrix}$	4 c's	4 b's	$e^{-(x-x_i)^2}$	24	51	106	128

Table A.1: Number of Eigenvalues Required to Attain a Specific Level of Significance

Appendix B

Controller

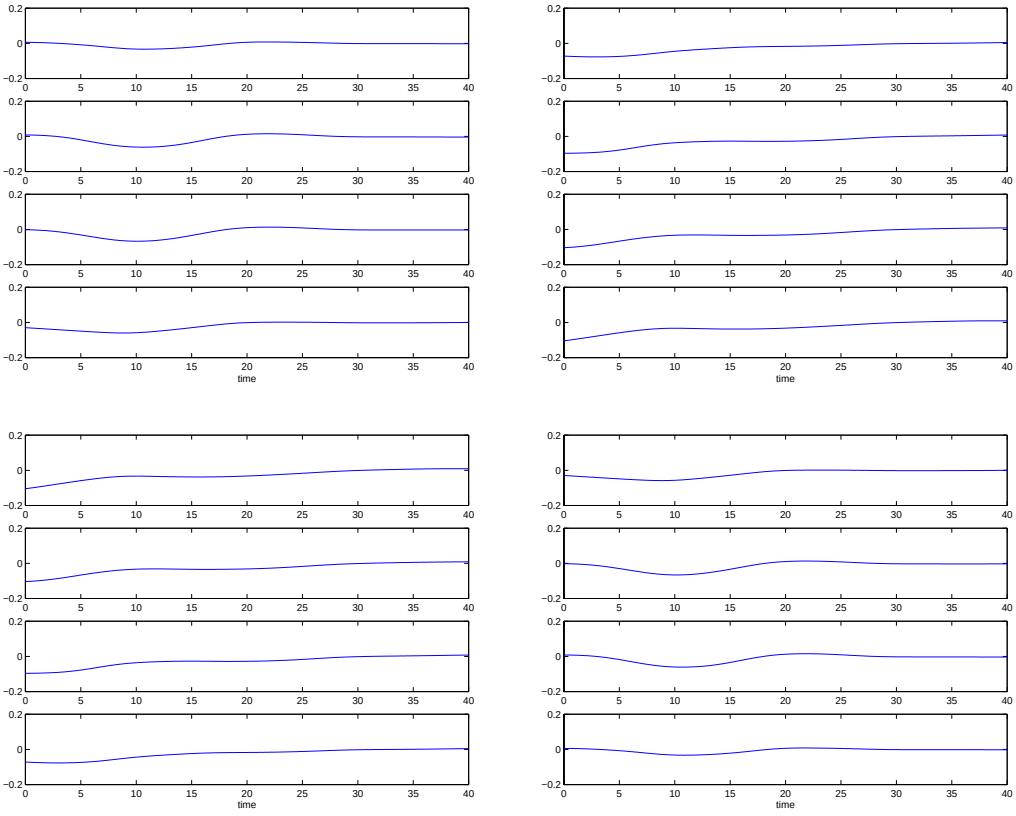

Figure B.1: Controller for Initial Condition Set 2(ordered top to bottom within each plot and ordered left to right and top to bottom overall)

Full Order Functional Gains

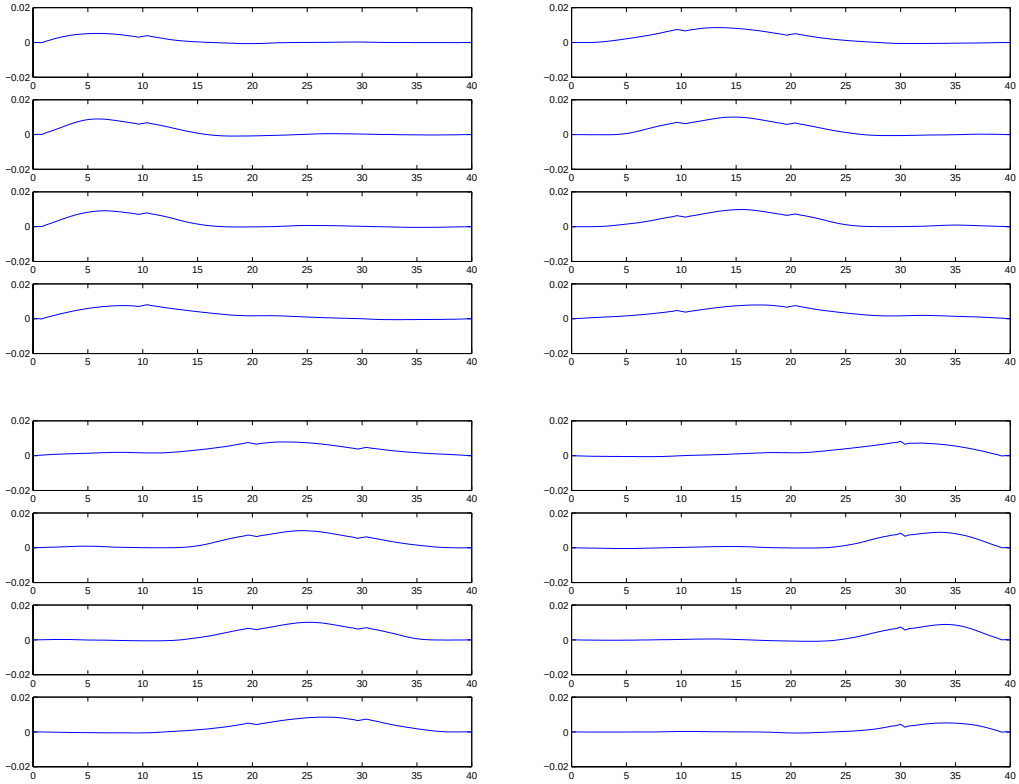

Figure B.2: Full Order Functional Gains for Position (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

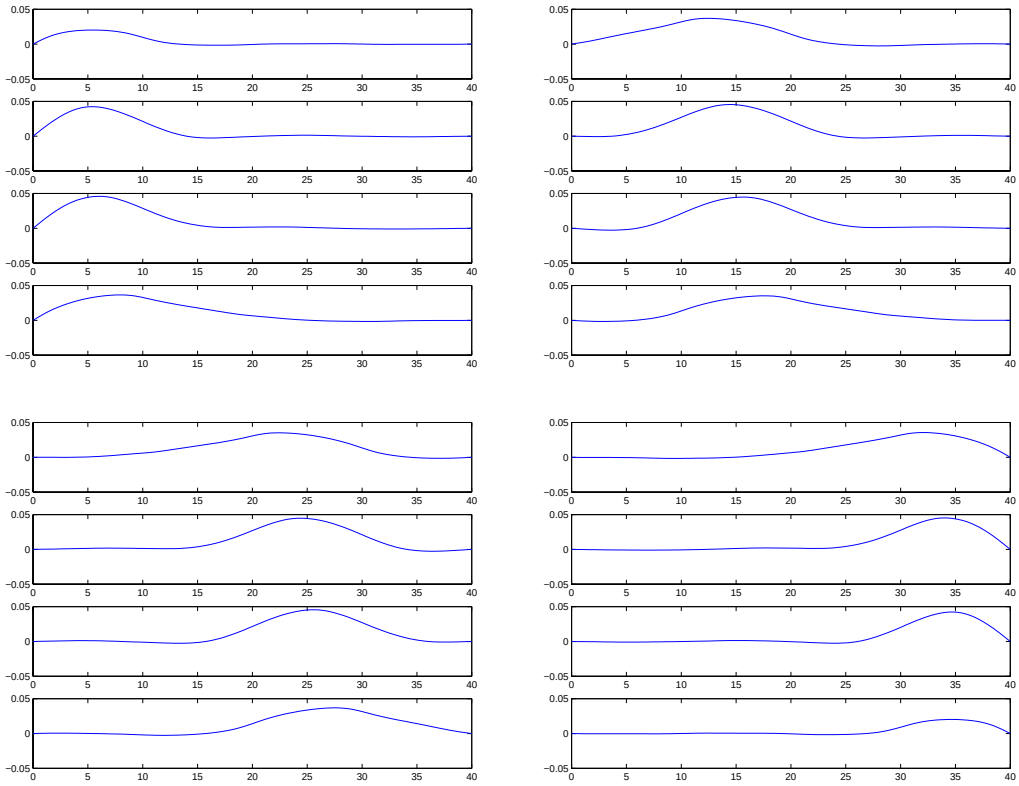

Figure B.3: Full Order Functional Gains for Velocity (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

Functional Gains for Ten States in State Estimate

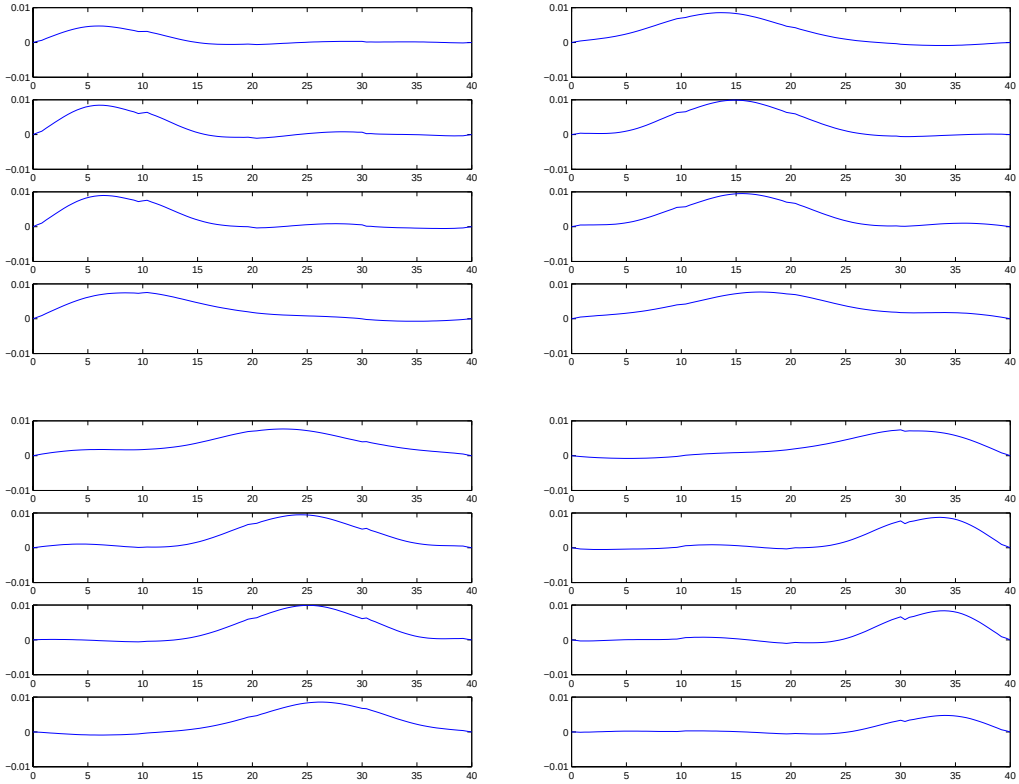

Figure B.4: Reduced Order Functional Gains for Position for Regular Balancing (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

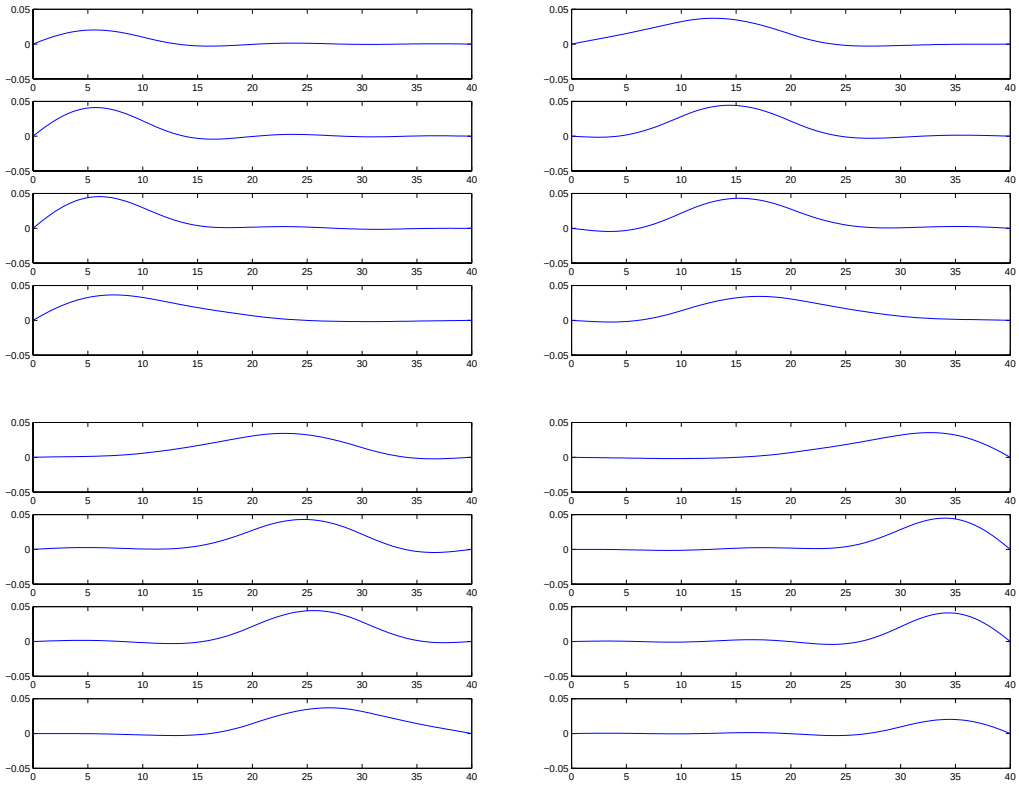

Figure B.5: Reduced Order Functional Gains for Velocity for Regular Balancing (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

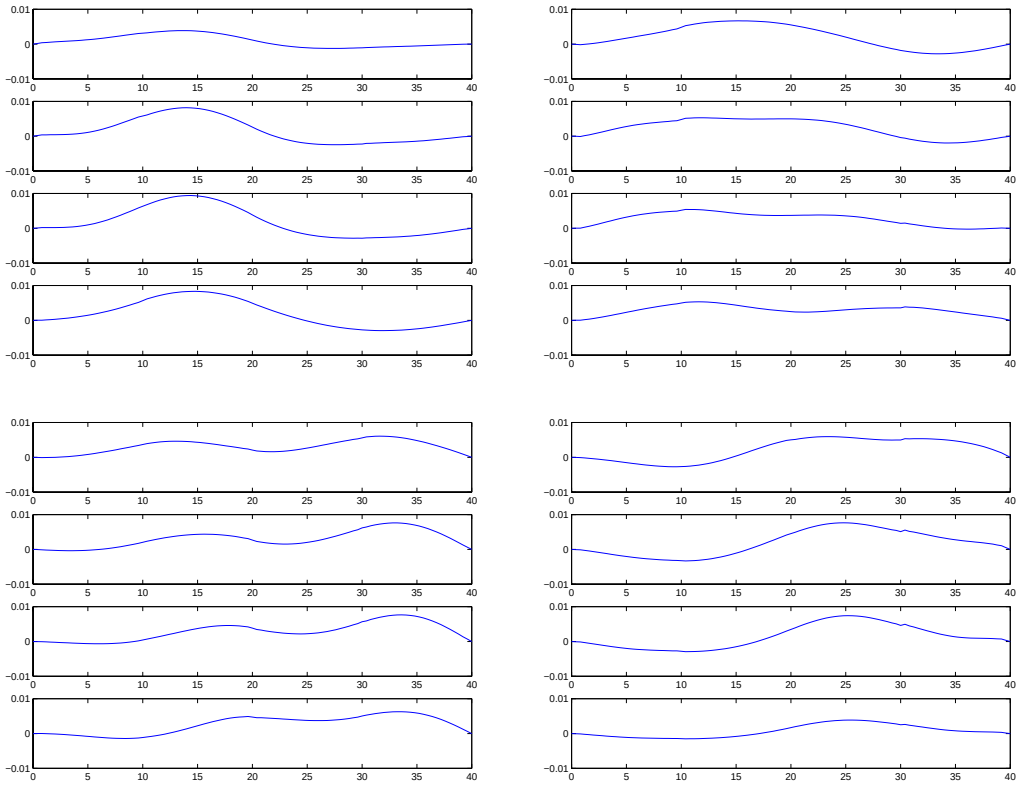

Figure B.6: Reduced Order Functional Gains for Position for LQG Balancing (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

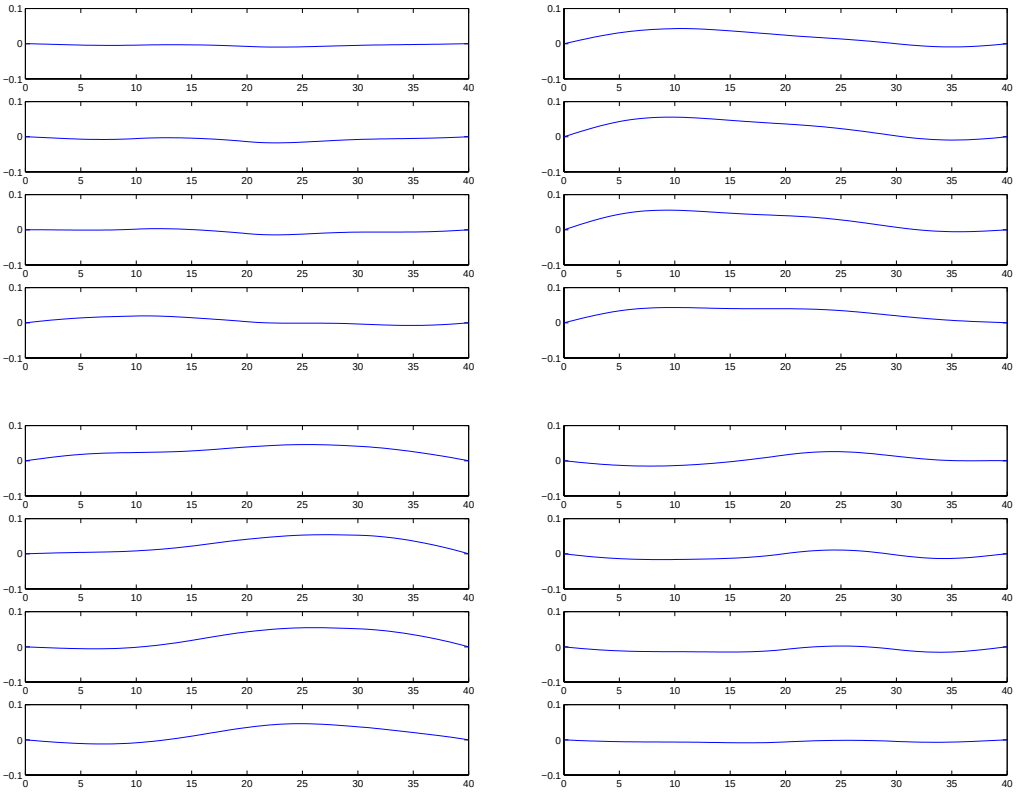

Figure B.7: Reduced Order Functional Gains for Velocity for LQG Balancing (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

Functional Gains for Fifteen States in State Estimate

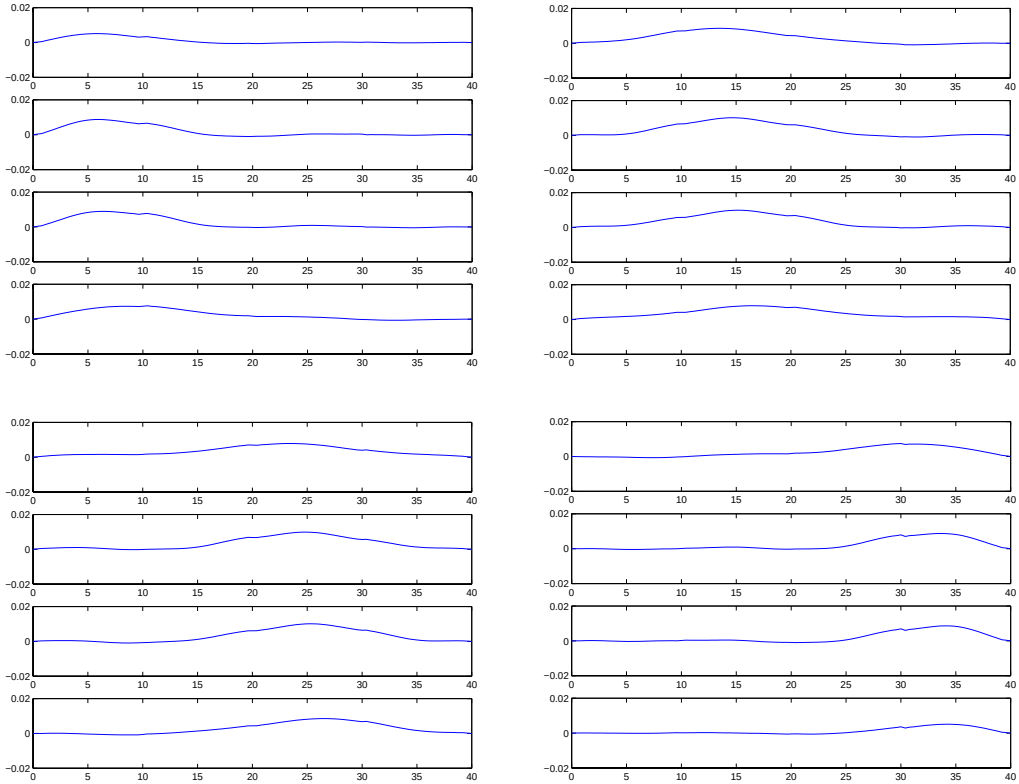

Figure B.8: Reduced Order Functional Gains for Position for Regular Balancing (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

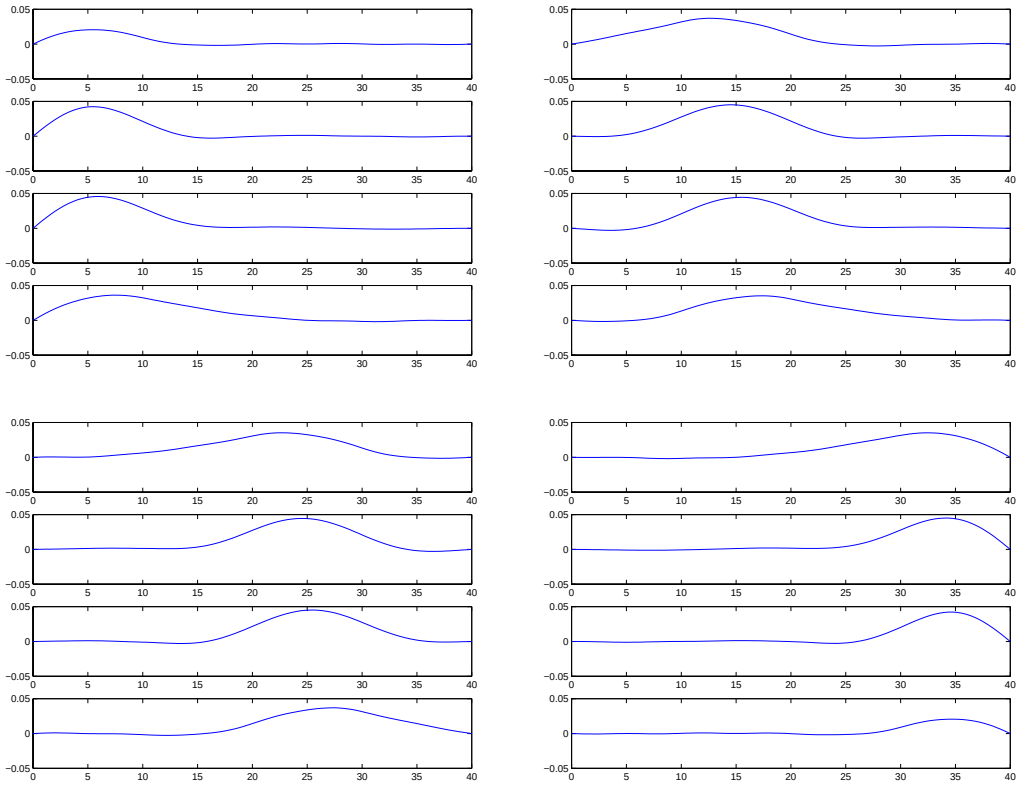

Figure B.9: Reduced Order Functional Gains for Velocity for Regular Balancing (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

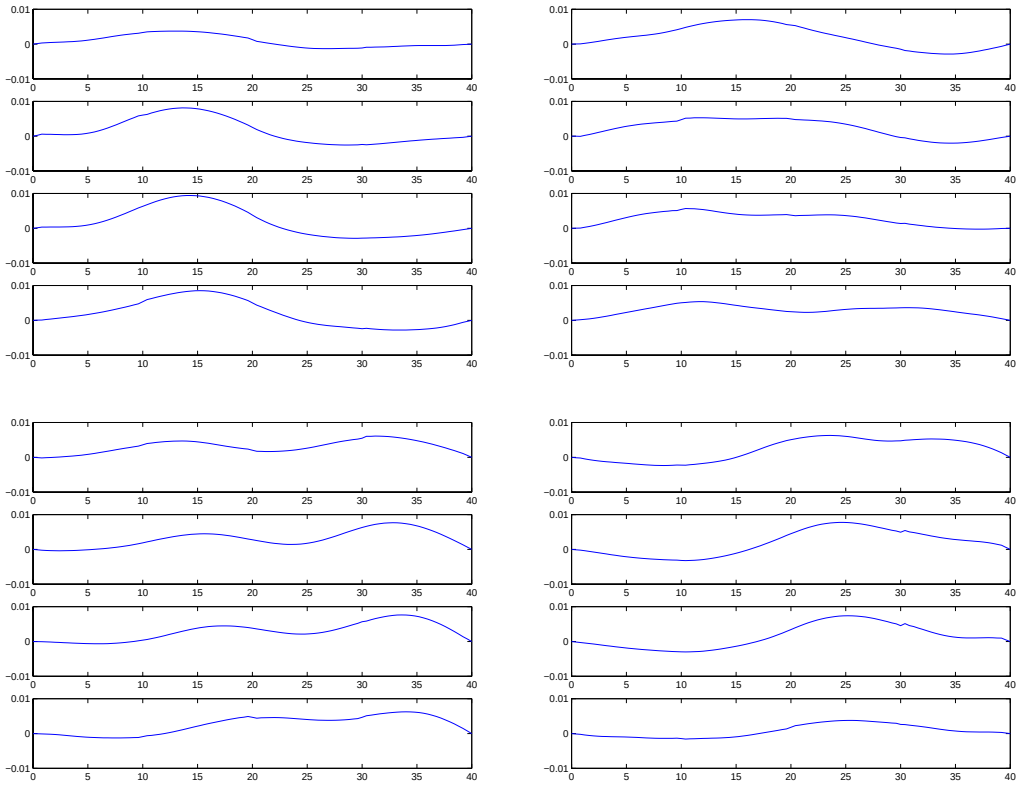

Figure B.10: Reduced Order Functional Gains for Position for LQG Balancing (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

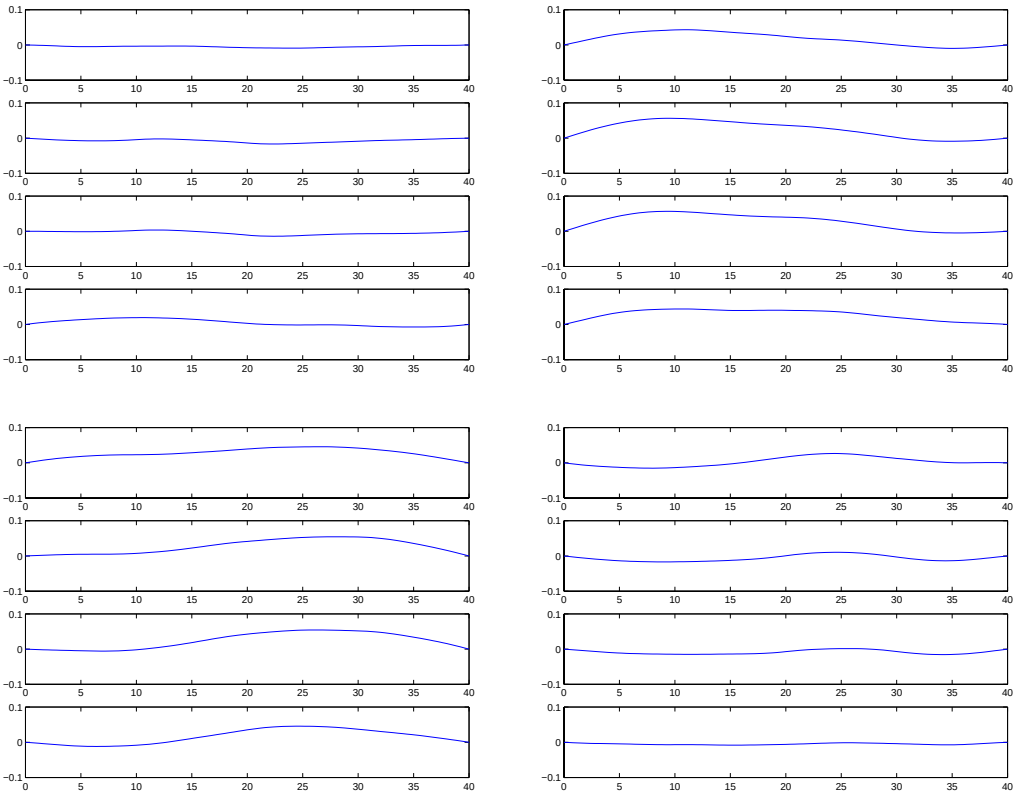

Figure B.11: Reduced Order Functional Gains for Velocity for LQG Balancing (ordered top to bottom within each plot and ordered left to right and top to bottom overall)

Vita

Katie Allison Evans Camp was born in Ashland, Kentucky on May 22, 1977. She graduated co-valedictorian from Boyd County High School in 1995 and received a full academic scholarship to attend Morehead State University in Morehead, Kentucky. Katie graduated summa cum laude with a Bachelor of Science in Mathematics in May 1999. After several research experiences as an undergraduate, she decided her interest was in applied mathematics, and she began her graduate studies at Virginia Polytechnic Institute and State University in Blacksburg, Virginia. Katie will enter the Ph.D. program at Virginia Tech in Fall 2001.