A Polynomial Chaos Approach to Control Design

Brian A. Templeton

Dissertation submitted to the faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

Mehdi Ahmadian, Chair

Steve C. Southward, Co-Chair

Adrian Sandu

Craig A. Woolsey

Donald J. Leo

August 17, 2009

Blacksburg, Virginia

Keywords: Control Design, Robust Control, Optimal Control, Parametric Uncertainty,

Polynomial Chaos

A Probabilistic Approach to Control Design using Polynomial Chaos

Brian A. Templeton

(ABSTRACT)

A method utilizing $\mathcal{H}_2$ control concepts and the numerical method of Polynomial Chaos was developed in order to create a novel robust probabilistically optimal control approach. This method was created for the practical reason that uncertainty in parameters tends to be inherent in system models. As such, the development of new methods utilizing probability density functions (PDFs) was desired.

From a more theoretical viewpoint, the utilization of Polynomial Chaos for studying and designing control systems has not been very thoroughly investigated. The current work looks at expanding the $\mathcal{H}_2$ and related Linear Quadratic Regulator (LQR) control problems for systems with parametric uncertainty. This allows solving deterministic linear equations that represent probabilistic linear differential equations. The application of common LTI (Linear Time Invariant) tools to these expanded systems are theoretically justified and investigated. Examples demonstrating the utilized optimization process for minimizing the $\mathcal{H}_2$ norm and parallels to LQR design are presented.

The dissertation begins with a thorough background section that reviews necessary probability theory. Also, the connection between Polynomial Chaos and dynamic systems is explained. Next, an overview of related control methods, as well as an in-depth review of current Polynomial Chaos literature is given. Following, formal analysis, related to the use of Polynomial Chaos, is provided. This lays the ground for the general method of control design using Polynomial Chaos and $\mathcal{H}_2$. Then an experimental section is included that demonstrates controller synthesis for a constructed probabilistic system. The experimental results lend support to the method.

I would like to gratefully acknowledge the support provided for this work under NASA Grant NNL05AA18A.

Dedication

To my family—both the two- and four-legged members.

Acknowledgments

During my PhD work there have been a great deal of extremely helpful people.

First I'd like to thank my family—they've been there for me for a long time before starting my PhD. Throughout the years they've been a huge help. This includes my parents, both brothers, and grandparents. Additionally, there have been a number of four-legged members (pets) in in my family. They too have given me a lot. Special items, like my current cat Holstein waking me up in the morning and my dog Cinnamon being by me during work and play, have always added great amounts of happiness. As such, I have dedicated my dissertation to my whole family.

Next, I'd like to thank my committee members for their support over the PhD process. In particular, my advisors Dr. Ahmadian and Dr. Southward have committed substantial time and effort and have been influential in my research. Also, Dr. Sandu originally came up with the grant proposal relating to exploring Polynomial Chaos for engineering applications that started the lab's VIPER group.

I'd like to specifically thank Dr. Sean Kenny, Dr. Luis Crespo, and Dr. David Cox at NASA Langley for having me for a summer of research at Langley and for the fruitful discussions that were important in forming many concepts. Dr. Cox had a background in LMI's that was very helpful. It was particularly amazing that he was so helpful even though he wasn't officially a grant supervisor and because I occasionally stopped by after the end of the day.

Further, there are many past and present members of the Center for Vehicle Systems & Safety (CVeSS) that I'd like to thank. All have been great in terms of discussing concepts. This has been particularly true in regards to Dr. Nima Mahmoodi who was also working on controls. Carvel Holton has been available throughout my entire time at Virginia Tech and very helpful in the area of circuits, power electronics, and noise reduction. Clement Nagode has helped with various items including designing fixtures (while still a visiting undergraduate researcher) to mount the linear electric motors on the oscillator. Dr. Mohammad Rastgaar was another graduate student researching the area of controls and is a good friend. Alireza Farjoud is another PhD student—we shared my interests in experimentation and theoretical modeling. Florin Marcu was helpful with lab tips and brainstorming. A fellow researcher, Michael Craft, provided assistance a large number of times. In addition to lending a hand in the lab a number of important times, Brad Hopkins was very helpful because he too tended to work later hours. William Burke's work and kindness in lending the variable-valve damper he designed are due a significant thank-you. Additionally, discussions with Dr. Brendan Chan in my first year as a PhD candidate were appreciated. Emmanuel Blanchard has worked on complimentary aspects of the VIPER project. Last but not least, Samuel Shimp started on the same broad research project and was very helpful because he too was learning about Polynomial Chaos at the same time. The help from all of these friends was greatly appreciated and I'd like to wish them the best of luck in their future endeavors!

In addition to the noted and greatly appreciated NASA grant, I'd like to thank a few companies, organizations, and individuals that supported the research. Mark Wilson and Oscar Guerra at H2W Technologies in Santa Clarita, CA were always very friendly and helpful in configuring the linear electric motors. Additionally, they were very supportive by giving significant discounts. David Conrad at PCB Piezotronics, Depew, NY donated an accelerometer amplifier. The A. B. Kreger Company in Roanoke, VA donated a selection of ferrites to help with noise suppression. Also,

Carolina Fluid Components was gracious enough to provide discounts on ordered products. Finally, the Virginia Tech Graduate Student Assembly (GSA) awarded a $400 grant for my research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this work, results relating to uncertainty in dynamic systems are presented. Specifically, the focus is on parametric uncertainty. Practitioners are very familiar with this problem when they perform system identifications—the question of "how good are the obtained parameter values?" Also, there are always concerns in regards to the "constant" parameters might not be truly constant. While the first question is directly answered, the second issue is also addressed for cases when the *quasi-stationary* assumption can be invoked.

More specifically, the research has an emphasis on the area of feedback control. In this area, it is important to ensure that designed controllers create stable closed loop systems, even in the face of uncertainty. As such, the work relates to Robust Control design.

Additionally, in many cases, it is felt that there is more knowledge than just simple bounds on parameters, as is typical in norm-bounded approaches. Thus, an approach of describing parameters based on probability density functions (PDFs) is developed in this dissertation.

There are multiple ways to interpret the use of PDFs to represent parameters. One may be based on experiments to support a given PDF. Another approach may be that the PDF is a measure of the

importance a designer gives for the system being able to perform well at given parameter values. Similarly, PDFs may simply be selected to cover an estimated bound, with low probabilities at the edges to represent the possibility of being at extremes, and higher probabilities in the center to represent increased confidence in that region. Regardless of the method, the additional information provided by the probability distributions lends itself to a notion of probabilistic optimality. In terms of Optimal Control, one approach relates to finding controllers that minimize expected cost. This concept can be taken further—including looking at measures of variance where the notion of how close solutions are to each other comes into play.

The path taken is to investigate the applicability of the numerical method of Polynomial Chaos. This method, generally agreed to have been introduced to the engineering community by Ghanem and Spanos around 1990, is relatively new. It involves allowing independent random variables to lie in separate dimensions. These additional dimensions are then utilized for creating orthogonal expansions of the random variables. The expansions, similar to discrete Fourier series expansions, are usually represented by coefficients associated with the basis function expansions. Polynomial Chaos Expansions (PCEs) serve to remap the PDFs of the random variables to a more tractable form for dynamic simulations and analysis.

The material presented in this document begins with an introduction to relevant concepts, including statistical and probabilistic background (Chapter 2). Additionally, Polynomial Chaos is presented and compared with several other methods for propagating uncertainty. This is performed through the example of an oscillator with probabilistic initial conditions. Also, since the work uses concepts common to the $\mathcal{H}_2$ and LQR problems, it is felt useful to present their connection.

In Chapter 3, a review of literature relating to trajectory sensitivity methods, probabilistic robust control, and Polynomial Chaos is presented. In the Polynomial Chaos section, there is not a substantial amount of literature relating to control and thus it was deemed that a worthy endeavor

would be to expand into related fields as well as further outside of the immediate focus. Thus, the intent is to allow the reader to get a better feel for the broad applications of Polynomial Chaos.

Chapter 4 presents results that help to add formality to the uncertainty analysis. It employs the theory of Hilbert spaces to help lay a foundation for the work. Further, numerical results are presented in terms of error analysis. The results assist in understanding relevant Polynomial Chaos convergence properties.

In Chapter 5, the development of an expanded state space model is presented in a more pragmatic framework. The presented procedure utilizes the orthogonality inherent in the PCEs. This form of the model takes advantage of the fact that the Galerkin projections of a linear state space model, with respect to random parameters, creates another linear state space system. Thus, there are many interesting possibilities, including analyzing stability of the whole set of systems and analyzing the form with regards to the $\mathcal{H}_2$ norm. It is shown that these interesting properties allow for the design of robust probabilistic optimal controllers.

Chapter 6 introduces an experimental test platform used to help validate designed controllers. A variety of tests utilizing this platform, a two-degree-of-freedom mechanical oscillator, are presented. In order to illustrate the experimental concept of "state uncertainty," a slowly time-varying parameter is introduced through slowly varying the current through a tunable Magneto-Rheologic (MR) damper. A more in-depth study is performed using a distribution of masses. Important results are obtained in regards to four controllers designed with the method and compared to the open loop system and a standard $\mathcal{H}_2$ design. A final study is performed on a system with a variable-valve damper.

Additionally, it should be noted that Appendix A includes Matlab functions for implementing the intrusive method presented in Chapter 5.

Novel contributions in this work include the first known implementable Probabilistic Robust controller derived with Polynomial Chaos. Additionally, through using this numerical approach, a special expanded state space formulation was derived. Although this was derived independently from Fisher and Bhattacharya [3], a related form was presented at a conference in June 2008, and thus can no longer completely be considered novel. But, there are attributes that are. One of these attributes is the specific formulation taken in this work since it lends itself better to a presentable derivation. It is easily possible to derive the approach taken in [3] from the work presented here, through adding an additional step using permutation matrices. However, directly deriving the related formulation in [3] is significantly more involved since it requires looking element-wise at matrices of matrices and vectors of vectors.

Additionally, while a few ways to perform trajectory optimization are discussed, it is believed that a derived measure relating to the square of the mean of errors and error covariances is reasonably novel in probabilistic optimal control. The spectral approach taken deviates from local approximations based on sensitivities and methods based on discrete sampling. This approach is explained as balancing the "closeness" of trajectories with performance of the mean (or center) trajectory. Further, being able to obtain this statistical information about the individual trajectories (as opposed to statistical information about the resulting cost function), through the Lyapunov solution to the $\mathcal{H}_2$ norm is believed to be novel as well. Thus, while the work presented in [3, 4] is related, and the insights gained from the independent research are gratefully acknowledged, it is believed that the presented body of knowledge significantly surpasses this other set of work.

Further, controlled experiments to validate the control designs for uncertain systems are presented. One of these experiments looks at calculating the experimental costs in terms of square mean and variance from the test's data for the new controllers. Also, a more qualitative design and test is presented. The experimental component is significant, since as is well known, many more theoretical developments are proposed than are tested experimentally. Additionally, they help demonstrate

the applicability of the mean and standard deviation trajectories for evaluating systems with large uncertainties.

Also, the code libraries provided are reasonably unique. Frequently the collocation method is used with Polynomial Chaos to avoid creating specialized libraries. However, the Galerkin method is an *intrusive* method and as such requires a more in-depth approach. The provided libraries allow easily studying a broad class of linear differential equations using the Galerkin method.

Chapter 2

Background

2.1 An Introduction to Probability and Distributions

Uncertainty in this work is studied through a probabilistic interpretation. The easiest way to introduce this aspect of the discussion is through a simple common example.

Imagine a "fair" coin toss. A coin is tossed upwards, and when it comes back down, typically it will either land on heads, H, or tails, T. Thus, the outcome of a coin flip, X, is expected to be a member of the set $S \in \{H, T\}$. This is a *trial*. Further, by saying that the outcome of the flip should be "fair," it is implied that for a large number of trials it is expected that half of the outcomes will be H and the other half T. In terms of probabilistic notation, $P\{X = H\} = P\{X = T\} = 0.5$. Note that sum of the probabilities of the outcomes, $P\{X = H\} + P\{X = T\} = 1$. This is consistent with an axiom of probability that the probability of all possible outcomes should add up to a probability of 1. In other words, by definition, a trial has an outcome.

Also, it is helpful to point out that $P\{X = H\} = P\{X = T\} = 0.5$ defines a probability mass function (PMF). This PMF can be denoted by $f(X)$. This mass function is plotted as a bar chart in

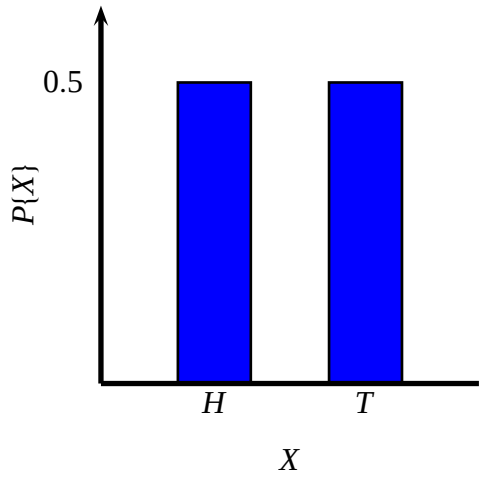


Figure 2.1: The PMF for a “fair” coin flip trial.

Figure 2.1. Here, f is a function of a *discrete random variable*, X, since there are a finite number of discrete outcomes.

However, what if the possible outcomes of a new random variable, V, form a continuum? Then it is a *continuous random variable*. For example, consider the initial velocity of a coin when it leaves a persons' thumb to be random. It will have a value, but it would be very difficult to know a priori the velocity or even guess a discrete set of velocities that the coin might have. Similar to before, it is desired to create a function $f(V)$ that relates outcomes to a measure of probability.

One method to assign probabilities is to discretize the outcome space and then assign probabilities to these defined regions. Thus, it might be supposed that the probability that the velocity will be less than or equal to v is given by $F(v) \equiv P\{V \leq v\}$. Further, the probability that the velocity will be less than or equal to $v + \Delta v$ is given by $F(v + \Delta v) \equiv P\{V \leq (v + \Delta v)\}$. From these two expressions, it is now possible to give the probability of a velocity that lies between v and $v + \Delta v$ as

$$P\{v < V \leq (v + \Delta v)\} = F(v + \Delta v) - F(v). \tag{2.1}$$

This allows a ratio of the change in probability of being less than v to the change in v to be written as

$$\frac{\Delta P\{V \leq v\}}{\Delta v} = \frac{F(v+\Delta v) - F(v)}{v + \Delta v - v} = \frac{F(v+\Delta v) - F(v)}{\Delta v}. \tag{2.2}$$

Taking the limit,

$$\lim_{\Delta v \to 0} \frac{\Delta P\{V \leq v\}}{\Delta v} = \frac{dF(v)}{dv} \equiv f(v). \tag{2.3}$$

Note that $f(v)$ has units of probability per the units of v, or a probability density. Hence, $f(v)$ is aptly named a *probability density function* (PDF). Thus, similar to being able to describe the density of any point in a bulk material, but not as easily a mass, it is possible to prescribe a probability density for any v.

Continuing the mass analogy, it is also possible to find the mass of a piece of material with a volume by integrating the density function of that volume. This relates to the *cumulative distribution function* (CDF),

$$F(v) = \int_{-\infty}^{v} f(\alpha)\, d\alpha, \tag{2.4}$$

which allows finding the probability of V occurring between any two v's.

Two common PDFs, a normal and a uniform distribution, are given in Figure 2.2.

Returning to the coin toss problem, a simple numerical experiment will be conducted as an example. Suppose that a coin starts heads up, at height $h = 0$ above the surface it will land on, it is given an initial upward velocity of v_0, and it is set spinning with an angular velocity of ω rotations/s. Assuming a constant downwards gravitational acceleration of $g = -9.81$ m/s and ignoring body forces due to moving through air, the time of travel and the angle are given by

$$\begin{aligned} t &= \frac{-v_0 - \sqrt{v_0^2 - 2gh}}{g} \\ \theta &= \omega t. \end{aligned} \tag{2.5}$$

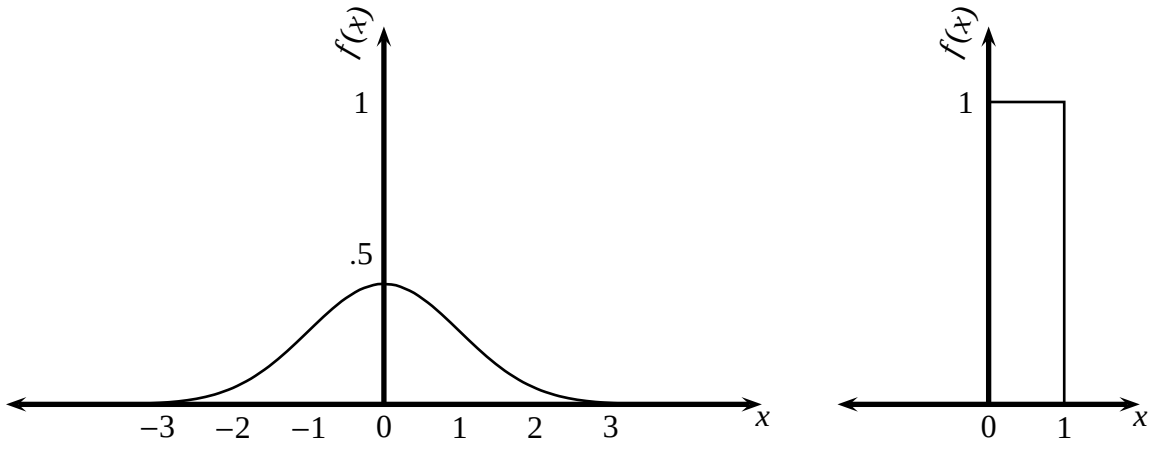


Figure 2.2: The standard normal distribution, N(0,1), on the left and a uniform distribution, u(0,1), on the right.

Since it is difficult to control the initial conditions, ω and v_0 are allowed to be continuous random variables. For simplicity, consider a situation where both are uniform. The random variable, ω, will be uniformly distributed between 10 and 30 rotations/s, or $\omega \sim u(10, 30)$. In the case of v_0, the random variable will be uniformly distributed between 2 and 5 m/s, or $v_0 \sim u(2, 5)$. Further, these two distributions are independent from each other. Thus, if a value of v_0 is randomly selected according to the given PDF, then the value of ω can be randomly selected without regards to v_0 (or vice-versa). Using a pseudo-random number generator, it is possible to run a numerical *Monte Carlo* experiment. Running a simple simulation with following Matlab code produces an approximate PDF for the angular displacement, given in Figure 2.3.

```
% Generate 1 million random variables, each
omega = 20*rand(1,1000000)+10;
v0 = 3*rand(1,1000000)+2;
g = -9.81; h = 0;   % constants
t = (-v0 - sqrt(v0.^2 - 2*g*h))/g; % time before hitting
theta = omega.*t; % angular displacement of coin

hist(theta,100)
```

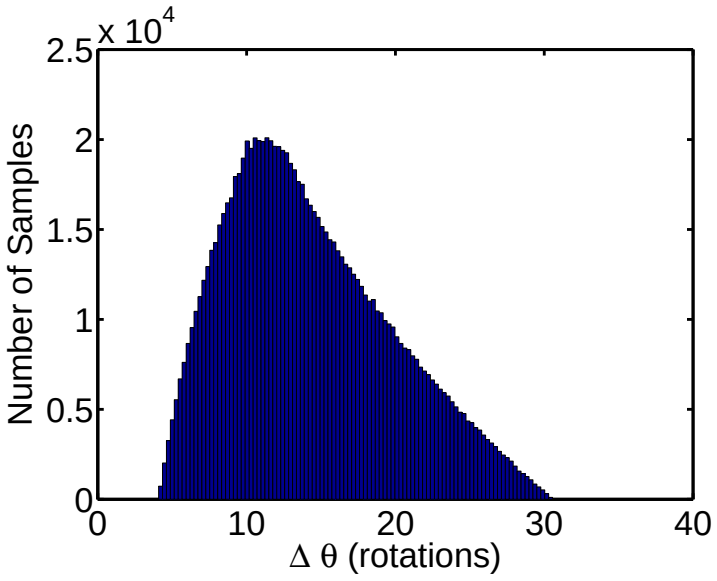


Figure 2.3: An approximate PDF, for the probabilistic coin toss experiment, created by binning one million Monte Carlo samples.

Further, if $\theta_2 \equiv \theta - n$, for integers n such that $\theta_2 \sim \left[-\frac{1}{4}, \frac{3}{4}\right)$, we can define an outcome of heads as being for angles on the subset interval $\left[-\frac{1}{4}, \frac{1}{4}\right)$ and the outcome of tails on the remaining interval $\left[\frac{1}{4}, \frac{3}{4}\right)$. Following this simple criteria, the simulation resulted in 499506 heads and 500494 tails. Thus, this simple example of a probabilistic dynamic system agrees well with our experience. Note that this example is not very computationally expensive since the samples do not require simulation in order to propagate the distribution.

2.2 Mean, Variance, and Covariance

The mean, variance, and covariance are very familiar and useful statistical measures. The concepts are briefly reviewed here as well as emphasizing that extreme care must be used when utilizing these measures. The need for caution is then demonstrated at the end of the section with an example framed in terms of a dynamic system.

Starting with the discrete case, for a set of m numbers x_1, x_2, x_3, $\ldots x_m$, each with a probability of p_i of occurring, the population mean and variance can be given by

$$\begin{aligned} \bar{x} &= \sum_{i=1}^{m} p_i x_i \\ \text{var}(x) &= \sum_{i=1}^{m} p_i (x_i - \bar{x})^2. \end{aligned} \tag{2.6}$$

Now suppose that all of the x_i are finite, except x_1 which has a fixed probability $0 < \inf\{p_1\} \leq 1$. Then it can be seen in the limit sense, the mean and variance *break down*:

$$\begin{aligned} \lim_{x_1 \to \infty} \bar{x} &= \sum_{i=1}^{m} p_i x_i = \infty \\ \lim_{x_1 \to \infty} \text{var}(x) &= \sum_{i=1}^{m} p_i (x_i - \bar{x})^2 = \infty. \end{aligned} \tag{2.7}$$

Similar to Equation (2.6), the mean and variance can be defined for a continuous random variable, x:

$$\begin{aligned} \bar{x} &= E[x] \equiv \int_{\Omega} x f(x)\, dx \\ \text{var}(x) &= E\left[(x - E[x])^2\right] \equiv \int_{\Omega} (x - E[x])^2 f(x)\, dx. \end{aligned} \tag{2.8}$$

Here $f(x)$ is a PDF and Ω is a bound that includes all non-zero probabilities in $f(x)$.

As an example, consider the Gaussian distribution, with a PDF given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}. \tag{2.9}$$

Here, μ is the mean of the distribution and σ^2 is the variance. Additionally the distribution often is referred to by its two parameters in the form $N(\mu, \sigma)$. By plotting three of the characteristic

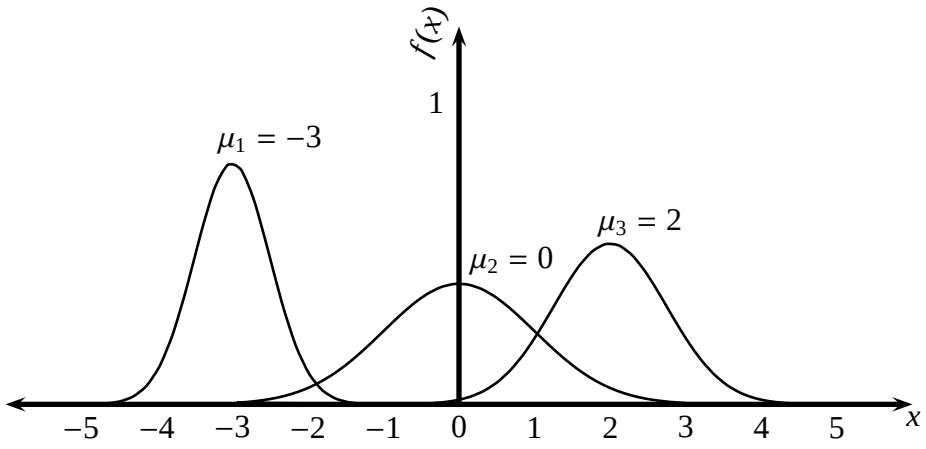


Figure 2.4: Three normal distributions, $N(\mu, \sigma)$, are plotted: $N(-3, 0.5)$, $N(0, 1)$, and $N(2, 0.75)$. The mean, μ, for each is marked so that it is possible to see that the value of the mean corresponds the the "location" of the distribution.

bell-shaped curves in Figures 2.4 and 2.5, it is possible to visually see that μ is located at the peak and σ relates to the "width" of the distribution. It is worthy to note that both the mean and standard deviation (the square root of the variance) can be seen to have the same units as the random variable of interest.

While the mean, when it exists,[1] is only guaranteed to coincide with the peak for distributions that are symmetric and uni-modal ("single-peaked"), for reasonably regular distributions, the mean can still offer a good indicator as to where the bulk lies. For this reason, the mean is known as a *parameter of location*. Further, for large numbers of samples of a distribution with finite mean and variance, the sample mean (average) does converge to the true mean. This is known as the *Law of Large Numbers* (e.g. Ross [5]). Additionally, the standard deviation, since it relates to the width of a distribution is known as a *parameter of scale*.

[1]The qualification on existence of the mean is made since there are distributions defined on an infinite domain, such as the Cauchy distribution, where the mean does not exist. However, this is not a complication with most commonly used distributions.

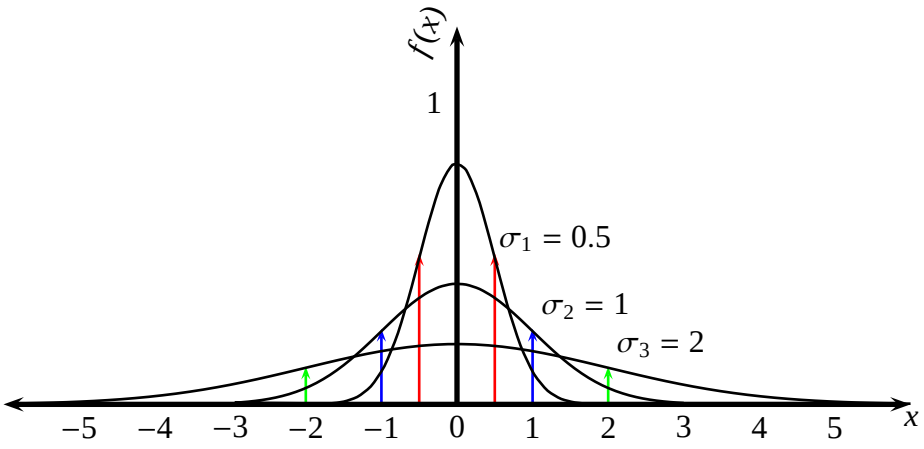


Figure 2.5: Three zero-mean normal distributions, $N(0, \sigma)$, are plotted: $N(0, 0.5)$, $N(0, 1)$, and $N(0, 2)$. The standard deviation, σ, for each is marked such that it is possible to see how this measure describes the "width" of the distribution.

The normal distribution is heavily used in this section as well as in many applications because of its nice mathematical properties. However, to conclude the section, a relevant cautionary example will be provided.

Suppose that after measuring the damping coefficients of many similar components, a mean damping coefficient of $\mu = 20$ N-s/m and a standard deviation of $\sigma = 0.5$ N-s/m is found. It is decided that the damping coefficients appear to have a bell-shaped trend and as such a normal distribution is a good representation. Thus, the damping coefficient is described as $d \sim N(20, 0.5)$. The model for the system is a damped oscillator:

$$m\ddot{x} + d\dot{x} + kx = 0. \tag{2.10}$$

For simplicity, it will be assumed that the mass $m = 20$ kg and the spring stiffness $k = 40$ N/m are deterministic. One hundred *realization* trajectories of this system, created by exciting the *realiza-*

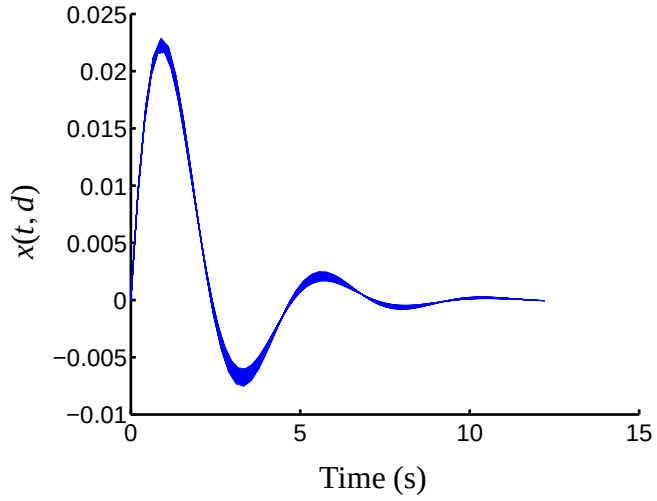


Figure 2.6: One hundred impulse response realizations of the system defined in Equation (2.10), with $d \sim N(20, .5)$

tion systems with an unit impulse, are shown in Figure 2.6. Each of these realizations were created by sampling the distribution for d and then simulating the impulse response for Equation (2.10).

Now suppose that the quantity of interest is the mean of $x(t)^2$. Intuitively, as $t \to \infty$, it would be expected for this value to go towards zero. The simulated trajectories in Figure 2.6 support this notion.

However, further thought brings up the issue that the system model only converges for realizations of $d > 0$. The mean for d is forty standard deviations away from 0, and thus is very close to zero. In fact by evaluating the CDF of the normal distribution,

$$P\{d < 0\} = F(x)|_{x=0} = \left[\frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{\text{x} - \mu}{\sigma\sqrt{2}}\right)\right]_{x=0} \approx \frac{1}{2}\left[1 + \text{erf}(-28.2843)\right] > 0. \tag{2.11}$$

Even though this expression is approximately zero, $P\{d < 0\}$ is still a distinctly non-zero constant. More formally, $\inf\left(P\{d < 0\}\right) > 0$. Thus, the mean of the squares of all of the asymptotically stable

trajectories converges towards zero as $t \to \infty$. Additionally, the mean of the squares of the unstable trajectories diverges as $t \to \infty$. For formality, the marginally stable trajectory created by $d = 0$ is known to be bounded. It is now possible to set up the mean square of the problem as

$$\begin{aligned}\overline{x^2}(t) =& p_{stable}\,\overline{x^2}_{stable}(t) + p_{unstable}\,\overline{x^2}_{unstable}(t) + p_{marg.stable}\,\overline{x^2}_{marg.stable}(t) \\ =& P\{d > 0\}\,\overline{x^2}_{stable}(t) + P\{d < 0\}\overline{x^2}_{unstable}(t) + P\{d = 0\}\,\overline{x^2}_{marg.stable}(t).\end{aligned} \tag{2.12}$$

Thus, it is now obvious that $\lim_{t\to\infty} \overline{x^2}(t)$ in fact diverges. Similarly, this can be shown shown for the variance. This fact can cause distributions with infinite bounds, like the normal, to be a theoretically problematic distribution. Through either trimming the tails of the distribution and renormalizing so that there remains a total probability of one (creating an *alpha-trim mean*) or selecting a distribution with an inherently bounded domain, this problem can be avoided. Further, it is worth mentioning that concepts of this type relate to a field known as *Robust Statistics* (i.e. see [6]).

2.3 A Further Note on the Relationship between PDFs and CDFs

In the previous section, PDFs and CDFs were defined. In this section, a relationship between the two will be demonstrated.

Starting with the relationship between a CDF and PDF, given in Equation (2.4) as $F(v) = \int_{-\infty}^{v} f(\alpha)\,d\alpha$, it easily follows that

$$F(v_0 + \Delta v) - F(v_0) = \int_{v_0}^{v_0+\Delta v} f(v)\,dv. \tag{2.13}$$

In a nutshell, Equation (2.13), demonstrates that an arbitrary CDF can be used to transform its associated PDF to the u[0,1] uniform distribution. The right side of this equation can be thought of as an area under the curve of the PDF; a probability of being between $v_0 + \Delta v$ and v_0. Likewise,

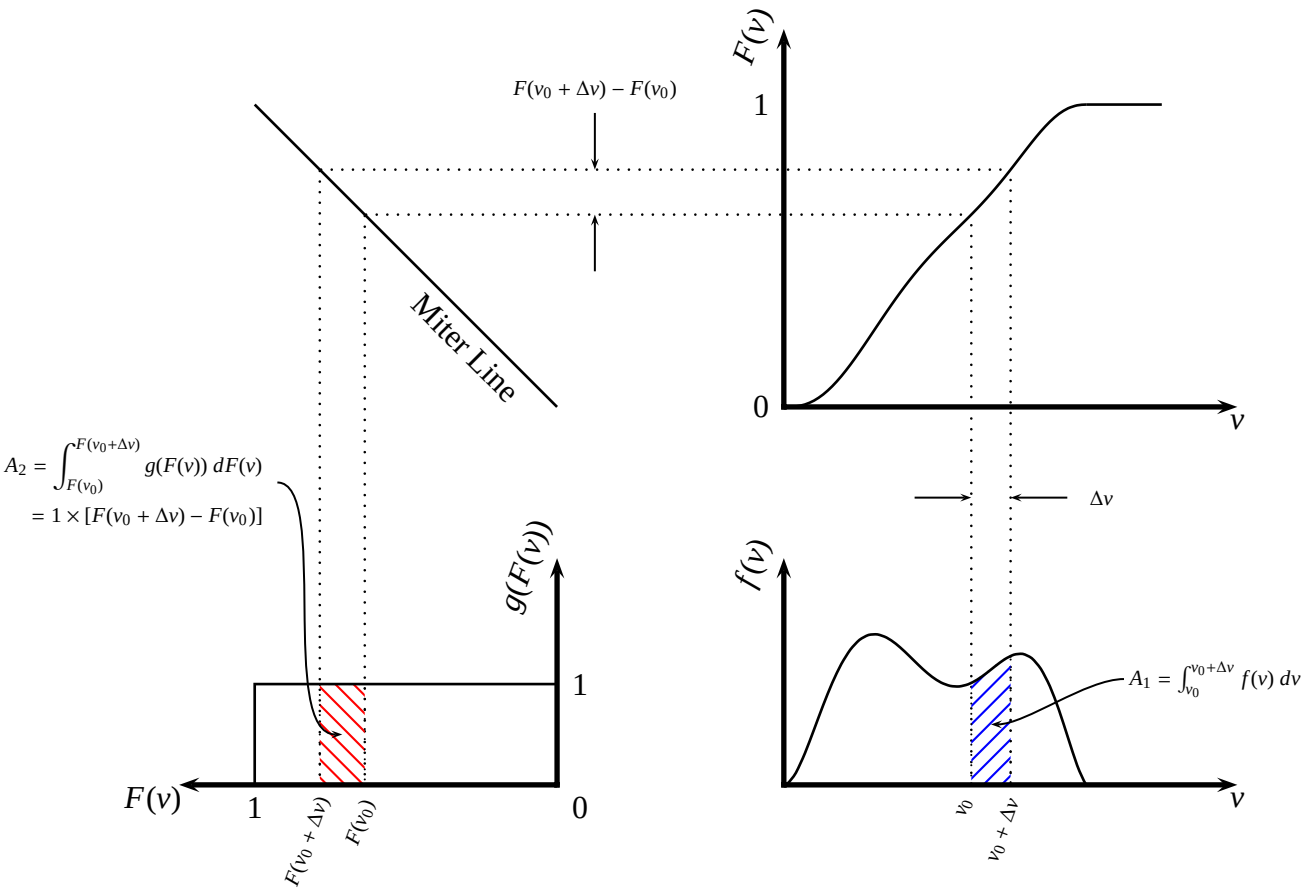


Figure 2.7: The relationship between, an arbitrary PDF ($f(v)$), its CDF ($F(v)$), and the u(0,1) uniform distribution. The y-axis for the u(0,1) distribution is labeled as $g(F(v))$ in order to emphasize that the probability density is a function of $F(v)$. This figure demonstrations how the CDF can transform a PDF to the uniform distribution. Additionally, following the process in reverse illustrates that the inverse CDF can transform a uniform distribution to the associated PDF.

this can be seen as being equivalent to a rectangle with an area defined as $1 \times [F(v_0 + \Delta v) - F(v_0)]$. Additionally, it is known that, by definition, the CDF (hereby denoted $F(v)$) is bounded such that $F(v) \in [0, 1]$. Thus, any interval on the PDF produces a rectangular probability region with height one and contained in the domain [0,1]. This is the uniform distribution, u[0,1]. The relationship is visually depicted in Figure 2.7.

Further, it is possible to use limits to demonstrate this relationship applies in infinitesimal cases. It can be understood that the CDF should be continuous for this relationship to apply. First, the limit of $A_1/\Delta v$, in Figure 2.7, as δv approaches zero:

$$\lim_{\Delta v \to 0} \frac{A_1}{\Delta v} = \lim_{\Delta v \to 0} \frac{\int_{v_0}^{v_0+\Delta v} f(v)\, dv}{\Delta v} = f(v_0). \tag{2.14}$$

For A_2, the integral of g is first defined as $\int g(\alpha)\, d\alpha \equiv G(\alpha)$. Using this notation, $A_2 = \int_{F(v_0)}^{F(v_0+\Delta v)} g(F(v))\, dF(v) = G(F(v_0 + \Delta v)) - G(F(v_0))$. Thus, similarly, for $A_2/\Delta v$:

$$\lim_{\Delta v \to 0} \frac{A_2}{\Delta v} = \lim_{\Delta v \to 0} \frac{G(F(v_0 + \Delta v)) - G(F(v_0))}{\Delta v} = \frac{d}{dv_0} G(F(v_0)) = g(F(v_0))\, f(v_0). \tag{2.15}$$

Now using that probability is conserved and as such $A_1 = A_2$:

$$\lim_{\Delta v \to 0} \frac{A_1}{\Delta v} = f(v_0) = \lim_{\Delta v \to 0} \frac{A_2}{\Delta v} = g(F(v_0))\, f(v_0). \tag{2.16}$$

Thus, for cases where $f(v_0) \neq 0$, it must hold that $g(F(v_0)) = 1$. Replacing v_0 by v to look at all v that are defined, it can be seen that $g(F(v)) = 1$. Additionally, since there is zero probability of $F(v) > 1$ or $F(v) < 0$, $g(F(v))$ can be defined as

$$g(F(v)) \equiv \begin{cases} 1 & \text{if } F(v) \in [0, 1] \\ 0 & \text{otherwise} \end{cases} \tag{2.17}$$

As such, given samples from an arbitrary PDF, $v \sim f(v)$, it is possible to transform them to samples from the u[0,1] distribution by evaluating $F(v)$.

By definition, CDFs monotonically increase. Thus, an inverse can be defined for all areas that the slope is non-zero. For any regions of the CDF that are zero, the inverse is undefined. But this is not a complication since it simply implies that there is zero probability of a sample occurring in

such a region. Hence, $F^{-1}(v)$ is defined as the inverse of the CDF $F(v)$, across all bounds where it exists.

As would be intuitively expected, it is possible to take samples, such as $x \sim u[0,1]$ and use the inverse CDF of an arbitrary distribution, $F^{-1}(x)$, to transform them to the associated PDF, $f(x)$. Practically, this is useful since pseudo-random number generators typically attempt to mimic uniform distributions, and thus this relationship can be used to transform these to samples of a different distribution.

2.4 Polynomial Chaos

At this point, it is possible to more formally introduce the tool extensively used throughout this work: Polynomial Chaos Expansions (PCEs). PCEs allow for the parameterization of equations and systems in terms of unknown variables. The independent random variables, hereby denoted by $\xi_1, \xi_2, \xi_3, \ldots, \xi_r$, define a space that for the purpose of this work is a subset of an r-dimensional space of real numbers, $\mathbb{R}^r$. One application is to represent uncertainty in parameters. A discussion of multiple random variables will be given in the next section in order generalize the results in this section. However, for now, it is easier to think of one random variable, ξ.

PCEs are formulated in terms of orthogonal basis function expansions. For example, using an arbitrary set of orthogonal basis functions, $\phi^0(\xi), \phi^1(\xi), \phi^2(\xi), \ldots$, an arbitrary function of a random variable, $q(\xi)$, is expanded as follows:

$$q(\xi) = \sum_{i=0}^{\infty} \hat{q}^i \phi^i(\xi) \approx \sum_{i=0}^{\mathrm{L}} \hat{q}^i \phi^i(\xi). \tag{2.18}$$

In practice, expansions are generally truncated in the manner shown in the right-hand portion of Equation (2.18). This approach is closely related to the familiar discrete Fourier transform, for which the basis functions are sinusoidal.

The utilized basis functions can be composed of many different orthogonal sets, including Hermite, Jacobi, and Legendre polynomials. The built in orthogonality is very useful in calculations. It can be defined in terms of the inner product

$$\delta_{ij} = \left\langle \phi^i(\xi), \phi^j(\xi) \right\rangle \equiv \int_\Omega \phi^i(\xi)\phi^j(\xi)w(\xi)\, d\xi \tag{2.19}$$

as a set where $\delta_{ij} = 0$ iff $i \neq j$. Note that the polynomials are orthogonal with respect to a weighting function, $w(\xi)$. The similar form of these weighting functions to frequently used PDFs imply optimality for representing random variables of corresponding distributions. Additionally, terms of orthogonal polynomial sets are generally conveniently defined through recurrence relations. A summary of these polynomials and useful properties are given in Table 2.1.

A simple example of a PCE is

$$q(\xi) = \sum_{i=0}^{\infty} \hat{q}^i \phi^i(\xi) = \hat{q}^0\phi^0(\xi) + \hat{q}^1\phi^1(\xi) + 0\phi^2(\xi) + \ldots. \tag{2.20}$$

For this example, ξ is made to be a uniform random variable, $\xi \sim u[-1, 1]$. From the given table, Legendre polynomials are associated with the uniform distribution. The first two polynomials are $\phi^0(\xi) = P^0(\xi) = 1$ and $\phi^1(\xi) = P^1(\xi) = \xi$. Thus, this expansions is a linear remapping of the uniform random variable. As such, it is easy to see that $q(\xi) \sim u\left[-\hat{q}^1 + \hat{q}^0, \hat{q}^1 + \hat{q}^0\right]$. Figure 2.8 provides a visual interpretation of this transformation. The same result could have been obtained through a using a large number of samples in a Monte Carlo simulation. In fact, for more complicated transformations involving higher order terms, especially with multiple random variables,

Table 2.1: Summary table for three orthogonal polynomial sets commonly used with Polynomial Chaos. A large number of additional orthogonal polynomial sets are available; i.e. see [1, 2]. Note $H^0(\xi) = P^0(\xi) = 1$. Also, the weighting functions have been normalized such that $\delta_{00} = 1$.

Set	Property	
Hermite	associated PDF	Gaussian
	weight	$w(\xi) = \frac{1}{\sqrt{2\pi}} e^{-\xi^2/2}$
	domain	$\Omega = (-\infty, \infty)$
	rec. relation	$H^{n+1}(\xi) = \xi H^n(\xi) - nH^{n-1}(\xi)$, $H^1(\xi) = \xi$
Jacobi	associated PDF	Beta$(\beta + 1, \alpha + 1)$, $\alpha > -1, \beta > -1$
	weight	$w(\xi) = \frac{\Gamma(\alpha+\beta+2)}{\Gamma(\alpha+1)\Gamma(\beta+1)2^{\alpha+\beta+1}}(1-\xi)^\alpha(1+\xi)^\beta$
	domain	$\Omega = [-1, 1]$
	rec. relation	$P^{n+1}(\xi) = \frac{(2n+\alpha+\beta+1)[(2n+\alpha+\beta+2)(2n+\alpha+\beta)\xi+\alpha^2-\beta^2]}{2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)}P^n(\xi)$ $-\frac{2(n+\alpha)(n+\beta)(2n+\alpha+\beta+2)}{2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)}P^{n-1}(\xi)$ $P^1(\xi) = \frac{1}{2}(\alpha+\beta+2)\xi + \frac{1}{2}(\alpha-\beta)$
Legendre	associated PDF	Uniform
	weight	$w(\xi) = \frac{1}{2}$
	domain	$\Omega = [-1, 1]$
	rec. relation	$P^{n+1}(\xi) = \frac{2n+1}{n+1}\xi P^n(\xi) - \frac{n}{n+1}P^{n-1}(\xi)$, $P^1(\xi) = \xi$

Monte Carlo simulations are in general the only practical method to obtain a PDF representation from a PCE.

Additionally, although it was not needed since the coefficient $\hat{q}^2 = 0$, the second basis function ($n + 1 = 2$) can be found from the recurrence relation provided in Table 2.1. As such, using $\phi^0(\xi)$ and $\phi^1(\xi)$,

$$P^2(\xi) = P^{n+1}(\xi) = \frac{2n+1}{n+1}\xi P^n(\xi) - \frac{n}{n+1}P^{n-1}(\xi) = \frac{3}{2}\xi(\xi) - \frac{1}{2}(1) = \frac{3}{2}\xi^2 - \frac{1}{2}. \tag{2.21}$$

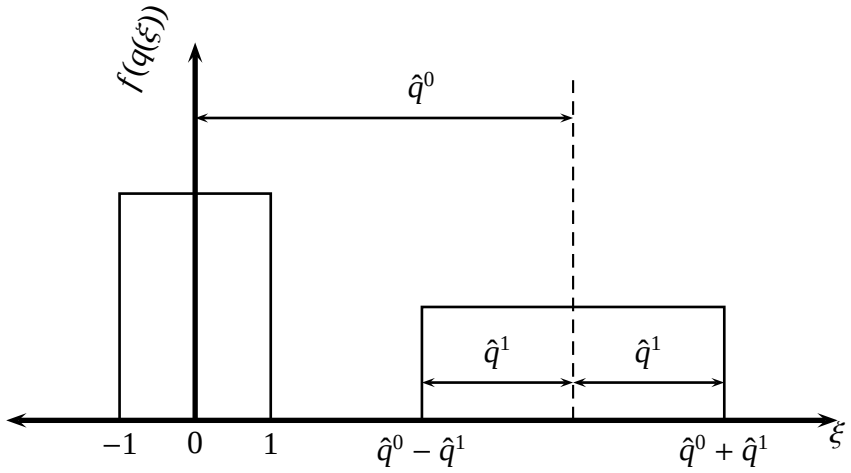


Figure 2.8: The PDFs for two uniform distributions are shown: one for $\xi \sim u[-1, 1]$ another for the linear transformation $\hat{q}^1\xi + \hat{q}^0 \sim u[-\hat{q}^1 + \hat{q}^0, \hat{q}^1 + \hat{q}^0]$. The y-axis represents probability density.

Further, PCEs have the nice property that the weighted expectations and covariances are easy to determine using properties of orthogonality. For $q(\xi) = \sum_{i=0}^{\infty} \hat{q}^i \phi^i(\xi)$, the expectation

$$E[q(\xi)] = \int_\Omega q(\xi) w(\xi) d\xi = \int_\Omega \phi^0(\xi) \sum_{i=0}^{\infty} \hat{q}^i \phi^i(\xi) w(\xi) d\xi = \hat{q}^0 \tag{2.22}$$

gives the mean of the PCE. Here, the property that polynomial sets are typically generated starting with $\phi^0(\xi) = 1$ is exploited.

Introducing a second expansion, $s(\xi) = \sum_{i=0}^{\infty} \hat{s}^i \phi^i(\xi)$, allows demonstrating that covariances can also be found:

$$\begin{aligned} &E[(q(\xi) - E[q(\xi)])(s(\xi) - E[s(\xi)])] \\ &= \int_\Omega \left[\sum_{i=0}^{\infty} \hat{q}^i \phi^i(\xi) - \hat{q}^0 \right] \left[\sum_{i=0}^{\infty} \hat{s}^i \phi^i(\xi) - \hat{s}^0 \right] w(\xi)\, d\xi \\ &= \sum_{i=1}^{\infty} \hat{q}^i \hat{s}^i \delta_{ii}. \end{aligned} \tag{2.23}$$

Approximate values of means and covariances are given by truncating the series. Note that performing finite series truncations is a commonly used practice when dealing with orthogonal sets since spectral series generally converge extremely rapidly for well-behaved problems (for example, see [2, 7]).

2.5 The Weighted-Least-Squares Interpretation of Polynomial Chaos

It is useful to briefly show how orthogonal expansions relate to the least squares fitting. Additionally, this helps to illustrate the meaning of the weighting function. For a C_1 (continuous first derivative with respect to ξ) function $g(\xi)$, it is possible to expand it as follows:

$$g(\xi) = \sum_{i=0}^{\infty} \hat{g}^i \phi^i(\xi) \tag{2.24}$$

where $\left\{\phi^i(\xi)\right\}$ are the terms of an orthogonal basis function. As mentioned, these polynomial sets are orthogonal with respect to a weighting function $w(\xi)$. (See Table 2.1.)

Moving the discussion towards regression, a residual, $r(\xi)$, can be defined as the difference between a function, $f(\xi)$, and the function being fit to it, $g(\xi)$. Using the expansion defined in Equation (2.24), the residual can be expressed as:

$$r(\xi) = f(\xi) - \sum_{i=0}^{\infty} \hat{g}^i \phi^i(\xi). \tag{2.25}$$

This residual can now be used to formulate a cost function with respect to a weighted squared residual:

$$J = \int_{\Omega} \left[r(\xi)\right]^2 w(\xi)\, d\xi. \tag{2.26}$$

Minimizing this cost function creates a weighted least-squares. This can be accomplished through first rearranging J:

$$\begin{aligned} J &= \int_{\Omega} \left[f(\xi) - \sum_{i=0}^{\infty} \hat{g}^i \phi^i(\xi) \right]^2 w(\xi)\, d\xi \\ &= \int_{\Omega} \left(f(\xi) \right)^2 w(\xi)\, d\xi - 2 \int_{\Omega} f(\xi) \left[\sum_{i=0}^{\infty} \hat{g}^i \phi^i(\xi) \right] d\xi + \sum_{i=0}^{\infty} \left(\hat{g}^i \right)^2 \delta_{ii}. \end{aligned} \tag{2.27}$$

Now the minimum is determined through differentiating with respect to $\hat{g}^i$ and setting the derivative equal to zero:

$$0 = \frac{\partial J}{\partial \hat{g}^i} = -2 \int_{\Omega} f(\xi) \left[\phi^i(\xi) \right] w(\xi)\, d\xi + 2 \left(\hat{g}^i \right) \delta_{ii}. \tag{2.28}$$

This yields the following solution for each parameter:

$$\hat{g}^i = \frac{1}{\delta_{ii}} \int_{\Omega} f(\xi) \phi^i(\xi) w(\xi)\, d\xi. \tag{2.29}$$

Note that the least-squares regression used to determine the coefficient for a basis function is independent from all other basis functions. This is a direct result of orthogonality and will be discussed more formally in Chapter 4. Since extra emphasis in the regression is received according to the weighting function, it is important to select the set of polynomials such that they are orthogonal with respect to the desired weighting function.

2.6 Representing Arbitrary Distributions with Polynomial Chaos Expansions

Building on the prior three sections, an interesting question to ask is: “given a distribution, and a set of orthogonal basis functions, how can a PCE be constructed to implicitly represent the PDF?”

A necessary related question is, "what is a function that could transform samples from one arbitrary distribution such that they become samples of a desired distribution?"

As discussed, PCEs utilize a weighting function that is represented by w and serves to weight the fit based upon probability densities of the random variable. Using the relationships developed in Section 2.3, it is known that the necessary function to transform the weighting distribution $w(\xi)$ to the new arbitrary PDF, $m(\xi)$, can be created from a CDF and an inverse CDF. Specifically for a continuous $\Omega \subset \mathbb{R}$, the random variable evaluated by its CDF $W(\xi) \equiv \int_{\inf \Omega}^{\xi} w(\alpha)\, d\alpha$ will follow a $u[0,1]$ distribution. This uniform distribution can now be transformed into the desired distribution, $h(\gamma)$, through utilizing the inverse CDF, $H^{-1}(\gamma)$. Thus, $\gamma \sim W(\xi)$ and as such the required function for the transformation is $g(\xi) = H^{-1}(W(\xi))$.

Now Equation (2.29) can be applied to find the PCE:

$$\hat{g}^i = \frac{1}{\delta_{ii}} \int_{\Omega} H^{-1}(W(\xi))\phi^i(\xi)w(\xi)\, d\xi. \tag{2.30}$$

In general, it is not always possible to find analytic solutions for $H^{-1}(\xi)$. However, the solutions for three elementary distributions are given in Table 2.2

Table 2.2: Three inverse CDFs. CDFs with finite support will have undefined endpoints for the inverse CDF and thus these end points have been defined for convenience. The distribution denoted $\beta^*(2,2)$ is the linear transformation of the $\beta(2,2)$ distribution that is the weighting function for the $J(1,1)$ Jacobi polynomial set.

Distribution	CDF	Defined Inverse CDF
$N(\mu, \sigma)$	$\frac{1}{2}\left[1 + \text{erf}\left(\frac{\xi-\mu}{\sigma\sqrt{2}}\right)\right]$	$\mu + \sigma\sqrt{2}\text{erf}^{-1}(2\xi - 1)$
$u[a,b]$	$\begin{cases} 0 & \text{if } \xi \in (-\infty, a) \\ \frac{\xi-a}{b-a} & \text{if } \xi \in [a,b] \\ 1 & \text{if } \xi \in (b, \infty) \end{cases}$	$(b-a)\xi + a$ for $\xi \in [0,1]$
$\beta^*(2,2)$	$-\frac{1}{4}\xi^3 + \frac{3}{4}\xi + \frac{1}{2}$	$\frac{(-[\delta(\xi)]^2 - 1 - \sqrt{3}i[\delta(\xi)]^2 + \sqrt{3}i)}{2\delta(\xi)}$ with $\delta(\xi) = (1 - 2\xi + 2(-\xi + \xi^2)^{\frac{1}{2}})^{\frac{1}{3}}$

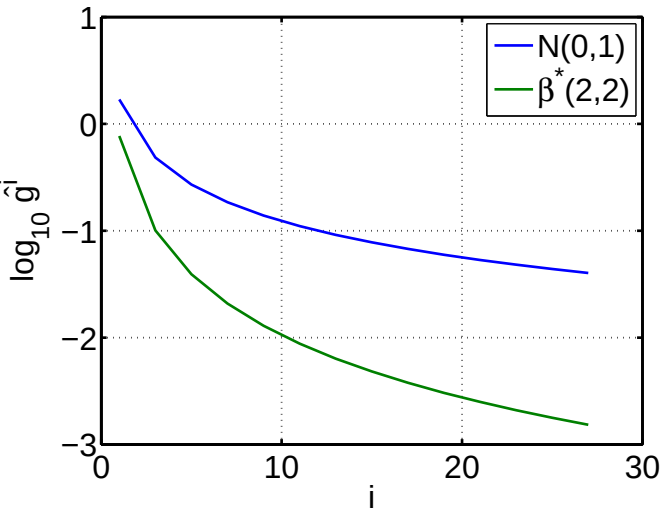


Figure 2.9: The convergence of Legendre polynomials for a PCE to represent the defined $\beta^*(2,2)$ and $N(0,1)$ distributions. This is shown by giving the magnitude of the expansions coefficients for the Legendre polynomial representation. Only odd coefficient terms are given since the even coefficient terms are zero. Note that convergence to the Beta-type distribution occurs faster.

As an example, the distribution $\beta^*(2,2)$, can be approximately represented using Legendre Polynomials. To do this, Equation (2.30) is used with $W(\xi) = \frac{\xi+1}{2}$ and $w(\xi) = \frac{1}{2}$; H^{-1} is the inverse CDF given in Table 2.2; and $\left\{\phi^i(\xi)\right\}$ are the orthogonal Legendre basis functions defined in Table 2.1. The convergence and rapid decrease of the coefficients associated with the basis functions are shown in Figure 2.9.

Additionally, the same procedure was performed in order to use a Legendre PCE to implicitly represent an $N(0,1)$ distribution. However, unlike the the case for the Beta type distribution, the Gaussian distribution's unboundedness is substantially different qualitatively in that the Legendre PCEs use a finite domain. Thus it can be seen that convergence occurs significantly slower.

2.7 Multi-Dimensional Basis Functions

Expanding on the previous section, it is possible to formulate new orthogonal sets in terms of other orthogonal sets of independent variables. Thus, even when dealing with multiple independent random variables, it is convenient to just refer to a single set of basis functions, $\phi(\xi)$, where ξ can include multiple independent random variables. This type of technique is analogous to a multi-dimensional discrete Fourier transform. Additionally, similar techniques have applications in spectral methods [8].

A convenient way to discuss an arbitrary number of independent random variables in the Polynomial Chaos framework is presented here. Without loss of generality, this discussion is given in terms of two independent random variables.

The discussion begins by expanding a function, $g(\xi_1, \xi_2)$, in terms of two sets of basis functions, $\phi_1(\xi)$ and $\phi_2(\xi)$, in a similar manner to the Askey-scheme [2]:

$$g(\xi_1, \xi_2) = \sum_{\forall (i_1, i_2):(0 \leq i_1 + i_2 \leq \mathrm{W})} \hat{g}_1^{i_1} \phi_1^{i_1}(\xi_1) \hat{g}_2^{i_2} \phi_2^{i_2}(\xi_2). \tag{2.31}$$

In this equation $\left\{\hat{g}_1^{i_1}\right\}$ and $\left\{\hat{g}_2^{i_2}\right\}$ are sets of coefficients corresponding to their respective basis function set. Additionally, i_1 and i_2 are indices (not exponents) corresponding to their respective basis function set.

Now a new index, i, from 0 to L, is created in one to one correspondence with index pairs (i_1, i_2). Note that these are countable sets and so this correspondence is possible in terms of a limit, such as the case of infinite order basis function sets. Also, for convenience and to follow convention, $i = 0$ should correspond to $(i_1, i_2) = (0, 0)$.

Letting $\hat{g}^i \equiv \hat{g}_1^{i_1}\hat{g}_2^{i_2}$ and $\psi^i(\xi_1,\xi_2) \equiv \phi_1^{i_1}(\xi_1)\phi_2^{i_2}(\xi_2)$, the expansion can be written in the following truncated form:

$$g(\xi_1,\xi_2) \approx \sum_{i=0}^{\mathrm{L}} \hat{g}^i\psi^i(\xi_1,\xi_2). \tag{2.32}$$

Additionally, define $w(\xi_1,\xi_2) \equiv w_1(\xi_1)w_2(\xi_2)$. Since the random variables are independent, it is easy to see that the newly defined set of basis functions, $\psi(\xi_1,\xi_2)$, is orthogonal with respect to $w(\xi_1,\xi_2)$. I.e.

$$\left\langle \psi^i(\xi_1,\xi_2), \psi^j(\xi_1,\xi_2) \right\rangle = \int_{\Omega_1}\int_{\Omega_2} \psi^i(\xi_1,\xi_2)\psi^j(\xi_1,\xi_2)w(\xi_1,\xi_2)\,d\xi_1 d\xi_2 = \delta_{ij}, \tag{2.33}$$

where $\delta_{ij} = 0$ iff $i \neq j$.

Further it is convenient to note that $\delta_{ij} = \delta_{1,(i_1,j_1)}\delta_{2,(i_2,j_2)}$. The first number in the subscript serves to distinguish the inner products of their respective basis function sets. The terms in the parenthesis correspond to the indices defined in Equation (2.19). Therefore, the inner product of the newly constructed orthogonal set is known in terms of single dimensional sets.

Thus, for notational convenience, $\phi^i(\xi)$ can refer to a single dimensional orthogonal basis function like $\phi^i(\xi_1)$, or an orthogonal basis function in terms of two or more independent random variables (i.e. $\psi^i(\xi_1,\xi_2)$). Also, it is noted that constituent single-domain basis functions do not have to be from the same orthogonal sets. As such, the given derivations apply to problems where the independent random variables follow different distributions.

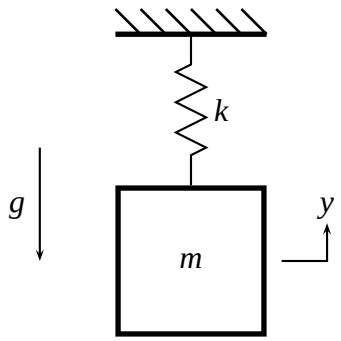


Figure 2.10: The spring-mass oscillator used as an example throughout the chapter. The parameters are defined as follows: m is the mass, k is the linear stiffness, and g is the gravitational acceleration.

2.8 Propagating Probabilistic Initial Conditions in a Simple Linear System

It is instructive to briefly look at several methods for handling uncertainty. To do this, a single degree-of-freedom (1-DOF) undamped oscillator with probabilistic initial conditions is examined (see Figure 2.10). Liouville's theorem, a simple Monte Carlo simulation, and analytical propagation of a Gaussian distribution are all used to solve the same uncertain initial condition problem. This same problem will be solved using Polynomial Chaos in the next section. It is worth mentioning that the potential complications with the Gaussian distribution demonstrated in the previous section are not a problem since initial conditions do not factor into the stability of linear systems.

The Newtonian formulation for the system's dynamics can be given by the following set of differential equations:

$$\begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}. \tag{2.34}$$

Alternately, the Hamiltonian formulation can be presented directly from the sum of potential (V) and kinetic (T) energies, in terms of the displacement of the mass from the equilibrium ($y(t)$) and its linear momentum ($\rho(t) = m\dot{y}(t)$):

$$H(y, \rho, t) = T + V = \left(\frac{1}{2}\frac{\rho^2}{m}\right) + \left(mgy(t) + \frac{1}{2}ky^2(t)\right). \tag{2.35}$$

Using the Hamiltonian, it is now possible to generate the equations of motion in terms of a position and conjugate momentum pair. This is accomplished through using $\dot{x} = J\frac{\partial H}{\partial x}$, where the symplectic matrix, J, for this case is taken to be $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. Thus,

$$\dot{x} = \begin{bmatrix} \dot{y} \\ \dot{\rho} \end{bmatrix} = J\frac{\partial H}{\partial y} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} \frac{\partial H}{\partial y} \\ \frac{\partial H}{\partial \rho} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} mg + ky \\ \frac{\rho}{m} \end{bmatrix} = \begin{bmatrix} \frac{\rho}{m} \\ -mg - ky \end{bmatrix}. \tag{2.36}$$

Additionally, it is easier to analyze the system by making the transformation $y = z - \frac{m}{k}g$. Using this transformation,

$$\dot{x} = \begin{bmatrix} \dot{z} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{bmatrix}\begin{bmatrix} z \\ \rho \end{bmatrix}. \tag{2.37}$$

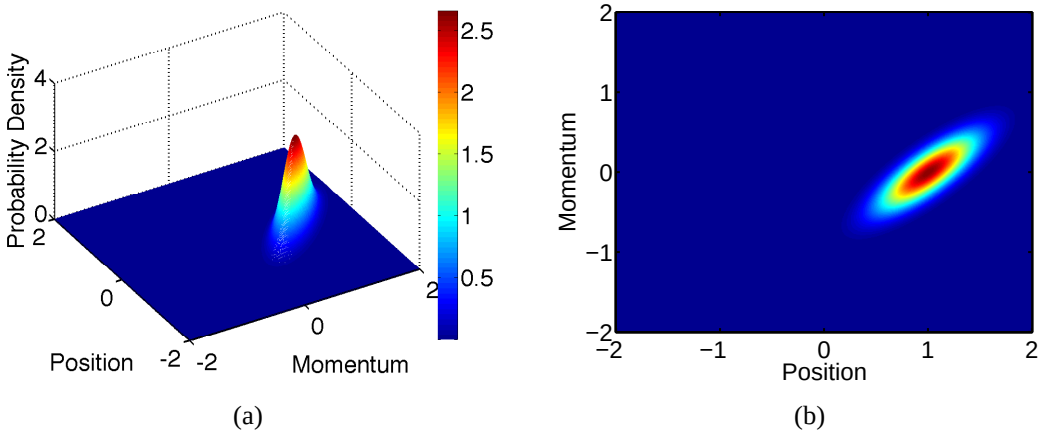
The solution in this case, in terms of the matrix exponential $\Phi(t, t_0)$ can be found to be

$$x(t) = \begin{bmatrix} y(t) \\ \rho(t) \end{bmatrix} = \Phi(t, t_0)x_0 \equiv \begin{bmatrix} cos(\omega_n t) & \frac{1}{m}sin(\omega_n t) \\ -ksin(\omega_n t) & cos(\omega_n t) \end{bmatrix}\begin{bmatrix} z(0) \\ \rho(0) \end{bmatrix}, \text{ where } \omega_{\mathrm{n}} \equiv \sqrt{\frac{k}{m}}. \tag{2.38}$$

The initial conditions for this system are given by the multivariate Gaussian PDF:

$$f(x_0) = \frac{1}{\sqrt{(2\pi)^n |P|}} e^{-\frac{1}{2}(x_0 - \overline{x_0})^T P^{-1}(x_0 - \overline{x_0})}. \tag{2.39}$$

Here, n refers to the number of random variables and thus $n = 2$. Additionally, the parameter P is the covariance matrix for the distribution. For the example, the mean is given by $\overline{x_0} = [\overline{z_0}, \overline{\rho_0}]^T = [1,\ 0]^T$, and the covariance matrix $P = \begin{bmatrix} 0.1 & 0.08 \\ 0.08 & 0.1 \end{bmatrix}$. This PDF is plotted in Figure 2.11.



(a) (b)

Figure 2.11: A 3D and a 2D plot of the PDF of initial conditions for momentum and displacement. Both are given since the 3D plot helps with visualizing the surface, while the 2D plot allows better seeing the location of the probability density. The latter is more important in the examples to come that illustrate the PDF evolving over time.

The first method of propagation, a method that the problem was specifically set up for, is Louisville's theorem. In its most simple form, Louisville's theorem states that *momentum/velocity phase volume* is conserved in canonical transforms (for example, see [9, 10]). Phrased a different way, one can define an arbitrary continuous volume in the phase space and this region may deform over time, but the volume will remain constant. Thus, in terms of the probabilistic initial value problem, one can trace a "particle" from its initial conditions and since there is not expansions or contraction of the phase space, the probability density will remain a constant for the trajectory. Following this particle interpretation, the evolution of probability distribution can be found, as is illustrated in Figure 2.12.

Secondly, it is possible to analytically propagate the Gaussian distributed initial conditions throughout time in linear systems. An easy geometric interpretation can be obtained by looking at the

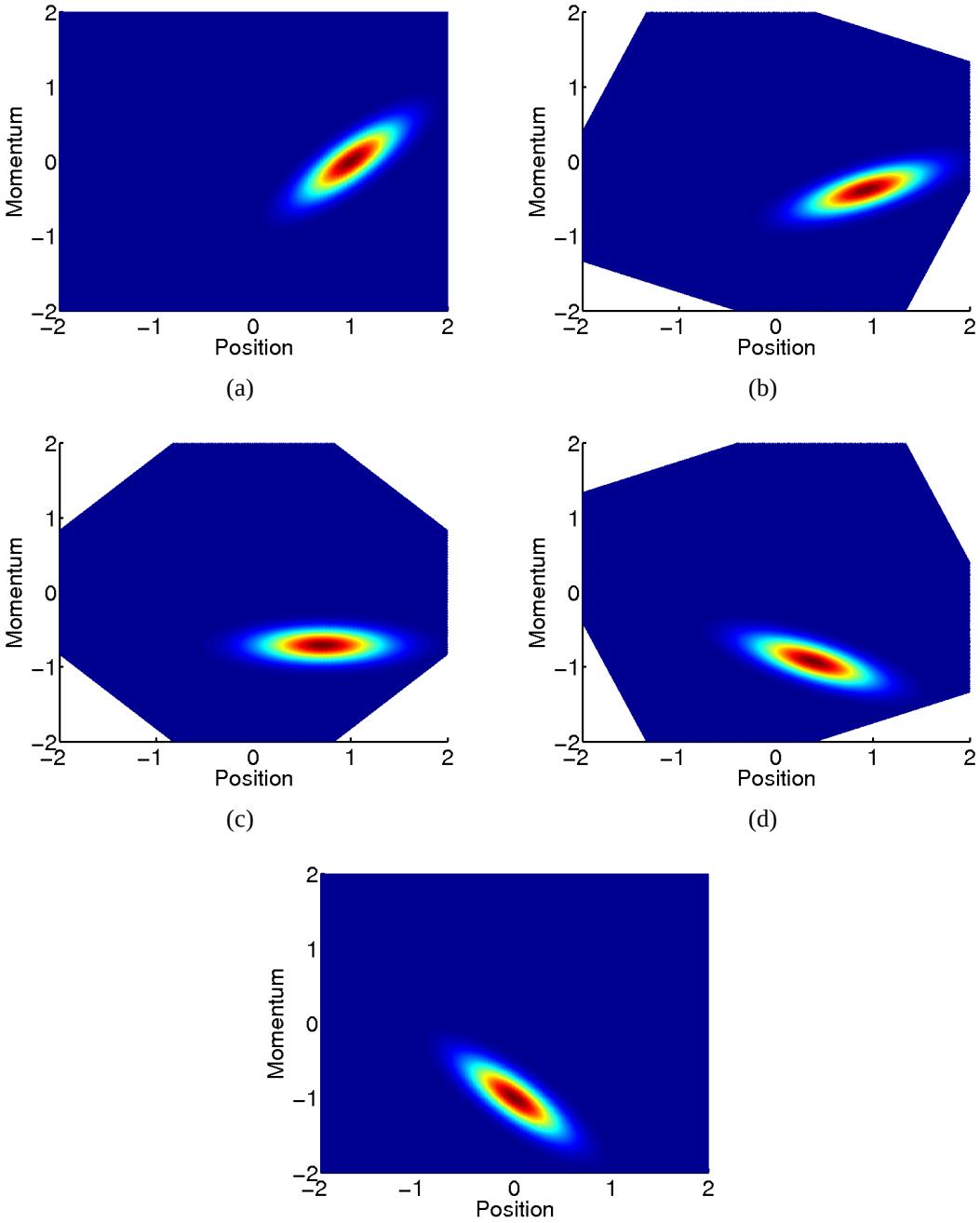


Figure 2.12: Propagation of uncertainty using Louisville's theorem. The plots show momentum vs. displacement from equilibrium. They evolve over time, starting with the initial conditions in (a), $t = T/16$ in (b), $t = T/8$ in (c), $t = 3T/16$ in (d), and $T = 1/4$ in (e), where T is the period of the system. The color scale is identical to the one given in Figure 2.11. Note the "cropped" edges within the bounds. This helps illustrate the uncertainty being propagated, from the initial conditions shown, along the *phase flow*.

distribution, such as the one shown in Figure 2.11. The level sets are ellipsoids. The matrix exponential, which exists for any linear system, performs linear transformations that preserve the ellipsoidal shape. In this example, it can be seen that the transformations consist of rotations in the phase space. More formally, the stability of the Gaussian distribution can be shown by defining a new random variable for the coordinates, $x_1 = Ax_0$, where A is a transformation matrix. Substituting $x_0 = A^{-1}x_1$ into Equation (2.39):

$$f(x_1) = \frac{c}{\sqrt{(2\,\pi)^n\,|P_0|}} e^{-\frac{1}{2}\left(A^{-1}x_1-\overline{x_0}\right)^T P_0^{-1}\left(A^{-1}x_1-\overline{x_0}\right)} = \frac{c}{\sqrt{(2\,\pi)^n\,|P_0|}} e^{-\frac{1}{2}(x_1-A\overline{x_0})^T A^{-T}P_0^{-1}A^{-1}\left(A^{-1}x_1-A\overline{x_0}\right)}. \quad (2.40)$$

Note that c was introduced as a constant since by definition of a PDF, it should be scaled so that $\int_{\mathbb{R}^n} f(x_1)dx_1 = 1$. Using the substitutions $\overline{x_1} = A\overline{x_0}$ and $P_1 = AP_0A^T$, creates the following expression:

$$f(x_1) = \frac{c}{\sqrt{(2\,\pi)^n\,|P_0|}} e^{-\frac{1}{2}(x_1-A\overline{x_1})^T P_1^{-1}(x_1-A\overline{x_1})}. \quad (2.41)$$

This makes the exponential component follow the original form of Equation (2.39). Thus, from that form, it is known that

$$f(x_1) = \frac{1}{\sqrt{(2\,\pi)^n\,|P_1|}} e^{-\frac{1}{2}(x_1-A\overline{x_1})^T P_1^{-1}(x_1-A\overline{x_1})} \quad (2.42)$$

is properly normalized to one. As such, if $A = \Phi(t, t_0)$ is set to be the the matrix exponential, the evolving mean and covariance matrix for the system can be given by

$$\begin{aligned} \overline{x}(t) &= \Phi(t, t_0)\overline{x_0} \\ P(t) &= \Phi(t, t_0)P_0\Phi^T(t, t_0). \end{aligned} \quad (2.43)$$

These exact relationships can be used to propagate the probabilistic initial conditions of the oscillator problem and generate the results in Figure 2.13.

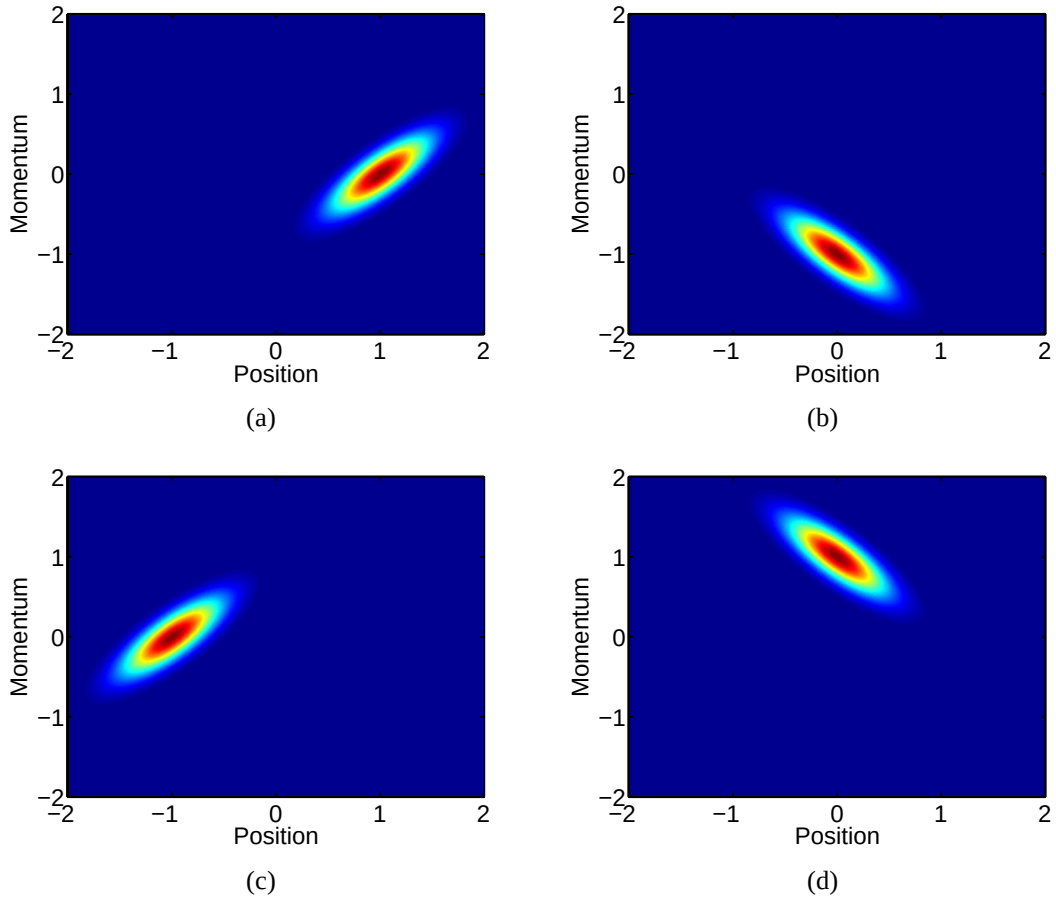


Figure 2.13: Analytical propagation of uncertainty, plotted as momentum vs. displacement from equilibrium, over time starting with the initial conditions in (a), $t = T/4$ in (b), $t = T/2$ in (c), and $t = 3T/4$ in (d), where T is the period of the system. The color scale is identical to that in Figure 2.11.

The final method of this section involves Monte Carlo simulations. In its simplest form, the distribution is represented by random samples generated according to the PDF. These samples can then be deterministically propagated by the given system. Standard normal samples (zero-mean, unit variance) can be generated by using a Box-Mueller transform on uniformly distributed pseudo-random numbers or the *randn* command in Matlab. To create pseudo-random numbers generated by the multi-variate Gaussian distribution, the Cholesky decomposition of the covariance matrix is used. Thus, the multi-variate samples, x_{mvg}, are created by transforming independent standard

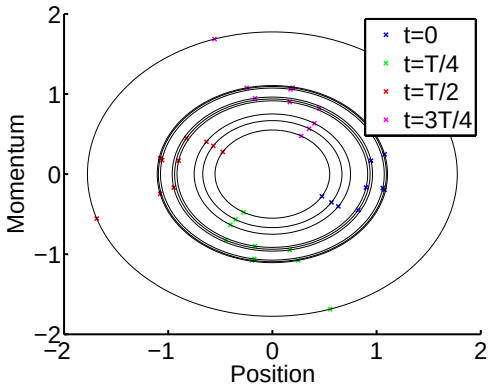


Figure 2.14: The trajectories for 10 samples plotted in momentum/position phase space.

normal random samples, x_s, with

$$x_{mvg} = \text{chol}(P)x_s + \overline{x_0}. \tag{2.44}$$

A schematic demonstrating the propagation of a few samples in phase space is given in Figure 2.14.

Further, through using a 2D histogram, it is possible to approximately recapture the propagated distribution from the samples (see Figure 2.15).

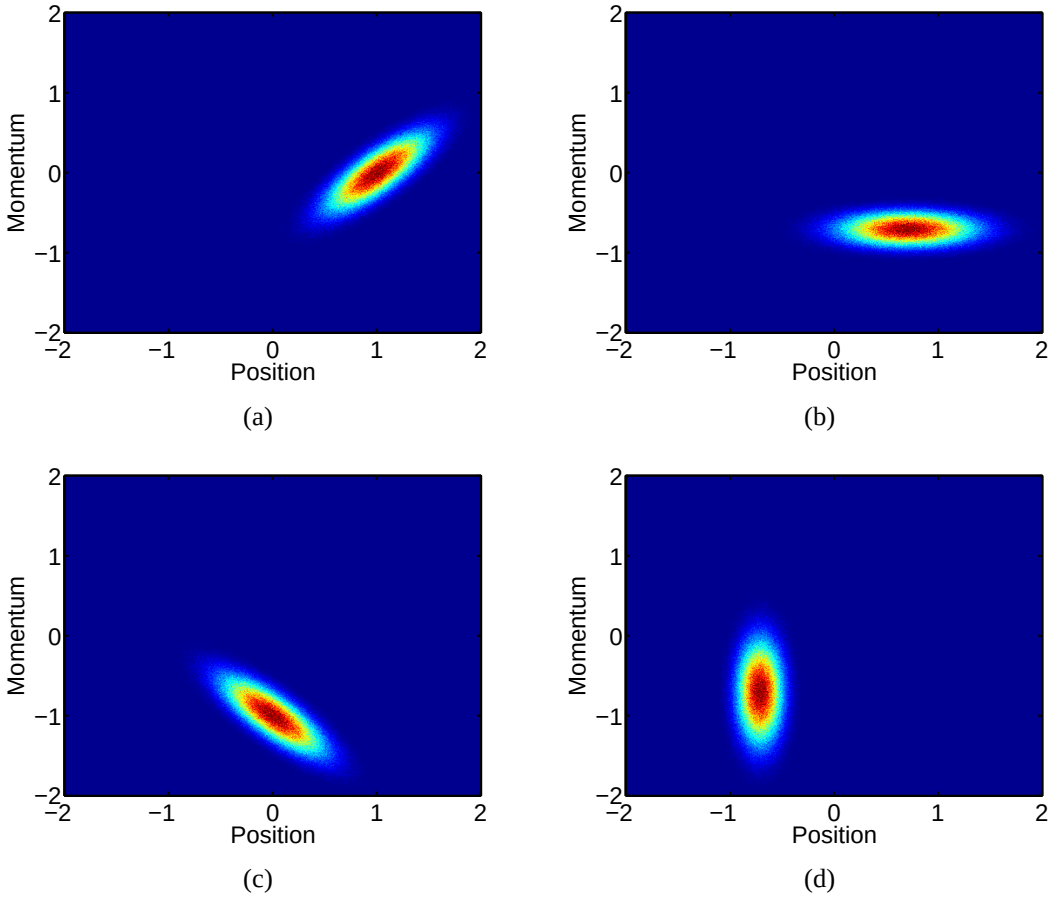


Figure 2.15: Propagation of uncertainty through a Monte Carlo simulation. The plots show momentum vs. displacement from equilibrium. They evolve over time, starting with the initial conditions in (a), $t = T/8$ in (b), $t = T/4$ in (c), and $t = 3T/8$ in (d), where T is the period of the system. The color scale is identical to that in Figure 2.11. The graininess is due to the Monte Carlo sampling approximation but the sampled distribution representations should converge to the true distributions as the number of samples approaches infinity.

2.9 Using Polynomial Chaos for Propagating Probabilistic Initial Conditions in a Simple Linear System

This section applies Polynomial Chaos to the same probabilistic initial value problem described in Section 2.8. This example is intended to help introduce the method and allow the reader to draw connections to the other presented methods.

Starting with Equation (2.37), which is repeated below,

$$\dot{x} = \begin{bmatrix} \dot{z} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{bmatrix} \begin{bmatrix} z \\ \rho \end{bmatrix},$$

it is again desired to make the initial conditions probabilistic. In the Polynomial Chaos approach, the initial conditions are explicitly made to be a function of two standard normal independent random variables: ξ_1 and ξ_2. From the discussion on Monte Carlo in the previous section, it is possible to write the initial conditions as

$$x_0(\xi_1, \xi_2) = \text{chol}(P) \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \overline{x_0}. \tag{2.45}$$

Additionally, in this approach, the states are then made to be a function of the same random variables. Thus, the states, explicitly as a function of time and the random variables, can be written as

$$x(\xi_1, \xi_2, t) \equiv \begin{bmatrix} z(\xi_1, \xi_2, t) \\ \rho(\xi_1, \xi_2, t) \end{bmatrix}. \tag{2.46}$$

Equation (2.37) is now

$$\dot{x}(\xi_1, \xi_2, t) \equiv \begin{bmatrix} \dot{z}(\xi_1, \xi_2, t) \\ \dot{\rho}(\xi_1, \xi_2, t) \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{bmatrix} \begin{bmatrix} z(\xi_1, \xi_2, t) \\ \rho(\xi_1, \xi_2, t) \end{bmatrix}. \tag{2.47}$$

It is desired to use the set orthogonal with respect to the PDF. Since Gaussian distributions are being utilized, the Hermite polynomials are chosen (see Table 2.1). These polynomials are orthogonal with respect to $w(\xi) = \frac{1}{\sqrt{2\pi}} e^{-\xi^2/2}$ over all of $\mathbb{R}$. Using the recurrence relation, the first few of these, in terms of ξ_1, can be given as

$$\begin{aligned} H^0(\xi_1) &= 1 \\ H^1(\xi_1) &= \xi_1 \\ H^2(\xi_1) &= \xi_1^2 - 1 \\ H^3(\xi_1) &= \xi_1^3 - 3\xi_1 \\ H^4(\xi_1) &= \xi_1^4 - 6\xi_1^2 + 3 \\ &\vdots \end{aligned} \tag{2.48}$$

Additionally, the polynomials can be written as functions of ξ_2. Each of these series individually forms a basis for a dimension in $\mathbb{R}$ that represents the probabilistic nature. However, it is desired to span $\mathbb{R}^2$ so that interactions between ξ_1 and ξ_2 can be represented. This concept was discussed in Section 2.7. Using the Askey scheme, the first few multi-dimensional orthogonal polynomials

can be defined as:

$$\begin{aligned}
\phi^0(\xi_1,\xi_2) &\equiv H^0(\xi_1)H^0(\xi_2) = 1 \\
\phi^1(\xi_1,\xi_2) &\equiv H^1(\xi_1)H^0(\xi_2) = \xi_1 \\
\phi^2(\xi_1,\xi_2) &\equiv H^0(\xi_1)H^1(\xi_2) = \xi_2 \\
\phi^3(\xi_2,\xi_2) &\equiv H^2(\xi_1)H^0(\xi_2) = \xi_1^2 - 1 \\
\phi^4(\xi_1,\xi_2) &\equiv H^1(\xi_1)H^1(\xi_2) = \xi_1\xi_2 \\
\phi^5(\xi_1,\xi_2) &\equiv H^0(\xi_1)H^2(\xi_2) = \xi_2^2 - 1 \\
&\vdots
\end{aligned} \tag{2.49}$$

It is now possible to decompose each of the states as a series of the orthogonal basis functions:

$$x(\xi_1,\xi_2,t) \equiv \begin{bmatrix} z(\xi_1,\xi_2,t) \\ \rho(\xi_1,\xi_2,t) \end{bmatrix} = \sum_{i=0}^{\infty} \hat{x}^i(t)\phi^i(\xi_1,\xi_2) = \sum_{i=0}^{\infty} \begin{bmatrix} \hat{z}^i(t) \\ \hat{\rho}^i(t) \end{bmatrix} \phi^i(\xi_1,\xi_2). \tag{2.50}$$

Substituting the expansions into Equation (2.47),

$$\begin{aligned}
\sum_{i=0}^{\infty} \dot{\hat{x}}^i(t)\phi^i(\xi_1,\xi_2) &= \begin{bmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{bmatrix} \sum_{i=0}^{\infty} \hat{x}^i(t)\phi^i(\xi_1,\xi_2) \\
&= \begin{bmatrix} \sum_{i=0}^{\infty} \dot{\hat{z}}^i(t)\phi^i(\xi_1,\xi_2) \\ \sum_{i=0}^{\infty} \dot{\hat{\rho}}^i(t)\phi^i(\xi_1,\xi_2) \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{bmatrix} \begin{bmatrix} \sum_{i=0}^{\infty} \hat{z}^i(t)\phi^i\xi_1,\xi_2) \\ \sum_{i=0}^{\infty} \hat{\rho}^i(t)\phi^i(\xi_1,\xi_2) \end{bmatrix}.
\end{aligned} \tag{2.51}$$

Next, Galerkin projections can be performed onto each of the multi-dimensional orthogonal basis functions, starting with the ones defined in Equation (2.49). For the zeroth projection, both sides of the first equality (the second set is obviously equivalent) in Equation (2.51) are multiplied by

$\phi^0(\xi_1, \xi_2)$ and then the inner product is taken:

$$\begin{aligned}&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_{i=0}^{\infty}\dot{\hat{x}}^i(t)\phi^i(\xi_1,\xi_2)\phi^0(\xi_1,\xi_2)w(\xi_1)w(\xi_2)d\xi_1 d\xi_2\\&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\begin{bmatrix}0 & \frac{1}{m}\\ -k & 0\end{bmatrix}\sum_{i=0}^{\infty}\hat{x}^i(t)\phi^i(\xi_1,\xi_2)\phi^0(\xi_1,\xi_2)w(\xi_1)w(\xi_2)d\xi_1 d\xi_2.\end{aligned} \tag{2.52}$$

Because of orthogonality, it is known a priori that the products resulting from every term in the summation—except the one with $\left(\phi^0(\xi_1, \xi_2)\right)^2$—will be zero. Thus, the zeroth projection produces the following equation:

$$\begin{aligned}&\dot{\hat{x}}^0(t)\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\phi^0(\xi_1,\xi_2)\right)^2 w(\xi_1)w(\xi_2)d\xi_1 d\xi_2\right]\\&=\begin{bmatrix}0 & \frac{1}{m}\\ -k & 0\end{bmatrix}\hat{x}^0(t)\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\phi^0(\xi_1,\xi_2)\right)^2 w(\xi_1)w(\xi_2)d\xi_1 d\xi_2\right].\end{aligned} \tag{2.53}$$

It is evident that the integrals in the brackets are non-zero and equal. Thus,

$$\dot{\hat{x}}^0(t) = \begin{bmatrix}0 & \frac{1}{m}\\ -k & 0\end{bmatrix}\hat{x}^0(t). \tag{2.54}$$

An equivalent procedure can be performed for all of the remaining projections. Therefore, the following set of equations can be used to see the expansions' coefficients in Equation (2.50) evolve

over time:

$$\begin{aligned}
\dot{\hat{x}}^0(t) &= \begin{bmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{bmatrix} \hat{x}^0(t) \\
\dot{\hat{x}}^1(t) &= \begin{bmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{bmatrix} \hat{x}^1(t) \\
\dot{\hat{x}}^2(t) &= \begin{bmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{bmatrix} \hat{x}^2(t) \\
\dot{\hat{x}}^3(t) &= \begin{bmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{bmatrix} \hat{x}^3(t) \\
&\vdots
\end{aligned} \tag{2.55}$$

Each one of these equations begins with the initial conditions defined in Equation (2.45).

Often at this point, it might be interesting to truncate the expansion at a finite number of terms. However, this is a special problem. Since the initial conditions only involve terms containing ξ_1, ξ_2, and deterministic components, which are fully spanned by $\phi^0(\xi_1, \xi_2)$, $\phi^1(\xi_1, \xi_2)$, and $\phi^2(\xi_1, \xi_2)$, only $\hat{x}^0(t)$, $\hat{x}^1(t)$, and $\hat{x}^2(t)$ will have non-zero initial conditions. Thus, all of the equations with an order of three or higher will remain zero. Further, the matrix exponential, $\Phi(t, t_0)$, for equations of this form has already been found (see Equation (2.38)). Thus, the complete system can be expressed in terms of the matrix exponential as:

$$\begin{aligned}
\hat{x}^0(t) &= \Phi(t, t_0)\hat{x}^0(0) \\
\begin{bmatrix} \hat{x}^1(t) & \hat{x}^2(t) \end{bmatrix} &= \Phi(t, t_0) \begin{bmatrix} \hat{x}^1(0) & \hat{x}^2(0) \end{bmatrix}.
\end{aligned} \tag{2.56}$$

Again returning to Equation (2.45),

$$\begin{aligned} \hat{x}^0(0) &= \overline{x_0} \\ \left[\begin{array}{cc} \hat{x}^1(0) & \hat{x}^2(0) \end{array}\right] &= \text{chol}(P). \end{aligned} \tag{2.57}$$

Substituting these initial conditions into Equation (2.56):

$$\begin{aligned} \hat{x}^0(t) &= \Phi(t, t_0)\overline{x_0} \\ \left[\begin{array}{cc} \hat{x}^1(t) & \hat{x}^2(t) \end{array}\right] &= \Phi(t, t_0)\text{chol}(P). \end{aligned} \tag{2.58}$$

Next, the second equation can be post-multiplied by a transpose of itself as

$$\left[\begin{array}{cc} \hat{x}^1(t) & \hat{x}^2(t) \end{array}\right]\left[\begin{array}{cc} \hat{x}^1(t) & \hat{x}^2(t) \end{array}\right]^T = \Phi(t, t_0)P\Phi^T(t, t_0). \tag{2.59}$$

Rewriting the left-hand side,

$$\begin{aligned} \left[\begin{array}{cc} \hat{x}^1(t) & \hat{x}^2(t) \end{array}\right]\left[\begin{array}{cc} \hat{x}^1(t) & \hat{x}^2(t) \end{array}\right]^T &= \left[\begin{array}{cc} \hat{z}^1(t) & \hat{z}^2(t) \\ \hat{\rho}^1(t) & \hat{\rho}^2(t) \end{array}\right]\left[\begin{array}{cc} \hat{z}^1(t) & \hat{\rho}^1(t) \\ \hat{z}^2(t) & \hat{\rho}^2(t) \end{array}\right] \\ &= \sum_{i=1}^{2}\left[\begin{array}{cc} \left(\hat{z}^i(t)\right)^2 & \hat{z}^i(t)\hat{\rho}^i(t) \\ \hat{z}^i(t)\hat{\rho}^i(t) & \left(\hat{\rho}^i(t)\right)^2 \end{array}\right]. \end{aligned} \tag{2.60}$$

Further, note that the way the multi-dimensional Hermite polynomial set is defined gives

$$\begin{aligned} \delta_{11} &\equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\phi^1(\xi_1, \xi_2)\right)^2 w(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 = 1 \\ \delta_{22} &\equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\phi^2(\xi_1, \xi_2)\right)^2 w(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 = 1. \end{aligned} \tag{2.61}$$

Additionally, using the information that $\hat{z}^i(t)$ and $\hat{\rho}^i(t)$ are zero for all orders greater than 2, Equation (2.60) can now be rewritten as

$$\sum_{i=1}^{\infty} \left\{ \delta_{ii} \begin{bmatrix} \left(\hat{z}^i(t)\right)^2 & \hat{z}^i(t)\hat{\rho}^i(t) \\ \hat{z}^i(t)\hat{\rho}^i(t) & \left(\hat{\rho}^i(t)\right)^2 \end{bmatrix} \right\} = \Phi(t, t_0) P \Phi^T(t, t_0). \tag{2.62}$$

It is evident from Equation (2.22) that the first equation in Equation (2.58) propagates the mean through time. Also, it is evident from Equation (2.23) that Equation (2.62) propagates the covariance matrix through time. Further, these equations agree with the analytic results for the Gaussian distribution presented in Equation (2.43).

As was previously mentioned, this is a special case. The Gaussian distribution can be fully represented by its mean and covariance matrix and thus the analytic solution can be obtained through Polynomial Chaos. However, as will be seen, Polynomial Chaos applies to much more general cases where analytic solutions are not known or easily obtained. Further, as is extensively discussed in the literature, there are many situations for which Polynomial Chaos is computationally superior to Monte Carlo simulations.

2.10 Introduction to $\mathcal{H}_2$ and LQR Control

The $\mathcal{H}_2$ and Linear Quadratic Regulator (LQR) control design problems are intimately related. While most of the work will be in the time domain, since the $\mathcal{H}_2$ norm is commonly expressed in terms of the frequency domain, both domains will be discussed in parallel.

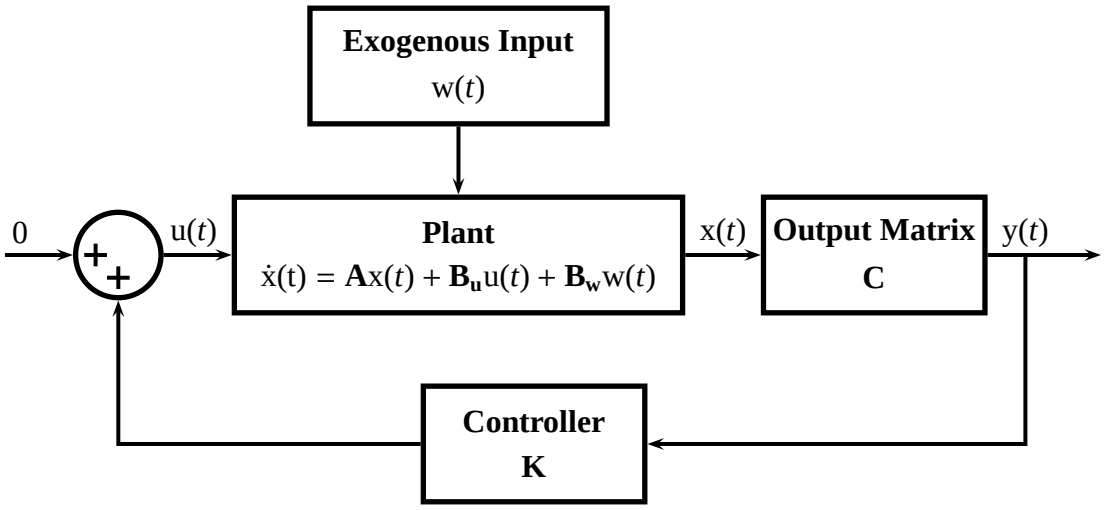


Figure 2.16: Linear control system diagram. The output vector y(t) is used for feedback. Also a convention of positive feedback is followed, thus changing signs in the matrix **K** when compared to the convention of negative feedback.

A block diagram for this section is defined in Figure 2.16. The linear state equations for the open loop system can be defined as

$$\begin{aligned} \dot{\mathrm{x}}(t) &= \mathbf{A}\mathrm{x}(t) + \mathbf{B_u}\mathrm{u}(t) + \mathbf{B_w}\mathrm{w}(t) \\ \mathrm{y}(t) &= \mathbf{C}\mathrm{x}(t). \end{aligned} \tag{2.63}$$

Here, x(t) is the state vector, y(t) is the output vector, u(t) is the input vector, and w(t) is an exogenous input.

In the Laplace domain, the open loop equations between inputs and outputs can be described as

$$\begin{aligned} \mathrm{X}(s) &= \mathbf{G}(s)\mathbf{B_w}\mathrm{W}(s) \\ \mathrm{X}(s) &= \mathbf{G}(s)\mathbf{B_u}\mathrm{U}(s) \\ \mathrm{Y}(s) &= \mathbf{C}\mathrm{X}(s), \end{aligned} \tag{2.64}$$

where $\mathbf{G}(s) \equiv (s\mathbf{I} - \mathbf{A})^{-1}$ is a matrix of transfer functions. The inverse Laplace transform of $\mathbf{G}(s)$ is the matrix exponential:

$$\mathbf{G}(t) = \mathcal{L}^{-1}\{\mathbf{G}(s)\} = e^{\mathbf{A}t}. \tag{2.65}$$

For the regulator problem, the feedback control law is given to be of the form

$$\mathrm{u}(t) = \mathbf{K}\mathrm{y}(t) = \mathbf{KC}\mathrm{x}(t). \tag{2.66}$$

Note that for now $\mathbf{C}$ will be considered to be square and non-singular. This makes it so that full state information can be assumed to exist, just in a transformed manner.

In closed loop form, this system can be written as

$$\begin{bmatrix} \dot{\mathrm{x}}(t) \\ \mathrm{y}(t) \\ \mathrm{u}(t) \\ \mathcal{X}(t) \\ \mathcal{U}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A_{cl}} & \mathbf{B_w} \\ \mathbf{C} & \mathbf{0} \\ \mathbf{KC} & \mathbf{0} \\ (\mathbf{Q})^{1/2} & \mathbf{0} \\ (\mathbf{R})^{1/2}\mathbf{KC} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathrm{x}(t) \\ \mathrm{w}(t) \end{bmatrix}, \tag{2.67}$$

where $\mathbf{A_{cl}} \equiv \mathbf{A} + \mathbf{B_u}\mathbf{KC}$. Also, the matrix square root is defined such that $(\mathbf{Q})^{1/2}(\mathbf{Q})^{1/2} = \mathbf{Q}$ and similarly for $\mathbf{R}$. Since the utilized matrix is symmetric, the matrix square root should be made a symmetric matrix. Also, the matrices are either positive semi-definite or positive definite and thus have a real matrix square root.

Equivalently, in the Laplace domain, the closed transfer functions are

$$\begin{bmatrix} \mathrm{x}(s) \\ \mathrm{y}(s) \\ \mathrm{u}(s) \\ \mathcal{X}(s) \\ \mathcal{U}(s) \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{C} \\ \mathbf{KC} \\ (\mathbf{Q})^{1/2} \\ (\mathbf{R})^{1/2}\,\mathbf{KC} \end{bmatrix} \begin{bmatrix} \mathbf{G_{cl}}(s)\mathbf{B_w}w(s) \end{bmatrix}, \tag{2.68}$$

where $\mathbf{G_{cl}}(s) \equiv (s\mathbf{I} - \mathbf{A_{cl}})^{-1}$.

The $\mathcal{H}_2$ control problem relates to finding a gain matrix that minimize the $\mathcal{H}_2$ norm. This norm, of a matrix of transfer functions here arbitrarily denoted $\mathbf{H}(s)$, in the frequency domain is defined as

$$\|\mathcal{H}\|_2^2 \equiv \frac{1}{2\pi}\mathrm{trace}\int_{-\infty}^{\infty} \mathbf{H}\,(j\omega)\,\mathbf{H}^H\,(j\omega)\;d\omega \tag{2.69}$$

The stated expression deserves comment before proceeding. First, thinking about $\mathbf{H}(s)$ as the special case of a single transfer function rather than the general case of a matrix of transfer functions, $\mathbf{H}\,(j\omega)\,\mathbf{H}^H\,(j\omega)$, evaluated at a single frequency $\omega = \Omega$ gives the square squared amplitude of the sinusoidal output from $\mathbf{H}(s)$ to a unit amplitude sinusoidal input with frequency Ω. The square of the amplitude relates to a power spectral density. Thus, the square of the $\mathcal{H}_2$ norm is an integral of the power spectral density across all frequencies. Generalizing to letting $\mathbf{H}(s)$ be a matrix of transfer functions, the trace operator in the expression for the norm makes it such that the norm is the sum of integrals of power spectral density from each matrix element of $\mathbf{H}(s)$.

Proceeding, let

$$\mathbf{H}(s) = \begin{bmatrix} \mathcal{X}(s) \\ \mathcal{U}(s) \end{bmatrix} = \begin{bmatrix} (\mathbf{Q})^{1/2} \\ (\mathbf{R})^{1/2}\,\mathbf{KC} \end{bmatrix} \begin{bmatrix} \mathbf{G_{cl}}(s)\mathbf{B_w} \end{bmatrix}. \tag{2.70}$$

Apply the $\mathcal{H}_2$ norm defined in Equation (2.69) and use the property trace [**AB**] = trace [**BA**]:

$$\begin{aligned}\|\mathcal{H}\|_2^2 =& \frac{1}{2\pi}\text{trace}\int_{-\infty}^{\infty}\left(\begin{bmatrix}(\mathbf{Q})^{1/2}\\(\mathbf{R})^{1/2}\mathbf{KC}\end{bmatrix}\begin{bmatrix}\mathbf{G_{cl}}(j\omega)\mathbf{B_w}\end{bmatrix}\right)^H\begin{bmatrix}(\mathbf{Q})^{1/2}\\(\mathbf{R})^{1/2}\mathbf{KC}\end{bmatrix}\begin{bmatrix}\mathbf{G_{cl}}(j\omega)\mathbf{B_w}\end{bmatrix}d\omega\\ =& \frac{1}{2\pi}\text{trace}\int_{-\infty}^{\infty}\begin{bmatrix}\mathbf{G_{cl}}(j\omega)\mathbf{B_w}\end{bmatrix}^H\left[\mathbf{Q}+(\mathbf{KC})^T\mathbf{R}(\mathbf{KC})\right]\begin{bmatrix}\mathbf{G_{cl}}(j\omega)\mathbf{B_w}\end{bmatrix}d\omega.\end{aligned} \tag{2.71}$$

Using Parseval's Theorem and then substituting in $e^{\mathbf{A_{cl}}t}$:

$$\begin{aligned}\|\mathcal{H}\|_2^2 =& \text{trace}\int_0^{\infty}\begin{bmatrix}\mathbf{G_{cl}}(t)\mathbf{B_w}\end{bmatrix}^T\left[\mathbf{Q}+(\mathbf{KC})^T\mathbf{R}(\mathbf{KC})\right]\begin{bmatrix}\mathbf{G_{cl}}(t)\mathbf{B_w}\end{bmatrix}dt\\ =& \text{trace}\,\mathbf{B_w}^T\left[\int_0^{\infty}e^{\mathbf{A_{cl}}^T t}\left[\mathbf{Q}+(\mathbf{KC})^T\mathbf{R}(\mathbf{KC})\right]e^{\mathbf{A_{cl}}t}\,dt\right]\mathbf{B_w}.\end{aligned} \tag{2.72}$$

At this point, it is useful to more formally talk about what $e^{\mathbf{A_{cl}}t}$ means in this context. To do this, let $\delta_i(t)$ denote an unit impulse in the i-th channel. For example,

$$\delta_2(t) = \begin{bmatrix}0 & \delta(t) & 0 & 0 & \dots\end{bmatrix}^T. \tag{2.73}$$

Additionally, the Laplace transform of this example is

$$\mathcal{L}\{\delta_2(t)\} = 1_2 = \begin{bmatrix}0 & 1 & 0 & 0 & \dots\end{bmatrix}^T. \tag{2.74}$$

Thus, it can be seen that by inputting unit impulses into w(t), the i-th impulse response, $\mathrm{x}_i(t)$ is given by

$$\mathrm{x}_i(t) = \mathcal{L}^{-1}\{\mathbf{G}(s)\mathbf{B_w}\mathcal{L}\{\delta_i(t)\}\} = e^{\mathbf{A_{cl}}t}\mathbf{B_w}1_i. \tag{2.75}$$

Further, it can be seen that the individual impulse responses can be summed and formulated as the following result:

$$\begin{bmatrix} \mathrm{x}_1(t) & \mathrm{x}_2(t) & \dots & \mathrm{x}_Q(t) \end{bmatrix} = \sum_{i=1}^{\mathrm{Q}} \mathcal{L}^{-1}\left\{\mathbf{G}(s)\mathbf{B_w}\mathcal{L}\left\{\delta_i(t)\right\}\right\} 1_i^T = \sum_{i=1}^{\mathrm{Q}} e^{\mathbf{A_{cl}}t}\mathbf{B_w}1_i 1_i^T = e^{\mathbf{A_{cl}}t}\mathbf{B_w}, \tag{2.76}$$

where 'Q' here denotes the number of columns in $\mathbf{B_w}$. Substituting the above results into Equation (2.72) and allowing $\mathbf{B_w}$ and $\mathbf{C}$ to be appropriately dimensioned identity matrices:

$$\|\mathcal{H}\|_2^2 = \text{trace}\left[\int_0^\infty \begin{bmatrix} \mathrm{x}_1(t) & \mathrm{x}_2(t) & \dots & \mathrm{x}_Q(t) \end{bmatrix}^T \left[\mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K}\right] \begin{bmatrix} \mathrm{x}_1(t) & \mathrm{x}_2(t) & \dots & \mathrm{x}_Q(t) \end{bmatrix} dt\right]. \tag{2.77}$$

Next, the trace is converted to a summation and then $\mathrm{u}_i(t) = \mathbf{K}\mathrm{x}_i(t)$ is used:

$$\begin{aligned} \mathrm{J} = \|\mathcal{H}\|_2^2 &= \sum_{i=1}^{\mathrm{Q}} \int_0^\infty \mathrm{x}_i^T(t)\left[\mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K}\right]\mathrm{x}_i(t)\,dt \\ &= \sum_{i=1}^{\mathrm{Q}} \int_0^\infty \mathrm{x}_i^T(t)\mathbf{Q}\mathrm{x}_i(t) + \mathrm{u}_i^T(t)\mathbf{R}\mathrm{u}_i(t)\,dt. \end{aligned} \tag{2.78}$$

Note that this is the form of the infinite horizon LQR problem and thus is denoted by J. However, in practice, the LQR problem is generally written as

$$\mathrm{J} = \int_0^\infty \mathrm{x}^T(t)\mathbf{Q}\mathrm{x}(t) + \mathrm{u}^T(t)\mathbf{R}\mathrm{u}(t)\,dt. \tag{2.79}$$

Thus, the connection can be seen between the two—the infinite horizon LQR problem is a subset of the $\mathcal{H}_2$ problem. Also, it can be noted that many programs, such as Matlab, utilize Riccati equations to solve both the $\mathcal{H}_2$ optimization and LQR problems.

Chapter 3

Literature Review

The research involved in this work consists of three main theoretical components: trajectory sensitivity methods, probabilistic based control, and the numerical method of Polynomial Chaos. The trajectory sensitivity methods in many ways most closely resemble the developed method in character. They involve augmenting the system with additional states and using them in the control design. However, this method differs because it is based on local approximations. The probabilistic methods in the next section overlap with the current research from a different perspective—there is an attempt to look at a set of systems, such as through using a finite set of systems. Finally, since there currently is not a strong link between the Polynomial Chaos and the first two control sections in the literature, an in-depth literature review on developing applications is given.

3.1 Trajectory Sensitivity Methods

From 1982 to 1990, Wagie and Skelton [11]; Yedavalli and Skelton [12]; and Okada and Skelton [13] published the cited papers relating to Trajectory Sensitivity Optimization. This concept works

by weighting nominal trajectories and sensitivity trajectories. These trajectories are given by a state space system, given in Equation (4.33). Finding the controller that minimizes the norm created by the trajectory weightings requires an iterative procedure using Lyapunov equations. The method possesses a computational advantage since the system's lower triangular form allows the equations to be decoupled into more, but smaller dimensioned, Lyapunov equations.

Trajectory Weighted LQG is an alternative approach from Trajectory Sensitivity Optimization that finds a controller for the full-state solution of the system augmented with sensitivity states. For this method, it is necessary to assume that the control feedback does not vary with parameter changes. This formulation allows using a Riccati solution to determine the gains. However, it is then necessary to attempt to observe these extra sensitivity states, such as with a Kalman Filter.

A good comparison between the methods, as well as μ-synthesis, H_∞, and the sampled methods mentioned in the next section, is given by Grocott [14].

3.2 Probabilistic Control

In robust control, there are a number of beautiful results. These include concepts like Kharitonov's theorem, which allows evaluating the stability of characteristic polynomials with perturbed coefficients. In its simplest form, where independent coefficients are assumed, this can be completed by checking the stability of four characteristic polynomials. However, it is possible to expand this result to $\mathcal{D}$ stability in order to check if the perturbed poles will lie within given regions of the complex plane. Also, by the connection between characteristic polynomials and matrix representations, it is possible to obtain some results for perturbed matrices. A good overview is given by Barmish and Kang [15]. Additionally, a well-known paper on an application to controlling the steering of a bus, by analyzing bounds of the closed loop poles in the complex plane, is given by Ackermann,

Kaesbauer, and Muench [16]. Eigenvalue analysis in probabilistic form can be found in the literature as well. In a simple form, eigenvalues for a set of systems are given Gaussian distributions and then the probability of closed loop eigenvalues protruding outside a given complex-plane region is evaluated (Tse et al. [17]). Further, in Ray and Stengel [18] closed loop locations of deterministic LQG and LTR controllers are evaluated on probabilistic models in order to obtain probabilistic closed loop pole locations. This is accomplished using Monte Carlo simulations. Additionally, Huerta-Ochoa and Johnson [19] use the gradient of eigenvalues in order to approximate the variance of poles. The authors then create a cost function for minimizing eigenvalue variation in a characteristic polynomial.

Overall, there appear to be a few distinct camps of thought in the in the robust probabilistic control literature. While minimax controllers such as H_∞ are interesting, they are mostly out of the scope of the current literature search. However, it is notable to point out that a substantial number of papers reference idealistic complications with the minimax approach as a motivation for taking a probabilistic approach. For example, from Crespo and Kenny [20], it is noted that the commonly used μ-synthesis and H_∞ methods

> *are based on the most pessimistic value of performance among the possible ones, usually referred to as "worst-case." This worst-case performance is usually realized only by a single member of the uncertain model set and by a particular input signal. No information is provided regarding the likelihood that this worst-case will ever occur in practice. In addition, the intrinsic mathematical requirements of the approach usually lead to conservative models of uncertainty, over-conservative designs and complicated compensators.*

Approaches partially related to "worst case" ideals involve a "guaranteed cost" approach, where the goal in the reviewed papers is to ensure that $J < \gamma$, for all probabilistic realizations, with γ

providing an upper limit on cost. For example, in Polyak and Tempo [21], random system realizations are created. It is stated that this is a convex problem and thus Quadratic Matrix Inequalities (QMIs) can be utilized to minimize infeasibility and arrive at robust solutions. A derivative work is presented in Kanev, Schutter, and Verhaegen [22], where a Linear Matrix Innequality (LMI) approach is used to successively narrow an ellipsoid feasibility region by successively creating new constraints from new samples of the probabilistic system.

A simple initial concept is given by Djavdan, Tulleken, Voetter, Verbruggen, and Olsder [23], which looks at robust tuning of controllers using distinct discrete "sets" of systems. In this work, the concept presented relates to minimizing a collective cost function:

$$\min_{\Psi} \sum_{i=1}^{m} \frac{1}{j_i} \left(J_i\left(\Psi, \hat{\theta}\right) - j_i + \left| J_i\left(\Psi, \hat{\theta}\right) - j_i \right| \right)^2,$$

where j_i denotes thresholds for individual cost functions for individual measures of performance.

One of the prevalent ideas in several works involves a reasonable approach to extending parametric uncertainty to LQR, LQG, and $\mathcal{H}_2$ concepts. The main idea here involves minimizing the expectation of a quadratic cost function relating systems with continuous parameter uncertainty distributions. For example, although this work only refers to feed-forward, Sternad and Ahlén [24] advocates minimizing $\overline{E}\left(E\,|z(t)|^2\right)$. Here $\overline{E}$ is the expectation with respect to the probabilistic model and E is the expectation with respect to the noisy inputs. Another work published within a close time interval, Öhrn et al. [25], also looks to minimize the cost defined by the mean due to parameter variation of the mean square cost resulting from noise into the system for open loop design. This approach uses spectral factorization and a Diophantine equation. Similarly, Apley [26] uses a "cautious minimum variance" controller by combining a concept of minimizing mean of a quadratic cost function with respect to model error, combined with a Bayesian approach to estimating uncertainty. This idea also appears in Boers, Weiland, and Damen [27], where the

authors take Monte Carlo sample realizations of a system and then optimize to minimize the average $\mathcal{H}_2$ norm. Further, there are a few works that advocate combining the variance of the resulting scalar from the cost function. One example is Tsumura [28]. A second example is Tenne and Singh [29, 30], where an unscented transform is utilized to approximate the mean and variance of the quadratic cost function. The unscented transform works through providing a set of weighted sampling points and a method to approximate the resulting mean and variance of the samples transformed by a function. This is then used to minimize $J(x, p, u) = m_J + \sigma_J$.

Another major concept in probabilistic control has been largely developed by Stengel and his colleagues. This approach involves utilizing large numbers of Monte Carlo simulations of closed loop probabilistic systems in order to infer properties. In general, this involves the concept of the probability meeting design goals, such as: percent overshoot, the actuator remaining within a set amplitude, and the settling time. In Marrison and Stengel [31] and Stengel and Wang [32] a genetic algorithm for optimizing controller parameters is discussed in detail, including with regard to selection, crossing-over, and mutations. The authors also apply a statistical criteria for estimating the variance created by the number of samples in order to better know the number of samples needed for each individual parameter vector of the mutation. This is done by recognizing that meeting or failing to meet a required level is a binomial distribution—there is a fixed and unknown probability that it is either met or not for each individual random Monte Carlo sample.

A relatively recent set of work has been produced by Crespo and Kenny. This approach starts with Crespo [33] looking at cost functions for instability, variability about a desired frequency response function, and probability of lying within a region of failure. It is proposed to analytically give probabilistic frequency response values. It also states that it is possible to optimize gains for minimizing given criteria. The work in Crespo and Kenny [20, 34] are notably differentiated from the work by Stengel and colleagues in that deterministic Hamersly sampling sequences are proposed in place of Monte Carlo sampling to reduce the number of samples needed. Additionally,

a hybrid approach, utilizing a First Order Reliability Method (FORM), is used to approximate a transformation to a Gaussian distribution in order to establish approximate confidence intervals for failures. Further, an assortment of metrics are presented, including conservative bounds on meeting reliability bounds. These are constructed from the Tchebyshev inequality, using mean and variance of the arbitrary metric. In these works, the overriding theme involves optimizing controllers to minimize the probability of a system encroaching into defined frequency- and time-domain regions of "failure."

3.3 Polynomial Chaos

The generally agreed upon origin of Polynomial Chaos is from Wiener [35] in 1938. Additionally, Cameron and Martin [36] laid fundamental groundwork by demonstrating convergence for Hermite-Fourier expansions in $L_2(C)$ for nonlinear equations over a finite Fourier domain. More recently, it is well-cited throughout the literature that early works by Ghanem and Spanos, such as [37] in 1990 and [38] in 1991, pioneered the introduction of Polynomial Chaos into engineering applications for studying uncertainty. Early applications of Polynomial Chaos were very heavily focused on mechanics and fluid problems. Another very influential paper, by Xiu and Karnadakis [2] in 2002, laid out a very general approach to representing uncertainty with Polynomial Chaos. This has resulted in a seemingly exponential rate of works utilizing Polynomial Chaos for evaluating uncertainty.

In Table 3.2, a substantial number of works are reviewed in order to help the reader capture and appreciate some of the excitement that has paralleled this rapid expansion of the method. Emphasis is placed on noting developments in the use of Polynomial Chaos, as well as specific applications proposed in the works. Preliminary concepts for control applications began appearing in 2004. Sandu et al. [39] proposed using PCEs in order to represent many possible gain values. Simula-

tions were performed on the Polynomial Chaos system, and the cost function was evaluated for realizations, thereby giving an efficient way to solve a classical optimal control problem. Monti, Ponci, and Lovett [40] assume that an "optimal feedforward" trajectory could be found and then used Polynomial Chaos to simulate responses of a system with the feedforward and parametric uncertainties. This group moved towards proposing observers to attempt to estimate modes resulting from Polynomial Chaos Expansions of linear models with uncertainty [41, 42]. Additionally, Sarma, Durlofsky, and Aziz [43] produced a conference proceeding in 2005 on using the expectations of Hermite polynomials for stochastic fields in the forward equations for control based on Bayesian inversion theory.

Hover and Triantafyllou investigated the application of Polynomial Chaos for examining stability of non-linear systems with unknown parameters through simulation [44]. In this work (noted as recieved Oct. 29, 2004), it was claimed that "[t]o our knowledge, this is the first time that Polynomial Chaos methods are applied to control problems." Additionally, Hover has investigated Polynomial Chaos' use for efficiently determining a set of optimal trajectories for a set of non-linear systems [45].

Recently, in June 2008, Fisher and Bhattacharya released two conference proceedings examining the use of Polynomial Chaos in linear systems [3, 4] that closely parallel the presented work (see discussion in the Introduction for differentiation). They have the first known publication of a result relating to a general Kronecker product form for the Galerkin projections of a state space system. Feasibility conditions for the parametrically robust LQR problem in the form of Bilinear Matrix Inequalities (BMIs) utilizing expanded matrices are given in [3]. Additionally, expanded forms are substituted into Riccati equations (without proof of appropriateness). Also, it is proposed that Layapunov equations can be used to evaluate stability of the expanded system, as well as eigenvalues of the expanded system are taken.

In addition to the control results discussed, there are results relating to the simulation and analysis of uncertain systems in Table 3.2, Also, the eigenvalue problem has been explored by multiple authors. These works include a major implied assumption, typically stated in forms similar to "eigenvalues must be well-spaced," since bifurcations in uncertain systems moving from a pair of real roots to a pair of complex roots creates a corner in the root locus that is not well represented by polynomial basis functions.

A key to symbols for categories for this table reviewing works is given in Table 3.1.

Table 3.1: Key for category symbols in Table 3.2.

Symbol	Represented Category
♠	Acoustic
♣	Biology & Medicine
‡	Chemical
◆	Circuit Modeling, Design, & Manufacturing
★	Control Design & Analysis
⋈	Discontinuous Stochastic Domains
◯	Dynamics (Mechanical)
×	Eigenvalue Problem
℧	Electrostatics & Electrodynamics
⊔	Filtering
□	Finite Element Method, Stochastic (SFEM)
■	Fluid & Flow Modeling
ℵ	Heat Transfer Modeling
√	Identification of Parameters
⊕	Multiscale Modeling
✠	Observers
∇	Optimization
¶	Software
♥	Theoretical Polynomial Chaos & Statistical Developments
▲	Vibrations

Table 3.2: Summary table of published Polynomial Chaos literature, organized by publication year and subject area. A key for subject areas is given in Table 3.1.

Subj.	Authors	Notes and Contributions
1999		
♥	Ghanem [46]	The presented work looks at representing log-normal stochastic processes. This goal is accomplished through the exponential relationship between the normal distribution and the log-normal distribution, and then utilizing Hermite polynomials in PCEs. The method is developed in parallel with a Karhunen-Loève decomposition approach.
2001		
¶	Wojtkiewicz, Eldred, Filed, Urbina, and Red-Horse [47]	The main purpose of this work is to introduce a numerical package under development with applications to uncertainty: DAKOTA (Design Analysis Kit for OpTimizAtion). Additionally, it is mentioned that "[w]hile efforts investigating non-probabilistic approaches to uncertainty quantification are being undertaken at Sandia, the only well-established methodology of propagating uncertainty through these models, at present, is the traditional probabilistic or statistical approach."

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
2002		
♥	Xiu and Karniadakis [2]	This classic paper begins by reviewing a result from the Cameron-Martin theorem that demonstrates that Hermite-chaos will converge to all distributions with finite second moments. Importantly, the Wiener-Askey scheme is given as a generalization of the Wiener PC. Additionally, an example of a stochastic exponential decay is presented; with $dy/dt = -ky$, such that k is a random variable, thereby creating a random process. Plots of the error in variance and means, as a function of the number of terms in an expansion are given for: 1) the Gaussian distribution and Hermite-chaos, 2) the Gamma distribution and Laguerre-chaos, 3) the Beta distribution and Jacobi-chaos, 4) the Poisson distribution and Charlier-chaos, 5) the Binomial distribution and Krawtchouk-chaos, 6) the Negative binomial distribution and Meixner-chaos, and 7) the Hypergeometric distribution and Hahn-chaos. Further, a discussion is given in regards to how to expand arbitrary distributions in terms of a certain type of random variable. Additionally, since the random variables are treated as expansions in stochastic dimensions, it is demonstrated that it is possible to have *stochastic Gibb's phenomena* for non-smooth stochastic functions. At the end, an appendix lists a collection of orthogonal polynomials for PC.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
2004		
♥	Debusschere, Najm, Pébay, Knio, Ghanem, and Le Maître [48]	This work gives an iterative solution to solve innerproducts of basis function terms multiplied by each other. It also looks at a Taylor series expansion approach, a novel integral approach, and a quadrature/sampling approach/pseudo-spectral approach for evaluating nonlinear functions using the Galerkin method. Numerical studies on these approaches are performed.
♥	Field and Grigoriu [49]	Numerical examples of convergence for non-linear maps of random Gaussian variables are used in a PC context to create log-normal, uniform, and reflected Gaussian distributions. Additionally, less-than desired convergence for functions with non-continuous first derivatives is demonstrated.
♥	Ghanem and Red-Horse [50]	A generalization of the Karhunen-Loève expansions to processes in Sobolev spaces is presented. This is proposed to be of interest for situations where solutions are "constrained with regards to smoothness and differentiability." The approach is briefly related to PC.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
∇	Giunta, Eldred, Swiler, Trucano, and Wojtkiewicz [51]	Philosophical arguments for stochastic optimal design are presented in this paper. One of the utilized design methods employs PC. The argued method relates to using statistical measures, including minimizing $E[f(d,u)]$ and minimizing Prob[f(d, u) > f*]. Some interesting quotes from the work include: "[o]ne of the criticisms of the traditional engineering design optimization process is that it often produces optimal designs that are not robust to perturbations in the design parameters or other uncertain parameters (e.g. , manufacturing tolerances, atmospheric conditions, etc);" "[t]hus it is a natural extension for engineers to incorporate statistical measures directly into the design optimization process;" and "[t]his has led to areas of study known as robust design, reliability-based design optimization, and optimization under uncertainty, among others." Also, the intrinsic large computational cost for optimizing statistical properties of high fidelity models is discussed.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
	Huyse and Thacker [52]	It is pointed out that arbitrary selection of distributions when estimating small failure probabilities, often resulting from tail probabilities, is problematic. An example that is intended to be paradoxical is given regarding different building codes mandating different distributions be assumed for the strength of concrete. Thus, it is argued that the reliability community is "ill-serve[d]" by distributions selected without adequate fitting to the tails. It is proposed that general classes of distributions, including the Exponential-Power Distributions, Pearson Family Distribution System, Johnson Transformation (transformations of the Normal distribution), and Gram-Charlier Expansions (PC is under this class) are used to help resolve this problem. Further, methods for obtaining the PDF from samples, including Moment and Quantile Estimation, (Weighted) Least Squares Estimation of the CDF, Maximum Likelihood Estimation (MLE), and Bayesian Estimation are given.
■∇	Isukapalli, Balakrishnan, and Georgopoulos [53]	It is proposed to combine using PC for the Stochastic Response Surface Method (SRSM) with information from partial derivative approximations to improve the method. It also proposes that the SRSM can be used in the inverse problem to determine how to reduce uncertainties. Additionally, there are brief mentions of applications in atmospheric and ground flows.
⋈■	Le Maître, Knio, Najm, and Ghanem [54]	A scheme for using the Wiener-Haar wavelets with Polynomial Chaos is presented. It is demonstrated that it has some good properties for in regards to representing discontinuities in the stochastic dimensions. This approach is applied to examples of the Rayleigh-Bernard fluid convection problem with uncertain temperature boundary conditions.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
⋈▲	Millman, King, Maple, and Beran [55]	This work uses PC to analyze limit cycles, which are very sensitive to parametric uncertainty. It looks to find a probabilistic response. In the introduction, Fourier Chaos Expansions are mentioned for periodicity reasons. The non-linear system model used relates to air flow over wing section and structural dynamics. Bifurcations are present. In order to try to work around the difficulties of discontinuities, B-splines are utilized as the PC basis functions. The results are compared to MC simulations with reasonably good consistency in regards to the shape of the produced bi-modal PDFs.
◆★	Monti, Ponci, and Lovett [40]	The authors look at feed-forward optimal control and try to minimize $J = x^T(\infty)Qx(\infty) + u^T(\infty)Ru(\infty)$. They refer to weighting enlarged Q and R matrices. They state that it is possible to define the Q matrix "by assigning significantly higher values to certain coefficients," in order to reduce the uncertainty level. Example applied to the control design for a DC/DC converter. Experimental results for a realization are given.
▲	Sandu, Sandu, Chan, and Ahmadian [56]	The use of PC is explored for determining the effects of uncertainty on Differential Algebraic Equations (DAEs). The application of interest applies to multibody dynamics, including vehicle/terrain interface. Stochastic terrain is represented through Karhunen-Loève expansions. Also, it is pointed out that finite support distributions, such as the beta and uniform distributions are the most applicable. Collocation and Galerkin methods are both discussed. Additionally, it is explained that it is possible to obtain the power spectral density by taking the FFT along the time dimension of the evolving PC coefficients.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
★	Sandu, Sandu, Chan, and Ahmadian [39]	Optimal control problems are analyzed by using PCEs to represent controller gains. An optimization is then performed to find the value of the expansion variables in the PCE that minimize an integral cost function of the states.
◆II	Wu, Zhu, and Najm [57]	A concept relating to decomposing inputs for ASICs and processors based on the Karhunen Loève expansion is presented. Polynomial Chaos is used to propagate the stochastic input throughout performed chip operations. The intent is to be able better optimize design for more efficient bitwidths such as to reduce circuit size, delay, and energy consumption. The design process is proposed for non-linear filters. The concepts are further developed in [58, 59].
■	Yang, Zhang, and Lu [60]	This work creates a stochastic flow model by stochastically representing soil parameters as well as the pressure head with a combination of PCEs and the Karhunen-Loève expansion.
2005		
◆♣	Geneser, Choe, Kirby, and MacLeod [61]	This application based paper looks at using the collocation approach to study statistical electrical propagation in a finite element model of the human body. It seeks to better model the forward conductivity problem such that the inverse problem used by electrocardiograms of determining cardiac function from surface electrical measurements can be better understood.
◆♣	Geneser, Kirby, and Sachse [62]	Three cardiac models, the FitzHugh-Nagumo model for cellular action potentials, one for sarcoplasmic ryanodine calcium release, and a Slowly-Activating Delayed Rectifier Potassium Current Model are analyzed with PC. The FitzHugh-Nagumo model involves bifurcations. Simulations of models with stochastic parameters are given.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
▲√	Ghanem, Masri, Pellissetti, and Wolfe [63]	This work looks at a 1-DOF system of the form $$m\ddot{x} = F(t) - f(x, \dot{x}) \quad (3.1)$$ The measured state response is normalized to within the bounds $[-1, 1]$. This allows $f(x, \dot{x}) = \sum_{i,j} C_{ij} T_i(x') T_j(\dot{x}')$, to be expanded in terms of Chebyshev polynomials. Then the coefficients for Chebyshev expansion functions are considered to be random variables represented by PC expansions. The identification problem is posed in terms of finding statistical properties of the Chebyshev coefficients.
×	Ghosh and Ghanem [64]	PC is applied to the eigenvalue problem where there are "widely spaced modes." The approach taken is to substitute expansions into eigenvalue problem matrices and perform Galerkin projections. Additionally, this approach advocates using Newton-Raphson iterations to converge on the eigenvalue expansions from an initial guess obtained by a different method.
▲	Lamarque and Gourdon [65]	Uncertainty in the parameters of non-linear oscillators for energy pumping is explored with the goal of ensuring robustness of design. This is pursued through PC with Hermite polynomials and Galerkin projections.
♥	Lovett, Ponci, and Monti [66]	The work examines measurement uncertainty in terms of independent measurements in relationship to ISO GUM.
◆	Monti, Ponci, and Lovett [67]	This is an introduction to PC modeling for circuit uncertainty analysis. The proceeding also mentions applications in measurement that are further developed in [68].

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
◆★✠	Monti, Ponci, Lovett, Smith, and Dougal [69]	An overview of PC is given, then it is proposed for use in applications relating to "automatic system modeling under uncertainty, uncertainty-based control, [and] uncertainty-based monitoring and diagnostics." A simple example relating to the Galerkin projection for a circuit model is given.
★■∇	Sarma, Durlofsky, and Aziz [43]	The article looks at expanding Bayesian inversion theory for the use in real time control of oil reservoir management. In here, a method to utilize Hermite Polynomial Chaos with the collocation method is presented in order to calculate expected states of the stochastic field. It is reported these are used to update a forward model.
◆	Su and Strunz [70]	The analogy between Hermite polynomials for Polynomial Chaos and Fourier series is briefly explored. It is stated that Hermite Polynomial Chaos could be useful for stochastic circuit analysis, such as when circuits are subjected to random effects, including thermal ones. An example of a linear bandpass filter is worked out.
♥	Wan and Karniadakis [71]	The authors propose an adaptive multi-element approach for Polynomial Chaos (ME-gPC). The approach has parallels to spectral elements.
♥	Wu, Zhu, and Najm [58]	This paper introduces the concept of using independent component analysis (ICA) and Karhunen-Loève expansions for non-Gaussian random processes. The method is based on using Newton's method to maximizes the absolute value of kurtosis and thus approximately maximize non-Gaussinity. The inverse of the determined ICA matrix and covariances are used to find a PCE. The concept is further explored in a later paper [59].

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
2006		
⋈▲	Beran, Pettit, and Millman [72]	A long discourse is given on applying intrusive and non-intrusive PC techniques to analyze limit cycle oscillations. The emphasis is on looking at how variability in configurations and model parameters effects flight safety. Hermite, B-spline, and Haar wavelet approaches are utilized. Additionally, a discussion on enforcing the periodic conditions needed for limit cycle oscillations is given. Example simulations are presented. This includes simulations with bifurcations that result in multi-modal response distributions.
♥	D'Antona, Monti, Ponci, and Rocc [73]	A maximum entropy approach to converting PCEs to PDFs is proposed. This concept is further developed in [74].
♥√	Das, Ghanem, and Spall [75]	A Fisher information approach, using "Karhunen-Loève decomposition and subsequent use of the maximum-entropy principle, Metropolis-Hastings Markov chain Monte Carlo algorithm and the Rosenblatt transformation," to estimating PCE coefficients from samples is presented. Also, the paper examines the problem from a maximum likelihood estimator (MLE) approach.
√	Ghanem and Doostan [76]	This work looks at a combination of Bayesian estimation and PC for estimating PDFs from samples.
▲□	Guedri, Bouhaddi, and Majed [77]	Two approaches using modal perturbation (MP) and PC are used to look at the mean, standard deviation, and "extreme statistics" for the frequency response of a stochastic FE model. Examples include a rotor and a frame vibration problem.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
★	Hover and Triantafyllou [44]	This paper is related to control and stability. It looks at how given initial conditions map to responses and how plant parameters map to responses. It is mentioned that "[t]he Polynomial Chaos expansion appears to be a completely new technique for these problems, and has not been reported in the literature." The PC method is described as noteworthy for the application since it can handle large variances. The goals presented in the paper are: "1. [p]rediction of short-term statistics and stability, given parameters or initial conditions with known distribution [and] 2. [c]ontroller design, enabled by treating the unknown gains as uncertain parameters." Galerkin projections, with Hermite polynomial expansions, are utilized and long term and short term stability are analyzed through simulation. Additionally, control design problems where the gain has been selected as a "random" variable for the expansion are discussed. It is proposed that PCEs would allow fast evaluation of individual deterministic gain realizations with regard to integral cost-functions.
■□	Knio and Le Maître [78]	This work includes a thorough review of the incorporation of PC with Navier-Stokes. Topics reviewed include Galerkin and collocation projections, integration of non-linear functions, Karhunen-Loève expansions, and a review of PC CFD literature.
♠	Lepage [79]	Sonar is examined through looking at PDEs for acoustic propagation. PCEs are utilized to consider uncertainty in environmental effects. Additionally, a "perterbation treatment" is adopted to address the exponential terms necessary to transform the Hermite chaos for log-normal distributions.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
□■	Lin, Su, and Karniadakis [80]	Two-dimensional supersonic flows into a wedge are analyzed for uncertainty in regards to uncertain flow velocity and uncertain wedge rotation. The motivation for this study relates to the experimental observation that small amounts of disturbances in wind tunnel flows and induced movement, such as flutter, create uncertainty in the results. Brief information is presented on setting up a "multi-element" stochastic-FEM-type PC approach (denoted as ME-gPC), by using separate uniform based expansions to split a stochastic domain into multiple joined stochastic domains. Information is given on decoupling the elements so that a standard deterministic Riemann solver can be used as well as a brief mention of the collocation method is given. Further, parallel information is presented relating to the creation of MC simulations. Simulation results for means and variance of shock paths are presented, for which the "multi-element" approach compares very favorably to analytic solutions and MC runs. Finally, a detailed PC derivation using the 2D Euler equations is presented in an appendix.
♥	Lin, Grinberg, and Karniadakis [81]	The Korteweg-de Vries (KdV) equation is a PDE that can be used to describe wave properties in plasma dynamics. An analytical solution to this equation only requires that additive noise be integral. Thus, the authors use this as an example to show the accuracy of a Finite Difference PC approach and a Discrete Galerkin PC approach. Additionally, thorough derivations of the models are provided in an appendix.
◆	Mi, Fan, and Tan [82]	This paper examines dividing integrated circuits into a finite number of spatial areas and then using Principal Component Analysis to transform the spatial uncertainty into independent random variables. The intent of the work is to study leakage currents.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
◆	Mi, Fan, and Tan [83]	This proceeding is related to another one of the authors' publications at the time. This one focuses more on an electrical circuit model.
×▲	Mulani, Kapania, and Walters [84]	This work takes a fairly straightforward approach to substituting PCEs into the eigenvalue problem. It appears to look at systems that do not contain damping, and thus it is not necessary to worry about bifurcations in the eigenvalues "breaking" the spectral approximation. Examples are given for a 3-DOF system and a simply supported beam.
□♥	Sachdeva, Nair, and Keane [85]	This work argues for the use of basis vectors from a Krylov subspace defined in terms of exponents of the random stiffness matrix for a stochastic FEM problem. The basis of the argument is that less memory is used in comparison to a Galerkin projection method of solving a stochastic FEM problem.
□♥	Sachdeva, Nair, and Keane [86]	The authors combine Krylov spaces with PC in order to try to reduce the number of basis functions needed for a stochastic FEM approach.
▲	Sinha [87]	Modal equations and a Galerkin PC procedure are used to examine the effects of mistuning on the statistical properties of vibrations in bladed disk assemblies.
★◆	Smith, Monti, and Ponci [41]	The authors combine a PI controller with the PC observer that the same authors also proposed in [88]. The Separation Principle is cited as a justification for designing the observer and controller independently. The controller is designed using an algebraic Riccati equation. An example relating to a Buck DC/DC converter is given.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
✠◆	Smith, Monti, and Ponci [88]	This work proposes a "Polynomial Chaos Observer." The intent of this observer is to observe states of a PC system that are not measured. Different strategies for selecting observer gains are presented, including Luenberger, "Optimal Kalman," and "Steady-state Kalman" approaches.
◆	Su and Strunz [89]	A stochastic circuit modeling problem is examined using PC and Galerkin projections. The specific application is Twelve-Pulse Diode Rectifiers for aircraft.
□■	Tartakovsky and Xiu [90]	An investigation of low Reynolds number steady state flow in a pipe with stochastic surface roughness is presented. The analysis makes the assumption of a parabolic velocity distribution without radial or angular velocity components. For the surface roughness in the radial dimension, it is enforced as being periodic through the use of random Fourier expansions. Overall, the problem resulted in a 21-dimensional random space for the PC model. The Smolyak sparse grid is used in a collocation approach. Simulations using the approach determined that "the effects of low to moderate roughness (the normalized standard deviation of the surface roughness below 5%) on mean dispersion of a passive scalar are negligible."
⊕	Velamur Asokan and Zabaras [91]	The paper describes a variational multiscale approach for modeling heterogeneous media described by random fields.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
□♥	Wan and Karniadakis [92]	A comparison between high order PC and multi-element PC (ME-gPC) is given. Inputs, such as sines with random frequencies or random amplitudes are applied to flow simulations past a 2D cylinder. It is claimed that the ME-gPC can capture asymptotic behavior that occurs in long simulations, whereas high order PCEs break down in the given simulation cases. It is worth noting that the authors conclude: "[h]owever, for high-dimensional random inputs, the effectiveness of ME-gPC will be weakened since the number of elements may increase fast. Thus, the long-term behavior of Polynomial Chaos deserves further study."
	Williams [93]	In this paper, the equation $$\frac{d\phi(x)}{dx} = \Sigma(x)\phi(x) = 0 \tag{3.2}$$ is examined because it has an analytical solution. $\Sigma(x)$ is allowed to be a random function of position and is expanded with a Karhunen-Loève expansion. The stochastic function is modeled as Gaussian noise, thus allowing an analytical Markov form for the covariance function as well as an analytic solution to the related eigenfunction problem. A Karhunen-Loève expansion and Hermite PC approach is then used to solve the problem. Results are compared to analytical solutions. It is noted that one of the points of the paper is to introduce the nuclear reactor community to PC.

Continued on Next Page…

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
◆LI	Wu, Zhu, and Najm [59]	This paper looks at expanding random processes, including from ARMA models, with the Karhunen-Loève expansion. Additionally, it discusses using independent component analysis to transform non-gaussian random variables before performing the Karhunen-Loève transformation. Also, it looks at propagating the random processes through linear and non-linear systems. Additionally, a good introductory discussion in regards to performing operations on PCEs is given. It is also presents that decision variables can be combined with Polynomial Chaos in terms of conditional probabilities. Throughout the paper, the application of PCEs to Digital Signal Processing (DSP) and Application-Specific Integrated Circuits (ASICs) is presented.
2007		
ʊ	Agarwal and Aluru [94]	This work examines Lagrangian electrostatics, using Green's functions combined with PC. Both random fields and random parameters (dimensions) are considered. Galerkin projections are used in the case of the random dimensions. Examples relating forces between charged conductors are given. This includes comb drives with variance in the gap sizes (can be used as an actuator in MEMs), charged conductors over the ground plane, and a simple beam problem relating to capacitance.
‡	Cheng and Sandu [95]	The authors compare Galerkin, collocation, and least-squares collocation methods of PC, as well as MC. It is pointed out that in stiff systems, ones where some states change quickly and others not quickly, that the order of the Galerkin method can be $O(S^3 n^3)$ for linear algebra. An example is given for stratosphere chemical reactions.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
◆	D'Antona, Monti, Ponci, and Rocca [74]	The presented research takes the approach of analytically computing moments of PCEs and then using the maximum entropy principle to determine the PDF from the moments. The paper utilizes an RC circuit for an example.
□■	Foo, Yosibash, and Karniadakis [96]	In this paper, the stochastic structural analysis of thin-walled riser (tubular) structures is developed in depth. Both stochastic material properties and stochastic external pressures are used in the FEM formulations. Galerkin methods are used in linear analysis. Collocation is used for non-linear properties. Example simulations, relating to stochastic loading fields, a random Young's modulus, and an example with non-linear elasticity are presented. It is noted that generating stochastic stiffness matrices are easy since linear stiffnesses can be factored out. The accuracy of the Galerkin and collocation methods are compared to each other as well as to MC. Difficulty was reported in the computational complexity of creating accurate enough MC simulations for comparison.
■	Ganapathysubramanian and Zabaras [97]	This work first reviews the Smolyak interpolation grid for the PC collocation method. Then the author proposes an adaptive method for determining collocation points based on interpolation error along each stochastic dimension. The method is applied to fluid convection problems, including the Rayleigh-Bénard convection.
♣	Geneser, Kirby, Xiu, and Sachse [98]	Here, PC with Galerkin projections is used to create uncertain rate constants in the ODEs used in the Markovian ion channel model. Characteristic simulations are given.

Continued on Next Page. . .

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
∇	Ghanmi, Bouazizi, and Bouhaddi [99]	This work looks at multi-objective Parento cost functions, including ones that incorporate the mean of a cost function and the standard deviation of a cost function. It utilizes the stochastic response surface method (SRSM), which is very closely related to collocation. The problem is pitched in terms of Hermite Polynomial Chaos Expansions as well as discussed and related to perturbations, including approximating variances. A genetics algorithm is used for minimization in order to create "robust" designs such that variations in parameters still yield good results. Examples include an FEM problem for a structure made from joined plates and one for a model of helicopter skids.
□	Ghanem, Saad, and Doostan [100]	The authors present a model reduction algorithm for stochastic FEM.
■	Hosder, Walters, and Balch [101]	This work looks at Non-Intrusive Polynomial Chaos (NIPC), a.k.a. the collocation method. In here, random sampling, Latin Hypercube Sampling (LHS), Hammersley Sampling, as well as a quadrature based approach are compared. It is shown that the quadrature is very inefficient in that it takes higher order polynomials to reach the same accuracy. The approach taken is to use an analytical function for comparison purposes. A further example looks at flow over an aircraft wing.
□	Huang, Mahadevan, and Rebba [102]	The paper contains a collocation based PC approach to a structural stochastic FEM problem. Karhunen-Loève expansions are used to allow incorporating stochastic fields. Comparisons in run time between the collocation and Galerkin methods are given—with the collocation generally being quicker for very large scale and Galerkin for smaller scale problems.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
□■	Le Maître and Knio [103]	Within this work, the Lagrangian meshless particle approach to fluid flow simulation is combined with PC using the Galerkin projection.
○	Li and Sandu [104]	This work presents a vehicle/terrain interaction model using PC to take into account stochastic 2D terrain profiles and vehicle model parameter distributions. The terrain profiles are handled with the combined Karhunen-Loève expansion and PC approach. Tire sinkage effects are taken into account and are used to transition the wheel model between rigid and flexible representations. Collocation PC methods are shown to give good agreement with MC in resulting simulations and to be substantially faster.
∇■	Lucor, Enaux, Jourdren, and Sagaut [105]	The authors describe optimization methods, utilizing PC theory, for stochastic models. Specifically, the discussion touches on gradient descent methods and simulated annealing. The concepts are applied to reacting flows. A black-box approach to modeling simulation, combined with collocation, is used. Stochastic objective functions are used to determine cost. The paper is presented as a method for optimizing parameters for reacting flows. Examples are given, but the optimization criteria are not. It is noted that deterministic optimization is faster, but stochastic optimization allows "more information on the solution (surface response, moments, statistics, sensitivity analysis, etc.) with considerable speed-up compared to brute force methods."
√	Marzouk, Najm, and Rahn [106]	In this paper, PC is hybridized with a Bayesian approach in order to look at the inverse problem of estimating parameters. It is noted that this method obtains its effectiveness by exploiting lower computational costs to sample realizations from a PC forward simulation compared to MC forward simulations. A Galerkin approach is utilized.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
○	Moon, Nabet, Leonard, Levin, and Kevrekidis [107]	The authors use PC to help compare course-grain population behavioral dynamic models to fine-scale dynamic models. Non-intrusive techniques are employed such that the course-grain models could be used as black-boxes.
★∇	Nagy and Braatz [108]	The work contains a general discourse on sensitivity, MC, and PC methods. It is noted that it is possible to use statistical measures, including ones obtained from MC and sensitivity analysis of PCEs from simulation, in optimization problems. An example relating to process control for a simulated batch crystal process is given. Also, it is notable that the author mentions that "[r]obust controller design can be formulated as a classical minmax optimization or as a multiobjective optimization problem where one objective accounts for the nominal performance and the other (usually a measure of the variance of the distribution) accounts for robustness."
□▲	Ngah and Young [109]	The author uses stochastic fields for stiffness in a stochastic FEM example problem. PC is reported to compare very favorably to the FOSM perturbation method with medium range uncertainties, as well as performing more quickly than MC with lower order PCEs. Additionally, it is stated for very large variance Gaussian fields, PC based stochastic FEM simulations became unstable, since unstable system realizations are implicitly present in the collective representation.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
♣	Paffrath and Wever [110]	Within this paper, a "windowed Hermite" polynomial expansion for PC is proposed. This method is based on a Gaussian weighting function, conditioned to be within a set of bounds. The orthogonal set is derived using a Gram-Schmidt orthogonalization procedure. The focus is on the distribution tails for computing failure probabilities. Examples relating to the Lotka-Volterra (predator prey) and Belousov-Zhabotinsky reaction models are given.
■□	Rupert and Miller [111]	This work explores the PDEs for modeling porous flows combined with PC and Karhunen-Loève expansions of flow properties resulting from stochastic porous media. The method is compared to MC and it is noted that for large stochastic domains, the curse of dimensionality is problematic for the PC approach.
✠	Saad, Ghanem, and Masri [112]	This paper looks at extending the Ensemble Kalman Filter (EnKF) to replace the Monte Carlo scheme of using samples to propagate uncertainty. It applies an uncertain model to observing the movement of a multi-floor structure with hysteresis. "The noise signals perturbing both the model and measurements are modeled as first order, one dimensional, independent, Polynomial Chaos expansions having zero-mean and an RMS of 0.5 and 0.001 respectively. The parameters' uncertainties on the other hand, are modeled as second order, one dimensional, Polynomial Chaos expansions whose coefficients are to be determined in accordance with the available observations." The authors report that the method works better than "standard Kalman Filtering methods." It also states that the method has relatively low computational cost.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
□♥	Seynaeve, Rosseel, Nicolaï, and Vandewalle [113]	The authors use a Fourier mode analysis approach to solving PDEs combined with a PC representation of the system. Galerkin projections are used and then it is shown that special orthogonal sets can be utilized for decoupling to create a more efficient solution.
◆✠	Smith, Monti, and Ponci [42]	This work uses Galerkin projections to expand a linear system. The expanded system is then used in the implementation of an observer. It discusses looking at an observability matrix for the expanded system. The gains are of expanded dimensions. Also, it appears the work proposes inputting system measurements into the zeroth order PC terms. An example using a Buck DC/DC converter is given. Also, from the examples, the authors appear to imply the use for observing a system with rapidly varying uncertain parameters.
◆	Smith, Monti, and Ponci [114]	The authors use this proceeding to point out that it is possible to figure out the approximation for the bounds of a PDF, or worst case, given by a PCE through looking at the supremum and infinum of the expansion. It is applied that the bound, Ω, is finite. Additionally, given the polynomial nature of PCEs, it is often possible to do this analytically. Results for circuit simulations are performed.
▲	Witteveen, Sarkar, and Bijl [115]	This paper presents a stochastic approach to studying parametric uncertainty in wind turbine blade flutter. While the examined system is capable of bifurcations, complications in the PC approximation are avoided by splitting the model into two separate regimes: the linear aerodynamic and the nonlinear stall regimes. Additionally Gram-Schmidt orthogonalization is utilized in order to create orthogonal polynomials based on statistical moments of an uncertain input.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
♣	Xiu and Sherwin [116]	This paper looks at applying uncertainty in the form of PCEs to an arterial network model. For the example, uncertainty is applied to a non-dimensional parameter that relates to pulse propagation speed. The Smolyak sparse grid is combined with a collocation approach.
□⊕	Xu [117]	The authors apply PC to a multiscale stochastic FEM problem.
◆	Zou, Cai, Zhou, Hong, Tan, and Kang [118]	This work explores system model reduction using PCEs. The application mentioned is for nanometer scale circuits.
2008		
℧▲	Agarwal and Aluru [119]	Problems are analyzed in this work related to electro-mechanically coupled cantilevers using uncertainty in the representations of the modulus of elasticity and the gap size as an electrical coupling parameter. A Lagrangian approach is used to expand random cantilever deformations. Additionally a comb-drive model is studied. The stated application is for MEMs.
♥	Blatman and Sudret [120]	This short paper proposes using an empirical "lack of fit" measure in order to adaptively determine the terms to use in a PCE to fit to collocation samples. A truss structure reliability example is given.
⊕‡	Chamoin, Oden, and Prudhomme [121]	An approach to muti-scale modeling is given. In this work, local bonding properties of polymers are simulated with a MC approach. The local model is coupled to a continuum represented by a combined PC and KL approach. The stated motivating application is for modeling the polymer component used in lithography for semiconductor manufacturing.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
★	Fisher and Bhattacharya [3]	This conference proceeding contains the first known publication proposing the form of a Galerkin projection of a state space system can produce another one with the form $\mathbf{A} = \sum_{\mathbf{k=0}}^{\mathbf{P}} \mathbf{A_k} \otimes \mathbf{\Phi_k}$. It also proposes placing the generated expanded system into an LQR cost function. Additionally, bilinear matrix inequalities (BMIs) are proposed for determining if it is feasible to determine gains, the dimensions of the original system, that minimize the expected cost function value. Also, the expanded system is substituted into an Algebraic Ricatti Equation and solved. It is conjectured that this latter approach produces an optimal set of gains, but more work needs to be performed on observing the state of the uncertainties. Also, an example, based on a linearized F16 at "a high angle of attack" is given.
★	Fisher and Bhattacharya [4]	The authors proposed that PC expanded state space formulations can be substituted into Lyapunov equations for determining stability. It is also mentioned that the Lyapunov equation can be formulated as a linear matrix inequality (LMI). Also, eigenvalues of the expanded state space are examined.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
√▲	Ghanem, Doostan, and Red-Horse [122]	The work relates model validation and analyzing the "predictive capabilities of a given model." It is stated that researchers are trying to do more with modeling, including assessing risk. The work in the paper takes a maximum likelihood approach, of $L(\gamma) := \prod_{j=1}^{M} p_{\hat{\eta}^{(k)}}\left(\eta_j^{(k)}|\gamma\right)$, where γ is a set of Polynomial Chaos coefficients and $\eta_j^{(0)}$, $\eta_j^{(1)}$, ... are observations of a vector. In this maximum likelihood approach, it is desired to find $\hat{\gamma} = \arg\max_{\gamma\in\Lambda} \mathrm{L}(\gamma)$. Additionally, as is common in maximum likelihood analysis, Fisher information is included in the approach. It states that simulated annealing is amenable to the type of search required. Finally, an example related to finding PCE represented distributions for a 3-DOF oscillator, coupled to an elastic foundation is presented.
‡★	Hauptmanns [123]	This applications oriented paper applies PC to analyzing uncertainty in the dynamics of the reaction process for producing the explosive chemical hexogen. A PI controller is considered in the dynamics. Non-linear differential equations are considered and integrals are evaluated using Gauss-Legendre quadrature. The intent is to provide an alternative to MC for evaluating probabilities of runaway reactions occurring.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
★∇	Hover [45]	This application of PC optimizes a non-linear set of system with respect to an integral, including final state values and integrals of the state value over a finite time integral. The presented method is used to find a corresponding set of optimal trajectories, including for feed-forward control. Both sets are represented by Legendre PC expansions. Gradient optimization is utilized for simplicity. Additionally, it is required to start with a stable trajectory. An approach of starting with a linear system and then increasing non-linearity is used. As reported by the author, *The key idea then is to gradually build the nonlinear terms from zero to their full values during a suitable number of gradient iterations, say* N. *It is the iteration itself that allows this entirely pragmatic procedure to be attempted; convergence to a static solution is effectively replaced by tracking a changing optimization problem.* The following system and cost function, with two uniformly distributed initial conditions $\dot{x}_1 = -\frac{1}{10}x_1 - x_2 - \frac{3}{4}x_1^2 - \frac{x_1}{x_2} - \frac{3}{2}x_2^2$ $\dot{x}_2 = x_1$ $x_1(t=0) = u\left(0, \frac{1}{3}\right) = \xi_1,$ $x_2(t=0) = u\left(0, \frac{1}{3}\right) = \xi_2,$ $\min_u J = \frac{1}{2}x_2^2(t=10) + \frac{1}{2}\int_0^{10}\left(x_1^2 + x_2^2 + u^2\right)dt,$ is used in an example to demonstrate the concept. The family of optimal trajectories is shown.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
ℵ√∇	Jin and Zou [124]	This is a stochastic finite element paper that looks at estimating the Robin coefficient, a parameter that is used in a model for convection in the ambient environment caused by boundary temperatures and thermal flux. The KLE is used to model random fields. An inverse stochastic optimization problem is posed. The solution relates to the minimization of a cost function combining the square of the errors between the simulation and observed, as well as Sobolev norm. A modified conjugate gradient method (CJG) is used as the optimization method. Sobolev inner products are used in the stopping condition for the optimization problem in order to "mitigate greatly the deleterious effect of data noise and numerical noise." The combination of the selected optimization method with the stopping condition are reported to serve as a good regularization method for handling the ill-posedness of the inverse problem. Multiple numerical examples relating to finding the distributed parameter are given.
▲	Laing and Kevrekidis [125]	In this work, coupled non-homogenous van der Pol oscillators are studied. These oscillators have been modified to produce asymmetric responses and have the following form: $\frac{dx_i}{dt} = y_i - x_i^2\left[x_i^2/3 - (\phi + \beta\mu_i)\right] + x_i^2/2 - \frac{\epsilon}{N}\sum_{j=1}^{N}(x_i - x_j)$ and $\frac{dy_i}{dt} = -x_i + A\sin(\omega t)$. In this model, N is the number of oscillators. The oscillators are subject to the same sinusoidal forcing. The article mentions that solutions of individual oscillators are integrated forward. Then the PC basis functions representing the solutions, as a function of the oscillator are found. The approximate PCE is used to extrapolate future points. Next, the integration is repeated. The stated point of the research is to find better ways to model mid-sized networks in order to close the gap between small networks and infinite networks.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
◆	Li, Wang, Tang, and Wu [126]	An approach for using the PC collocation method for analyzing circuit cross talk is given in this paper. Results using collocation with SPICE and the MC implemented in HSPICE are compared. It is proposed that PC is much more efficient than MC to examine the effects of process variation.
¶	Lovett, Monti, and Ponci [127]	Automated uncertainty modeling of linear circuits using PC and PSPICE is discussed in this paper. An example of a Buck DC/DC converter is given. Further, an example of a closed loop converter is given, where gain values are deterministic and chosen by an unspecified method.
◆	Mi, Fan, Tan, Cai, and Hong [128]	This is a full paper on using PC to analyze spatial based circuit variations, including inter-die inconsistencies. It uses much work from the authors' other papers [82, 83, 129]. It is performed using the Karhunen-Loève transformation and principal component analysis. Log-normal distributions are used for representing uncertainties.
♥	Monti, Ponci, and Valtorta [130]	The authors use this proceeding to propose a relationship between PC and alfa-cuts of Random Fuzzy variables.
◆LI	Smith, Monti, and Ponci [131]	In this work, a known method of finding a PDF from a PCE, based on the inverse method, is used to find a CDF and thus determine confidence intervals. An example is applied to circuit impedance measurement. The concept is presented through application to a Sallen-Key filter.
♥	Sudret [132]	PC is shown to apply to sensitivity analysis through using PC to calculate the variances in Sobol indices.

Continued on Next Page...

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
▲	Witteveen, Loeven, Sarkar, and Bijl [133]	The authors develop a PC approach to studying Limit Cycle Oscillations—that they coin PCLCO. This method is based upon a non-intrusive collocation approach to simulating harmonically excited flutter models. Frequency responses are obtained after the model has reached a steady state. It is reported that the analysis agrees well with MC for a presented 2-DOF flutter model with a deterministic initial condition and log-normal spring constant.
□	Chen and Guedes Soares [134]	In this paper, a stochastic FEM model for a laminated plate is developed using PC. For this model, the elastic moduli along the fiber direction and shear moduli are regarded as independent Gaussian random fields. Simulations using low order KL and PC approximations are given. Results are compared to MC results. The stated goal is for improving structure reliability analysis, including buckling analysis.
√∇	Guilleminot, Soize, Kondo, and Binetruy [135]	This work presents an experimental procedure, using non-destructive ultrasonic testing, to measure fiber volume fraction in a composite. A random search is employed in a maximum-likelihood approach to determine PC coefficients for a KL expansion of the field of volume fractions.
□♥	Nouy, Clément, Schoefs, and Moës [136]	In this work, a stochastic FEM approach for solving PDEs is given. The research relates specifically to stochastic dimensions: when sampling, such as for collocation methods, it is very computationally expensive to have to recalculate the mesh. The new approach of an extended stochastic finite element method (X-SFEM) relies on Galerkin projections and random level-sets. Results are given for a randomly dimensioned plate in tension; a plate with a hole of random radius and location; and a weld with random leg lengths.

Continued on Next Page. . .

Table 3.2 – Continued

Subj.	Authors	Notes and Contributions
□	Wei, Cui, and Chen [137]	The authors of the paper propose using points from monomial cubature rules (MCR) to create PCEs. Using MCRs is being suggested over a Smallayak sparse grid collocation approach for the reasons that the "approach suffers from its lack of universal applicability as it only works for certain combinations of the state space dimension and practical experience is still limited." Cubature approaches, involving sampling points and associated weights, for use in integrations relating to Hermite based PC, are given. The method is applied to two mathematical expressions of random variables to find means and variances. The agreement with MC and known answers is very good. Additionally, the approach is applied to a stochastic model of crushing a tube. It is concluded that the method is very advantageous for situations where the stochastic response surface can be modeled by low order polynomials since very accurate results can be obtained with many fewer points than used in quadrature approaches.
2009		
■	Najm [138]	This review paper provides a small sample of the Polynomial Chaos developments in the area of fluid modeling and simulation. Additionally, a portion of the paper is dedicated to Polynomial Chaos and uncertainty fundamentals.

Chapter 4

Analytical and Numerical Convergence Results

4.1 Introduction

In this chapter, a formal approach is used to discuss Polynomial Chaos Expansions of probabilistic linear differential equations. The presented results are related to those found by Cameron and Martin [36]. Additionally, properties of the systems of interest allow the approach to be generalized as well as demonstrate the existence of integrals over times from zero to infinity. Also, square mean, variance, and mean square trajectories are discussed. The presented discussion is focused in such a way as to help explain the developed probabilistic $\mathcal{H}_2$ control design method.

In general, the equations of interest for the chapter can be expressed as:

$$\dot{\mathrm{x}}(t,\xi) = \mathbf{A}(\xi)\mathrm{x}(t,\xi) + \mathbf{B}_w(\xi)\delta(t). \tag{4.1}$$

Here, $\xi \in \Omega$ is a parameter. Additionally, a powerful theorem is cited at the end of the chapter that justifies allowing ξ to be a set of parameters. Also, $\delta(t)$ is an impulse at $t = 0$. As discussed in Section 2.10, this is equivalent to a set of impulse responses or initial condition problems. The case will be expanded in the subsequent chapter to a full probabilistic linear system with feedback. Since the equations can be expressed as differential equations of this form, when coupled with algebraic manipulation for the output equation, the presented analysis is applicable there as well.

The remainder of this chapter starts by discussing stability and well-posedness for this class of problems. Next, some necessary functional analysis preliminaries are presented. It then makes sense to move to Hilbert spaces and demonstrate the convergence of Galerkin projections. Following, Sobolev spaces for general problems of this class are discussed. Next, numerical examples are given to illustrate the concepts. Lastly, a discussion of the application to larger-dimension linear ODEs is given.

4.2 Well-Posedness

For the type of problems addressed, standard linear differential equation theory indicates the well-posedness of the solution. For every finitely defined parameter, a member of a set of ODEs is defined. The solution to the examined problem is simply the set of ODEs. Hence, if every member of the set is well-defined, then the solution to the probabilistic system problem is a well-posed response surface.

It is now significant to point out two important classes of well-posed (finite magnitude parameter) systems:

I. Sets of systems that exclusively contain strictly stable systems

II. Sets of systems that contain stable, marginally stable, and/or unstable systems

For the first category, every realization system is a stable LTI system and thus globally exponentially stable. It follows that there exist constants $a > 0$ and $b > 0$ for each system such that for finite initial states, $||\mathrm{x}_0||$, at time t_0

$$||\mathrm{x}(t; \mathrm{x}_0, t_0)|| \leq b\,||\mathrm{x}_0||\,e^{-a(t-t_0)}. \tag{4.2}$$

Therefore, for the set of constants that parameterize the group of systems, the infinium and supremum, given by $\inf\{a\}$ and $\sup\{b\}$ denote constants for an exponential bound for all realization systems of the set. This can be expressed as:

$$P\left\{||\mathrm{x}(t; \mathrm{x}_0, t_0, \xi)|| \leq \sup\{b\}\,||\mathrm{x}_0||\,e^{-(\inf\{a\})(t-t_0)}\right\} = 1. \tag{4.3}$$

Since the probability of exponential stability is equal to one, the set of systems exhibits *almost sure exponential stability* [139]. Further, it is obvious every single system has a well posed $\mathcal{H}_2$ norm, and by extension, a well-posed L_2 integral of any state for an impulse response.

Additionally, the second set of systems is still interesting. For linear systems, unstable realizations take infinite time to diverge. Hence, for finite time, the mean and variation of sets of systems of Class II are finite for finite time. That said, individual systems diverge exponentially and hence could drastically effect the well-posedness of finite spectral approximations. One such example is the L_2 and $\mathcal{H}_2$ norms are in general infinite (see Section 2.2).

4.3 Hilbert Space Preliminaries

The method chosen to solve the specified type of equations is one utilizing PCEs and Galerkin projections. In order to establish a deeper understanding of what this is and why it can be done, it is necessary to delve into Hilbert spaces.

Hilbert spaces are complete inner product spaces. Two such spaces are particularly important to the current derivation: the $L_2(\Omega)$ and a weighted extension, the $L_{2,w}(\Omega)$ space.

Definition 4.3.1. *$L_2(\Omega)$ space: This space is defined with respect to the inner product*

$$\langle f(\xi), g(\xi)\rangle \equiv \int_\Omega f(\xi)\overline{g(\xi)}\, d\xi. \tag{4.4}$$

Additionally, the L_2 norm is defined by

$$\|\mathrm{x}(\xi)\| \equiv \sqrt{\langle \mathrm{x}(\xi), \mathrm{x}(\xi)\rangle}. \tag{4.5}$$

The $L_2(\Omega)$ space is composed of all functions that are square integrable—with respect to the domain $\Omega \subset \mathbb{R}^n$.

Definition 4.3.2. *$L_{2,w}$ space: Using a measurable function, $w(\xi)$, the concept of an $L_2(\Omega)$ space can be expanded. The weighted inner product is*

$$\langle f(\xi), g(\xi)\rangle \equiv \int_\Omega f(\xi)\overline{g(\xi)}w(\xi)\, d\xi. \tag{4.6}$$

In this equation, the weighting function should be defined such that $w(\xi) > 0$ almost everywhere (Debnath and Mikusiński [140], p. 96). Again, the required norm is defined with respect to the

related inner product:

$$\|\mathrm{x}(\xi)\| \equiv \sqrt{\langle \mathrm{x}(\xi), \mathrm{x}(\xi)\rangle}. \tag{4.7}$$

Note that the norms are defined with respect to the Lebesque integral. Also L_2 is obviously an example of $L_{2,w}$ and hence it is possible to simply talk in general about L_2 *spaces*. Additionally, in regards to utilized polynomial sets, the w is the same as the *weighting functions* described in Table 2.1. As before, it is going to be mandated that w is normalized such that $\|1\| = 1$.

Additionally, orthonormal basis are important to the discussion.

Definition 4.3.3. *A complete orthonormal basis* $\left\{\phi^i(\xi)\right\}$ *implies that for every* $\mathrm{x}(\xi)$*, there is a unique representation of the form*

$$\mathrm{x}(\xi) = \sum_{i=0}^{\infty} \hat{\alpha}^i \phi^i(\xi). \tag{4.8}$$

It is distinguished from simply orthogonal since the sets are normalized as

$$\left\|\phi^i(\xi)\right\| = 1, \ \forall i, \tag{4.9}$$

with regards to the appropriate norm.

Lemma 4.3.4. *For a Hilbert space,* $\hat{\alpha}^i = \left\langle \mathrm{x}(\xi), \phi^i(\xi)\right\rangle \equiv \hat{\mathrm{x}}^i$ *in Definition 4.3.3 (Debnath and Mikusiński [140], p. 110).*

Since Hilbert spaces are complete, it is possible to say this inner product forms the unique solutions– without involving Cauchy sequences. Completeness implies every Cauchy sequence converges to a member of the space. Krall [141] provides proofs for completeness in the appropriate $L_{2,w}(\Omega)$ space for many of orthogonal (normalized to be orthonormal) sets used in the dissertation.

Definition 4.3.5. *Weak equality of expansions implies that for the orthonormal set* $\left\{\phi^i(\xi)\right\}$,

$$\lim_{n\to\infty}\left\|\mathrm{x}(\xi)-\sum_{i=0}^{n}\hat{\mathrm{x}}^i\phi^i(\xi)\right\|=0 \tag{4.10}$$

(Debnath and Mikusiński [140], p. 109).

This differs from *point-wise* equality in that it is not guaranteed that the expansion and the original function are equal for all ξ. For example, consider the case where $\xi \in \Omega$ and Ω is a closed continuous interval in $\mathbb{R}$. Now, let $\mathrm{x}(\xi) = \xi$. Additionally, let $\mathrm{x}_p(\xi) = \mathrm{x}(\xi)$ except at a countable number of ξ, where it is infinite. $\mathrm{x}_p(\xi)$ is still L_2 integrable. Also, the inner products with basis functions are equal. As such, $\lim_{n\to\infty}\left\|\mathrm{x}_p(\xi)-\sum_{i=0}^{n}\hat{\mathrm{x}}^i\phi^i(\xi)\right\| = 0$ as well, even though it obviously is not equal at every ξ. While this may seem like a niche case, Hilbert space analysis applies to all members of the space, including ones like $\mathrm{x}_p(\xi)$.

Lemma 4.3.6. *Let* $\mathrm{x}(\xi) = \sum_{i=0}^{\infty}\hat{\mathrm{x}}^i\phi^i(\xi)$ *lie within* $L_{2,w}(\Omega)$*. Then it follows that if* $a(\xi)$ *is in* $L_{2,w}(\Omega)$ *and* $c \equiv \sup\{|a(\xi)| : \xi \in \Omega\} < \infty$*, then* $a(\xi)\mathrm{x}(\xi) = a(\xi)\sum_{i=0}^{\infty}\hat{\mathrm{x}}^i\phi^i(\xi)$*.*

Proof. Following from the weak equality,

$$c^2\lim_{n\to\infty}\left\|\mathrm{x}(\xi)-\sum_{i=0}^{n}\hat{\mathrm{x}}^i\phi^i(\xi)\right\|^2=0. \tag{4.11}$$

Using Definition 4.3.2 and writing the norm equivalently in terms of a magnitude,

$$\lim_{n\to\infty}\int_\Omega c^2\left|\mathrm{x}(\xi)-\sum_{i=0}^{n}\hat{\mathrm{x}}^i\phi^i(\xi)\right|^2 w(\xi)\,d\xi=0. \tag{4.12}$$

Since $c \geq |a(\xi)|\ \forall\xi$ and $\lim_{n\to\infty}\left|\mathrm{x}(\xi)-\sum_{i=0}^{n}\hat{\mathrm{x}}^i\phi^i(\xi)\right|^2 \geq 0$, zero obviously serves as an upper-bound for

$$\lim_{n\to\infty}\int_\Omega |a(\xi)|^2\left|\mathrm{x}(\xi)-\sum_{i=0}^{n}\hat{\mathrm{x}}^i\phi^i(\xi)\right|^2 w(\xi)\,d\xi\leq 0. \tag{4.13}$$

Additionally, the norm cannot be less than zero and hence the equality holds. As such

$$\lim_{n\to\infty}\int_{\Omega}\left|a(\xi)\mathrm{x}(\xi)-a(\xi)\sum_{i=0}^{n}\hat{\mathrm{x}}^{i}\phi^{i}(\xi)\right|^{2}w(\xi)\,d\xi=\lim_{n\to\infty}\left\|a(\xi)\mathrm{x}(\xi)-a(\xi)\sum_{i=0}^{n}\hat{\mathrm{x}}^{i}\phi^{i}(\xi)\right\|^{2}=0. \tag{4.14}$$

This concludes the proof. □

Parseval's Equality, commonly used in controls to relate frequency and time domain is more broadly defined in terms of Hilbert spaces.

Theorem 4.3.7. *Parseval's Equality: A set is a complete orthogonal set iff*

$$\|\mathrm{x}(\xi)\|^{2}=\sum_{i=0}^{\infty}\left|\left\langle \mathrm{x}(\xi),\phi^{i}(\xi)\right\rangle\right|^{2} \tag{4.15}$$

(from Krall [141], p. 15).

Lemma 4.3.8. *Parseval's Equality is bounded (finite) iff for any positive and finite set of constants,* $\left\{c^{i}\right\}$*, the following sum is bounded:*

$$\sum_{i=0}^{\infty}c^{i}\left|\left\langle \mathrm{x}(\xi),\phi^{i}(\xi)\right\rangle\right|^{2}. \tag{4.16}$$

Proof.

$$\inf\left\{c^{i}\right\}\sum_{i=0}^{\infty}\left|\left\langle \mathrm{x}(\xi),\phi^{i}(\xi)\right\rangle\right|^{2}\leq\sum_{i=0}^{\infty}c^{i}\left|\left\langle \mathrm{x}(\xi),\phi^{i}(\xi)\right\rangle\right|^{2}\leq\sup\left\{c^{i}\right\}\sum_{i=0}^{\infty}\left|\left\langle \mathrm{x}(\xi),\phi^{i}(\xi)\right\rangle\right|^{2} \tag{4.17}$$

□

Corollary 4.3.9. *The finite and positive weights* c^{0} *and* c^{1}

$$c^{0}\left|\left\langle \mathrm{x}(\xi),\phi^{0}(\xi)\right\rangle\right|^{2}+c^{1}\sum_{i=1}^{\infty}\left|\left\langle \mathrm{x}(\xi),\phi^{i}(\xi)\right\rangle\right|^{2} \tag{4.18}$$

are bounded iff $\|\mathrm{x}(\xi)\|^2$ *is bounded.*

The following definitions, in terms of the L_2 spaces, demonstrate the significance of the above corollary: weighting the square mean and variance. These definitions are the reason for requiring that $w(\xi)$ is normalized such that $\|1\| = 1$.

Definition 4.3.10. *Mean Square:* $E\left[\mathrm{x}(\xi)^2\right] \equiv \|\mathrm{x}(\xi)\|^2$

Definition 4.3.11. *Square Mean:* $E\left[\mathrm{x}(\xi)\right]^2 \equiv \left|\left\langle \mathrm{x}(\xi), \phi^0(\xi)\right\rangle\right|^2$*, where the zeroth term is a constant such that* $\phi^0(\xi) \equiv \left\|\phi^0(\xi)\right\|$*. The constant* $\phi^0(\xi)$ *properly normalizes the mean and by extension the square mean.*

Definition 4.3.12. *Variance:* $\mathrm{var}\left[\mathrm{x}(\xi)\right] = E\left[\left(\mathrm{x}(\xi) - E\left[\mathrm{x}(\xi)\right]\right)^2\right] = \sum_{i=1}^{\infty}\left|\left\langle \mathrm{x}(\xi), \phi^i(\xi)\right\rangle\right|^2$

From the set of definitions, the mean square can be found from the solution $\mathrm{x}(\xi)$, and the latter two by projection of the solution. The defined relations naturally agree with the basic probability result: $E\left[\mathrm{x}(\xi)^2\right] = E\left[\mathrm{x}(\xi)\right]^2 + \mathrm{var}\left[\mathrm{x}(\xi)\right]$.

Based on Parseval's Equality (Theorem 4.3.7) and Lemma 4.3.4, the projection of both the solution and series expansion are guaranteed to be unique and equal.

4.4 On Derivatives

The independence of the probabilistic domain, denoted ξ, from the temporal domain, t, allows for nice results. For example, it can be seen the derivative of a projection is equal to the weighted mean of the derivative.

Theorem 4.4.1. *Given the* L_2 *space,* H*,* $\mathrm{x}(t,\xi) \in H$ *for* $t \in [a,b]$*, and* $\Psi(\xi) \in H$*, the projection operation and derivative, with respect to independent variables, commute. This applies if the derivative of* $\mathrm{x}(t,\xi)$ *with respect to* t *on the interval exists.*

Proof.

$$\begin{aligned}\frac{d}{dt}\langle \mathrm{x}(t,\xi), \Psi(\xi)\rangle &= \frac{d}{dt}\int_\Omega \mathrm{x}(t,\xi)\Psi(\xi)w(\xi)\,d\xi \\ &= \int_\Omega \frac{d}{dt}\mathrm{x}(t,\xi)\Psi(\xi)w(\xi)\,d\xi = \left\langle \frac{d}{dt}\mathrm{x}(t,\xi), \Psi(\xi)\right\rangle\end{aligned} \tag{4.19}$$

□

Definition 4.4.2. *Weighted Mean:*

$$\overline{\mathrm{x}(\xi)} = \int_\Omega \mathrm{x}(\xi)\, w(\xi)\, d\xi, \tag{4.20}$$

when $\int_\Omega w(\xi)\,d\xi = 1$. It is important to ensure that all functions, parameterized by ξ are differentiable with respect to t. Along the time domain this is easy to verify by freezing the probabilistic variables.

Corollary 4.4.3. *The weighted mean operation and derivative, with respect to independent variables, commute.*

Proof. This follows directly from Theorem 4.4.1 by letting $\Psi(\xi) = 1$. □

4.5 Hilbert Space

To begin simply, consider the single probabilistic linear differential equation

$$\frac{d}{dt}\mathrm{x}(t,\xi) = a(\xi)\mathrm{x}(t,\xi) + b(\xi)\delta(t). \tag{4.21}$$

In this equation, $\xi \in \Omega$, where Ω is a continuous and finite subset of $\mathbb{R}$. Further, a is defined such that $a(\xi) < 0$. Hence, all solutions, parameterized by ξ, are bounded for all $t \geq 0$. This implies the L_2 integral over Ω, at any $t \geq 0$, exists.

Rewriting Equation (4.21) as an integral equation:

$$\mathrm{x}(t,\xi) = \int_0^t \left[a(\xi)\mathrm{x}(T,\xi) + b(\xi)\delta(T)\right]\, dT. \tag{4.22}$$

This is equivalent to

$$\mathrm{x}(t,\xi) = \int_0^t a(\xi)\mathrm{x}(T,\xi)\, dT \tag{4.23}$$

with $\mathrm{x}(t,\xi)|_{t=0} = b(\xi)$. The conversion to an integral equation is completed because it allows temporarily sidestepping the question of existence of the derivative with respect to time.

Since the L_2 integral over the domain Ω exists, by Theorem 4.3.3 and Lemma 4.3.4 there is a unique decomposition $\mathrm{x}(t,\xi) = \sum_{i=0}^{\infty} \hat{\mathrm{x}}^i(t,\xi)\phi^i(\xi)$. Additionally, by construction, $a(\xi)$ is finite and L_2 integrable—implying that $a(\xi)\mathrm{x}(T,\xi)$ at time T can be expanded as $a(\xi)\sum_{i=0}^{\infty} \hat{\mathrm{x}}^i(t,\xi)\phi^i(\xi)$ (Theorem 4.3.6). Substituting these expressions in:

$$\sum_{i=0}^{\infty} \hat{\mathrm{x}}^i(t,\xi)\phi^i(\xi) = \int_0^t a(\xi) \sum_{i=0}^{\infty} \hat{\mathrm{x}}^i(t,\xi)\phi^i(\xi)\, dT \tag{4.24}$$

For a given $t \geq 0$, the L_2 space for $\mathrm{x}(t,\xi)$ is spanned by the set $\left\{\phi^i(\xi)\right\}$. Hence, it is possible to perform Galerkin projections of the integral equation onto the complete basis set:

$$\hat{\mathrm{x}}^k(t) = \left\langle \sum_{i=0}^{\infty} \hat{\mathrm{x}}^i(t,\xi)\phi^i(\xi), \phi^k(\xi) \right\rangle = \int_0^t \left\langle a(\xi)\mathrm{x}(T,\xi), \phi^k(\xi) \right\rangle\, dT. \tag{4.25}$$

The integral equation is an acceptable form to solve the equation. But, since it is known that the solution to the original probabilistic ODE is differentiable with respect to t, by Theorem 4.4.1 the

derivative of the projection with respect to t exists:

$$\dot{\hat{\mathrm{x}}}^k(t) = \frac{d}{dt}\left\langle \hat{\mathrm{x}}^i(t,\xi)\phi^i(\xi), \phi^k(\xi)\right\rangle = \left\langle a(\xi)\mathrm{x}(t,\xi), \phi^k(\xi)\right\rangle. \tag{4.26}$$

As such, it is possible to use the basis function coefficient trajectories to form useful terms. The mean square trajectory, from Definition 4.3.10,

$$E\left[\mathrm{x}(t,\xi)^2\right] = \lim_{n\to\infty}\sum_{i=0}^{n}\left|\hat{\mathrm{x}}^i(t)\right|^2 \tag{4.27}$$

the square mean trajectory, from Definition 4.3.11,

$$E\left[\mathrm{x}(t,\xi)\right]^2 = \left|\hat{\mathrm{x}}^0(t)\right|^2 \tag{4.28}$$

(where the zeroth term is a constant such that $\phi^0(\xi) = \left\|\phi^0(\xi)\right\|$), and the variance trajectory, from Definition 4.3.12,

$$\lim_{n\to\infty}\sum_{i=1}^{n}\left|\hat{\mathrm{x}}^i(t)\right|^2 \tag{4.29}$$

converge as the number of basis functions goes towards infinity. Even though the mean only contains one term instead of an infinite summation, in a strict sense, it only in general holds when the complete basis (countably infinite terms for $L_2(\Omega)$) for the space is used.

Since this is a set of stable systems, the stability bound given in Equation (4.3), indicates the integral with respect to time for $t \geq 0$ exists. Based on Parseval's Equality (Theorem 4.3.7), for all $t \geq 0$, the expansion solution to the mean square trajectory is equal to the true mean square trajectory. Based on Corollary 4.3.9, the integrals for the square mean trajectories and variance trajectories are finite as well. This follows from the fact that the two trajectories are always bounded below by zero and above by the mean square.

4.6 Smoothness of Solutions

Using the theory of ordinary differential equations, it is easy to show that solutions to problems with "smooth" parameter distributions are members of Sobolev spaces.

Definition 4.6.1. *Sobolev Space*

A Sobolev space, $\tilde{H}^m(\Omega)$, is the space of all functions of the open set Ω for which the derivatives

$$\frac{\partial^n f(\gamma_1, \gamma_2, \ldots \gamma_N)}{\partial \gamma_1^{n_1} \partial \gamma_2^{n_2} \ldots \partial \gamma_N^{n_N}}, \tag{4.30}$$

for all $n = \sum_{i=1}^{N} n_i$ such that the positive integer $n \leq m$, have a finite L_2 norm. The required inner product is the sum of the integral, over the domain Ω of all of the required partial derivatives. Note that all possible non-negative integer permutations for n_i must hold. Additionally, the derivatives up to the necessary order should be defined.

Using this definition, it is possible to show that solutions to a finite dimension stochastic linear ODEs (with a sufficiently smooth parameter space) at time t are members of Sobolev spaces.

Theorem 4.6.2. *Finite dimensional stochastic linear ODEs, with sufficiently differentiable parameter matrices, have solutions at time t that lie within Sobolev spaces.*

Proof. To begin, the linear systems are expressed in terms of state space notation. Such a system, in terms of the parameter γ is

$$\begin{aligned} \frac{d}{dt}\mathrm{x}(\gamma, t) = \mathbf{A}(\gamma)\,\mathrm{x}(\gamma, t) + \mathbf{B}(\gamma)\,\mathrm{u}(\gamma, t) = f(t, \mathrm{x}, \gamma) \\ \mathrm{y}(\gamma, t) = \mathbf{C}(\gamma)\,\mathrm{x}(\gamma, t) + \mathbf{D}(\gamma)\,\mathrm{u}(\gamma, t). \end{aligned} \tag{4.31}$$

From Khalil [142], p. 99, if $f(t, \mathrm{x}, \gamma)$ has continuous first partial derivatives with respect to x and t for a given interval of time, then the partial derivative with respect to γ exists within those bounds. This is equivalent to the "sufficiently differentiable parameter matrices" requirement. The partial derivative, with respect to the parameter γ (and simplifying the notation by not specifically denoting functions of t and γ), is

$$\begin{aligned} \frac{d}{dt}\frac{\partial \mathrm{x}}{\partial \gamma} &= \frac{\partial \mathbf{A}}{\partial \gamma}\mathrm{x} + \mathbf{A}\frac{\partial \mathrm{x}}{\partial \gamma} + \frac{\partial \mathbf{B}}{\partial \gamma}\mathrm{u} + \mathbf{B}\frac{\partial \mathrm{u}}{\partial \gamma} \\ \frac{\partial \mathrm{y}}{\partial \gamma} &= \frac{\partial \mathbf{C}}{\partial \gamma}\mathrm{x} + \mathbf{C}\frac{\partial \mathrm{x}}{\partial \gamma} + \frac{\partial \mathbf{D}}{\partial \gamma}\mathrm{u} + \mathbf{D}\frac{\partial \mathrm{u}}{\partial \gamma}. \end{aligned} \tag{4.32}$$

Combining the two sets of equations:

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} \mathrm{x} \\ \frac{\partial \mathrm{x}}{\partial \gamma} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \frac{\partial \mathbf{A}}{\partial \gamma} & \mathbf{A} \end{bmatrix}\begin{bmatrix} \mathrm{x} \\ \frac{\partial \mathrm{x}}{\partial \gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \frac{\partial \mathbf{B}}{\partial \gamma} & \mathbf{B} \end{bmatrix}\begin{bmatrix} \mathrm{u} \\ \frac{\partial \mathrm{u}}{\partial \gamma} \end{bmatrix} \\ \begin{bmatrix} \mathrm{y} \\ \frac{\partial \mathrm{y}}{\partial \gamma} \end{bmatrix} &= \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \frac{\partial \mathbf{C}}{\partial \gamma} & \mathbf{C} \end{bmatrix}\begin{bmatrix} \mathrm{x} \\ \frac{\partial \mathrm{x}}{\partial \gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \frac{\partial \mathbf{D}}{\partial \gamma} & \mathbf{D} \end{bmatrix}\begin{bmatrix} \mathrm{u} \\ \frac{\partial \mathrm{u}}{\partial \gamma} \end{bmatrix}. \end{aligned} \tag{4.33}$$

Crucially, this is also a state space system. It is easy to see that it is possible to incorporate any desired partial derivatives with respect to parameters into the equations—as long as the partial derivatives with respect to the parameter matrices and input are well-defined.

Further, when the new state matrix is well-defined, note that the structure ensures that the eigenvalues solely depend on **A**. Hence, for a stable system, any finite order partial derivative is stable too. Additionally, bounded input, bounded output (BIBO) stability applies for such systems as well as properties of exponential rate of decay in initial condition problems. As such, the derivatives are square integrable. □

Demonstrating such linear problems lie in a Sobolev space helps to give a general idea of the numerical well-posedness of the problem. It is well known that smooth functions (i.e. continuous and

continuous derivative) in practice tend to converge *point-wise* substantially faster. One example of the converse is *Gibbs phenomenon*—the "ringing" in an expansion approximation (including, but not limited to, the Fourier series) using continuous basis functions to approximate a discontinuous function.

Further, it is possible to find results for the *order* of convergence in Sobolev spaces. A couple of references for results of this type are given in [143, 144, 145]. Since many of the results were derived for spectral methods, it makes sense that the orders are derived for Sobolev spaces. Spectral methods are very commonly applied to solving PDEs. As such, it is important for the spatial derivatives to be well-posed. An example of possible consequences is given by Gottlieb and Orszag [146]. Additionally, the book serves as yet another resource on convergence order.

4.7 Numerical Error Analysis

In this section, Monte Carlo and Polynomial Chaos are analyzed to look at convergence toward the mean and standard deviation. For Monte Carlo, it is necessary to represent the distribution with a finite set of samples. With Polynomial Chaos, it is necessary to use a finite size subset of the orthogonal series. As such, neither of these methods are exact—but can produce very accurate solutions.

The exact solution is known for Equation (4.21):

$$\mathrm{x}(t,\xi) = b(\xi)e^{a(\xi)t}. \tag{4.34}$$

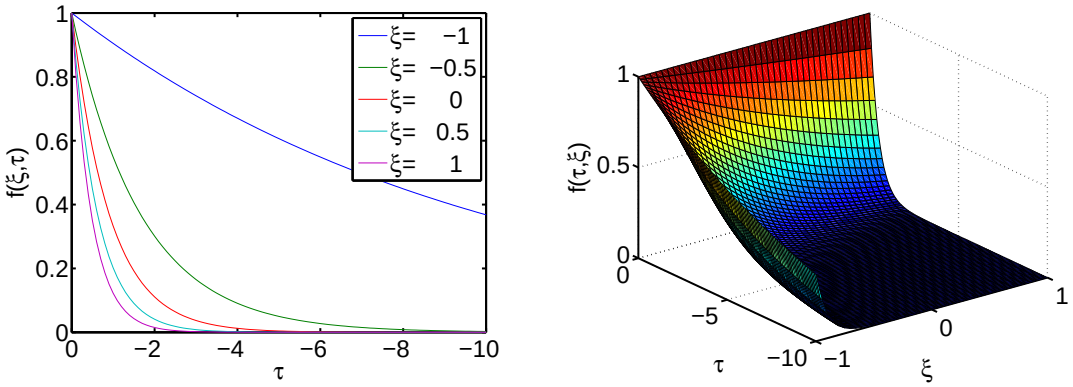
The problem used is similar to the one used by Xiu and Karniadakis [2]–but with new results.

Setting $b(\xi) = 1$ and making $a(\xi)$ a linear combination allows non-dimensionalizing along t so that only one additional parameter needs to be evaluated for this analysis. Hence, the equation becomes,

$$\mathrm{x}(\tau, \xi) = e^{(q+\xi)(\tau)} \tag{4.35}$$

with $\tau = rt$, and q the parameter of interest.

The solution can be viewed in terms of individual trajectories for a set ξ, or as a response surface (see Figure 4.1).



(a) (b)

Figure 4.1: Non-dimensionalized solutions: (a) the solution given as trajectories and (b) the solution illustrated as a surface

For this analysis, a number of weighting cases will be explored (Table 4.1). These functions serve as a probability distribution—they weight based on probabilistic likelihood.

Weighting functions from Table 4.1 are plotted in Figure 4.2.

The analytical solutions, in Table 4.1, for the mean trajectory can be transformed to the trajectory of the mean square. The following relations allow computing solutions to the necessary trajectories:

$$\begin{aligned} E\left[\mathrm{x}(\tau, q, \xi)\right] &= m(\tau, q) \\ E\left[\mathrm{x}(\tau, q, \xi)^2\right] &= m(2\tau, q). \end{aligned} \tag{4.36}$$

Table 4.1: Test cases with analytic solution for non-dimensionalized mean trajectory.

Case		Weight, w	$\overline{w}$	$\sigma(w)$	Non-Dimensionalized Mean
I.	ODE	$w_1(\xi) = \delta(\xi)$	ξ	0	$m_1(\tau, q, \xi) = e^{(1+q\xi)\tau}$
II.	$\beta(1,1)$	$w_2(\xi) = \frac{1}{2}$	0	$\frac{\sqrt{300}}{30}$	$m_2(\tau, q) = e^{(q-1)\tau}\,\frac{\left(-1+e^{2\tau}\right)}{2\tau}$
III.	$\beta(1,2)$	$w_3(\xi) = \frac{1}{2}(1-\xi)$	$-\frac{1}{3}$	$\frac{\sqrt{200}}{30}$	$m_3(\tau, q) = e^{(q-1)\tau}\,\frac{\left(-1+e^{2\tau}-2\tau\right)}{2\tau^2}$
IV.	$\beta(2,1)$	$w_4(\xi) = \frac{1}{2}(1+\xi)$	$\frac{1}{3}$	$\frac{\sqrt{200}}{30}$	$m_4(\tau, q) = e^{(q-1)\tau}\,\frac{\left(1-e^{2\tau}+2e^{2\tau}\tau\right)}{2\tau^2}$
V.	$\beta(2,2)$	$w_5(\xi) = \frac{3}{4}(1-\xi)(1+\xi)$	0	$\frac{\sqrt{180}}{30}$	$m_5(\tau, q) = e^{(q-1)\tau}\,\frac{3\left(1-e^{2\tau}+\tau+e^{2\tau}\tau\right)}{2\tau^3}$
VI.	$\beta(3,1)$	$w_6(\xi) = \frac{3}{8}(1+\xi)^2$	$\frac{1}{2}$	$\frac{\sqrt{135}}{30}$	$m_6(\tau, q) = e^{(q-1)\tau}\,\frac{3\left(-1+e^{2\tau}-2e^{2\tau}\tau+2e^{2\tau}\tau^2\right)}{4\tau^3}$
VII.	$\beta(1,3)$	$w_7(\xi) = \frac{3}{8}(1-\xi)^2$	$-\frac{1}{2}$	$\frac{\sqrt{135}}{30}$	$m_7(\tau, q) = e^{(q-1)\tau}\,\frac{3\left(-1+e^{2\tau}-2\tau-2\tau^2\right)}{4\tau^3}$
VIII.	$\beta(3,2)$	$w_8(\xi) = \frac{3}{4}(1-\xi)(1+\xi)^2$	$\frac{1}{5}$	$\frac{\sqrt{144}}{30}$	$m_8(\tau, q) = e^{(q-1)\tau}\,\frac{3\left(-3+3e^{2\tau}-2\tau-4e^{2\tau}\tau+2e^{2\tau}\tau^2\right)}{2\tau^4}$
IX.	$\beta(2,3)$	$w_9(\xi) = \frac{3}{4}(1-\xi)^2(1+\xi)$	$-\frac{1}{5}$	$\frac{\sqrt{144}}{30}$	$m_9(\tau, q) = e^{(q-1)\tau}\,\frac{3\left(3-3e^{2\tau}+4\tau+2e^{2\tau}\tau+2\tau^2\right)}{2\tau^4}$

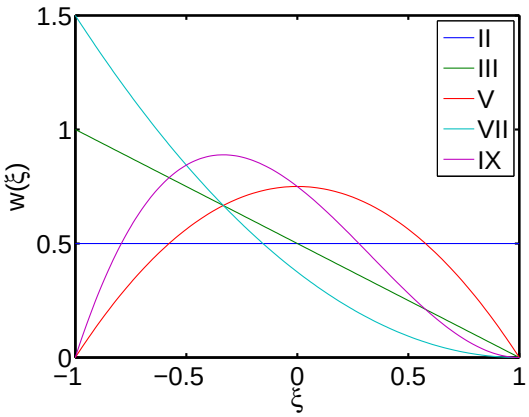


Figure 4.2: Plots of $w(\xi)$. The missing functions are mirror images (about $\xi = 0$) of the non-symmetric weighting functions shown.

The second relationship in Equation (4.36) is special to this problem—it would not typically apply for larger dimensional systems. In these equations, the expectation is of course with respect to ξ. Additionally, these can obviously be used to compute the variance and standard deviation trajectories.

The next step in the comparison is computing the Polynomial Chaos approximations. To do this, using the correct polynomial set for the given weighting function, let

$$\begin{aligned} \hat{a}^0 &= \left\langle q + \xi, \phi^0(\xi) \right\rangle \\ \hat{a}^1 &= \left\langle q + \xi, \phi^1(\xi) \right\rangle . \end{aligned} \tag{4.37}$$

Additionally, the following Nth-order series approximation is used:

$$\begin{aligned} \mathrm{x}(t,\xi) &\approx \sum_{i=0}^{N} \hat{\mathrm{x}}^i(t)\phi^i(\xi) \\ \hat{x}(t,\xi) &\approx \sum_{i=0}^{N} \hat{\dot{\mathrm{x}}}^i(t)\phi^i(\xi). \end{aligned} \tag{4.38}$$

The approximation for the derivative follows from Theorem 4.19.

Substituting these approximations into Equation (4.21):

$$\sum_{i=0}^{N} \hat{\dot{\mathrm{x}}}^i(t)\phi^i(\xi) = \sum_{j=0}^{1} \hat{a}^j\phi^j(\xi) \sum_{i=0}^{N} \hat{\mathrm{x}}(t)^i\phi^i(\xi) + \delta(t). \tag{4.39}$$

Using the inner product to project onto the orthogonal set:

$$\left\langle \sum_{i=0}^{N} \hat{\dot{\mathrm{x}}}^i(t)\phi^k(\xi), \phi^i(\xi) \right\rangle = \left\langle \sum_{i=0}^{1} \hat{a}^i\phi^i(\xi) \sum_{j=0}^{N} \hat{\mathrm{x}}^j(t)\phi^j(\xi), \phi^k(\xi) \right\rangle + \left\langle \delta(t), \phi^k(\xi) \right\rangle . \tag{4.40}$$

For a given order N, these truncated expansions have the property of being the best fit in the least-squares sense (see Section 2.5). However, the truncation also introduces some error into the derivative of the expansion coefficients, hence adding complexity to the approximation.

Projecting with the basis functions $\phi^0(\xi)$ to $\phi^N(\xi)$ produces the following set of equations (in vector form) :

$$\begin{bmatrix} \hat{\dot{x}}^0(t)\left\|\phi^0(\xi)\right\|^2 \\ \hat{\dot{x}}^1(t)\left\|\phi^1(\xi)\right\|^2 \\ \vdots \\ \hat{\dot{x}}^N(t)\left\|\phi^N(\xi)\right\|^2 \end{bmatrix} = \mathbf{A_{exp}}' \begin{bmatrix} \hat{x}^0(t) \\ \hat{x}^1(t) \\ \vdots \\ \hat{x}^N(t) \end{bmatrix} + \begin{bmatrix} \left\langle 1, \phi^0(\xi)\right\rangle \\ 0 \\ \vdots \\ 0 \end{bmatrix} \delta(t) \tag{4.41}$$

with the matrix

$$\mathbf{A_{exp}}' = \sum_{j=0}^{1} \hat{a}^i \begin{bmatrix} \left\langle \phi^i(\xi)\phi^0(\xi), \phi^0(\xi)\right\rangle & \left\langle \phi^i(\xi)\phi^0(\xi), \phi^1(\xi)\right\rangle & \cdots & \left\langle \phi^i(\xi)\phi^0(\xi), \phi^N(\xi)\right\rangle \\ \left\langle \phi^i(\xi)\phi^1(\xi), \phi^0(\xi)\right\rangle & \left\langle \phi^i(\xi)\phi^1(\xi), \phi^1(\xi)\right\rangle & \cdots & \left\langle \phi^i(\xi)\phi^1(\xi), \phi^N(\xi)\right\rangle \\ \vdots & \vdots & & \vdots \\ \left\langle \phi^i(\xi)\phi^N(\xi), \phi^0(\xi)\right\rangle & \left\langle \phi^i(\xi)\phi^N(\xi), \phi^1(\xi)\right\rangle & \cdots & \left\langle \phi^i(\xi)\phi^N(\xi), \phi^N(\xi)\right\rangle \end{bmatrix}. \tag{4.42}$$

If the set is orthonormal, this is the final form. However, it is frequently convenient to use simply orthogonal sets. In this case,

$$\begin{bmatrix} \hat{\dot{x}}^0(t) \\ \hat{\dot{x}}^1(t) \\ \vdots \\ \hat{\dot{x}}^N(t) \end{bmatrix} = \mathbf{A_{exp}} \begin{bmatrix} \hat{x}^0(t) \\ \hat{x}^1(t) \\ \vdots \\ \hat{x}^N(t) \end{bmatrix} + \mathbf{B_{exp,w}}\delta(t) \tag{4.43}$$

with the matrices

$$\mathbf{A_{exp}} = \sum_{j=0}^{1} \hat{a}^i \begin{bmatrix} \frac{\left\langle \phi^i(\xi)\phi^0(\xi), \phi^0(\xi)\right\rangle}{\left\|\phi^0(\xi)\right\|^2} & \frac{\left\langle \phi^i(\xi)\phi^0(\xi), \phi^1(\xi)\right\rangle}{\left\|\phi^0(\xi)\right\|^2} & \cdots & \frac{\left\langle \phi^i(\xi)\phi^0(\xi), \phi^N(\xi)\right\rangle}{\left\|\phi^0(\xi)\right\|^2} \\ \frac{\left\langle \phi^i(\xi)\phi^1(\xi), \phi^0(\xi)\right\rangle}{\left\|\phi^1(\xi)\right\|^2} & \frac{\left\langle \phi^i(\xi)\phi^1(\xi), \phi^1(\xi)\right\rangle}{\left\|\phi^1(\xi)\right\|^2} & \cdots & \frac{\left\langle \phi^i(\xi)\phi^1(\xi), \phi^N(\xi)\right\rangle}{\left\|\phi^1(\xi)\right\|^2} \\ \vdots & \vdots & & \vdots \\ \frac{\left\langle \phi^i(\xi)\phi^N(\xi), \phi^0(\xi)\right\rangle}{\left\|\phi^N(\xi)\right\|^2} & \frac{\left\langle \phi^i(\xi)\phi^N(\xi), \phi^1(\xi)\right\rangle}{\left\|\phi^N(\xi)\right\|^2} & \cdots & \frac{\left\langle \phi^i(\xi)\phi^N(\xi), \phi^N(\xi)\right\rangle}{\left\|\phi^N(\xi)\right\|^2} \end{bmatrix} \tag{4.44}$$

and

$$\mathbf{B}_{\mathbf{exp,w}} = \begin{bmatrix} \frac{1}{\phi^0(\xi)} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4.45}$$

The mean and variance trajectory approximations are calculated as:

$$\begin{aligned} \text{mean}_{PC}\ \text{x}(t) &= \left\|\phi^0(\xi)\right\| \hat{\text{x}}^0(t) \\ \text{var}_{PC}\ \text{x}(t) &= \sum_{i=1}^{N} \left\|\phi^i(\xi)\right\|^2 \left|\hat{\text{x}}^i(t)\right|^2 \\ \sigma_{PC}\ \text{x}(t) &= \sqrt{\text{var}_{PC}\ \text{x}(t)}. \end{aligned} \tag{4.46}$$

The error between the analytical and Polynomial Chaos approximate trajectories for mean and standard deviation can be defined as:

$$\begin{aligned} \text{Error}\left(\text{mean}_{\text{PC}}\ \text{x(t)}\right) &= \left|\text{mean}_{analyt}\ \text{x}(t) - \text{mean}_{PC}\ \text{x}(t)\right| \\ \text{Error}\left(\sigma_{\text{PC}}\ \text{x(t)}\right) &= \left|\sigma_{analyt}\ \text{x}(t) - \sigma_{PC}\ \text{x}(t)\right|. \end{aligned} \tag{4.47}$$

Here, the magnitude is used simply to make the trajectories positive. Plots of these error trajectories for the outlined cases, using first ($N = 1$) and second ($N = 2$) order approximations, are given in Figure 4.3.

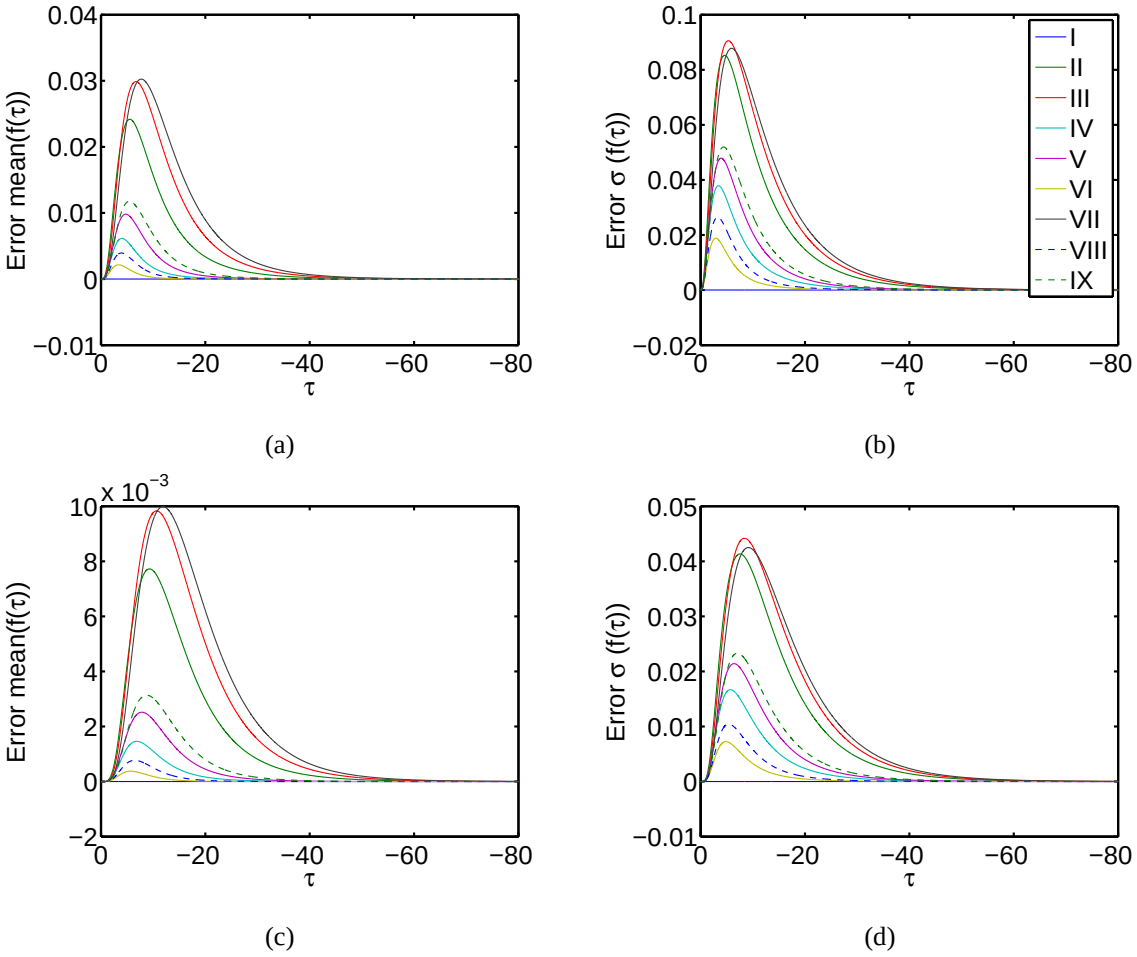
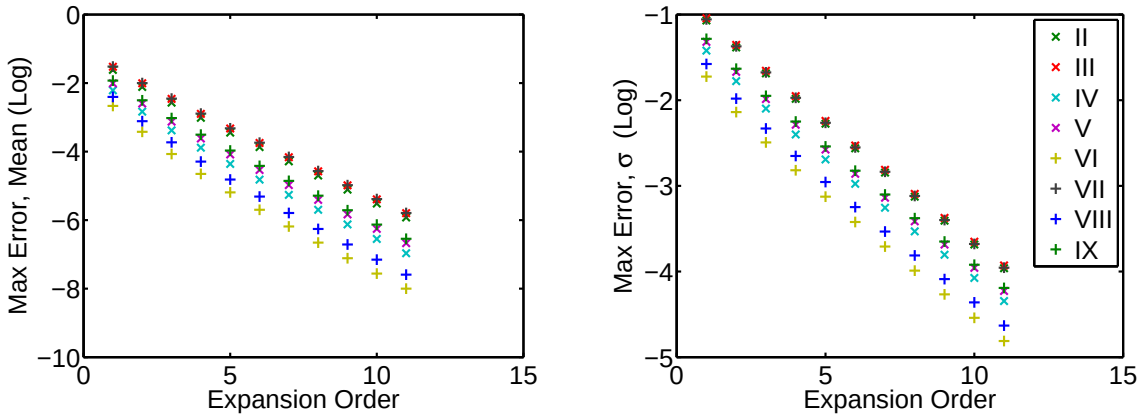


Figure 4.3: Error in the mean and standard deviation trajectories. The errors were obtained through comparing the approximation to the analytical solution. (a) Error in mean, order 1 PCE; (b) error in standard deviation, order 1 PCE; (c) error in mean, order 2 PCE; (d) error in standard deviation, order 2 PCE

As can be seen in these plots, the exponential convergence of the individual trajectories in the solution space creates a nice feature: both the approximated Polynomial Chaos trajectories and analytical trajectories converge towards zero, thereby effectively bounding the error propagation. This manifests itself as a peak. Hence, it made sense to look at the maximum values of the error trajectories. These are plotted in Figure 4.4 for Polynomial Chaos approximations of orders $N = 1$ to $N = 11$.



(a) (b)

Figure 4.4: Maximum error in the mean and standard deviation trajectories. These were obtained by comparing the PC approximations to the analytical solution. (a) Log_{10} of the maximum error in the mean trajectory; (b) Log_{10} of the maximum error in the standard deviation trajectory

These plots demonstrate a nearly exponential rate of convergence.

For comparison, the same process was performed using Monte Carlo simulations for Cases II and III. Calculations were performed with 10, 100, 1000, 10000, and 100000 samples. Additionally, to demonstrate the notion that the Monte Carlo estimates are probabilistic, every one of these tests was run ten times. The results for these series of tests are given in Figure 4.5.

It can be seen the general trend is the error decreases proportionally to the square root of the number of samples. In these two cases, to approach being equivalent to the 5th order PC approximation takes about 1000 samples for the standard deviation and 100000 for the mean.

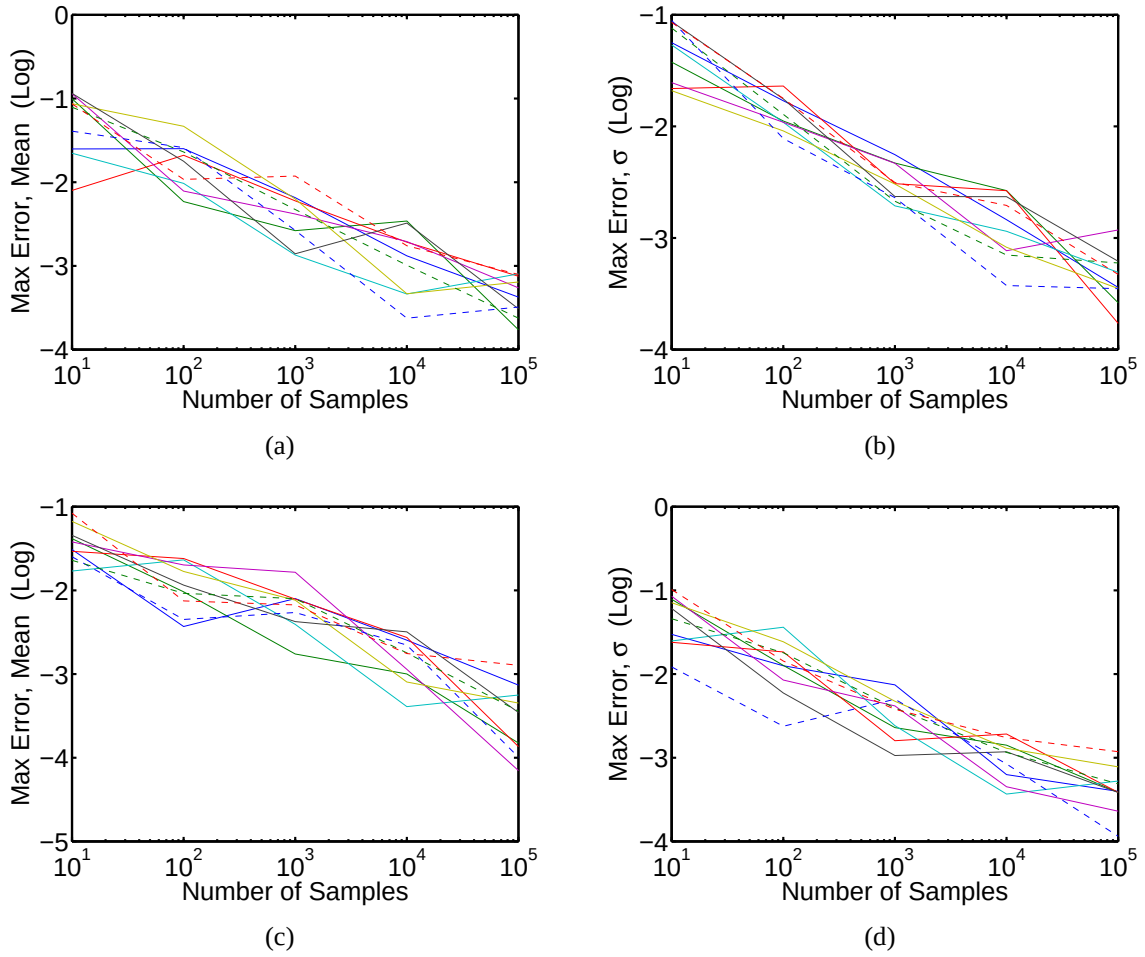


Figure 4.5: Maximum error in mean and standard deviation trajectories using Monte Carlo simulations. Ten independent tests were run at each denoted "Number of Samples." The error was obtained by comparing to the analytical solution. (a) Log_{10} of the maximum error in the mean, Case I; (b) Log_{10} of the maximum error in the standard deviation, Case II; (c) Log_{10} of the maximum error in the mean trajectory for Case III; (d) Log_{10} of the maximum error in the standard deviation trajectory for Case III

Finally it is interesting to use Lyapunov equations to calculate the expected $\mathcal{H}_2^2$ norm, as well as the square mean and variance trajectory components. These are calculated as

$$\begin{aligned} \mathbf{A_{exp}}^T\mathbf{P} + \mathbf{P}\mathbf{A_{exp}} + \mathbf{Q} = 0 \\ \mathcal{H}_2^2 = \text{trace}\, \mathbf{B_{exp,w}}^T\mathbf{P}\mathbf{B_{exp,w}} \end{aligned} \tag{4.48}$$

using the matrices defined in Equations (4.44) and (4.45). Also, different $\mathbf{Q}$ matrices allows obtaining the estimates for the integral over time of the expectation of the square trajectory, the variance trajectory, and the mean square trajectory:

$$\begin{aligned} \mathbf{Q}_E &= \text{diag}\left[\begin{array}{cccc} \left\|\phi^0(\xi)\right\|^2 & \left\|\phi^1(\xi)\right\|^2 & \cdots & \left\|\phi^N(\xi)\right\|^2 \end{array} \right] \\ \mathbf{Q}_{var} &= \text{diag}\left[\begin{array}{cccc} 0 & \left\|\phi^1(\xi)\right\|^2 & \cdots & \left\|\phi^N(\xi)\right\|^2 \end{array} \right] \\ \mathbf{Q}_{sq.mean} &= \mathbf{Q}_E - \mathbf{Q}_{var}. \end{aligned} \tag{4.49}$$

Error for these metrics was estimated by comparing to the 11-th order Polynomial Chaos approximation. Plots of the relative errors are given in Figure 4.6.

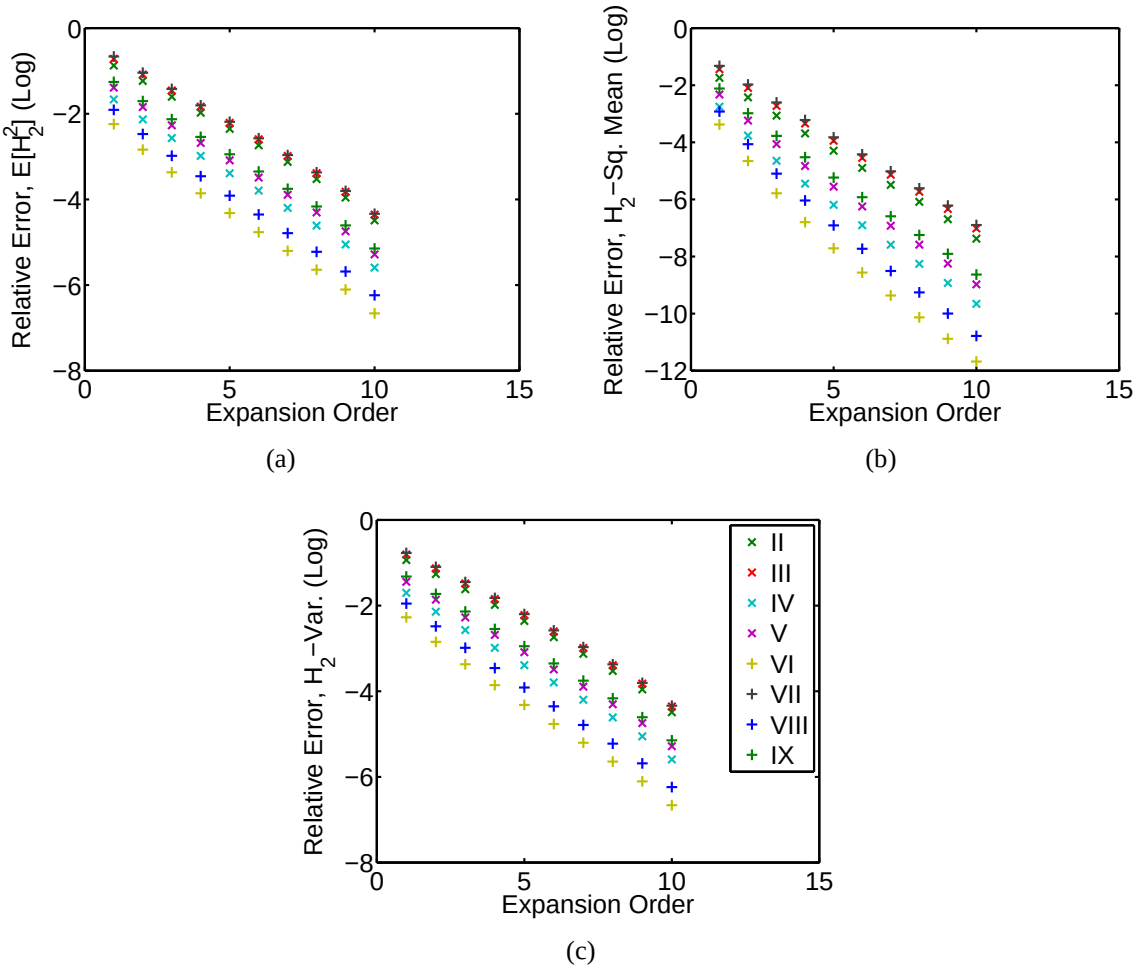


Figure 4.6: Error in $\mathcal{H}_2$ terms. These were obtained by comparing to the solution obtained from the 11-th order expansion. (a) Error in $E\left[\mathcal{H}_2^2\right]$; (b) error in square mean portion; (c) error in variance portion

4.8 Larger Dimension Systems

The next chapter looks at larger dimensional systems. Thus, it is useful to conclude this chapter by demonstrating the close ties between the analysis on the single state system and the larger, finite state system, described in the beginning by Equation (4.1). For stable systems of this type, with finite matrix elements, the *almost sure exponential stability* condition given by Equation (4.3) still applies. As such, for an impulse response, at all $t > 0$, each state is integrable with respect to the proper $L_{2,w}(\Omega)$ norm and therefore belongs within this Hilbert space. This allows knowing that in the limit, as the number of basis functions goes to infinity, the mean square converges, as well as can be separated into the square mean trajectories and variance trajectories. Additionally, keeping the two latter components bounded is necessary and sufficient for keeping the mean square bounded.

However, it will still be the case that in order to solve the system, it is necessary to approximate the expansion with a finite length expansion. As such, the method depends on the rapid convergence demonstrated in Section 4.7.

Finally, the section will be concluded by citing a theorem from Sard [147], p. 355.

Theorem 4.8.1. *Let $\left\{\phi_1^i(\xi_1)\right\}$ and $\left\{\phi_2^j(\xi_2)\right\}$ form bases for the Hilbert spaces H_1 and H_2, respectively. Both of these basis should have a countable number of terms. Then the cartesian product of the sets of basis functions, $\left\{\phi_1^i(\xi_1) \times \phi_2^j(\xi_2)\right\}$, is a basis for the Hilbert space $H = H_1 \times H_2$.*

By induction, this process can be used to combine a finite number of bases. This is important since it rigorously justifies combining orthogonal basis functions, such as by the Askey scheme described in Section 2.7, to span multiple dimensions along the parameter space.

Chapter 5

$\mathcal{H}_2$ and LQR Design utilizing Polynomial Chaos

This chapter introduces a generalized state space approach to considering uncertainty in linear state space systems. This derivation, while based on the theory of the previous chapter, takes a more utilitarian approach. A standard set of nomenclature is given for use throughout the chapter, the generalized state space form is derived, a realization matrix is discussed, applicability to the $\mathcal{H}_2$ problem is demonstrated, and examples are shown. The Matlab code for utilized libraries are given in Appendix A.

5.1 Notation

Problems utilizing Polynomial Chaos can get very complicated in terms of notation. Thus, a standardized list of nomenclature is provided as a reference for throughout the chapter.

Nomenclature

$\mathbf{A}$ State Matrix, dim $M \times M$

$\mathbf{A_{exp}}$ Expanded State Matrix, dim $M\,(L+1) \times M\,(L+1)$

$\mathbf{A_{exp,cl}}$ Expanded State Matrix for the closed-loop expanded system, dim $M\,(L+1) \times M\,(L+1)$

$\mathbf{B}$ Input Matrix, dim $M \times N$

$\mathbf{B_{exp}}$ Expanded Input Matrix, dim $M\,(L+1) \times N\,(L+1)$

$\mathbf{B_{exp,u}}$ Expanded Input Matrix specifically for control inputs, dim $M\,(L+1) \times N\,(L+1)$

$\mathbf{B_{exp,w}}$ Expanded Input Matrix specifically for exogenous inputs

$\mathbf{B_{u}}$ Input Matrix specifically for control inputs, dim $M \times N$

$\mathbf{B_{w}}$ Input Matrix specifically for exogenous inputs, dim $M \times Q$

$\mathbf{C}$ Output Matrix, dim $P \times M$

$\mathbf{C_{exp}}$ Output Matrix, dim $P\,(L+1) \times M\,(L+1)$

$\underline{\mathbf{C}}_{\mathbf{Q}}$ Associated covariance matrix with $\mathbf{Q}$

$\underline{\mathbf{C}}_{\mathbf{R}}$ Associated covariance matrix with $\mathbf{R}$

E Expectation Operator

F Matrix of inner product terms, dim (L+1) × (L+1)

f, g Functions

$\hat{f}, \hat{g}$ Coefficients for Polynomial Chaos Expansions

H Realization Matrix, dim P × P (L + 1)

I Identity matrix, dim specified in subscript

J Cost function

K Gain Matrix, dim N × M

$\mathbf{K_{exp}}$ Expanded Gain Matrix, dim N (L + 1) × M (L + 1)

L Basis functions sets are truncated to order L and numbered 0 to L

$\underline{\mathbf{M}}_{\mathbf{Q}}$ Associated mean matrix with **Q**

$\underline{\mathbf{M}}_{\mathbf{R}}$ Associated mean matrix with **R**

P A real positive definite and symmetric matrix, dim M (L + 1) × M (L + 1)

Q Output weighting matrix

R Feedback weighting matrix

r Number of random variables

S Defined as $\mathbf{P}^{-1}$ and thus also a real positive definite and symmetric matrix, dim M (L + 1) × M (L + 1)

$\mathbf{S_s}$ A real positive definite and symmetric matrix, dim M × M

u Input Vector, u(t) implied, dim N

u_{exp} Expanded Input Vector, $u_{exp}(t)$ implied, dim N (L + 1)

$\mathcal{U}_{exp}$ A vector resulting from multiplying u_{exp} by the matrix square root of a weighting matrix

W Maximum order for one-dimensional polynomial basis functions

w Weighting function for orthogonality, also used to represent a PDF

w Exogenous Input Vector, dim Q

w_{exp} Expanded Exogenous Input Vector, dim Q if w is not a function of ξ; dim Q (L + 1) if w is a function of ξ.

x State Vector, x(t) implied, dim M

x_{exp} Expanded State Vector, $x_{exp}(t)$ implied, dim M (L + 1)

$x_{exp,i}(t)$ Expanded State Vector representing the response from an impulse into the i-th input, dim M (L + 1)

$\mathbf{Y}$ Created such that $\mathbf{Y} = \mathbf{K_{exp}S}$

$\mathbf{Y_s}$ Created such that $\mathbf{Y_s} = \mathbf{KS_s}$

y Output Vector, y(t) implied, dim P

y_{exp} Expanded Output Vector, $y_{exp}(t)$ implied, dim P (L + 1)

$\mathcal{Y}_{exp}$ A vector resulting from multiplying y_{exp} by the matrix square root of a weighting matrix

$\mathbf{Z}$ A free real matrix variable

δ_i The dirac delta used as an inpulse inputted into the i-th input.

δ_{ij} Inner Product of the i-th and j-th basis functions

δ_{ijk} "Inner product" of the i-th, j-th, and k-th basis functions

μ A scalar term relating to the step size used in a gradient descent optimization

Ξ An element or realization of ξ

ξ Random variable; can also denote a vector of random variables

ϕ^k k-th basis function of an orthogonal set

ψ^k k-th basis function of a multi-dimensional orthogonal set

Ω Domain for random variables

5.2 Problem Description

Let's consider a system modeled by a probabilistic set of linear equations of the form:

$$\begin{aligned} \dot{\mathrm{x}}(t,\xi) &= \mathbf{A}(\xi)\mathrm{x}(t,\xi) + \mathbf{B_u}(\xi)\mathrm{u}(t,\xi) + \mathbf{B_w}(\xi)\mathrm{w}(t,\xi) \\ \mathrm{y}(t,\xi) &= \mathbf{C}(\xi)\mathrm{x}(t,\xi). \end{aligned} \tag{5.1}$$

Here $\mathrm{w}(t,\xi)$ is the exogenous input.

Thus, state space realizations are a function of one or more random parameters. Note that while ξ could represent one or multiple random variables, in a strict sense, none of them vary with time. One place that a probabilistic framework makes sense is for a set of similar systems, such as many realizations of the same item created on an assembly line or natural variation in biological systems.

The states are functions of the random variables since they are dependent on the state matrices. Also, observe that the input can be a function of the random parameters as well. This is significant if the input is a function of the states; i.e. in feedback control.

Additionally, in a less strict sense, if a parameter varies slowly enough, then a quasi-stationary argument can be invoked. In this case it is assumed that the dynamics of the system are mainly governed by the values of the parameters at any instant and not by the change in the values of the parameters. An example could be parameters varying slowly due to external thermal fluctuations.

5.3 Polynomial Chaos General State Space Derivation

In this section, a detailed derivation of the expanded state equation is given. It can be seen that deriving the output and state feedback equations follows the same form and thus the derivation is omitted. Also, for the purpose of the derivation, w is explicitly not a function of ξ in order to make the derivation more interesting. If it is a function of ξ, then the section of the equations with $\mathbf{B_w}(\xi)\mathrm{w}(t,\xi)$ proceeds in the same manner as given for $\mathbf{B_u}(\xi)\mathrm{u}(t,\xi)$.

To begin, the state equation is given to be of the following form:

$$\dot{\mathrm{x}}(t,\xi) = \mathbf{A}(\xi)\mathrm{x}(t,\xi) + \mathbf{B_u}(\xi)\mathrm{u}(t,\xi) + \mathbf{B_w}(\xi)\mathrm{w}(t). \tag{5.2}$$

Next, PCEs in terms of orthogonal basis functions are substituted in:

$$\sum_{j=0}^{\mathrm{L}} \hat{x}^j(t)\phi^j(\xi) = \sum_{i=0}^{\mathrm{L}} \hat{\mathbf{A}}^i\phi^i(\xi) \sum_{j=0}^{\mathrm{L}} \hat{\mathrm{x}}^j(t)\phi^j(\xi) + \sum_{i=0}^{\mathrm{L}} \hat{\mathbf{B}_\mathbf{u}}^{\,i}\phi^i(\xi) \sum_{j=0}^{\mathrm{L}} \hat{\mathrm{u}}^j(t)\phi^j(\xi) + \sum_{j=0}^{\mathrm{L}} \hat{\mathbf{B}_\mathbf{w}}^{\,j}\phi^j(\xi)\mathrm{w}(t). \tag{5.3}$$

By rearranging into double summations:

$$\sum_{j=0}^{\mathrm{L}} \hat{\mathrm{x}}^j(t)\phi^j(\xi) = \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}} \hat{\mathbf{A}}^i \hat{\mathrm{x}}^j(t)\phi^i(\xi)\phi^j(\xi) + \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}} \hat{\mathbf{B}}_{\mathbf{u}}{}^i \hat{\mathrm{u}}^j(t)\phi^i(\xi)\phi^j(\xi) + \sum_{j=0}^{\mathrm{L}} \hat{\mathbf{B}}_{\mathbf{w}}{}^j \mathrm{w}(t)\phi^j(\xi). \quad (5.4)$$

Following, the Galerkin projection is used to project the equation onto the k-th basis function, $\phi^k(\xi)$:

$$\begin{aligned} &\left\langle \sum_{j=0}^{\mathrm{L}} \hat{\mathrm{x}}^j(t)\phi^j(\xi), \phi^k(\xi) \right\rangle \\ &= \left\langle \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}} \hat{\mathbf{A}}^i \hat{\mathrm{x}}^j(t)\phi^i(\xi)\phi^j(\xi) + \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}} \hat{\mathbf{B}}_{\mathbf{u}}{}^i \hat{\mathrm{u}}^j(t)\phi^i(\xi)\phi^j(\xi) + \sum_{j=0}^{\mathrm{L}} \hat{\mathbf{B}}_{\mathbf{w}}{}^j \mathrm{w}(t)\phi^j(\xi), \phi^k(\xi) \right\rangle. \end{aligned} \quad (5.5)$$

Linearity is used to "distribute" the inner product:

$$\begin{aligned} \sum_{j=0}^{\mathrm{L}} \hat{\mathrm{x}}^j(t) \left\langle \phi^j(\xi), \phi^k(\xi) \right\rangle &= \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}} \hat{\mathbf{A}}^i \mathrm{x}^j(t) \left\langle \phi^i(\xi)\phi^j(\xi), \phi^k(\xi) \right\rangle \\ &+ \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}} \hat{\mathbf{B}}_{\mathbf{u}}{}^i \hat{\mathrm{u}}^j(t) \left\langle \phi^i(\xi)\phi^j(\xi), \phi^k(\xi) \right\rangle + \sum_{j=0}^{\mathrm{L}} \hat{\mathbf{B}}_{\mathbf{w}}{}^j \mathrm{w}(t) \left\langle \phi^j(\xi), \phi^k(\xi) \right\rangle. \end{aligned} \quad (5.6)$$

Next, orthogonality is utilized as well as the δ notation to represent the inner products:

$$\hat{\mathrm{x}}^k(t)\delta_{kk} = \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}} \hat{\mathbf{A}}^i \hat{\mathrm{x}}^j(t)\delta_{ijk} + \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}} \hat{\mathbf{B}}_{\mathbf{u}}{}^i \hat{\mathrm{u}}^j(t)\delta_{ijk} + \hat{\mathbf{B}}_{\mathbf{w}}{}^k \mathrm{w}(t)\delta_{kk}. \quad (5.7)$$

It is now possible to express one of the summations in terms of the Kronecker product and matrix multiplication:

$$\hat{\mathrm{x}}^k(t)\delta_{kk} = \sum_{i=0}^{\mathrm{L}} \left[\mathbf{1}_{1\times \mathrm{M}} \otimes \hat{\mathbf{A}}^i\right] \begin{bmatrix} \hat{\mathrm{x}}^0(t)\delta_{i0k} \\ \hat{\mathrm{x}}^1(t)\delta_{i1k} \\ \vdots \\ \hat{\mathrm{x}}^{\mathrm{L}}(t)\delta_{i\mathrm{L}k} \end{bmatrix} + \sum_{i=0}^{\mathrm{L}} \left[\mathbf{1}_{1\times \mathrm{M}} \otimes \hat{\mathbf{B}}_{\mathbf{u}}{}^i\right] \begin{bmatrix} \hat{\mathrm{u}}^0(t)\delta_{i0k} \\ \hat{\mathrm{u}}^1(t)\delta_{i1k} \\ \vdots \\ \hat{\mathrm{u}}^{\mathrm{L}}(t)\delta_{i\mathrm{L}k} \end{bmatrix} + \hat{\mathbf{B}}_{\mathbf{w}}{}^k \mathrm{w}(t)\delta_{kk}. \quad (5.8)$$

The equation is rearranged:

$$\begin{aligned}\hat{\dot{x}}^k(t) = \sum_{i=0}^{\mathrm{L}} \left[\begin{bmatrix} \frac{\delta_{i0k}}{\delta_{kk}} & \frac{\delta_{i1k}}{\delta_{kk}} & \dots & \frac{\delta_{i\mathrm{L}k}}{\delta_{kk}} \end{bmatrix} \otimes \hat{\mathbf{A}}^i \right] \begin{bmatrix} \mathrm{x}^0(t) \\ \mathrm{x}^1(t) \\ \vdots \\ \mathrm{x}^{\mathrm{L}}(t) \end{bmatrix} \\ + \sum_{i=0}^{\mathrm{L}} \left[\begin{bmatrix} \frac{\delta_{i0k}}{\delta_{kk}} & \frac{\delta_{i1k}}{\delta_{kk}} & \dots & \frac{\delta_{i\mathrm{L}k}}{\delta_{kk}} \end{bmatrix} \otimes \hat{\mathbf{B}}_{\mathbf{u}}{}^i \right] \begin{bmatrix} \hat{\mathrm{u}}^0(t) \\ \hat{\mathrm{u}}^1(t) \\ \vdots \\ \hat{\mathrm{u}}^{\mathrm{L}}(t) \end{bmatrix} + \hat{\mathbf{B}}_{\mathbf{w}}{}^k \mathrm{w}(t).\end{aligned} \tag{5.9}$$

Finally, all of the "k-th" projections for the zeroth to the L-th are combined:

$$\begin{aligned}\begin{bmatrix} \hat{\dot{x}}^0(t) \\ \hat{\dot{x}}^1(t) \\ \vdots \\ \hat{\dot{x}}^{\mathrm{L}}(t) \end{bmatrix} = \sum_{i=0}^{\mathrm{L}} \left[\begin{bmatrix} \frac{\delta_{i00}}{\delta_{00}} & \frac{\delta_{i10}}{\delta_{00}} & \dots & \frac{\delta_{i\mathrm{L}0}}{\delta_{00}} \\ \frac{\delta_{i01}}{\delta_{11}} & \frac{\delta_{i11}}{\delta_{11}} & \dots & \frac{\delta_{i\mathrm{L}1}}{\delta_{11}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta_{i0\mathrm{L}}}{\delta_{\mathrm{LL}}} & \frac{\delta_{i1\mathrm{L}}}{\delta_{\mathrm{LL}}} & \dots & \frac{\delta_{i\mathrm{LL}}}{\delta_{\mathrm{LL}}} \end{bmatrix} \otimes \hat{\mathbf{A}}^i \right] \begin{bmatrix} \hat{\mathrm{x}}^0(t) \\ \hat{\mathrm{x}}^1(t) \\ \vdots \\ \hat{\mathrm{x}}^{\mathrm{L}}(t) \end{bmatrix} \\ + \sum_{i=0}^{\mathrm{L}} \left[\begin{bmatrix} \frac{\delta_{i00}}{\delta_{00}} & \frac{\delta_{i10}}{\delta_{00}} & \dots & \frac{\delta_{i\mathrm{L}0}}{\delta_{00}} \\ \frac{\delta_{i01}}{\delta_{11}} & \frac{\delta_{i11}}{\delta_{11}} & \dots & \frac{\delta_{i\mathrm{L}1}}{\delta_{11}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta_{i0\mathrm{L}}}{\delta_{\mathrm{LL}}} & \frac{\delta_{i1\mathrm{L}}}{\delta_{\mathrm{LL}}} & \dots & \frac{\delta_{i\mathrm{LL}}}{\delta_{\mathrm{LL}}} \end{bmatrix} \otimes \hat{\mathbf{B}}_{\mathbf{u}}{}^i \right] \begin{bmatrix} \hat{\mathrm{u}}^0(t) \\ \hat{\mathrm{u}}^1(t) \\ \vdots \\ \hat{\mathrm{u}}^{\mathrm{L}}(t) \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{B}}_{\mathbf{w}}{}^0 \\ \hat{\mathbf{B}}_{\mathbf{w}}{}^1 \\ \vdots \\ \hat{\mathbf{B}}_{\mathbf{w}}{}^{\mathrm{L}} \end{bmatrix} \mathrm{w}(t).\end{aligned} \tag{5.10}$$

As a side note, in order to see the structural similarity of the right-hand term to the rest of the equation, it is interesting to rewrite the term as

$$\begin{bmatrix} \hat{\mathbf{B}}_{\mathbf{w}}{}^{0} \\ \hat{\mathbf{B}}_{\mathbf{w}}{}^{1} \\ \dots \\ \hat{\mathbf{B}}_{\mathbf{w}}{}^{\mathrm{L}} \end{bmatrix} = \sum_{i=0}^{\mathrm{L}} \left[\begin{bmatrix} \frac{\delta_{i00}}{\delta_{00}} \\ \frac{\delta_{i01}}{\delta_{11}} \\ \dots \\ \frac{\delta_{i0\mathrm{L}}}{\delta_{\mathrm{LL}}} \end{bmatrix} \otimes \hat{\mathbf{B}}_{\mathbf{w}}{}^{i} \right]. \tag{5.11}$$

This equality assumes a typical convention for orthogonal polynomials that $\phi^i(\xi) = 1$. If this is not true, adapting the expression is simply a matter of correctly normalizing the right hand side.

5.4 Applying Polynomial Chaos

It can now be seen that Equations 5.1 can be expanded in terms of basis functions of the random parameters by a relatively easy procedure—hence bypassing the intricacies of the Galerkin projection. All of the state matrices are assumed to be expanded in the following form:

$$\mathbf{A}(\xi) = \sum_{i=0}^{\mathrm{L}} \hat{\mathbf{A}}^i \phi^i(\xi). \tag{5.12}$$

Additionally, the state, input, and output vectors can be expanded as follows:

$$\mathrm{x}(t,\xi) = \sum_{i=0}^{\mathrm{L}} \hat{\mathrm{x}}^i(t)\phi^i(\xi), \;\; \mathrm{u}(t,\xi) = \sum_{i=0}^{\mathrm{L}} \hat{\mathrm{u}}^i(t)\phi^i(\xi), \;\; \text{and} \;\; \mathrm{y}(t,\xi) = \sum_{i=0}^{\mathrm{L}} \hat{\mathrm{y}}^i(t)\phi^i(\xi). \tag{5.13}$$

Further, if created as a function of a random variable, the exogenous input can be expanded in this form as well. Note that the explicit reference to time will be dropped in the future for the purpose notational convenience. Also, the common notation of an equality is now being used in favor of the strictly correct approximation given in Equation (2.18).

One way to solve the given problem is to project the basis functions onto the state equations using a Galerkin projection. This consists of applying the inner product to both sides to each of the M state equations and P output equations with respect to each of the L + 1 basis functions. This process was performed in the previous section. The result is the following expanded system (see the Nomenclature for dimensions):

$$\begin{aligned} \dot{\mathrm{x}}_{\mathrm{exp}} &= \mathbf{A_{exp}}\mathrm{x}_{\mathrm{exp}} + \mathbf{B_{exp,u}}\mathrm{u}_{\mathrm{exp}} + \mathbf{B_{exp,w}}\mathrm{w}_{\mathrm{exp}} \\ \mathrm{y}_{\mathrm{exp}} &= \mathbf{C_{exp}}\mathrm{x}_{\mathrm{exp}}. \end{aligned} \tag{5.14}$$

Further, convenient forms for the expanded system can be written in terms of the Kronecker product (similar to one first published in [3]):

$$\mathbf{A_{exp}} = \sum_{i=0}^{\mathrm{L}} \mathbf{F}^i \otimes \hat{\mathbf{A}}^i \tag{5.15}$$

$$\mathbf{B_{exp,u}} = \sum_{i=0}^{\mathrm{L}} \mathbf{F}^i \otimes \hat{\mathbf{B}_{\mathbf{u}}}^i \tag{5.16}$$

where

$$\mathbf{F}^i = \begin{bmatrix} \frac{\delta_{i00}}{\delta_{00}} & \frac{\delta_{i10}}{\delta_{00}} & \cdots & \frac{\delta_{iL0}}{\delta_{00}} \\ \frac{\delta_{i10}}{\delta_{11}} & \frac{\delta_{i11}}{\delta_{11}} & \cdots & \frac{\delta_{iL1}}{\delta_{11}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta_{iL0}}{\delta_{LL}} & \frac{\delta_{iL1}}{\delta_{LL}} & \cdots & \frac{\delta_{iLL}}{\delta_{LL}} \end{bmatrix} \tag{5.17}$$

Similarly, the **C** matrix can be represented in the same form such that $\mathbf{C_{exp}} = \sum_{i=0}^{\mathrm{L}} \mathbf{F}^i \otimes \hat{\mathbf{C}}^i$.

This implies that the states are ordered such that

$$x_{exp} = \begin{bmatrix} \hat{x}^0 \\ \hat{x}^1 \\ \vdots \\ \hat{x}^L \end{bmatrix}, \quad u_{exp} = \begin{bmatrix} \hat{u}^0 \\ \hat{u}^1 \\ \vdots \\ \hat{u}^L \end{bmatrix}, \text{ and } \quad y_{exp} = \begin{bmatrix} \hat{y}^0 \\ \hat{y}^1 \\ \vdots \\ \hat{y}^L \end{bmatrix}. \tag{5.18}$$

These equations are vectors of coefficient vectors defined in Equation (5.13).

The $\mathbf{B_{exp,w}}$ matrix is a special case. It can always take the form of the type given in Equation (5.15). If $w(t, \xi)$ is used, then this form should be used as well as letting $w_{exp} = \begin{bmatrix} w^0(t)^T & w^1(t)^T & \dots & w^L(t)^T \end{bmatrix}^T$. However, if $w(t)$ is used, then it is possible to let $w_{exp} = \begin{bmatrix} \hat{w}^0(t)^T & 0^T_{Q\times 1} & \dots & 0^T_{Q\times 1} \end{bmatrix}^T$. Alternately, in this case, it is possible to use the simplified form of $\mathbf{B_{exp,w}}$ and $w_{exp}(t) = w(t)$, as was derived in Section 5.3.

Additionally, it can easily be seen that $\mathbf{F}^0 = \mathbf{I}_{L+1}$. Therefore, if any non-expanded state matrices are completely deterministic, and thus not a function of ξ, the expanded form is significantly simplified. For example, if $\mathbf{C}$ is not a function of ξ, then $\mathbf{C_{exp}} = \mathbf{I}_{L+1} \otimes \mathbf{C}$. This form is true for all of the other state matrices as well.

5.5 The Realization Matrix, H

Further, it is instructive to define a new output equation, similar in spirit to a concept presented in [148]:

$$y(\Xi) = \mathbf{H}(\Xi) y_{exp} \tag{5.19}$$

where

$$\mathbf{H}(\Xi) = \left[\begin{bmatrix} \phi^0(\Xi) & \phi^1(\Xi) & \cdots & \phi^L(\Xi) \end{bmatrix} \otimes \mathbf{I}_P \right]. \tag{5.20}$$

Fixing a value (or values) of $\Xi \in \Omega$, the defined Realization Matrix, $\mathbf{H}(\Xi)$, transforms a "flow" of the spectrally represented solution space captured by the PCEs to the approximation for that fixed realization.

As discussed in the previous chapter, the spectral approximation is not guaranteed to converge point-wise. However, in general, as long the parameters are smoothly defined, the spectral approximation often can be taken to be a very good representation of the system.

5.6 Realizations

For any matrix expanded in the method of Equations (5.15) and (5.16), a deterministic realization of this matrix can be found:

$$\mathbf{A}(\xi) = \mathbf{H}(\xi)\mathbf{A}_{\mathbf{exp}}\left[[1, 0^1, 0^2, \ldots, 0^{\mathrm{L}}]^T \otimes \mathbf{I}_{\mathrm{M}}\right]. \tag{5.21}$$

This can be verified by substituting in the definitions of the matrices given in Equations (5.15) and (5.20). Looking at the first two matrix terms:

$$\begin{aligned}
\mathbf{H}(\xi)\mathbf{A}_{\mathbf{exp}} &= \left[\left[\begin{array}{cccc} \phi^0(\xi) & \phi^1(\xi) & \cdots & \phi^{\mathrm{L}}(\xi) \end{array}\right] \otimes \mathbf{I}_{\mathrm{M}}\right]\left[\sum_{i=0}^{\mathrm{L}} \mathbf{F}^i \otimes \hat{\mathbf{A}}^i\right] \\
&= \sum_{i=0}^{\mathrm{L}}\left[\left[\left[\begin{array}{cccc} \phi^0(\xi) & \phi^1(\xi) & \cdots & \phi^{\mathrm{L}}(\xi) \end{array}\right]\mathbf{F}^i\right] \otimes \hat{\mathbf{A}}^i\right] \\
&= \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}}\left[\left[\begin{array}{cccc} \frac{\delta^{ij0}}{\delta^{jj}}\phi^j(\xi) & \frac{\delta^{ij1}}{\delta^{jj}}\phi^j(\xi) & \cdots & \frac{\delta^{ij\mathrm{L}}}{\delta^{jj}}\phi^j(\xi) \end{array}\right] \otimes \hat{\mathbf{A}}^i\right].
\end{aligned} \tag{5.22}$$

Bringing in the remaining terms confirms that the original expansion is recreated:

$$\begin{aligned}\mathbf{A}(\xi) &= \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}}\left[\left[\begin{array}{cccc}\frac{\delta^{ij0}}{\delta jj}\phi^j(\xi) & \frac{\delta^{ij1}}{\delta jj}\phi^j(\xi) & \cdots & \frac{\delta^{ij\mathrm{L}}}{\delta jj}\phi^j(\xi)\end{array}\right]\otimes\hat{\mathbf{A}}^i\right]\left[[1,0^1,0^2,\ldots,0^{\mathrm{L}}]^T\otimes\mathbf{I}_{\mathrm{M}}\right] \\ &= \sum_{i=0}^{\mathrm{L}}\sum_{j=0}^{\mathrm{L}}\left[\left(\frac{\delta^{ij0}}{\delta jj}\phi^j(\xi)\right)\hat{\mathbf{A}}^i\right] = \sum_{i=0}^{\mathrm{L}}\hat{\mathbf{A}}^i\phi^i(\xi).\end{aligned} \tag{5.23}$$

Thus, evaluating $\mathbf{H}(\xi)$ can serve as a convenient way to evaluate realizations of PCEs. If $\mathbf{A}(\xi)$ was expanded and truncated such that it is a least-squares approximation, obviously only this approximation will be recaptured. However, if the series used is exact, such as in the case of the matrix being defined as a finite length expansion, then the algebraic manipulation shows that the exact finite length expansion is recreated.

The collocation method can be loosely thought of as an inverse problem to finding realizations of the Polynomial Chaos system. Simply stated, given a set of realizations, it is desired to find the expanded Polynomial Chaos states.

We define a set of collocation points, $[\Xi_1,\Xi_2,\ldots]\in\xi$. For any one of these realizations, the following is true to within the spectral approximation (i.e. the first realization):

$$\mathrm{y}(\Xi_1) = \mathbf{H}(\Xi_1)\mathrm{y}_{\mathrm{exp}}. \tag{5.24}$$

Additionally, the same is obviously true for state information:

$$\mathrm{x}(\Xi_1) = \mathbf{H}(\Xi_1)\mathrm{x}_{\mathrm{exp}}. \tag{5.25}$$

However, in each of these cases, L + 1 distinct realizations are necessary in order to uniquely determine the expansion.

Rephrasing Equation (5.25) with L + 1 distinct realizations yields the following:

$$x_{exp} = \begin{bmatrix} \mathbf{H}(\Xi_1) \\ \mathbf{H}(\Xi_2) \\ \mathbf{H}(\Xi_3) \\ \vdots \\ \mathbf{H}(\Xi_{L+1}) \end{bmatrix}^{-1} \begin{bmatrix} x(\Xi_1) \\ x(\Xi_2) \\ x(\Xi_3) \\ \vdots \\ x(\Xi_{L+1}) \end{bmatrix}. \tag{5.26}$$

Note that the inverted matrix is constant once the collocation points have been chosen. Additionally, it is important to point out that there has been substantial work into the selection of collocation points.

This connection is significant in that it is much easier to simulate realizations of nonlinear systems than collocation schemes using the Galerkin projection method. However, for the desired fast rate of convergence, it is still important for the solution space represented by the polynomial sets to be C_1.

5.7 $\mathcal{H}_2$ Formulation

Formulating the $\mathcal{H}_2$ control problem starts with the system model defined in Equation (5.14). Note that in many cases, the exogenous input may not be a function of a random variable and thus just $w(t)$. However, it is written in the more general form since there are examples in the literature, e.g. [149], where excitations are made to be a function of random variables.

Also, for the approach taken in this dissertation, the feedback is made to be independent of the uncertain parameters. Thus the feedback is expressed as

$$\mathrm{u}(t,\xi) = \mathbf{K}\mathrm{x}(t,\xi), \tag{5.27}$$

where $\mathbf{K}$ is explicitly not a function of ξ. From a control design standpoint, this has important implications in that the controller is not required to "know" the state of the uncertain parameters in order to control the system. Rather, the approach taken is to find a controller that is robust in regards to being stable within the region of support for the parameter distributions as well as optimal with respect to the PDFs chosen for the parameters. For the infinite length series expansions, arguments were provided in the previous chapter for being able to link stability to PCEs. Further, as discussed in Section 4.7, PCEs tend to converge quickly. Thus, it is assumed that the finite length PCEs hold the same properties and as such generally ensure stability. However, a statistical check is provided at the end of the Chapter to allow testing this claim.

To begin down this path, the probabilistic system model, given in Equation (5.14), is rewritten in closed loop form as

$$\begin{bmatrix} \dot{\mathrm{x}}(t,\xi) \\ \mathrm{y}(t,\xi) \\ \mathrm{u}(t,\xi) \\ \mathcal{Y}(t,\xi) \\ \mathcal{U}(t,\xi) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\xi) + \mathbf{B_u}(\xi)\mathbf{K} & \mathbf{B_w}(\xi) \\ \mathbf{C}(\xi) & \mathbf{0} \\ \mathbf{K} & \mathbf{0} \\ (\mathbf{Q})^{1/2}\,\mathbf{C}(\xi) & \mathbf{0} \\ (\mathbf{R})^{1/2}\,\mathbf{K} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathrm{x}(t,\xi) \\ \mathrm{w}(t,\xi) \end{bmatrix}. \tag{5.28}$$

Here, the states $\mathcal{U}(t,\xi)$ and $\mathcal{Y}(t,\xi)$ are created for the purpose of weighting the outputs and the feedback.

Next, an $\mathcal{H}_2$ cost function in terms of weighted states can be created as

$$\begin{aligned} \mathrm{J}(\xi) = \|H\|_2^2(\xi) &= \int_0^\infty \mathcal{Y}(t,\xi)^T \mathcal{Y}(t,\xi) + \mathcal{U}(t,\xi)^T \mathcal{U}(t,\xi)\, dt \\ &= \int_0^\infty \mathrm{y}(t,\xi)^T \mathbf{Q} \mathrm{y}(t,\xi) + \mathrm{u}(t,\xi)^T \mathbf{R} \mathrm{u}(t,\xi)\, dt, \end{aligned} \tag{5.29}$$

where **Q** is positive semidefinite, **R** is positive definite, and both are symmetric. Notice the relation to the weighted L_2 norm created through squaring the state vectors.

In order to enforce the approach of optimality with respect to the probability distributions for the parameters in the model, the expectation

$$E(\mathrm{J}(\xi)) \equiv \int_\Omega \mathrm{J}(\xi) w(\xi)\, d\xi = \int_\Omega \int_0^\infty \mathrm{y}(\xi)^T \mathbf{Q} \mathrm{y}(\xi) + \mathrm{u}(\xi)^T \mathbf{R} \mathrm{u}(\xi)\, dt\, w(\xi)\, d\xi \tag{5.30}$$

is taken. This has the effect of weighting the cost by the probability of a realization occurring.

Next, it is possible to substitute in PCEs from Equation (5.13), into Equation (5.30), and rearrange the equation as

$$\begin{aligned} \int_\Omega \mathrm{J}(\xi) w(\xi)\, d\xi = \int_0^\infty \int_\Omega &\left[\sum_{j=0}^{\mathrm{L}} \hat{\mathrm{y}}^j \phi^j(\xi)\right]^T \mathbf{Q} \left[\sum_{j=0}^{\mathrm{L}} \hat{\mathrm{y}}^j \phi^j(\xi)\right] w(\xi) \\ &+ \left[\sum_{j=0}^{\mathrm{L}} \hat{\mathrm{u}}^j \phi^j(\xi)\right]^T \mathbf{R} \left[\sum_{j=0}^{\mathrm{L}} \hat{\mathrm{u}}^j \phi^j(\xi)\right] w(\xi)\, d\xi\, dt. \end{aligned} \tag{5.31}$$

This represents the transformation from the sum of the square of the states to the sum of the square of the expansions. Since the right-hand side is in the form of the inner product, orthogonality can be utilized to produce the following result:

$$\int_\Omega \mathrm{J}(\xi) w(\xi)\, d\xi = \int_0^\infty \sum_{j=0}^{\mathrm{L}} \left[\hat{\mathrm{y}}^j\right]^T \delta_{jj} \mathbf{Q} \left[\hat{\mathrm{y}}^j\right] + \sum_{j=0}^{\mathrm{L}} \left[\hat{\mathrm{u}}^j\right]^T \delta_{jj} \mathbf{R} \left[\hat{\mathrm{u}}^j\right] dt. \tag{5.32}$$

The result is related to one obtained in [3].

This equation can then be written in terms of Kronecker products and the expanded vectors as

$$\begin{aligned}\int_{\Omega} \mathrm{J}(\xi) w(\xi)\, d\xi = \int_0^{\infty} {\mathrm{y}_{\mathrm{exp}}}^T \left[\mathrm{diag}\left(\delta_{00}, \delta_{11}, \ldots, \delta_{\mathrm{LL}}\right) \otimes \mathbf{Q}\right] \mathrm{y}_{\mathrm{exp}} \\ + {\mathrm{u}_{\mathrm{exp}}}^T \left[\mathrm{diag}\left(\delta_{00}, \delta_{11}, \ldots, \delta_{\mathrm{LL}}\right) \otimes \mathbf{R}\right] \mathrm{u}_{\mathrm{exp}}\, dt.\end{aligned} \tag{5.33}$$

Further, it can be seen, from the given form, that the following cost function separately weights mean state terms and covariance terms:

$$\begin{aligned}\mathrm{J}_2 = \int_0^{\infty} {\mathrm{y}_{\mathrm{exp}}}^T \left[\mathrm{diag}\left(\delta_{00}, 0, \ldots, 0\right) \otimes \underline{\mathbf{M}}_{\mathbf{Q}} + \mathrm{diag}\left(0, \delta_{11}, \ldots, \delta_{\mathrm{LL}}\right) \otimes \underline{\mathbf{C}}_{\mathbf{Q}}\right] \mathrm{y}_{\mathrm{exp}} \\ + {\mathrm{u}_{\mathrm{exp}}}^T \left[\mathrm{diag}\left(\delta_{00}, 0, \ldots, 0\right) \otimes \underline{\mathbf{M}}_{\mathbf{R}} + \mathrm{diag}\left(0, \delta_{11}, \ldots, \delta_{\mathrm{LL}}\right) \otimes \underline{\mathbf{C}}_{\mathbf{R}}\right] \mathrm{u}_{\mathrm{exp}}\, dt.\end{aligned} \tag{5.34}$$

Conveniently, the expanded form of the model, given in Equation (5.14), can be written in closed loop form as

$$\begin{bmatrix} \dot{\mathrm{x}}_{\mathrm{exp}} \\ \mathrm{y}_{\mathrm{exp}} \\ \mathrm{u}_{\mathrm{exp}} \\ \mathcal{Y}_{\mathrm{exp}} \\ \mathcal{U}_{\mathrm{exp}} \end{bmatrix} = \begin{bmatrix} \mathbf{A_{exp}} + \mathbf{B_{exp,u}}\mathbf{K_{exp}} & \mathbf{B_{exp,w}} \\ \mathbf{C_{exp}} & \mathbf{0} \\ \mathbf{K_{exp}} & \mathbf{0} \\ \left(\mathbf{Q_{exp}}\right)^{1/2} \mathbf{C_{exp}} & \mathbf{0} \\ \left(\mathbf{R_{exp}}\right)^{1/2} \mathbf{K_{exp}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathrm{x}_{\mathrm{exp}} \\ \mathrm{w}_{\mathrm{exp}} \end{bmatrix}. \tag{5.35}$$

In the equation above, it is used that the expanded form, produced by Galerkin projection of the feedback equation, is

$$\mathrm{u}_{\mathrm{exp}}(t) = \mathbf{K_{exp}} \mathrm{x}_{\mathrm{exp}}(t), \tag{5.36}$$

where $\mathbf{K_{exp}} \equiv \mathbf{I}_{\mathrm{L}+1} \otimes \mathbf{K}$.

It is possible to write a new $\mathcal{H}_2$ norm for the expanded system as

$$\begin{aligned} \mathrm{J} = \|H\|_2^2 &= \int_0^\infty \mathcal{Y}_{\mathrm{exp}}{}^T \mathcal{Y}_{\mathrm{exp}} + \mathcal{U}_{\mathrm{exp}}{}^T \mathcal{U}_{\mathrm{exp}}\, dt \\ &= \int_0^\infty \mathrm{y}_{\mathrm{exp}}{}^T \mathbf{Q_{exp}} \mathrm{y}_{\mathrm{exp}} + \mathrm{u}_{\mathrm{exp}}{}^T \mathbf{R_{exp}} \mathrm{u}_{\mathrm{exp}}\, dt, \end{aligned} \tag{5.37}$$

where $\mathbf{Q_{exp}}$ is positive semidefinite, $\mathbf{R_{exp}}$ is positive definite, and both are symmetric. This is the same structure as the original $\mathcal{H}_2$ norm presented in Equation (5.29). Note that although the symbols are reused to show similarity, the $\mathcal{H}_2$ norm and cost have been redefined from Equation (5.29). Further, the fact that the norm is of the form given in Equations (5.33) and (5.34) demonstrates that the expanded system can be used for design. Thus, the following weighting matrix forms are particularly useful:

$$\begin{aligned} \mathbf{Q_{exp}} &= \left[\mathrm{diag}\,(\delta_{00}, 0, \ldots, 0) \otimes \underline{\mathbf{M}}_{\mathbf{Q}} + \mathrm{diag}\,(0, \delta_{11}, \ldots, \delta_{\mathrm{LL}}) \otimes \underline{\mathbf{C}}_{\mathbf{Q}}\right] \\ \mathbf{R_{exp}} &= \left[\mathrm{diag}\,(\delta_{00}, 0, \ldots, 0) \otimes \underline{\mathbf{M}}_{\mathbf{R}} + \mathrm{diag}\,(0, \delta_{11}, \ldots, \delta_{\mathrm{LL}}) \otimes \underline{\mathbf{C}}_{\mathbf{R}}\right]. \end{aligned} \tag{5.38}$$

For $\mathcal{H}_2$ control design, an assumption of broadband excitation into the exogenous input is typically made. One interpretation involves unit impulse responses [150, 151]. In this interpretation, the norm is calculated from the sum of impulse responses created by inputting impulses into each exogenous input channel. For the closed loop system, with $\mathbf{A_{exp,cl}} \equiv \mathbf{A_{exp}} + \mathbf{B_{exp,u}}\mathbf{K_{exp}}$, individual state response can be expressed as

$$\mathrm{x}_{\mathrm{exp,i}}(\mathrm{t}) = e^{\mathbf{A_{exp,cl}}t}\mathbf{B_{exp,w}}\delta_i\,(t) \tag{5.39}$$

$$\begin{aligned} \|H\|_2^2 = \mathrm{trace}\, \mathbf{B_{exp,w}}{}^T \int_0^\infty \left[e^{\mathbf{A_{exp,cl}}t}\right]^T (&\mathbf{C_{exp}}{}^T \mathbf{Q} \mathbf{C_{exp}} \\ &+ \mathbf{K_{exp}}{}^T \mathbf{R} \mathbf{K_{exp}}) e^{\mathbf{A_{exp,cl}}t}\, dt\, \mathbf{B_{exp,w}}. \end{aligned} \tag{5.40}$$

For stable systems, the integral term can be formulated in terms of a Lyapunov equation [152] such that

$$\mathbf{A_{exp,cl}}^T\mathbf{P} + \mathbf{P}\mathbf{A_{exp,cl}} + \mathbf{C_{exp}}^T\mathbf{Q}\mathbf{C_{exp}} + \mathbf{K_{exp}}^T\mathbf{R}\mathbf{K_{exp}} = 0 \tag{5.41}$$

allows solving the represented $\mathcal{H}_2$ norm:

$$||H||_2^2 = \operatorname{trace} \mathbf{B_{exp,w}}^T\mathbf{P}\mathbf{B_{exp,w}}. \tag{5.42}$$

The Lyapunov equation has the convenient property that $\mathbf{P} > 0$, iff $\mathbf{A_{exp,cl}}$ represents a system with a finite $\mathcal{H}_2$ norm. Additionally, for the application, this condition is equivalent to $\mathbf{A_{exp,cl}}$ possessing negative real eigenvalues. Further, because of the symmetry of the problem, $\mathbf{P}$ is symmetric.

This problem can be posed as a matrix inequality in order to establish a bound on the norm. If there exists a $\mathbf{P} > 0$ such that

$$\begin{aligned}\left[\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\mathbf{K_{exp}}\right]^T\mathbf{P} + \mathbf{P}\left[\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\mathbf{K_{exp}}\right] \\ +\mathbf{C_{exp}}^T\mathbf{Q}\mathbf{C_{exp}} + \mathbf{K_{exp}}^T\mathbf{R}\mathbf{K_{exp}} < 0,\end{aligned} \tag{5.43}$$

then

$$\gamma^2 > \operatorname{trace} \mathbf{B_{exp,w}}^T\mathbf{P}\mathbf{B_{exp,w}} \tag{5.44}$$

such that $||H||_2$ is below some level γ. Let $\mathbf{S} \equiv \mathbf{P}^{-1}$ and then pre- and post-multiply Equation (5.43) by $\mathbf{S}$. The following set of matrix inequalities results:

$$\begin{aligned}\mathbf{S}\left[\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\mathbf{K_{exp}}\right]^T + \left[\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\mathbf{K_{exp}}\right]\mathbf{S} \\ +\mathbf{S}\mathbf{C_{exp}}^T\mathbf{Q}\mathbf{C_{exp}}\mathbf{S} + \mathbf{S}\mathbf{K_{exp}}^T\mathbf{R}\mathbf{K_{exp}}\mathbf{S} < 0\end{aligned} \tag{5.45}$$

$$\mathbf{B_{exp,w}}^T\mathbf{S}^{-1}\mathbf{B_{exp,w}} - \mathbf{Z} < 0 \tag{5.46}$$

$$\operatorname{trace} \mathbf{Z} < \gamma^2 \tag{5.47}$$

where $\mathbf{S} > 0$.

The Schur complement [153] can be applied to Equations (5.45) and (5.46).

$$\begin{bmatrix} \mathbf{S}\left[\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\mathbf{K_{exp}}\right]^T & \mathbf{S}\left[\mathbf{C_{exp}}\,(\mathbf{Q})^{1/2}\right]^T & \mathbf{S}\left[\mathbf{K_{exp}}\,(\mathbf{R})^{1/2}\right]^T \\ +\left[\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\mathbf{K_{exp}}\right]\mathbf{S} & & \\ \left[\mathbf{C_{exp}}\,(\mathbf{Q})^{1/2}\right]\mathbf{S} & -\mathbf{I} & \mathbf{0} \\ \left[\mathbf{K_{exp}}\,(\mathbf{R})^{1/2}\right]\mathbf{S} & \mathbf{0} & -\mathbf{I} \end{bmatrix} < 0 \tag{5.48}$$

$$\begin{bmatrix} \mathbf{Z} & \mathbf{B_{exp,w}}^T \\ \mathbf{B_{exp,w}} & \mathbf{S} \end{bmatrix} > 0 \tag{5.49}$$

At this point, two different substitutions can be made to make this equation a linear matrix inequality (LMI): 1) let $\mathbf{Y} = \mathbf{K_{exp}}\mathbf{S}$ and find $\mathbf{Z}$, for this case denoted $\mathbf{Z}_1$, and 2) let $\mathbf{S} = \mathbf{I}_{L+1} \otimes \mathbf{S_s}$, $\mathbf{K_{exp}} = \mathbf{I}_{L+1} \otimes \mathbf{K}$, such that $\mathbf{Y} = \mathbf{K_{exp}}\mathbf{S} = \mathbf{I}_{L+1} \otimes [\mathbf{K}\mathbf{S_s}] \equiv \mathbf{I}_{L+1} \otimes \mathbf{Y}_s$ and find $\mathbf{Z}$, for this case denoted as $\mathbf{Z}_2$. For both of these cases, the $\mathbf{Z}$'s are implied to be the ones that minimize trace $\mathbf{Z}$ such that there exists a $\mathbf{S} > 0$ that satisfies Equations (5.48) and (5.49).

Ordinarily, solving an LMI in the sense of an optimization problem in order to minimize trace $\mathbf{Z}$ could allow determining the $\mathcal{H}_2$ norm. However, in the present formulations, the first case is under-constrained and the second case is over-constrained. Thus if both exist, then $\min \text{trace}\,\mathbf{Z}_1 \leq \min \|H\|_2^2 \leq \min \text{trace}\,\mathbf{Z}_2$, for gains of the form $\mathbf{K_{exp}} = \mathbf{I}_{L+1} \otimes \mathbf{K}$. The second case is a feasibility condition in that if a solution exists, then there exists a set of gains of the form $\mathbf{K_{exp}} = \mathbf{I}_{L+1} \otimes \mathbf{K}$ that stabilizes the expanded system. Both of these problems can be solved with available LMI software, such as the freely available combination of YALMIP [154] and SeDuMi [155]. In reality, the lower bound expressed in the first case can be reasonably difficult to solve because of the large number of variables to be optimized.

The bounds on the $\mathcal{H}_2$ norm for the second case can be further lowered by substituting in the gains determined from $\mathbf{K} = \mathbf{Y}_s\mathbf{S}_s^{-1}$, in the form $\mathbf{K_{exp}} = \mathbf{I}_{\mathrm{L}+1} \otimes \mathbf{K}$, into the original Lyapunov equation in Equation (5.41) and then using Equation (5.42). This process leads to a method for numerically optimizing the problem in order to obtain implementable solutions.

5.8 Numerical Optimization

The following proposed approach involves starting from a stable set of gains and converging on a local minimum. To achieve this goal, a gradient of the $\mathcal{H}_2$ norm, with respect to the gains can be found.

Taking the partial derivative of the following Lyapunov equation,

$$\begin{aligned}\left(\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\left(\mathbf{I}\otimes\mathbf{K}\right)\right)^T\mathbf{P} + \mathbf{P}\left(\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\left(\mathbf{I}\otimes\mathbf{K}\right)\right)\\ +\mathbf{C_{exp}}^T\mathbf{Q}\mathbf{C_{exp}} + \left(\mathbf{I}\otimes\mathbf{K}\right)^T\mathbf{R}\left(\mathbf{I}\otimes\mathbf{K}\right) = 0\end{aligned} \tag{5.50}$$

with respect to the α index of $\mathbf{K}$, yields

$$\begin{aligned}\left(\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\left(\mathbf{I}\otimes\mathbf{K}\right)\right)^T\frac{\partial\mathbf{P}}{\partial\mathbf{K}_\alpha} + \frac{\partial\mathbf{P}}{\partial\mathbf{K}_\alpha}\left(\mathbf{A_{exp}} + \mathbf{B_{exp,u}}\left(\mathbf{I}\otimes\mathbf{K}\right)\right) + \left(\mathbf{B_{exp,u}}\left(\mathbf{I}\otimes\mathbf{1}_\alpha\right)\right)^T\mathbf{P}\\ +\mathbf{P}\left(\mathbf{B_{exp,u}}\left(\mathbf{I}\otimes\mathbf{1}_\alpha\right)\right) + \left(\mathbf{I}\otimes\mathbf{1}_\alpha\right)^T\mathbf{R}\left(\mathbf{I}\otimes\mathbf{K}\right) + \left(\mathbf{I}\otimes\mathbf{K}\right)^T\mathbf{R}\left(\mathbf{I}\otimes\mathbf{1}_\alpha\right) = 0\end{aligned} \tag{5.51}$$

where $\mathbf{1}_\alpha \equiv \frac{\partial\mathbf{K}}{\partial\mathbf{K}_\alpha}$. As defined, $\mathbf{1}_\alpha$ is a matrix of zeros, the same dimension as $\mathbf{K}$, except the α element of the matrix is a one. Also,

$$\frac{\partial\left\|H\right\|_2^2}{\partial\mathbf{K}_\alpha} = \mathrm{trace}\,\mathbf{B_{exp,w}}^T\frac{\partial\mathbf{P}}{\partial\mathbf{K}_\alpha}\mathbf{B_{exp,w}}. \tag{5.52}$$

The partial derivatives with respect to the gains defines a gradient, $\nabla \|H\|_2^2$. This gradient can be found for given sets of gains by solving the Lyapunov equation in Equation (5.50) and then solving the Lyapunov equation in Equation (5.51). This result is then substituted into Equation (5.52). Assembling all of the partial derivatives produces the gradient. A gradient descent of the form

$$\mathbf{K}_{(k+1)} = \mathbf{K}_{(k)} - \mu \left[\nabla \|H\|_2^2 \right]_{\mathbf{K}_{(k)}} \tag{5.53}$$

can be utilized in order to find a set of gains that locally minimizes the $\mathcal{H}_2$ norm. In this equation, μ is a positive scalar. Additionally, the gradient enables the use of Newtonian based optimization algorithms.

However, it has been found in many cases that greater numerical accuracy in regard to calculating the gradient of the $\mathcal{H}_2$ norm can be obtained by performing a finite difference approximation:

$$\frac{\partial \|H\|_2^2}{\partial \mathbf{K}_\alpha} \approx \frac{\delta \|H\|_2^2}{\delta \mathbf{K}_\alpha} = \text{trace}\, \mathbf{B_{exp,w}}^T \frac{\delta \mathbf{P}}{\delta \mathbf{K}_\alpha} \mathbf{B_{exp,w}}, \tag{5.54}$$

for small $\delta \mathbf{K}_\alpha$. As such, it can be useful to use this approximation in optimization.

Since the Lyapunov equation to determine the $\mathcal{H}_2$ norm is only valid for stable systems, iterations should start with initial choice of gains that create a stable closed loop system. Additionally an argument must be put forth as to why the gradient descent should arrive at closed loop stable systems.

To begin this argument, if a system is strictly stable and the weightings are finite, then the weighted $\mathcal{H}_2$ norm is bounded. Typically, as a system moves closer to marginal stability, the square of the outputs increase asymptotically towards infinity. Conceptually, this can be seen in the case of a marginally stable system, where oscillations will never die out and thus the integral defined in Equation (5.40) will then typically diverge. In the case of a fully observable system, any ongoing

oscillations will drastically effect the $\mathcal{H}_2$ norm and thus the norm will asymptotically diverge as the system approaches marginal stability.

Further, in terms of a root locus, if the gains are changed slowly enough, the poles should change slowly as well. As the poles approach the imaginary axis, the $\mathcal{H}_2$ norm will asymptotically increase. Thus, for a fully observable system, the negative of the gradient should point in a direction as to move the changing gain matrix such that the closed loop poles do not cross the imaginary axis. This completes the argument by implying that for an observable system, as long as 1) the gradient descent begins from a stable gain, and 2) small enough steps in the direction of the gradient are taken to faithfully follow the gradient, then the resulting local minimum should be stable as well.

To conclude the section, one additional piece of information is noted. As mentioned, the LQR problem is a subset of the $\mathcal{H}_2$ problem. This is in fact the case for the presented probabilistic interpretation of the $\mathcal{H}_2$ problem as well. The derivation and optimization method obviously applies for the identity matrix, $\mathbf{B_w} = \mathbf{I}_\mathrm{M}$. In the presented framework, this selection functions as an interpretation of the LQR input assumptions [150] and thus also allows solving the probabilistic LQR problem.

5.9 Examples

In order to demonstrate the theory developed in this chapter, two examples are presented in this section. The first example focuses on the optimization process and stability. The second example is presented to demonstrate the ability to optimize with respect to weighting variances in states.

First, the optimization process can be illustrated using a one degree of freedom oscillator, shown in Figure 5.1. This example is chosen because there are only two physical states. Utilizing only two gains, one for each state, allows easy visualization of the $\mathcal{H}_2$ norm as a function of the gains.

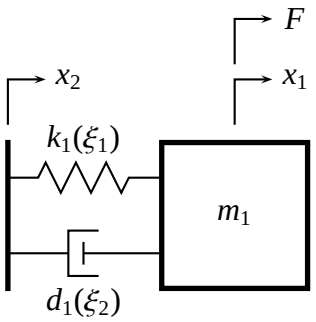


Figure 5.1: Uncertain 1-degree-of-freedom system schematic

Additionally, the parameters for this system, shown in Table 5.1, have been deliberately selected to include stable and unstable realizations.

Table 5.1: Parameters for the system in Figure 5.1. Note that λ is simply used to convert the standard notation for the β distribution to the domain, $\Omega_1 \in [-1, 1]$, used by the Jacobi polynomials.

$m_1 = 1$	$k_1(\xi_1) = \xi_1 + 1$	$d_1(\xi_2) = \frac{1}{2}\xi_2$
$\xi_1 = 2\lambda - 1$	$\lambda \sim \beta(2, 2)$	$\xi_2 \sim u(-1, 1)$

The state equation for the system can be given by

$$\begin{aligned}\dot{\mathrm{x}}(\xi_1, \xi_2, t) &= \begin{bmatrix} \dot{\mathrm{x}}_1(\xi_1, \xi_2, t) \\ \ddot{\mathrm{x}}_1(\xi_1, \xi_2, t) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ -\frac{k_1(\xi_1)}{m_1} & -\frac{d_1(\xi_2)}{m_1} \end{bmatrix} \begin{bmatrix} \mathrm{x}_1(\xi_1, \xi_2, t) \\ \dot{\mathrm{x}}_1(\xi_1, \xi_2, t) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 \\ \frac{1}{m_1} \end{bmatrix} \begin{bmatrix} F(\xi_1, \xi_2, t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_1(\xi_1)}{m_1} & \frac{d_1(\xi_2)}{m_1} \end{bmatrix} \begin{bmatrix} \mathrm{x}_2(t) \\ \dot{\mathrm{x}}_2(t) \end{bmatrix}.\end{aligned} \tag{5.55}$$

This state equation can expressed in terms of basis functions and placed in expanded form. For simplicity, let $\mathbf{C} = \mathbf{I}$ in the output equation, thus, making $\mathbf{C}_{\mathbf{exp}}$ an identity matrix as well.

After selecting a weighting matrix, it is possible to create a plot of $\|H\|_2^2$ as a function of the gains. This is shown in Figure 5.2. Additionally, a gradient descent to the optimal gain is plotted in this figure. Further, it can be seen that if the gains are moved in the positive direction towards creating unstable closed loop systems, there is the desirable sharp increase in $\|H\|_2$.

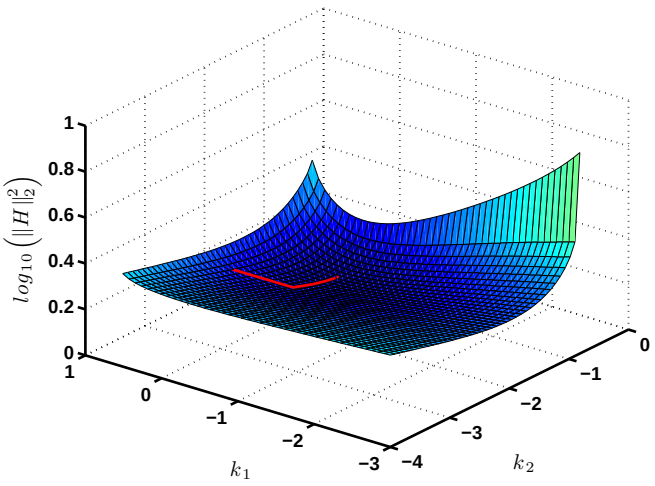


Figure 5.2: Plotted is $log_{10}\left(\|H\|_2^2\right)$ in the stable region of the system. The weighting matrices are $\mathbf{Q} = \text{diag}\,[\,1 \quad 1\,]$ and $\mathbf{R} = \text{diag}\,[\,1 \quad 0.1\,]$. Also, the line on the surface shows a gradient descent with $\mu = 0.2$, starting from $\mathbf{K} = [\,k_1 \quad k_2\,] = [\,0.3 \quad -2.5\,]$. The gradient descent converges to $\mathbf{K} = [\,-0.3649 \quad -1.4757\,]$. Fourth order polynomials are used, thereby creating fifteen two-dimensional basis functions.

To help illustrate that the determined gains stabilize unstable realizations of the system, the following open loop realization is chosen:

$$\mathbf{A}(\Xi_1, \Xi_2) = \begin{bmatrix} 0 & 1 \\ -1.5 & 0.5 \end{bmatrix}, \tag{5.56}$$

such that it can be seen $eig(\mathbf{A}(\Xi_1, \Xi_2))$ produces the unstable poles $0.2500 \pm 1.1990i$.

Closing the loop with the determined gains

$$\mathbf{A}(\Xi_1, \Xi_2) + \mathbf{B_u K} = \begin{bmatrix} 0 & 1 \\ -1.5 & 0.5 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} -0.3649 & -1.4757 \end{bmatrix} \tag{5.57}$$

produces the stable eigenvalues $-0.4879 \pm 1.2755i$.

Next, a second example uses a two-degree-of-freedom system shown in Figure 5.3 with its model parameters given in Table 5.2. The state equation for design can be written as

$$\begin{aligned} \dot{\mathrm{x}}(\xi_1, \xi_2, t) &= \begin{bmatrix} \dot{\mathrm{x}}_1(\xi_1, \xi_2, t) \\ \dot{\mathrm{x}}_2(\xi_1, \xi_2, t) \\ \ddot{\mathrm{x}}_1(\xi_1, \xi_2, t) \\ \ddot{\mathrm{x}}_2(\xi_1, \xi_2, t) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_1(\xi_1)}{m_1} & \frac{k_1(\xi_1)}{m_1} & -\frac{d_1(\xi_2)}{m_1} & \frac{d_1(\xi_2)}{m_1} \\ \frac{k_1(\xi_1)}{m_2} & -\frac{k_1(\xi_1)+k_2}{m_2} & \frac{d_1(\xi_2)}{m_2} & -\frac{d_1(\xi_2)+d_2}{m_2} \end{bmatrix} \begin{bmatrix} \mathrm{x}_1(\xi_1, \xi_2, t) \\ \mathrm{x}_2(\xi_1, \xi_2, t) \\ \dot{\mathrm{x}}_1(\xi_1, \xi_2, t) \\ \dot{\mathrm{x}}_2(\xi_1, \xi_2, t) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_1} \\ 0 \end{bmatrix} \begin{bmatrix} F(\xi_1, \xi_2, t) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \delta_i(t). \end{aligned} \tag{5.58}$$

The δ_i notation denotes a dirac delta input for the i-th state. Note that the manner in which the exogenous input is arranged relates to the LQR problem. Also, for simplicity, let $\mathbf{C} = \mathbf{I}$.

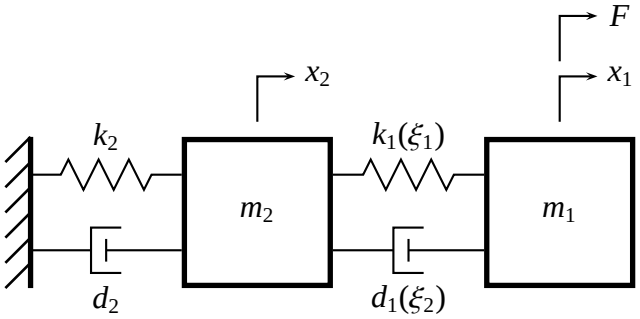


Figure 5.3: Uncertain 2-degree-of-freedom system schematic

Table 5.2: Parameters for the system in Figure 5.3. Note that λ is simply used to convert the standard notation for the β distribution to one with a support with $\Omega_1 = [-1, 1]$ used by the Jacobi polynomials.

$m_1 = 1$	$m_2 = 1$
$k_1(\xi_1) = \xi_1 + 1$	$k_2 = 1$
$d_1(\xi_2) = \frac{1}{2}\xi_2 + 1$	$d_2 = 1$
$\xi_1 = 2\lambda - 1$	$\xi_2 \sim u(-1, 1)$
$\lambda \sim \beta(2, 2)$	

The random parameters can then be expanded in terms of their basis functions as

$$\begin{aligned} k_1(\xi_1) &= 1 + \xi_1 = 1\phi_1^0(\xi_1) + \tfrac{1}{2}\phi_1^1(\xi_1) + 0\phi_1^2(\xi_1) + 0\phi_1^3(\xi_1) + \dots \\ &= \hat{k}_1^0\phi_1^0(\xi_1) + \hat{k}_1^1\phi_1^1(\xi_1) + \hat{k}_1^2\phi_1^2(\xi_1) + \hat{k}_1^3\phi^3(\xi_1) + \dots = \textstyle\sum_{i_1=0}^{\mathrm{L}_1} \hat{k}_1^{i_1}\phi_1^{i_1}(\xi_1) \end{aligned} \tag{5.59}$$

and

$$\begin{aligned} d_1(\xi_2) &= 1 + \tfrac{1}{2}\xi_2 = 1\phi_2^0(\xi_2) + \tfrac{1}{2}\phi_2^1(\xi_2) + 0\phi_2^2(\xi_2) + 0\phi_2^3(\xi_2) + \dots \\ &= \hat{d}_1^0\phi_2^0(\xi_2) + \hat{d}_1^1\phi_2^1(\xi_2) + \hat{d}_1^2\phi_2^2(\xi_2) + \hat{d}_1^3\phi_2^3(\xi_2) + \dots = \textstyle\sum_{i_2=0}^{\mathrm{L}_2} \hat{d}_1^{i_2}\phi_2^{i_2}(\xi_2) \end{aligned} \tag{5.60}$$

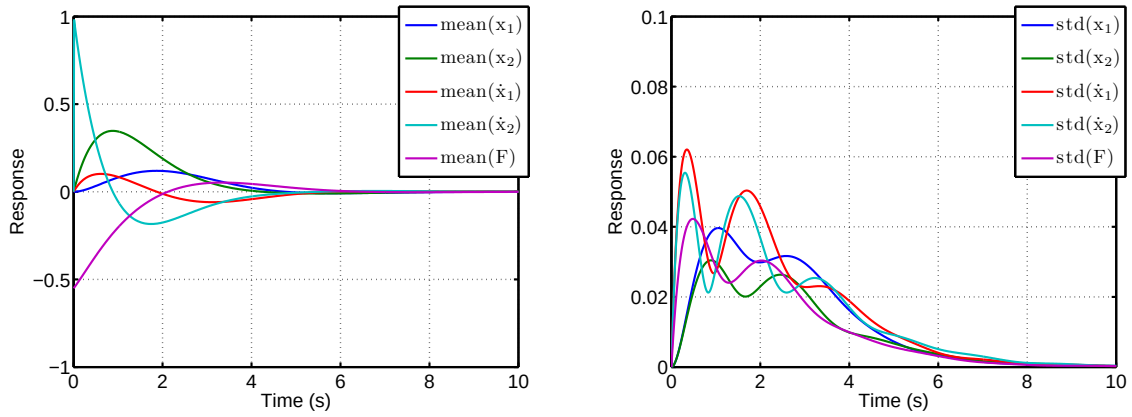
where the polynomial sets represented by $\left\{\phi_1^{i_1}(\xi_1)\right\}$ are Jacobi(1,1) polynomials and the polynomial sets represented by $\left\{\phi_2^{i_2}(\xi_2)\right\}$ are Legendre polynomials.

In order to create the matrix expansion for $\mathbf{A}(\xi_1, \xi_2)$, a set of multi-dimensional polynomials, $\left\{\phi^i(\xi_1, \xi_2)\right\}$, needs to be defined. These are created from combinations of the single-dimensional polynomials $\phi_1^{i_1}(\xi_1)$ and $\phi_2^{i_2}(\xi_2)$. Full details were given in Section 2.7. Once these multi-dimensional polynomial sets have been defined, it is now possible to create a matrix expansion of the type shown in Equation (5.12). Further, the inner products of the multi-dimensional basis functions can be calculated and then used to create the required $\mathbf{F}^i$ matrices defined in Equation (5.17). The $\hat{\mathbf{A}}^i$ terms and the $\mathbf{F}^i$ terms can be used in Equation (5.15) in order to create the expanded state matrix $\mathbf{A_{exp}}$. Since the $\mathbf{B_u}$ and $\mathbf{C}$ matrices in this example are not a function of the random variables, the expanded matrices can simply be computed as $\mathbf{B_{exp,u}} = \mathbf{I}_{L+1} \otimes \mathbf{B_u}$ and $\mathbf{C_{exp}} = \mathbf{I}_{L+1} \otimes \mathbf{C}$.

As mentioned earlier, the $\mathbf{B_{exp,w}}$ matrix is a special case. For this example, the exogenous input, the base excitation has been defined as independent from the random parameters. Thus, this matrix can be expanded in the manner given in Equation (5.11). Further, since $\mathbf{B_w}$ is not a function of the random variables, $\hat{\mathbf{B}_\mathbf{w}}^0 = \mathbf{B_w}$ and the remainder of the matrix coefficients are zero matrices.

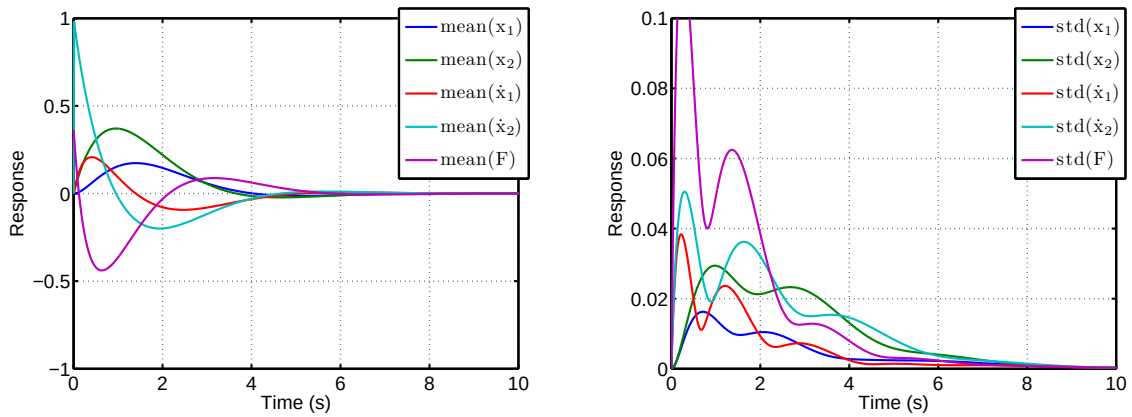
The procedure outlined in Section 5.8 can now be used to find the local minimum for $\mathcal{H}_2$ norms defined by weighting matrices described in Equation (5.38). In order to attempt to find a global minimum, 100 initial stable gains were randomly chosen and then the local minimums were converged upon. This large number was chosen in order to attempt to converge to different local minimums. However, within numerical error, these all converged to the same local minimums representing the same sets of gains. To illustrate the results, two different sets of weighting matrices have been chosen. The results for the stochastic system are shown in the figures below.

In these figures, the standard deviation trajectories are plotted, since they correspond to the spread of the closed loop trajectories. This corresponds to the interpretation of the standard deviation functioning as a *parameter of scale*.



(a) Mean (b) Standard Deviation

Figure 5.4: Mean and standard deviation trajectories for the response to the unit impulse into the fourth state, δ_4, are presented. The weighting matrices $\underline{\mathbf{M}}_{\mathbf{Q}} = \text{diag}\,[\,1 \;\; 1 \;\; 1 \;\; 1\,]$, $\underline{\mathbf{M}}_{\mathbf{R}} = [1]$, $\underline{\mathbf{C}}_{\mathbf{Q}} = \text{diag}\,[\,0 \;\; 0 \;\; 0 \;\; 0\,]$, and $\underline{\mathbf{C}}_{\mathbf{R}} = [1]$ are used. The determined gains are $\mathbf{K} = [\,-0.569 \quad -0.285 \quad -1.058 \quad -0.551\,]$.



(a) Mean (b) Standard Deviation

Figure 5.5: Mean and standard deviation trajectories for the unit impulse into the fourth state, δ_4, are presented. The weighting matrices $\underline{\mathbf{M}}_{\mathbf{Q}} = \text{diag}\left[\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array}\right]$, $\underline{\mathbf{M}}_{\mathbf{R}} = [1]$, $\underline{\mathbf{C}}_{\mathbf{Q}} = \text{diag}\left[\begin{array}{cccc} 0 & 0 & 100 & 0 \end{array}\right]$, and $\underline{\mathbf{C}}_{\mathbf{R}} = [1]$ are used. The determined gains are $\mathbf{K} = \left[\begin{array}{cccc} -1.226 & 0.149 & -2.391 & 0.363 \end{array}\right]$. Also, $\max \text{std}(F) = 0.1187$.

Note that the only difference in the design criteria was that the variance of the velocity of m_1 was heavily weighted in the second set of figures. It can be seen that this did truly have the effect of substantially lowering the standard deviation of this state. Also, while the weighting for the standard deviation seems large, it is necessary since the square of the plotted state, the variance, is what is actually weighted. This large weighting corresponds to the large difference between the magnitude of the variances and the magnitude of the square of the means.

Further, it is not surprising that since the variation in the velocity of m_1 decreased, variation in the position decreased as well. Also, it makes sense that since the variance for the control effort was not weighted, that it could greatly increase. However, the mean of the control effort did not greatly increase since it was still weighted. A physical interpretation of the second system is that for the probabilistic set of systems, there will be some realizations in the set that require much greater control effort and some that require much less control effort. In effect, there needs to be greater variation in the control effort and a slight increase in the expected (mean) control effort in order to create the desired result of decreasing the variation in the velocity of m_1. Thus, the presence of this tradeoff intuitively makes sense.

5.10 Discussion

It is well known that the number of terms in Polynomial Chaos Expansions grows very quickly as the number of independent uncertain parameters increases. In fact, the total number of terms in a multi-dimensional basis expansion is given by $\mathrm{L} + 1 = (\mathrm{W} + r)!/(\mathrm{W}!r!)$, where r is the number of random variables used and W is the maximum order of the one-dimensional orthogonal basis functions. In the work performed in this chapter, good approximations were obtained with relatively low order PCEs. This was very helpful in keeping the total number of required multi-dimensional basis functions reasonably low. However, as r, the number of random variables used,

increases, it is easy to see that $L+1$ rapidly increases. By extension, a large number of states for an expanded system can be created. This commonly creates a large burden for simulating responses. However, the method presented bypasses simulation through the use of Lyapunov equations.

Additionally, while the finite approximations to the Galerkin projection converged without problems for the given example and the previous chapter demonstrated the rapid convergence, it is worth pointing out that if desired, probabilistic guarantees can still be obtained by combining the method with sampling techniques. One such example of this would be to represent possible stable and unstable systems as a binomial distribution [31, 32]. To do this, the following steps can be used:

Step 1: Pick a required probability of stability, π_0.

Step 2: Take n samples of the parameter space and check the stability of the designed system, such as by using eigenvalues. To follow good practice, $n\pi_0 \geq 5$ (and $n(1-\pi_0) \geq 5$).

Step 3: Calculate the sample probability of stability, $\hat{\pi} = \frac{n_{stable}}{n}$. If the system is always stable, this will have a value of one.

Step 4: Calculate $z = (\hat{\pi} - \pi_0)\sqrt{\frac{n}{\pi_0(1-\pi_0)}}$.

Step 5: The normal approximation for the true probability of stability being greater than the sample probability, $\pi > \pi_0$, is given by the inverse CDF of the standard normal distribution: $N^{-1}(z)$. This is related to the *level of significance* for the statistical test.

This type of test is very well accepted and can be found in introductory statistics books, such as [156].

Hence, it is possible to design a system using the presented Polynomial Chaos Method. If the rapid convergence of the Galerkin method is not sufficient for ensuring robustness for the given application, it is possible to a posteriori confirm robustness by sampling. If the system is always stable in the parameter space, then $\hat{\pi} = 1$. An example close to this ideal can be seen that when testing 5000 sample systems, there should be less than 0.08% of the time a probabilistic system with less than 99.8% of actual stable realizations is expected to create a sample probability appearing as always stable. Thus, a posteriori testing can help reinforce confidence in (closed loop) model stability.

Chapter 6

Experimental Platform for Validation

From the literature survey, it is evident that experimental work for methods derived with Polynomial Chaos is rare. Additionally, since the concept of using probabilistic optimality on systems with large uncertainties is reasonably novel, there is little controlled experimental work in this area as well. However, it is believed that the ultimate goal is for controllers to be applied in applications beyond simulations. Through using the probabilistic framework, the work in this chapter helps resolve a paradox of running controlled tests on uncertain systems—by designing experiments where it is possible to sample the probability space. While it is numerically efficient and useful to use PCEs, experimental results are ultimately obtained in terms of samples.

The uncertainties used are created in several ways. Section 6.1 uses a slowly time-varying current through a tunable magnetorheological (MR) damper. The damper is included as a component on a 2-degree-of-freedom oscillator. This method is used to illustrate the concept of probability distributions for state trajectories as a function of time. In Section 6.2, a grid sampling approach is used to add a range of masses to the oscillator. The process for designing and implementing the probabilistic controller on the system is presented. Extensive controller analysis and experimental

results are given. In Section 6.3, a controller is designed for a set of systems created by a variable-valve damper. For all of these systems, the uncertain element is intentionally introduced as an experimental device—not as a component to be directly controlled.

6.1 Magnetorheological (MR) Damper

The underlying mechanism for the first "uncertain system" is a tunable magnetorheological (MR) damper. MR dampers are semi-active devices in that they can only dissipate energy from the system. The working fluid in the damper is an MR fluid: generally this is an oil based fluid with large quantities of ferrous particles. In addition to the fluid, electromagnets allow the creation of magnetic fields in the working fluid. These fields cause the ferrous particles to align, thus altering the fluid properties, including increasing its resistance to shearing. This results in allowing dampers that change properties with a change in electrical current moving through the magnetic coils. Thus, by placing an MR damper into a 2-degree-of-freedom mechanical oscillator and applying an "uncertain current" to the damper, an uncertain system has been created. Further, since it is possible to specify the uncertain current as well as to record the time history of the current, controlled uncertain experiments can be created. A schematic of the system and a picture can be seen in Figure 6.1. Additionally, specifics on the instrumentation are given in Table 6.1. A picture with many of the main components labeled is given in Figure 6.2. Also, more views of the experiment are shown in Figure 6.3.

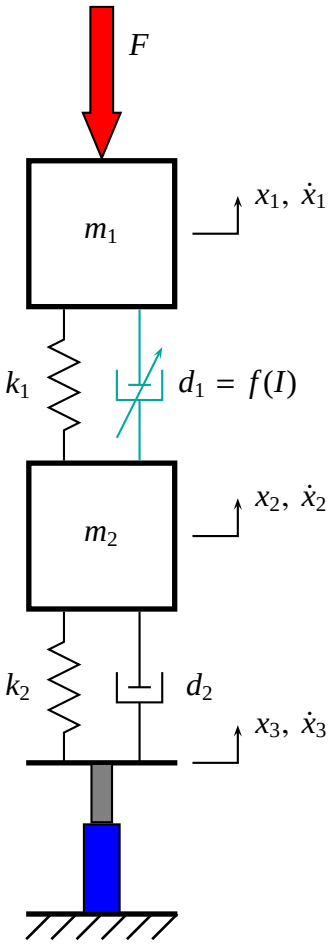


Figure 6.1: A schematic of the 2-degree-of-freedom test platform. In the schematic, F denotes the force from linear electric motors; $d_1 = f(I)$ denotes a damping as a function of an inputted current; and the position, x_3, and velocity, $\dot{x}_3$, are governed by a controlled hydraulic actuator.

Table 6.1: Description of components used in experimental test platform

#	Component	Description
3	Frequency Devices, model 9016	2 port digital filter/amplifiers
1	Quanser, LCAM	Linear current amplifier, continuous current 7 A, modified by adding a Pentium II heat sink to decrease temperature
1	Lambda regulated power supply, model LA 100-03BM	DC power supply, rated at up to 35 V and up to 10 A
2	Glentek SMA-9807	Brushless servo amplifiers
2	H2W BLDD-04	Brushless, cogless, 3-phase linear electric motors, rated 263 W constant power and 2367 W peak power
1	MagneRide MR damper	
1	MTS 242.02	Hydraulic actuator, rated up to 10 kN
2	VP510-10	String-pot with tachometer
2	Goodyear Super Cushion	Airspring
1	Air Reservoir	30 gallon
1	dSPACE AutoBox	DAQ
1	Sola MCB 100	Isolation transformer

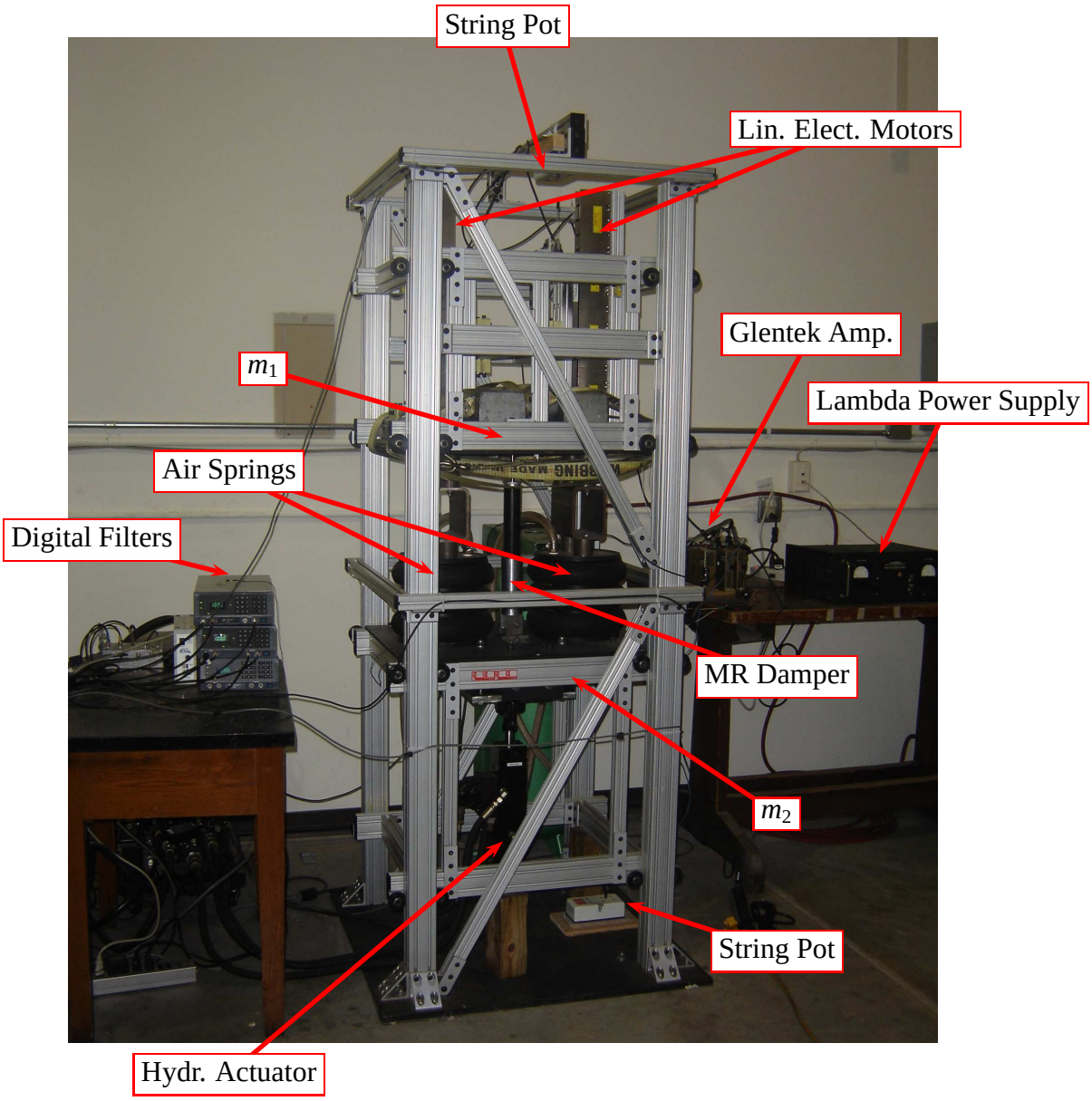
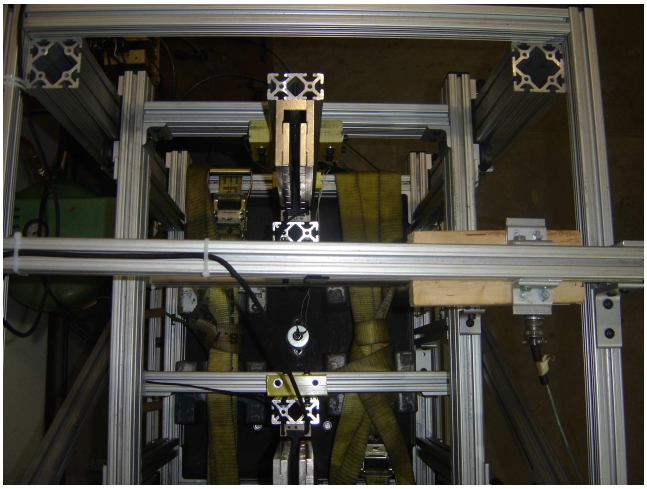
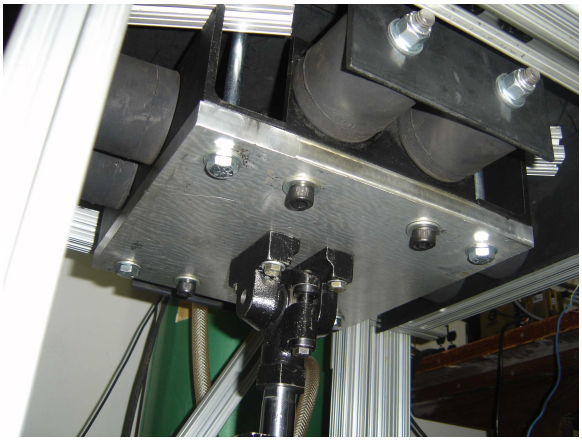


Figure 6.2: An image of the 2-degree-of-freedom test platform with components labeled.



(a) Top View with Linear Electric Motors

(b) MR Damper and Air Springs



(c) Bottom Suspension

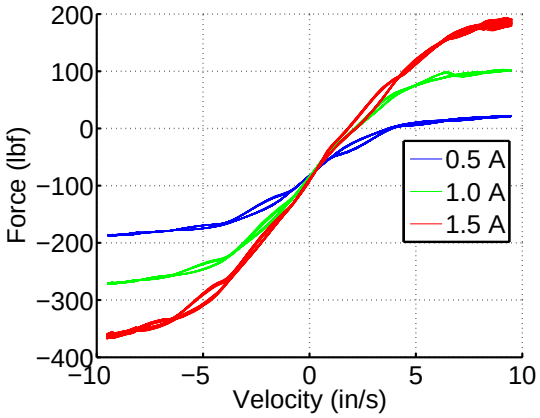
Figure 6.3: Additional views of the 2-degree-of-freedom test platform are given. In (a) a top view of the platform is shown, such that it is possible to see the top of a linear motor, as well as additional masses added to the top mass. Subfigure (b) is a closer view of the the air springs and MR damper. Additionally, a bottom view is given in (c), such that it is possible to see the hydraulic actuator interface with a plate. This plate is attached to m_2 by stiff rubber bushings that can also be seen in this view. Also, this picture shows how it is possible to create a stiff connection in the place of the rubber bushings by inserting rigid material (aluminum) and using bolts to clamp the two plates to the sandwiched rigid material.

In order to determine the relationship between damping and current for the MR damper, constant currents were inputted with low velocity sinusoidal displacements. The test set-up can be seen in Figure 6.4 and plots showing force versus velocity can be seen in Figure 6.1.

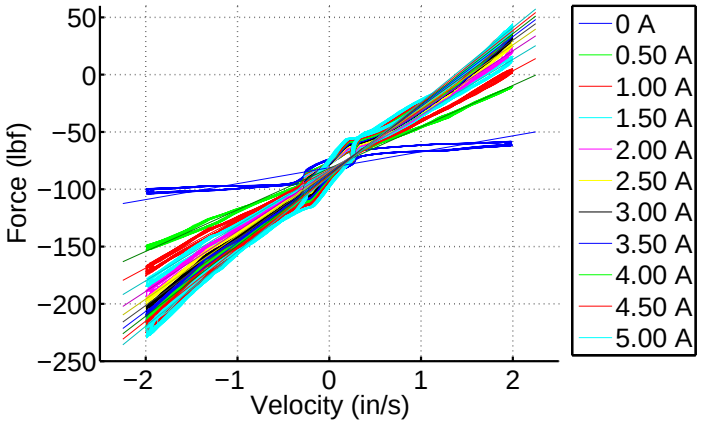
Figure 6.4: Magneto Rheologic Damper in load frame, with a load cell, to undergo low velocity characterization. A sinusoidal stroke of 1 inch (2.54 cm) peak to peak displacement at 1.5 Hz was used.

Table 6.2: Identified parameter values

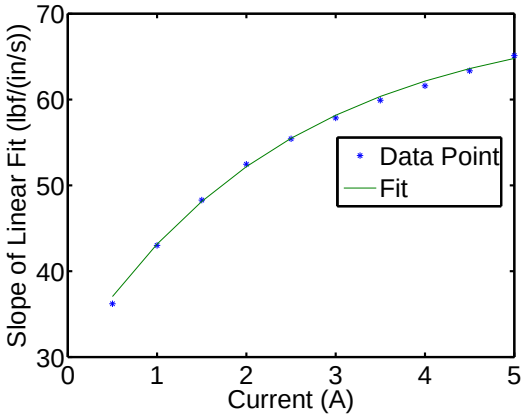
Parameter Type	**Values**	
Spring Constants	$k_1 = 11.9$ kN/m	$k_2 = 870$ kN/m
Damping Ratios	$d_1 \sim [6.6\ \text{kN s/m}, 11.4\ \text{kN s/m}]$	$d_2 = 0.3$ kN s/m
Masses	$m_1 = 175$ kg	$m_2 = 170$ kg



(a)



(b)



(c)

Figure 6.5: In (a) the force vs. velocity relationship obtained from a 0.5 inch (1.27 cm) amplitude, 3.0 Hz, sinusoidal displacement input of the damper at a few currents. The test was re-run for a 0.5 inch (1.27 cm) amplitude, 1.0 Hz, sinusoidal displacement input. A linear region was then selected for (b). The curve corresponding to 0 Amps is considered to not fit a linear trend well. Lines were fit to the linear portions. The slopes vs. the current are plotted in (c). A curve, $d_1(I) = 70 - 40.45\exp(-0.41I)$ lbf-s/in (equivalent to $d_1(I) = 12.26 - 7.083\exp(-0.41I)$ kN-s/m), was fitted.

As a test of the "uncertain system," in a simple experiment, the hydraulic cylinder was directly blocked to m_2, such that $x_2(t)$ was forced to be equal to $x_3(t)$. Then, using the inverse of the relationship determined between damping and current, a nearly uniform distribution of damping over time was created. The signal was designed to dwell, and was low-pass filtered ($\tau = 0.01$) in order to decrease high frequency content. A sample of the input can be seen in Figure 6.6. Additionally, the system was excited with the hydraulic actuators with a sinusoidal disturbance of 4.3 mm (0.17 in) amplitude and 100/33 Hz.

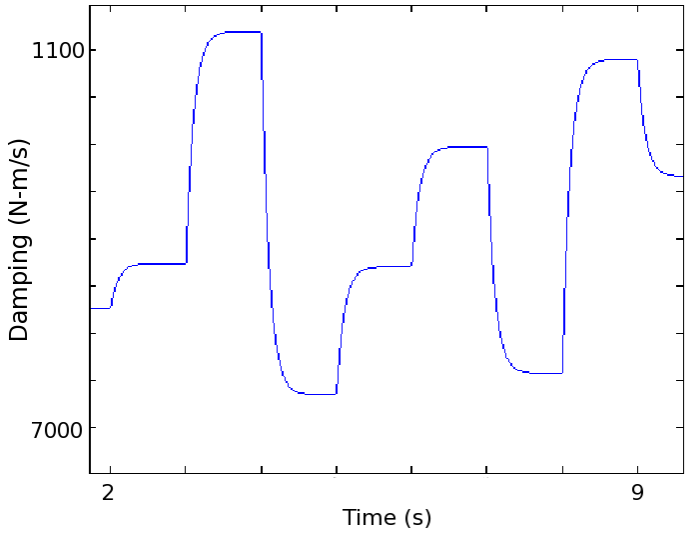


Figure 6.6: An eight second sample of the inputted signal for damping

By organizing approximately 3000 sampled cycles, it was possible to create the time evolving histogram shown in Figure 6.7. This can be compared to Polynomial Chaos and Monte Carlo simulations shown in Figure 6.8. Note that the reason why the Polynomial Chaos Simulations are slightly different from the Monte Carlo simulation is that in the former, an assumption of a slowly time-varying parameter was made and in the latter, the same type of signal used in the experiment was used.

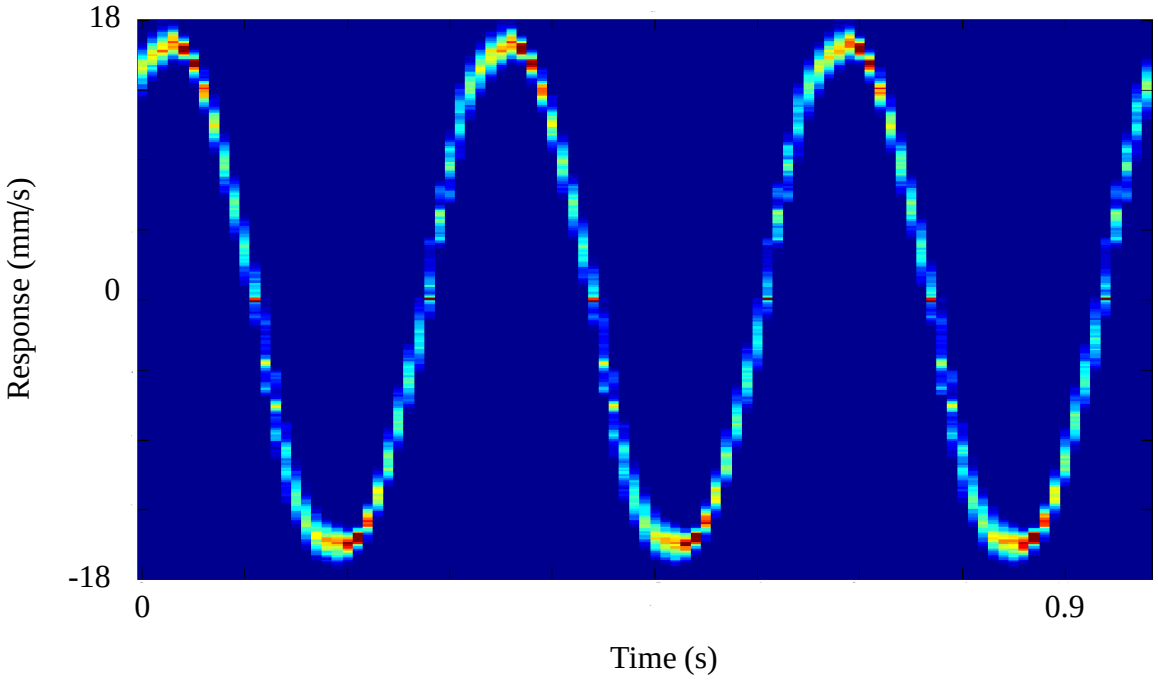


Figure 6.7: The experimental histogram is shown and can be compared to the simulations in Figure 6.8. Sections of vertical slices with red coloration show high probability density at a given time, while blue shows lower probability density. Further, dark blue implies zero probability density.

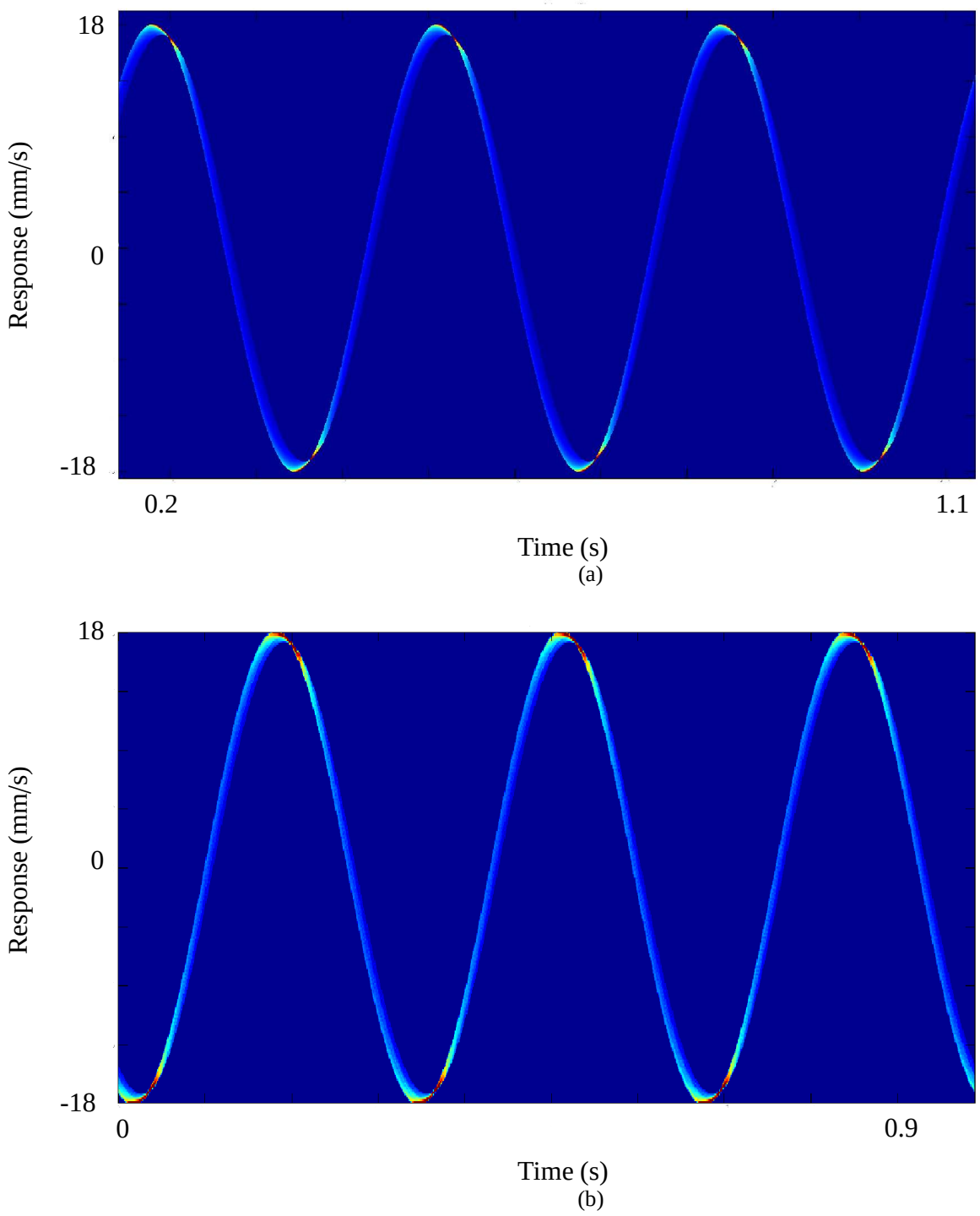


Figure 6.8: Histograms from (a) a Polynomial Chaos simulation and (b) a Monte Carlo simulation are shown so that they can be compared with the experimentally obtained time-evolving histogram in Figure 6.7. This data is mostly qualitative and thus it is sufficient to say that the scales for the probability densities are identical–where blue corresponds to a density of zero and red is the maximum density.

Many similarities between Figures 6.7 and 6.8 can be seen. In general, probability densities are more concentrated right after the sine peaks. Additionally, for the most part, the low velocity areas are reasonably well distributed due to the distribution of phases caused by the distribution for damping.

Also, a few discrepancies between these figures are evident. The most obvious one is that the sampling rate for the figure shown is lower. Additionally, a "smearing" of the distribution, causing it to be broader than normal, was produced by a high level of signal noise in this section. However, the model uncertainty—and not the process noise—is still the dominating effect. While interesting in this section, the signal noise is intentionally reduced in later sections. However, distinguishing between the notions of variability caused by system uncertainty and variability caused by parametric uncertainty is important. Hence, the distinction is further touched on in the next section.

Finally, at regions of velocities approximately zero, it can be seen that the experimental PDFs show very concentrated probability densities. This is due to briefly sticking and dwelling at zero velocity from slight static friction effects. Effects like this are the reality of actual mechanical systems.

6.2 Added Set of Masses

The control design method derived in Chapter 5 was applied to a set of systems created by varying a mass. Further, the system was augmented with an electronic filter in order to create a system that the controller had reasonable authority over. Hence, the overall system model was created to combine the mechanical oscillator with a second order filter. The combined system is described by Equations (6.1) to (6.10). These equations are partitioned to emphasize the boundary between the mechanical and filter components. Since the purpose of the experiment was to validate the design method, it was considered prudent to use full state feedback, so that results could be isolated

from additional issues inherent to $\mathcal{H}_2$ with partial state feedback. Four controllers, with different variance weights, are compared to a nominal $\mathcal{H}_2$ controller and the open loop response.

$$\left[\begin{array}{c}\dot{\mathrm{x}}_{mech}\\ \hline \dot{\mathrm{x}}_{filter}\end{array}\right] = \left[\begin{array}{c|c}\mathbf{A}_{mech} & \bar{\mathbf{B}}_{w,mech}\\ \hline \mathbf{0}_{2\times4} & \mathbf{A}_{filter}\end{array}\right]\left[\begin{array}{c}\mathrm{x}_{mech}\\ \hline \mathrm{x}_{filter}\end{array}\right] + \left[\begin{array}{c}\mathbf{B}_{u,mech}\\ \hline \mathbf{0}_{2\times1}\end{array}\right]\mathrm{u} + \left[\begin{array}{cc}\mathbf{B}_{w,mech,design} & \mathbf{0}_{4\times2}\\ \mathbf{0}_{2\times4} & \mathbf{B}_{w,filter}\end{array}\right]\mathrm{w} \tag{6.1}$$

$$\mathrm{x}_{mech}(\xi_1, t) = \left[\begin{array}{cccc}\mathrm{x}_1(\xi_1,t) & \mathrm{x}_2(\xi_1,t) & \dot{\mathrm{x}}_1(\xi_1,t) & \dot{\mathrm{x}}_2(\xi_1,t)\end{array}\right]^T \tag{6.2}$$

$$\mathrm{x}_{filter}(t) = \left[\begin{array}{cc}\mathrm{x}_3(t) & \dot{\mathrm{x}}_3(t)\end{array}\right]^T \tag{6.3}$$

$$\mathbf{A}_{mech}(\xi_1) = \left[\begin{array}{cccc} 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -\frac{k_1(\xi_1)}{m_1(\xi_1)} & \frac{k_1(\xi_1)}{m_1(\xi_1)} & -\frac{d_1(\xi_1)}{m_1(\xi_1)} & \frac{d_1(\xi_1)}{m_1(\xi_1)}\\ \frac{k_1(\xi_1)}{m_2} & -\frac{k_1(\xi_1)+k_2}{m_2} & \frac{d_1(\xi_1)}{m_2} & -\frac{d_1(\xi_1)+d_2}{m_2}\end{array}\right] \tag{6.4}$$

$$\mathbf{B}_{u,mech}(\xi_1) = \left[\begin{array}{c} 0\\ 0\\ \frac{F_k}{m_1(\xi_1)}\\ 0\end{array}\right] \tag{6.5}$$

$$\mathbf{B}_{w,mech} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_2}{m_2} \end{bmatrix}, \quad \bar{\mathbf{B}}_{w,mech} = \left[\begin{array}{c|c} \mathbf{B}_{w,mech} & \mathbf{0}_{4\times 1} \end{array} \right], \quad \mathbf{B}_{w,mech,design} = 0.01\,\mathbf{I} \tag{6.6}$$

$$\mathbf{C}_{mech} = \mathbf{I} \tag{6.7}$$

$$\mathbf{A}_{filter} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \tag{6.8}$$

$$\mathbf{B}_{w,filter} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.9}$$

$$\mathbf{C}_{filter} = \mathbf{I} \tag{6.10}$$

6.2.1 Experimental Setup

An overall system schematic of the six state equations presented is given in Figure 6.9.

The filter was implemented on the data acquisition and control hardware. The mechanical system is composed of two masses riding on rigid extruded aluminum rails. These masses are coupled by a set of air springs. Also, the bottom mass, m_2, is excited—through a set of stiff rubber bushings—by a displacement controlled hydraulic actuator. A schematic of this setup is given in Figure 6.9

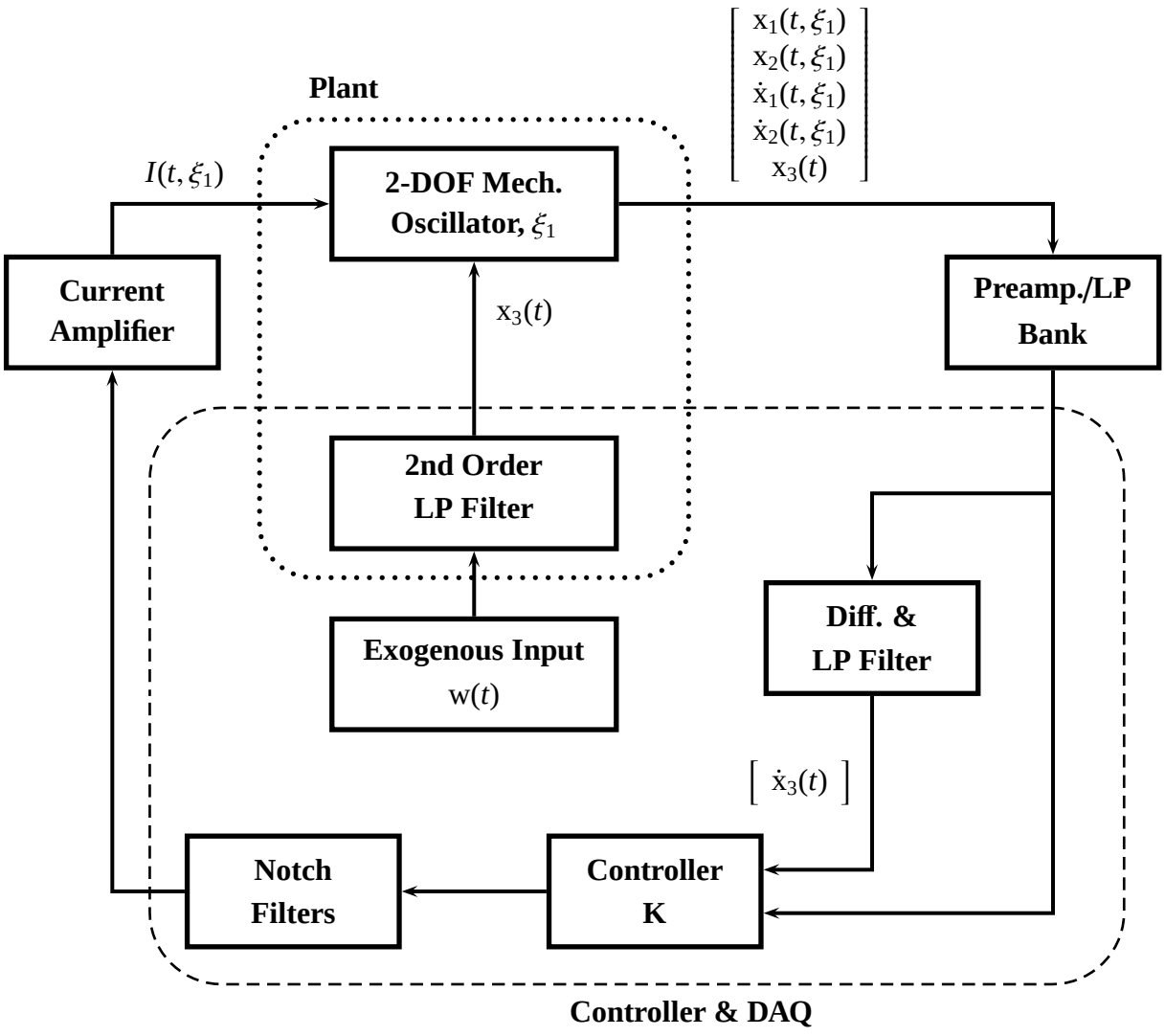


Figure 6.9: Overview of experimental setup.

and a picture of the mechanical oscillator is shown in Figure 6.10. Also, a table of equipment is given in Table 6.3.

Figure 6.10: The mechanical oscillator configured for testing a distribution of masses for m_1

Table 6.3: Description of experimental equipment

#	Component	Description
3	Frequency Devices, model 9016	2 port digital filter/amplifiers
2	Glentek SMA-9807	Brushless servo amplifiers
2	H2W BLDD-04	Brushless, cogless, 3-phase linear electric motors, rated 263 W constant power and 2367 W peak power.
1	MTS 242.02	Hydraulic actuator, rated up to 10 kN
2	VP510-10	String-pot with tachometer
2	Goodyear Super Cushion	Airspring
1	dSPACE AutoBox	DAQ
1	Sola MCB 100	Isolation transformer

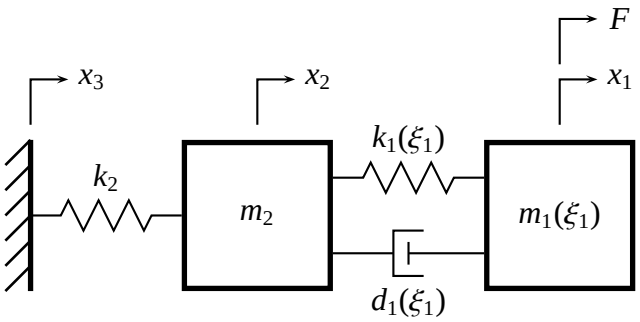


Figure 6.11: Uncertain 2-degree-of-freedom system schematic. Note that changes in the mass, m_1 drives changes in the stiffness and damping. Thus, they are all a function of ξ_1.

Figure 6.12: An additional 155 lb (70.307 kg) of weights as part of m_1

In order to obtain a family of systems resulting from a changing parameter, m_1 was altered through adding masses (Figure 6.12).

6.2.2 System Identification

In order to optimize controllers for the system, it was first necessary to possess a good model. This was completed using the Least Squares Method [157]. From first principles, the state equation for mechanical system should have the form:

$$\dot{\mathrm{x}}_{mech} = \mathbf{A}_{mech}(\xi_1)\mathrm{x}_{mech} + \mathbf{B}_{w,mech}\mathrm{x}_3(t). \tag{6.11}$$

The best parameter fit was defined in terms of the least squares cost function:

$$\mathrm{J}(\xi_1) = \frac{1}{2}\int_{t_0}^{t_1} \left(\mathrm{x}_{model}(t,\xi_1) - \mathrm{x}_{exper}(t,\xi_1)\right)^2 \, dt. \tag{6.12}$$

Two states were used in this cost function: the displacements of the two masses, $\mathrm{x}_1(t,\xi_1)$ and $\mathrm{x}_2(t,\xi_1)$. The minimum was determined through using the gradient of the cost function, described in terms of the generic parameter γ:

$$\frac{\partial \mathrm{J}(\xi_1)}{\partial \gamma} = \int_{t_0}^{t_1} \frac{\partial \mathrm{x}_{model}(t,\xi_1)}{\partial \gamma} \left(\mathrm{x}_{model}(t,\xi_1) - \mathrm{x}_{exper}(t,\xi_1)\right)^2 \, dt. \tag{6.13}$$

The required model sensitivity to parameters were computed using

$$\frac{\partial \dot{\mathrm{x}}(\xi_1,t)}{\partial \gamma} = \frac{\partial \mathbf{A}_{mech}(\xi_1)}{\partial \gamma}\mathrm{x}(\xi_1,t) + \mathbf{A}_{mech}(\xi_1)\frac{\partial \mathrm{x}(\xi_1,t)}{\partial \gamma} + \frac{\partial \mathbf{B}_{w,mech}}{\partial \gamma}\mathrm{w}(t). \tag{6.14}$$

These equations form a single state space model:

$$\begin{bmatrix} \dot{\mathrm{x}}(\xi_1,t) \\ \frac{\partial \dot{\mathrm{x}}(\xi_1,t)}{\partial \gamma_1} \\ \frac{\partial \dot{\mathrm{x}}(\xi_1,t)}{\partial \gamma_2} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{mech}(\xi_1) & \mathbf{0} & \mathbf{0} & \\ \frac{\partial \mathbf{A}_{mech}(\xi_1)}{\partial \gamma_1} & \mathbf{A}_{mech}(\xi_1) & \mathbf{0} & \\ \frac{\partial \mathbf{A}_{mech}(\xi_1)}{\partial \gamma_2} & \mathbf{0} & \mathbf{A}_{mech}(\xi_1) & \\ \vdots & & & \ddots \end{bmatrix} \begin{bmatrix} \mathrm{x}(\xi_1,t) \\ \frac{\partial \mathrm{x}(\xi_1,t)}{\partial \gamma_1} \\ \frac{\partial \mathrm{x}(\xi_1,t)}{\partial \gamma_2} \\ \vdots \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{w,mech} \\ \frac{\partial \mathbf{B}_{w,mech}}{\partial \gamma_1} \\ \frac{\partial \mathbf{B}_{w,mech}}{\partial \gamma_2} \\ \vdots \end{bmatrix} \mathrm{w}(t). \tag{6.15}$$

Through combining Equations (6.13) and (6.15) and using a numerical optimization routine, such as Matlab's *fminunc*, it is possible to minimize the original cost function in Equation (6.12).

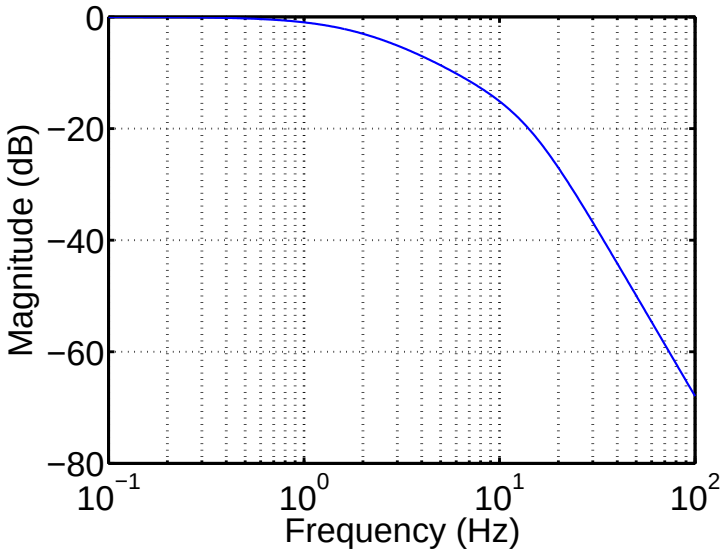


Figure 6.13: Magnitude plot of the transfer function used to color the noise for the displacement input created for the system ID of the mechanical system.

In order to perform this procedure on the mechanical system, data was obtained from a colored noise input shaped by the transfer function:

$$T_{nf}(s) = \frac{99252.72}{(s + 12.57)(s^2 + 125.7s + 7896)}. \tag{6.16}$$

A plot of the transfer function magnitude is given in Figure 6.13. The signal used for testing was created by filtering 100 seconds of white noise, sampled at 1 kHz. Further, a Hanning window was used such that the system ID was performed across a wide range of amplitudes. The resulting signal is shown in Figure 6.15.

This signal was then used to specify the displacement trajectory of the hydraulic actuator, x_3. The response was recorded in order to obtain x_{exper}. In post processing, the actual measured hydraulic cylinder displacement was used to excite the system model. This was due to the desire to factor out the effect of the hydraulic lag (Figure 6.15). A comparison of the model to the experimental data is shown in Figures 6.16 and 6.17.

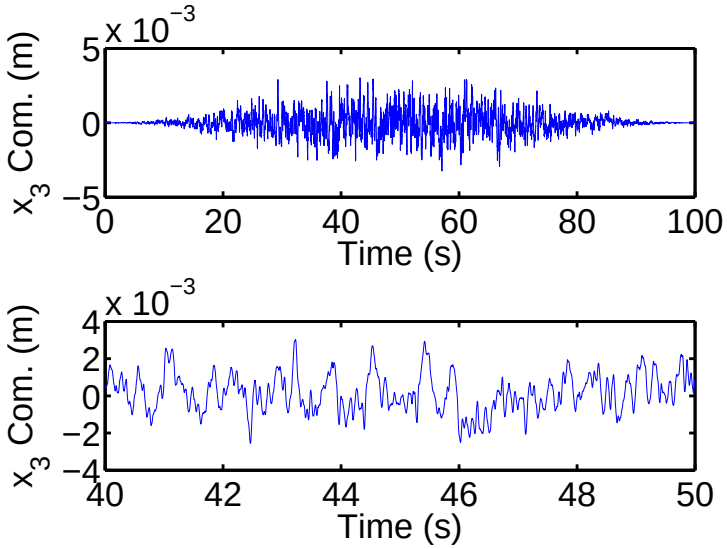


Figure 6.14: Time trace of colored noise displacement input used for the system ID of the mechanical system.

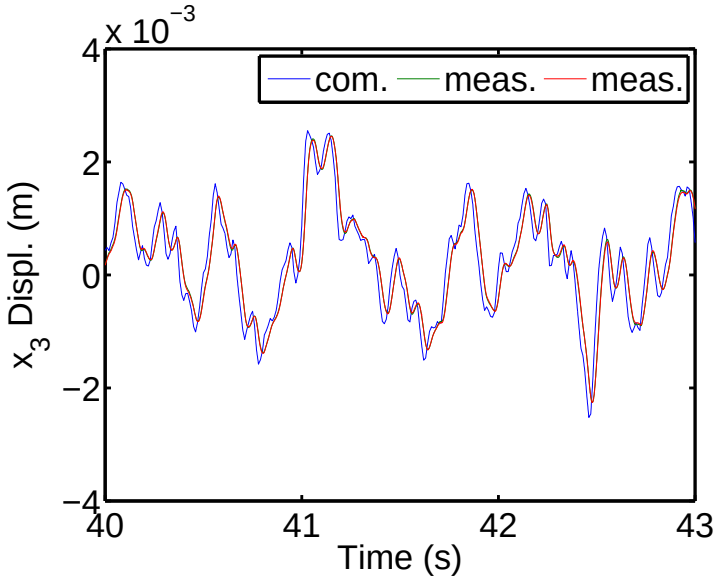


Figure 6.15: The input and measured output from the hydraulic actuator for two independent tests. Note the signals are similar, with the exception of an introduced lag.

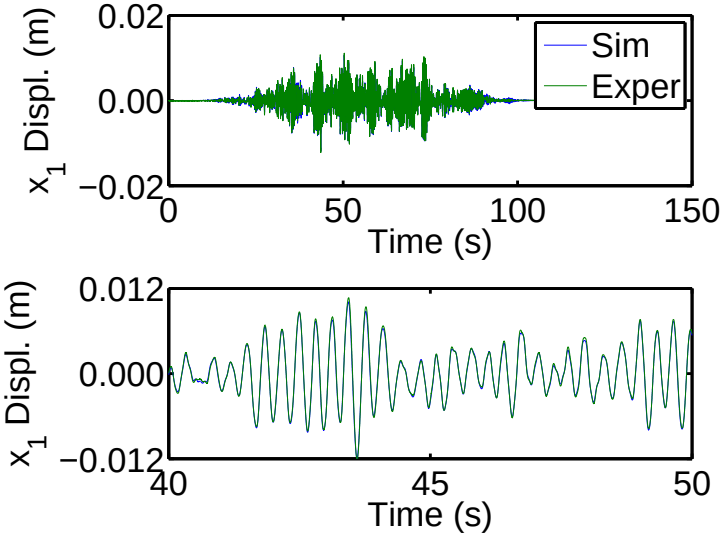


Figure 6.16: Displacement output, x_1, for the experimental system and the model, using the same excitation.

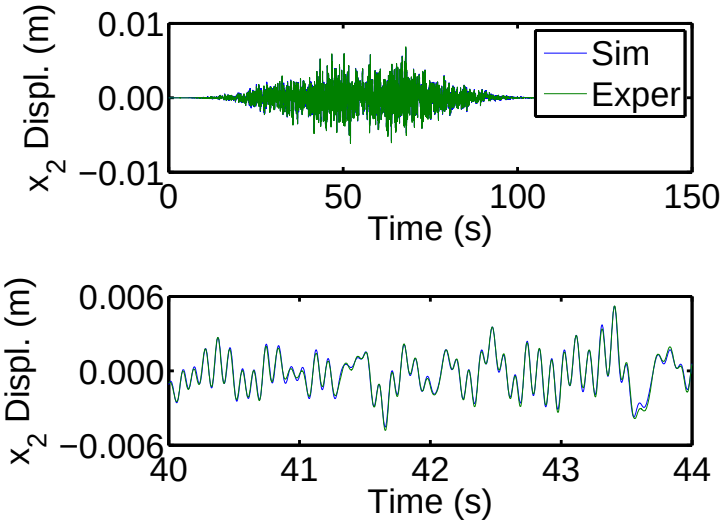


Figure 6.17: Displacement output, x_2, for the experimental system and the model, using the same excitation.

The system ID procedure was performed three times: at 0 lb, 100 lb (45.359 kg), and 200 lb (90.718 kg), added to m_1. These three tests represented the mean added mass and edges of interest for the family of systems controlled. The results are given in Table 6.4.

Table 6.4: Values from optimization

nom. added weight (lb)	0	100	200
m_1 (kg)	128.83	174.19	219.55
k_1 (N/m)	5.808e+04	6.180e+04	6.488e+04
d_1 (N/m-s)	203	237	274
k_2 (N/m)	3.489e+05	3.503e+05	3.466e+05

As can be seen, the parameters k_1 and d_1 appear to be reasonably strongly correlated with m_1. Intuitively, this makes sense since the resting pressure in the air springs is a function of m_1. While the spring constant is close to constant at a given mass (since the test is low displacement), there is a calculably noticeable difference at the pressures that result from the different masses. In terms of the damping, it is assumed that the different masses place more pressure on the bearings. However, the stiffness of the rubber bushings, denoted by the spring constant k_2, does not appear to significantly vary.

Thus, it was taken to be a reasonable control design decision to make m_1, k_1, and d_1 linear functions of the same variable ξ_1. The fitting was accomplished by standard least squares. Also, k_2 appeared to be constant and thus was taken to be the average of the three obtained values. Since the first two terms of the Legendre polynomial set is a basis for linear functions, it was trivial to express the parameters in terms of the basis functions. These expansions are given in Table 6.5.

Table 6.5: Function and expansions in terms of Legendre Polynomials with $\phi^0(\xi_1) = 1$, $\phi^0(\xi_1) = \xi_1$, ..., and $\xi_1 \in [-1, 1]$

$m_1(\xi_1) = 174.19\phi^0(\xi_1) + 45.36\phi^1(\xi_1)$, (kg)
$k_1(\xi_1) = 61580\phi^0(\xi_1) + 3398\phi^1(\xi_1)$, (N/m)
$d_1(\xi_1) = 237.9\phi^0(\xi_1) + 35.77\phi^1(\xi_1)$, (N/m-s)
$k_2 =$, (N/m)

For the derivations being employed, it is desired to place the parameters into state matrices that can be directly expanded in terms of the basis functions. Thus, Galerkin projections were used to eliminate the complication of having a mass denominator. For example:

$$\frac{k_1(\xi_1)}{m_1(\xi_1)} = 360.13\phi^0(\xi_1) - 75.66\phi^1(\xi_1) + 13.37\phi^2(\xi_1) - 2.13\phi^3(\xi_1) + 0.32\phi^4(\xi_1) + \ldots \quad (6.17)$$

6.2.3 Optimization

Four controllers were optimized using the iterative Polynomial Chaos methods. In these designs, the disturbance input matrix $\mathbf{B}_w$, incorporating $\mathbf{B}_{w,mech,design}$, used in the optimization processes is slightly different from that used for the system ID. The matrix $\mathbf{B}_{w,mech,design}$ contains small magnitude elements to acknowledge that disturbances could be produced from other sources besides the hydraulic actuator. However, these other disturbances would obviously be small compared to the hydraulic actuator and thus are weighted accordingly.

For each of these four optimizations, a finite difference approach, described in Section 5.8, was used to find the gradient since the resulting independent set of Lyapunov equations appeared to have better numerical stability than dependent equations that result from directly utilizing the gradient. Since the problems are not proven to have a single minimum, 10 separate optimizations—starting from random stable gains were performed. Stability was checked based on the closed loop

eigenvalues for the expanded system. For each individual optimizations, the process was stopped at or before 10001 iterations.

Optimizations for the following test cases are presented in Tables 6.6 to 6.9.

Case 1: Open Loop

Case 2: Standard $\mathcal{H}_2$

$$Q = \text{diag}\begin{bmatrix} 1 & 100 & .0001 & .0001 & 0 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 2e-5 \end{bmatrix}$$

Case 3: Minimum E[$\mathcal{H}_2^2$]

$$Q_M = \text{diag}\begin{bmatrix} 1 & 100 & .0001 & .0001 & 0 & 0 \end{bmatrix}, \quad R_M = \begin{bmatrix} 2e-5 \end{bmatrix}$$
$$Q_C = Q_M = \text{diag}\begin{bmatrix} 1 & 100 & .0001 & .0001 & 0 & 0 \end{bmatrix}, \quad R_C = R_M$$

Case 4: Low Variance Weight for x_2

$$Q_M = \text{diag}\begin{bmatrix} 1 & 100 & .0001 & .0001 & 0 & 0 \end{bmatrix}, \quad R_M = \begin{bmatrix} 2e-5 \end{bmatrix}$$
$$Q_C = \text{diag}\begin{bmatrix} 1 & 2000 & .0001 & .0001 & 0 & 0 \end{bmatrix}, \quad R_C = R_M$$

Case 5: Medium Variance Weight for x_2

$$Q_M = \text{diag}\begin{bmatrix} 1 & 100 & .0001 & .0001 & 0 & 0 \end{bmatrix}, \quad R_M = \begin{bmatrix} 2e-5 \end{bmatrix}$$
$$Q_C = \text{diag}\begin{bmatrix} 1 & 10000 & .0001 & .0001 & 0 & 0 \end{bmatrix}, \quad R_C = R_M$$

Case 6: High Variance Weight for x_2

$$Q_M = \text{diag}\begin{bmatrix} 1 & 100 & .0001 & .0001 & 0 & 0 \end{bmatrix}, \quad R_M = \begin{bmatrix} 2e-5 \end{bmatrix}$$
$$Q_C = \text{diag}\begin{bmatrix} 1 & 50000 & .0001 & .0001 & 0 & 0 \end{bmatrix}, \quad R_C = R_M$$

Each of the five controllers is nominally weighted such as to place emphasis on keeping m_2 close to its equilibrium point. Additionally, the Polynomial Chaos $\mathcal{H}_2$ controllers place various levels of emphasis on x_2's trajectory variance.

Table 6.6: Case 3: Results for 10 gain optimizations. Optimization stopped at or before 10001 iterations. Optimization number 8 is the minimum.

Opt	Gain Vals.	# of Iter.	$\mathcal{H}_2$
1	$K_0 = \begin{bmatrix} -89.73 & 393.65 & -442.11 & -147.13 & 313.17 & -490.14 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -343.88 & 1006.85 & -24.24 & 11.08 & -1644.73 & -85.55 \end{bmatrix}$	565	11.267
2	$K_0 = \begin{bmatrix} -361.11 & -297.23 & -301.28 & 103.79 & -227.81 & -301.19 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -344.75 & 1008.75 & -24.25 & 11.08 & -1648.28 & -85.54 \end{bmatrix}$	490	11.267
3	$K_0 = \begin{bmatrix} -195.38 & -310.35 & -306.57 & 182.22 & -197.24 & 41.67 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -343.36 & 1005.83 & -24.24 & 11.09 & -1642.59 & -85.55 \end{bmatrix}$	495	11.267
4	$K_0 = \begin{bmatrix} -349.13 & 197.90 & -121.63 & 360.01 & 353.66 & 93.56 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -343.63 & 1006.25 & -24.24 & 11.08 & -1643.62 & -85.55 \end{bmatrix}$	566	11.267
5	$K_0 = \begin{bmatrix} -158.03 & -210.27 & -158.81 & 34.08 & 227.11 & -190.71 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -344.00 & 1007.83 & -24.24 & 11.08 & -1645.49 & -85.55 \end{bmatrix}$	547	11.267
6	$K_0 = \begin{bmatrix} -327.04 & 479.75 & -228.55 & -247.67 & 375.74 & 237.31 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -344.68 & 1009.07 & -24.25 & 11.08 & -1648.19 & -85.54 \end{bmatrix}$	577	11.267
7	$K_0 = \begin{bmatrix} -215.59 & -30.78 & -435.22 & 488.33 & 82.79 & -76.50 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -341.75 & 1002.34 & -24.23 & 11.09 & -1635.95 & -85.56 \end{bmatrix}$	544	11.267
8	$K_0 = \begin{bmatrix} 29.82 & 140.53 & -290.93 & -120.18 & 283.33 & 180.85 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -344.64 & 1009.79 & -24.25 & 11.09 & -1648.36 & -85.55 \end{bmatrix}$	597	11.267
9	$K_0 = \begin{bmatrix} -48.58 & -456.10 & -472.81 & -187.31 & -487.14 & -116.03 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -344.21 & 1007.97 & -24.24 & 11.08 & -1646.19 & -85.55 \end{bmatrix}$	533	11.267
10	$K_0 = \begin{bmatrix} 183.12 & -407.16 & -464.66 & 112.40 & 108.54 & -484.24 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -343.86 & 1006.81 & -24.24 & 11.07 & -1644.55 & -85.55 \end{bmatrix}$	602	11.267

Table 6.7: Case 4: Results for 10 gain optimizations. Optimization stopped at or before 10001 iterations. Optimization number 5 is the minimum.

Opt	Gain Vals.	# of Iter.	$\mathcal{H}_2$
1	$K_0 = \begin{bmatrix} -451.71 & -119.85 & -87.21 & -98.61 & -79.00 & -123.05 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -194.12 & 1025.73 & -33.50 & 6.62 & -1325.18 & -106.51 \end{bmatrix}$	861	11.692
2	$K_0 = \begin{bmatrix} -117.09 & -247.21 & -157.07 & 467.80 & -20.19 & -131.67 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -195.13 & 1027.95 & -33.52 & 6.61 & -1329.16 & -106.52 \end{bmatrix}$	944	11.692
3	$K_0 = \begin{bmatrix} 250.46 & -373.80 & -456.94 & -129.06 & 193.30 & 435.82 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -194.08 & 1025.55 & -33.50 & 6.62 & -1324.93 & -106.51 \end{bmatrix}$	1045	11.692
4	$K_0 = \begin{bmatrix} -6.13 & -82.46 & -207.74 & -210.34 & 253.85 & -403.20 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -194.61 & 1026.42 & -33.51 & 6.62 & -1326.79 & -106.51 \end{bmatrix}$	873	11.692
5	$K_0 = \begin{bmatrix} -133.80 & -360.57 & -485.24 & 140.64 & 237.72 & -473.24 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -195.29 & 1029.66 & -33.52 & 6.61 & -1330.38 & -106.53 \end{bmatrix}$	1024	11.692
6	$K_0 = \begin{bmatrix} -441.56 & 38.51 & -309.85 & 99.48 & -207.74 & -408.71 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -193.39 & 1021.08 & -33.48 & 6.63 & -1320.69 & -106.47 \end{bmatrix}$	756	11.692
7	$K_0 = \begin{bmatrix} -26.59 & 183.18 & -366.70 & -35.92 & -428.75 & 81.22 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -193.35 & 1022.64 & -33.49 & 6.62 & -1321.30 & -106.49 \end{bmatrix}$	787	11.692
8	$K_0 = \begin{bmatrix} 65.98 & -244.74 & -261.51 & -484.05 & -115.26 & 257.30 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -193.67 & 1022.50 & -33.49 & 6.62 & -1322.09 & -106.49 \end{bmatrix}$	1023	11.692
9	$K_0 = \begin{bmatrix} 75.16 & -91.91 & -304.29 & 12.17 & 213.35 & 367.39 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -193.71 & 1023.22 & -33.49 & 6.62 & -1322.43 & -106.49 \end{bmatrix}$	919	11.692
10	$K_0 = \begin{bmatrix} 136.79 & 444.06 & -459.22 & -214.33 & 177.25 & 86.18 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -194.15 & 1023.65 & -33.50 & 6.63 & -1324.04 & -106.49 \end{bmatrix}$	1022	11.692

Table 6.8: Case 5: Results for 10 gain optimizations. Optimization stopped at or before 10001 iterations. Optimization number 1 is the minimum.

Opt	Gain Vals.	# of Iter.	$\mathcal{H}_2$
1	$K_0 = \begin{bmatrix} 3.01 & 221.98 & -193.79 & -387.84 & -56.71 & -33.24 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -42.93 & 1141.47 & -41.45 & 0.22 & -1193.15 & -115.01 \end{bmatrix}$	10001	12.449
2	$K_0 = \begin{bmatrix} -39.19 & -54.69 & -412.26 & -56.52 & -133.70 & -197.47 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -40.38 & 1125.20 & -41.33 & 0.26 & -1176.68 & -114.79 \end{bmatrix}$	10001	12.450
3	$K_0 = \begin{bmatrix} -127.82 & -426.31 & -300.16 & -450.51 & 66.71 & -378.08 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -42.21 & 1136.36 & -41.42 & 0.24 & -1188.28 & -114.95 \end{bmatrix}$	10001	12.450
4	$K_0 = \begin{bmatrix} 97.91 & 449.15 & -211.20 & 388.83 & -398.41 & -434.69 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -41.80 & 1134.13 & -41.38 & 0.24 & -1185.66 & -114.91 \end{bmatrix}$	10001	12.450
5	$K_0 = \begin{bmatrix} -265.70 & 433.10 & -436.87 & -235.78 & 499.53 & -288.01 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -39.70 & 1121.94 & -41.27 & 0.37 & -1173.37 & -114.71 \end{bmatrix}$	10001	12.450
6	$K_0 = \begin{bmatrix} -369.06 & -404.59 & -485.14 & -211.81 & 316.73 & 485.48 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -38.59 & 1115.29 & -41.23 & 0.29 & -1166.24 & -114.64 \end{bmatrix}$	10001	12.451
7	$K_0 = \begin{bmatrix} -272.85 & 16.25 & -41.80 & 203.20 & 82.48 & 9.21 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -40.52 & 1128.05 & -41.35 & 0.28 & -1178.79 & -114.90 \end{bmatrix}$	10001	12.450
8	$K_0 = \begin{bmatrix} -425.71 & -306.76 & -120.40 & -223.57 & 270.88 & -186.07 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -38.26 & 1111.42 & -41.21 & 0.32 & -1163.10 & -114.58 \end{bmatrix}$	10001	12.451
9	$K_0 = \begin{bmatrix} -80.57 & -287.01 & -464.40 & -418.84 & 350.57 & -159.80 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -39.11 & 1118.35 & -41.25 & 0.28 & -1169.02 & -114.66 \end{bmatrix}$	10001	12.451
10	$K_0 = \begin{bmatrix} -33.85 & 413.76 & -271.42 & 362.04 & 156.62 & 391.18 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -40.96 & 1130.75 & -41.35 & 0.26 & -1180.95 & -114.84 \end{bmatrix}$	10001	12.450

Table 6.9: Case 6: Results for 10 gain optimizations. Optimization stopped at or before 10001 iterations. Optimization number 1 is the minimum.

Opt	Gain Vals.	# of Iter.	$\mathcal{H}_2$
1	$K_0 = \begin{bmatrix} -11.86 & 492.65 & -126.67 & 31.38 & -318.68 & 1.94 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} 40.07 & 1048.74 & -32.35 & -10.58 & -1121.67 & -58.50 \end{bmatrix}$	1989	14.189
2	$K_0 = \begin{bmatrix} 65.05 & 469.17 & -476.26 & 370.22 & -473.12 & 19.53 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} 40.57 & 1044.82 & -32.32 & -10.60 & -1117.61 & -58.45 \end{bmatrix}$	1946	14.189
3	$K_0 = \begin{bmatrix} -307.71 & 215.69 & -249.33 & 433.86 & -362.81 & 21.62 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} 40.70 & 1043.58 & -32.31 & -10.60 & -1116.34 & -58.43 \end{bmatrix}$	1801	14.189
4	$K_0 = \begin{bmatrix} 181.34 & 165.82 & -365.28 & -477.51 & -237.80 & -383.48 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} 37.02 & 1074.14 & -32.54 & -10.47 & -1146.92 & -58.81 \end{bmatrix}$	9507	14.189
5	$K_0 = \begin{bmatrix} -430.68 & 352.93 & -319.67 & -467.58 & 233.93 & 36.52 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} 36.72 & 1076.20 & -32.56 & -10.47 & -1149.37 & -58.86 \end{bmatrix}$	8362	14.190
6	$K_0 = \begin{bmatrix} -223.97 & -131.54 & -487.11 & 389.21 & 366.02 & -245.75 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -36.18 & 1577.08 & -40.20 & -7.75 & -1621.70 & -74.76 \end{bmatrix}$	10001	14.293
7	$K_0 = \begin{bmatrix} 108.11 & -324.00 & -497.97 & 290.22 & 13.61 & -286.77 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -139.85 & 2082.54 & -51.89 & -7.63 & -2094.11 & -96.15 \end{bmatrix}$	10001	14.412
8	$K_0 = \begin{bmatrix} -396.55 & -342.66 & -92.49 & -92.24 & -447.31 & 441.82 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -168.03 & 2215.13 & -54.86 & -8.22 & -2219.45 & -100.86 \end{bmatrix}$	10001	14.435
9	$K_0 = \begin{bmatrix} -350.03 & -115.63 & -188.94 & -331.47 & 396.65 & -177.28 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} -156.91 & 2161.90 & -53.68 & -7.95 & -2171.13 & -98.97 \end{bmatrix}$	10001	14.426
10	$K_0 = \begin{bmatrix} 234.00 & -89.10 & -100.21 & 5.52 & -330.69 & 24.75 \end{bmatrix}$		
	$K_{min} = \begin{bmatrix} 40.38 & 1046.26 & -32.32 & -10.59 & -1118.80 & -58.46 \end{bmatrix}$	2245	14.189

As can be seen for these four optimizations, when compared to the initial starting conditions, all 10 tests for each case converge to the same approximate set of gains. This hints at a possible tendency for the required optimization problems to have a unique global minimum.

The gains resulting from the process are:

Case 1: Open Loop

Case 2: Standard $\mathcal{H}_2$

$$\mathbf{K}_2 = \begin{bmatrix} -382.31 & 978.72 & -22.89 & 10.43 & -1729.66 & -79.41 \end{bmatrix}$$

Case 3: Minimum E[$\mathcal{H}_2^2$]

$$\mathbf{K}_3 = \begin{bmatrix} -344.64 & 1009.79 & -24.25 & 11.09 & -1648.36 & -85.55 \end{bmatrix}$$

Case 4: Low Variance Weight for x_2

$$\mathbf{K}_4 = \begin{bmatrix} -195.29 & 1029.66 & -33.52 & 6.61 & -1330.38 & -106.53 \end{bmatrix}$$

Case 5: Medium Variance Weight for x_2

$$\mathbf{K}_5 = \begin{bmatrix} -42.93 & 1141.47 & -41.45 & 0.22 & -1193.15 & -115.01 \end{bmatrix}$$

Case 6: High Variance Weight for x_2

$$\mathbf{K}_6 = \begin{bmatrix} 40.07 & 1048.74 & -32.35 & -10.58 & -1121.67 & -58.50 \end{bmatrix}$$

6.2.4 Gain Analysis

The selected gains are now analytically examined in three different ways. These include Bode plots examining L_2 and $\mathcal{H}_2$ norms, and looking at effects on the eigenvalues.

Bode plots for the six test cases are shown in Figures 6.18 to 6.20. These figures all show the system for five representative realizations across the span of mass added to m_1. As can be seen in these plots, the change in mass drastically effects the open loop states. Changes occur in the

shifting of the resonant peak, the amplitude of the resonant peak, and the phase. That said, by the definition of open loop, the required current for the controller is zero for all open loop realization systems. Further, for the experimental system, the $\mathcal{H}_2$ controller designed from the mean m_1 value is very similar to the minimum expected $\mathcal{H}_2^2$ controller. This is not surprising given that there are minimal differences in the two gain matrices. However, they are not exactly the same and slight difference in the first resonant peak can be observed for x_1 and x_2. That said, both do provide substantial decrease to the variability present in the first visible resonance peak.

The most significant trend present can be seen in Figure 6.19 (c) to (f). In this plot, it can be seen that as the weighting for the trajectory variance of x_2 is increased, the variability of the Bode plots decreases. Additionally, optimal control's characteristic tradeoff is visible—as the variance decreases, the first resonant peak's mean amplitude can be seen to increase.

These changes can be quantitatively examined by looking at the L_2 norms presented in Table 6.10. This figure gives the L_2 norm for the mean trajectory and standard deviation trajectory for the displacement and velocity of m_1 and m_2, as well as the current command, I. The norms are calculated for all six systems formed from the six test cases. The decrease in the L_2 norm of the variance of x_2 from System 3 to System 6 can be seen. Additionally, the tradeoff with the increase in the L_2 norm of the mean trajectory for x_2 can be seen. Further, the previously noted small difference in the heights of the first resonance peak between the minimal $\mathcal{H}_2^2$ solution and the $\mathcal{H}_2$ solution using the mean m_1 value are visible in the L_2 norms.

The results for the L_2 integrals along time also have meaning in that they can obviously be used with the state weightings to obtain $\mathcal{H}_2$ metrics. These metrices are presented in Table 6.11. The table presents the cost different controllers incur from the five cost functions. The lowest cost of this set of cotrollers is denoted for each cost function. Not surprisingly, the *optimal* is the lowest one. It is significant to note that the greatest meaning is obtained from looking at the associated costs for multiple systems over a single cost function. For example, the large discrepancies in

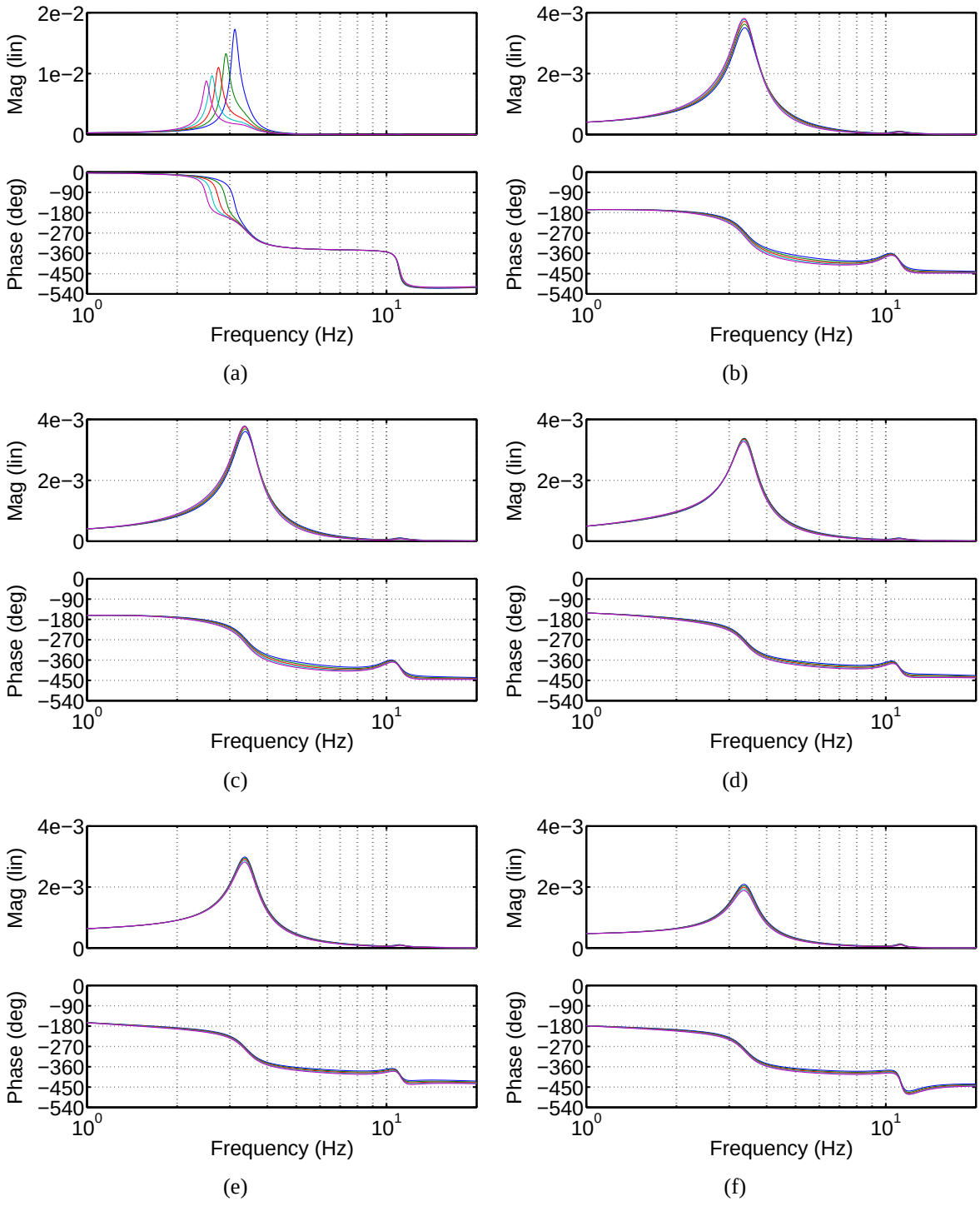


Figure 6.18: Transfer function from x_3 to x_1. (a) Case 1: Open Loop, (b) Case 2: Standard $\mathcal{H}_2$, (c) Case 3: Minimum E$[\mathcal{H}_2^2]$, (d) Case 4: Low Variance Weight for x_2, (e) Case 5: Medium Variance Weight for x_2, and (f) Case 6: High Variance Weight for x_2. Legend: — 0 lb added, — 50 lb (22.680 kg) added, — 100 lb (45.359 kg) added, — 150 lb (68.039 kg) added, — 200 lb (90.718 kg) added

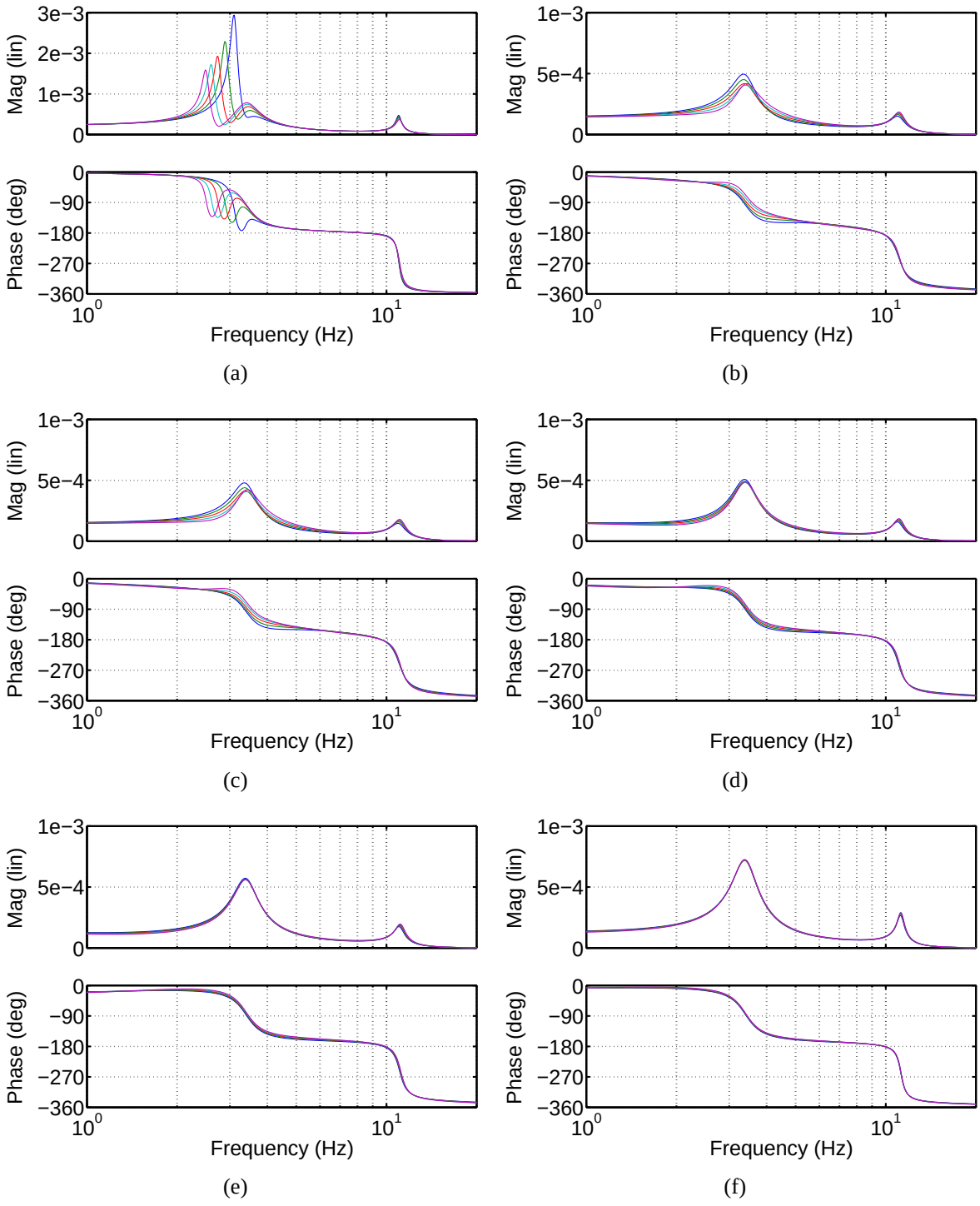


Figure 6.19: Transfer function from x_3 to x_2. (a) Case 1: Open Loop, (b) Case 2: Standard $\mathcal{H}_2$, (c) Case 3: Minimum E$[\mathcal{H}_2^2]$, (d) Case 4: Low Variance Weight for x_2, (e) Case 5: Medium Variance Weight for x_2, and (f) Case 6: High Variance Weight for x_2. Legend: — 0 lb added, — 50 lb (22.680 kg) added, — 100 lb (45.359 kg) added, — 150 lb (68.039 kg) added, — 200 lb (90.718 kg) added

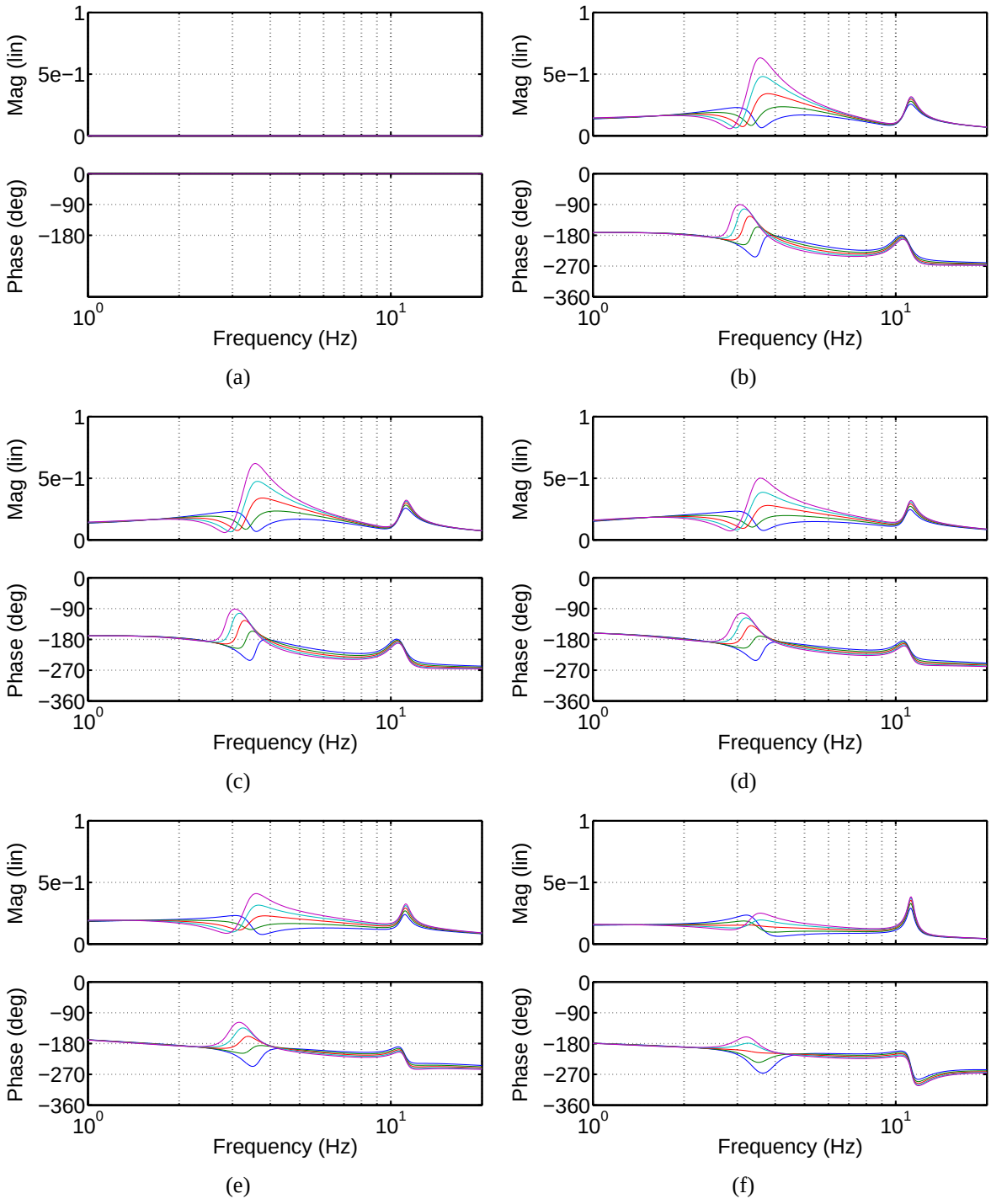


Figure 6.20: Transfer function from x_3 to u. (a) Case 1: Open Loop, (b) Case 2: Standard $\mathcal{H}_2$, (c) Case 3: Minimum E$[\mathcal{H}_2^2]$, (d) Case 4: Low Variance Weight for x_2, (e) Case 5: Medium Variance Weight for x_2, and (f) Case 6: High Variance Weight for x_2. Legend: — 0 lb added, — 50 lb (22.680 kg) added, — 100 lb (45.359 kg) added, — 150 lb (68.039 kg) added, — 200 lb (90.718 kg) added

Table 6.10: Theoretical L_2 norms, induced solely by input into filter, calculated using Lyapunov equations. Note: $\mathrm{E}\left[|\mathrm{x}_1(\xi_1,\mathrm{t})|^2\right] = \left|\mathrm{E}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right|^2 + \left|\mathrm{std}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right|^2$, as well as analogous relationships for the other states.

	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Sys 6
$\left\|\mathrm{E}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right\|^2$	3.38626e-05	2.47493e-05	2.54213e-05	2.18838e-05	1.74993e-05	8.39568e-06
$\left\|\mathrm{std}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right\|^2$	3.66053e-05	2.87453e-07	2.69923e-07	1.10927e-07	5.21418e-08	3.20922e-08
$\left\|\mathrm{E}\left[\mathrm{x}_2(\xi_1,\mathrm{t})\right]\right\|^2$	2.47832e-06	6.34082e-07	6.15028e-07	6.74177e-07	7.98314e-07	1.26589e-06
$\left\|\mathrm{std}\left[\mathrm{x}_2(\xi_1,\mathrm{t})\right]\right\|^2$	9.25272e-07	9.35619e-09	8.47968e-09	3.30539e-09	1.53264e-09	6.30239e-10
$\left\|\mathrm{E}\left[\dot{\mathrm{x}}_1(\xi_1,\mathrm{t})\right]\right\|^2$	1.16619e-02	1.13499e-02	1.15845e-02	9.75223e-03	7.62746e-03	3.65238e-03
$\left\|\mathrm{std}\left[\dot{\mathrm{x}}_1(\xi_1,\mathrm{t})\right]\right\|^2$	1.20062e-02	1.53892e-04	1.42696e-04	5.99226e-05	2.95513e-05	1.84872e-05
$\left\|\mathrm{E}\left[\dot{\mathrm{x}}_2(\xi_1,\mathrm{t})\right]\right\|^2$	2.11695e-03	7.13332e-04	6.82615e-04	6.96931e-04	7.76463e-04	1.29869e-03
$\left\|\mathrm{std}\left[\dot{\mathrm{x}}_2(\xi_1,\mathrm{t})\right]\right\|^2$	3.08011e-04	5.90563e-06	5.42280e-06	2.92753e-06	2.09955e-06	8.30360e-07
$\left\|\mathrm{E}\left[\mathrm{u}(\xi_1,\mathrm{t})\right]\right\|^2$	0.00000e+00	1.25250e+00	1.31396e+00	1.38814e+00	1.34526e+00	6.96783e-01
$\left\|\mathrm{std}\left[\mathrm{u}(\xi_1,\mathrm{t})\right]\right\|^2$	0.00000e+00	7.03247e-02	6.73692e-02	5.18190e-02	4.13631e-02	1.79586e-02

costs for J_6 from the optimal solution allows understanding the overall driving force of the heavy penalty for variance of x_2 in this this cost function. Also, it is interesting to note that the converse is not true—the effect of the heavy variance weight does not appear to be as dramatic in terms of increases to $\mathcal{H}_2^2$ for other cost functions applied to System 6.

Table 6.11: Theoretical $\mathcal{H}_2^2$, calculated using Lyapunov equations. The table shows the norm for each cost function for the open loop system and each closed loop system. A box is placed around the minimum norm for each cost function—corresponding to the closed loop system with lowest norm.

	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Sys 6
J_2	3.76294e-04	**1.14819e-04**	1.14846e-04	1.18431e-04	1.25376e-04	1.49724e-04
J_3	4.13437e-04	1.17059e-04	**1.16910e-04**	1.19593e-04	1.26112e-04	1.49872e-04
J_4	2.17145e-03	1.34836e-04	1.33022e-04	**1.25874e-04**	1.29024e-04	1.51069e-04
J_5	9.57363e-03	2.09686e-04	2.00859e-04	1.52317e-04	**1.41285e-04**	1.56111e-04
J_6	4.65845e-02	5.83933e-04	5.40046e-04	2.84532e-04	2.02591e-04	**1.81321e-04**

A plot of eigenvalues for five realizations of each the six test systems is presented in Figure 6.21. This plot illustrates the increased "closeness" in the systems' dominant poles. However, as can be seen in the most heavily weighted x_2 variance design (Case 6), a cost for this "closeness" is paid in terms of severely slowing down the dominant poles.

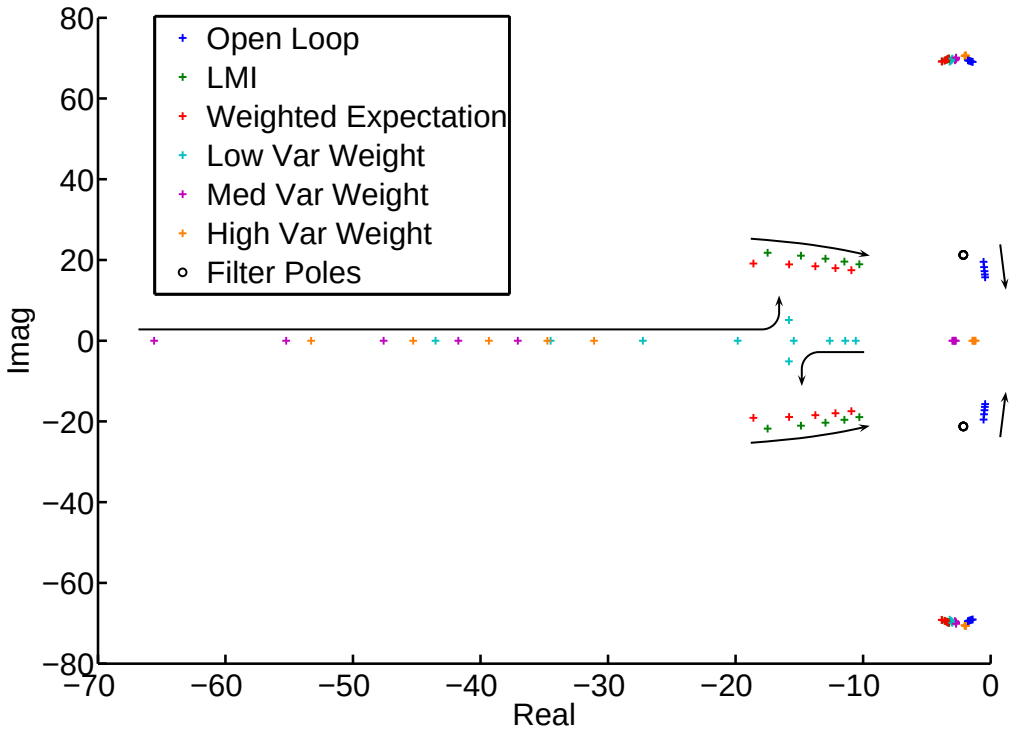


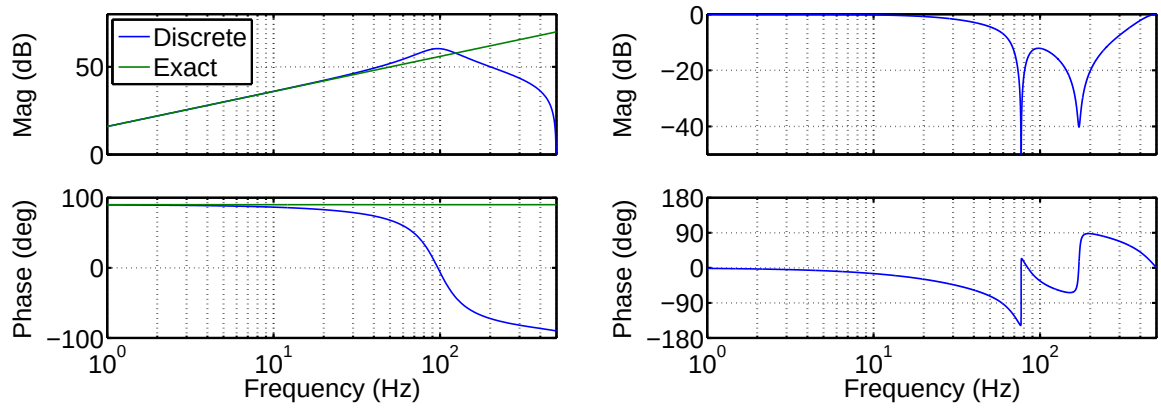
Figure 6.21: Poles for 0 lb, 50 lb (22.680 kg), 100 lb (45.359 kg), 150 lb (68.039 kg), and 200 lb (90.718 kg) added to m_1. The arrows denote closed loop systems' pole movements along the increase in m_1. All six test cases are overlaid.

6.2.5 Filter Design

In order to implement the controllers, it was necessary to design several filters. These were implemented in discrete time, with a sample rate of 0.001 s. An attempt was made to closely match the filters in amplitude and phase for frequencies below approximately 10 Hz. This was done to minimize the effect of the filters on the test

The first of these, a differentiator, allowed for obtaining a velocity of the motion of the hydraulic cylinder, $\dot{x}_3$. This filter was designed by combining a second-order low pass filter ($\omega_n = 200\pi$, $\zeta = .3$) with a differentiator. The Tustin transform was then used to convert to the following discrete system:

$$D(z) = \frac{153.3510 - 153.3510z^{-2}}{1 - 1.4004z^{-1} - 0.7071z^{-2}}. \tag{6.18}$$



(a) (b)

Figure 6.22: Discrete filters: (a) differentiation filter as well as a lines for an exact differentiator, (b) notch filters

The discrete transfer function is plotted in Figure 6.22(a).

Additionally, the string potentiometers utilized contain a spring and rotational mass. As such, they slightly resonate at approximately 180 Hz. The high bandwidth of the linear electric motors allowed for the signal to be fed back and reinforced. Also, a structural mode was found slightly around 80 Hz that was desirable to notch. In order to eliminate these, a pair of notch filters was designed to shape the control actuator current command signal:

$$N(z) = \left[\frac{0.3686 - 0.3439z^{-1} + 0.3559z^{-2}}{1 - 0.3439z^{-1} - 0.2755z^{-2}}\right]\left[\frac{0.9093 - 1.6100z^{-1} + 0.9091z^{-2}}{1 - 1.615z^{-1} + 0.8214z^{-2}}\right]. \tag{6.19}$$

These filters were designed to minimally effect lower frequencies while sufficiently lowering the response around the required frequencies. Also, the filters were designed to partially cancel each other's phase effects in the lower frequencies. This caused a positive phase for the filter at some of the higher frequencies, but was not a problem because of the adequate phase margin of the overall system. These filters are shown in Figure 6.22(b).

6.2.6 Test Results

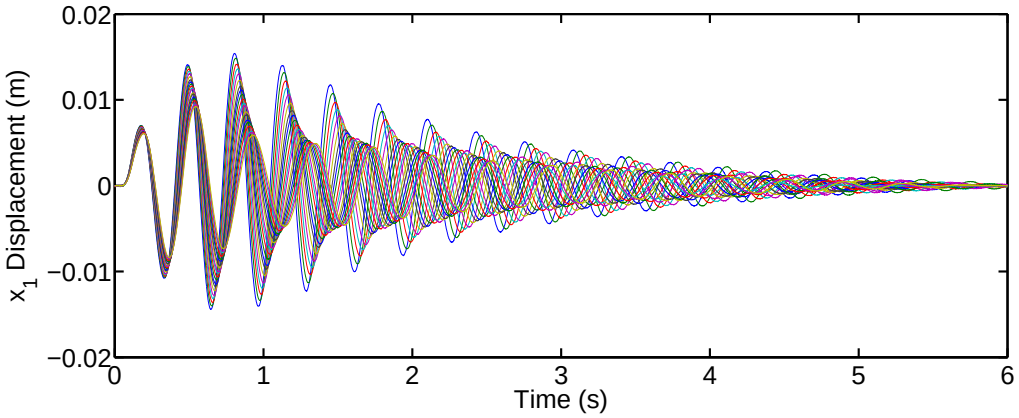
Next, all six control test cases were experimentally run using an evenly-spaced grid of weights. In order to represent a uniformly distributed mass, 20 grid points were selected to run a range from 5 lb (2.268 kg) to 195 lb (88.451 kg), in 10 lb (4.535 kg) increments. The points were equally spaced in order to ensure equal importance of the tests runs. As defined in the model, these test runs were created through adding the denoted mass to m_1. In order to minimize any hypothetical significance in the order the tests were run, the tests for the different mass values were run according to a random order.

The open loop system impulse response trajectories for $\mathrm{x}_1(t,\xi)$ and $\mathrm{x}_2(t,\xi)$, are shown in Figure 6.23. These responses represent the output when ξ takes the value of the defined set of grid points.

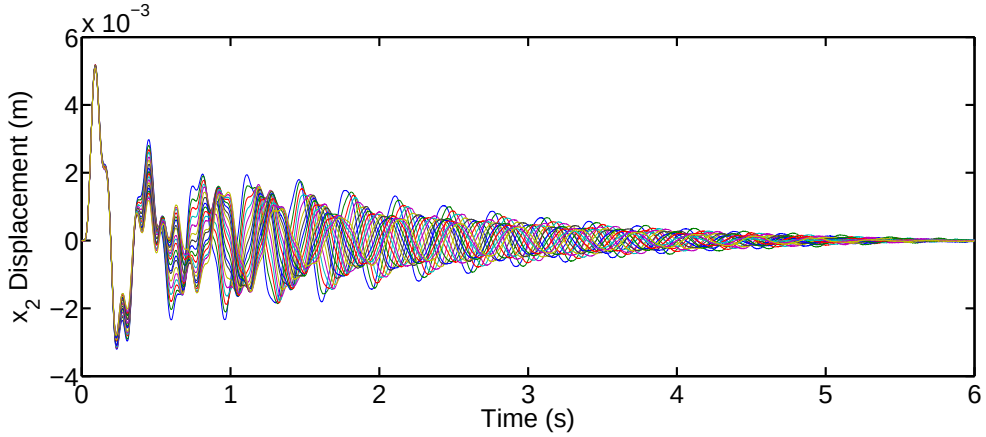
The experimental responses for all six control cases are shown in Figures 6.24 to 6.29. Additionally, these figures contain the simulated model response adjacent to the experimental results. Only four of the twenty obtained trajectories are displayed in each plot in order to allow easily interpreting the figures.

As can be seen, for the most part, nominal trajectories for the experimental results are similar to those predicted by the model. One exception is that obviously Case 6 does not agree well with the expected results in that it demonstrates continued oscillations. Additionally, while the trajectories for x_2 appear to be moving closer together as expected in test Cases 4 and 5, it is interesting to quantify the trend.

The results are quantified through L_2 and $\mathcal{H}_2$ norms. To accomplish this, the square mean and population variance, at each sampled time (0.001 second increments) for the tests, are calculated. The population variance is used in preference to a sample variance since a grid, instead of a random sample, was selected to represent the population. The mean is squared at each time value to create



(a)



(b)

Figure 6.23: The set of twenty open loop trajectories created by setting m_1 according to a grid.

the square mean trajectory. Next, the integrals can be approximated through trapezoidal integration. In order to put all of the tests on equal footing, the trapezoidal integration was performed over three seconds. The one exception to this was that the open loop test was integrated over eight seconds. It could have been argued to integrate Case 6 over a larger bound—however, it is already known that this case represents a less desirable controller and so it was a goal to maintain consistency in the analysis. This creates the L_2 norms for square mean and variance trajectories given in Table 6.12.

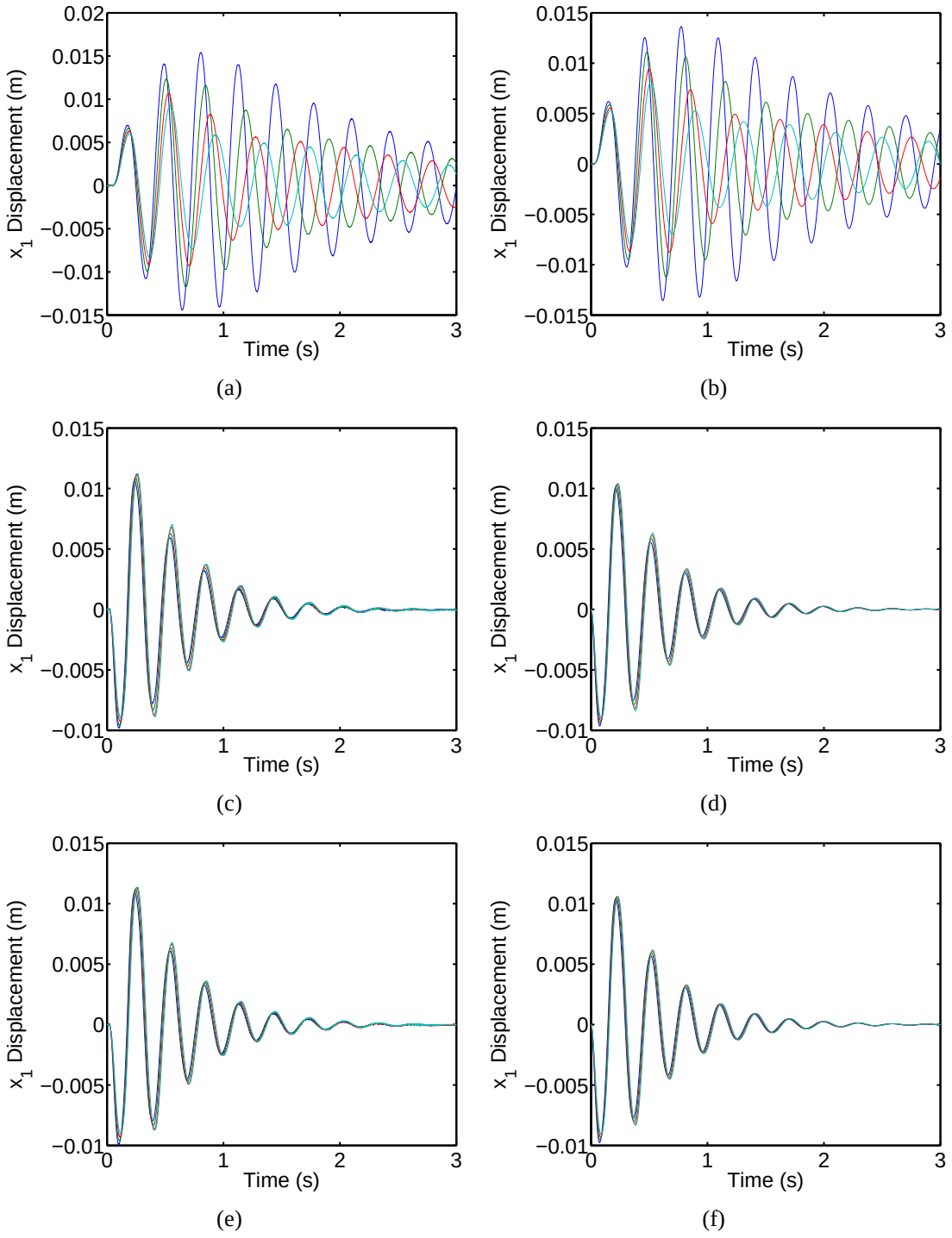


Figure 6.24: Responses for x_1. Case 1: Open Loop–(a) experimental, (b) simulation. Case 2: Standard $\mathcal{H}_2$–(c) experimental, (d) simulation. Case 3: Minimum E[$\mathcal{H}_2^2$]–(e) experimental, (f) simulation. Legend: — 5 lb (2.268 kg) added, — 65 lb (29.483 kg) added, — 135 lb (61.235 kg) added, — 195 lb (88.451 kg) added

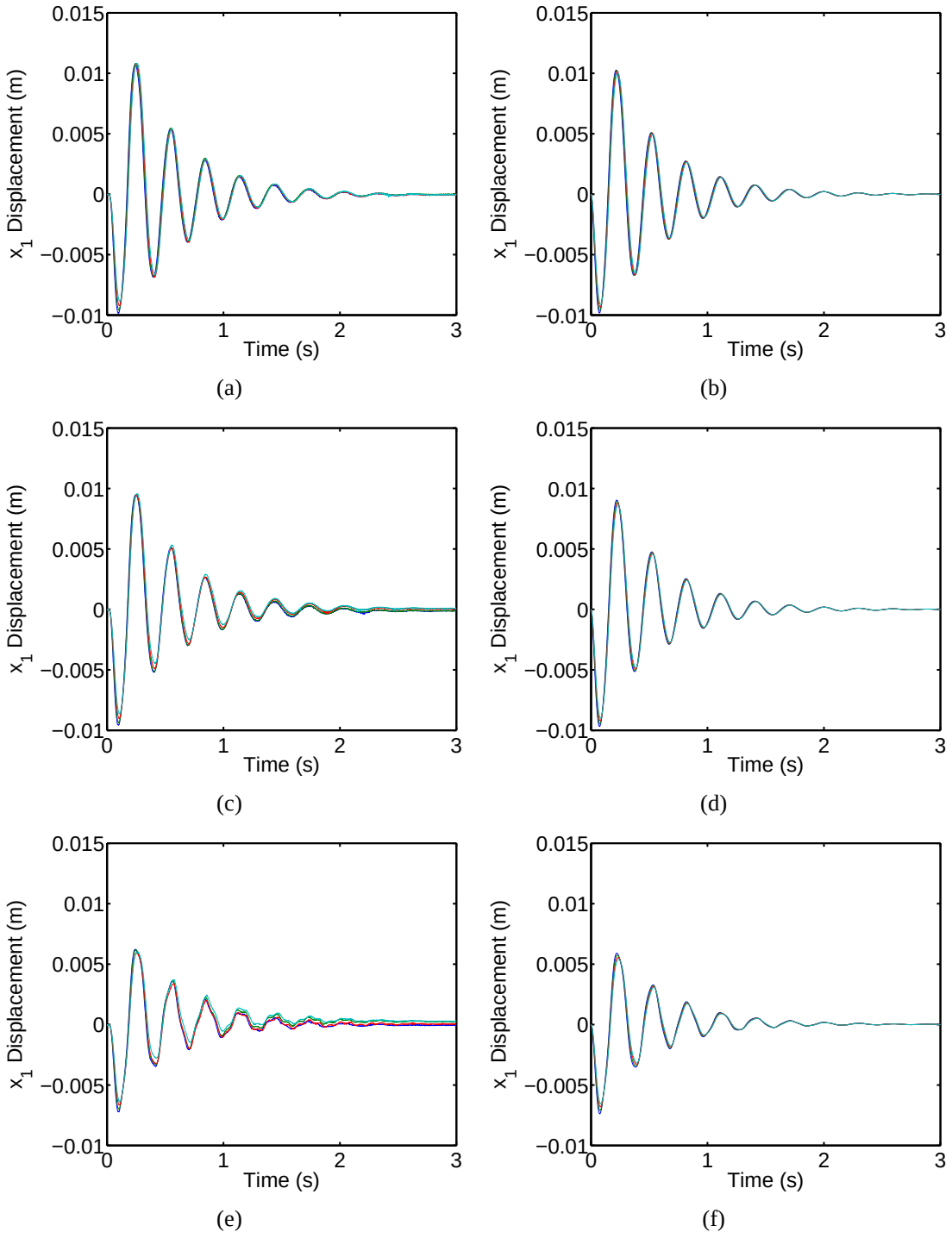


Figure 6.25: Responses for x_1. Case 4: Low Variance Weight for x_2–(a) experimental, (b) simulation. Case 5: Medium Variance Weight for x_2–(c) experimental, (d) simulation. Case 6: High Variance Weight for x_2–(e) experimental, (f) simulation. Legend: — 5 lb (2.268 kg) added, — 65 lb (29.483 kg) added, — 135 lb (61.235 kg) added, — 195 lb (88.451 kg) added

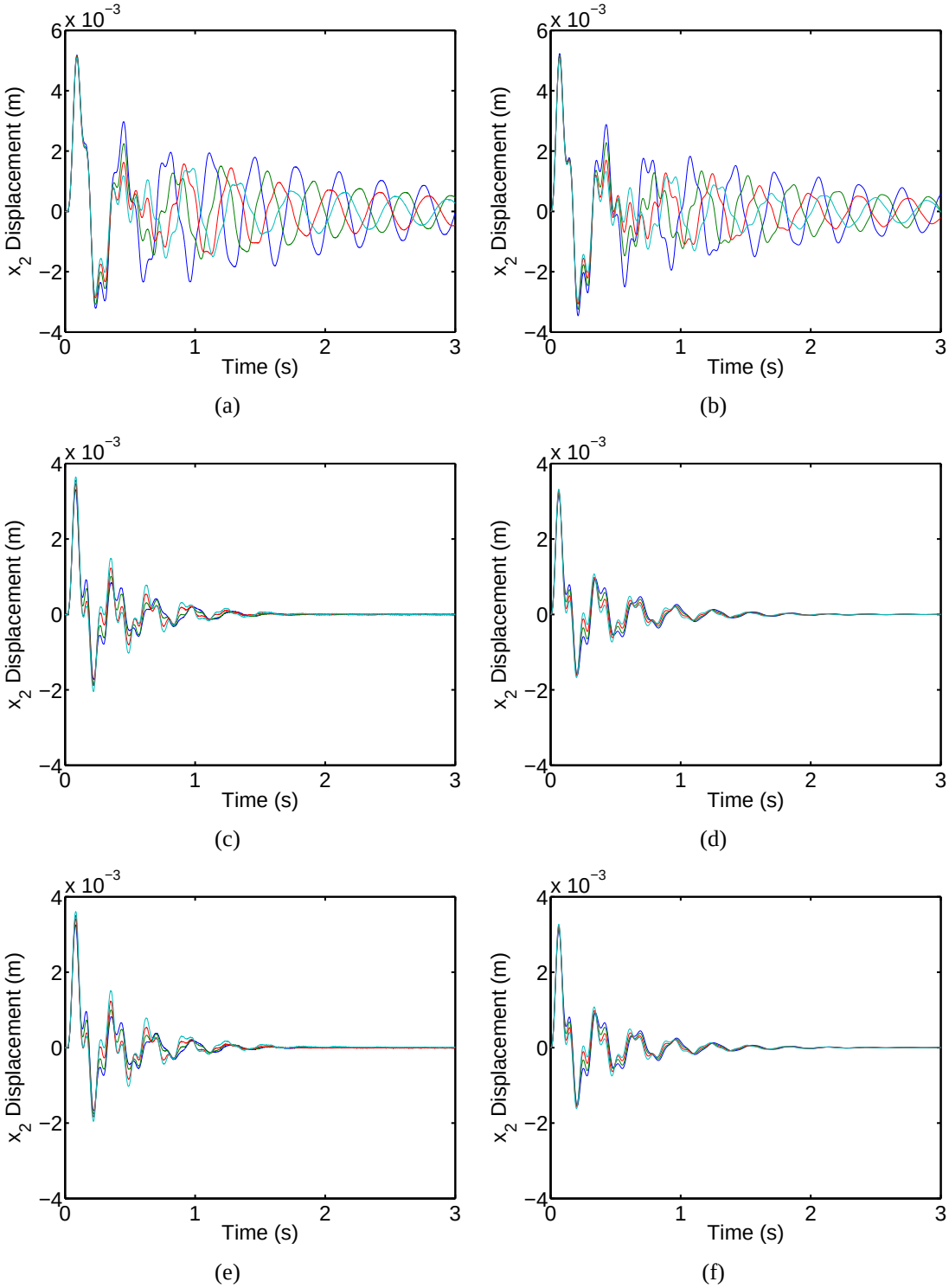


Figure 6.26: Responses for x_2. Case 1: Open Loop–(a) experimental, (b) simulation. Case 2: Standard $\mathcal{H}_2$–(c) experimental, (d) simulation. Case 3: Minimum E[$\mathcal{H}_2^2$]–(e) experimental, (f) simulation. Legend: — 5 lb (2.268 kg) added, — 65 lb (29.483 kg) added, — 135 lb (61.235 kg) added, — 195 lb (88.451 kg) added

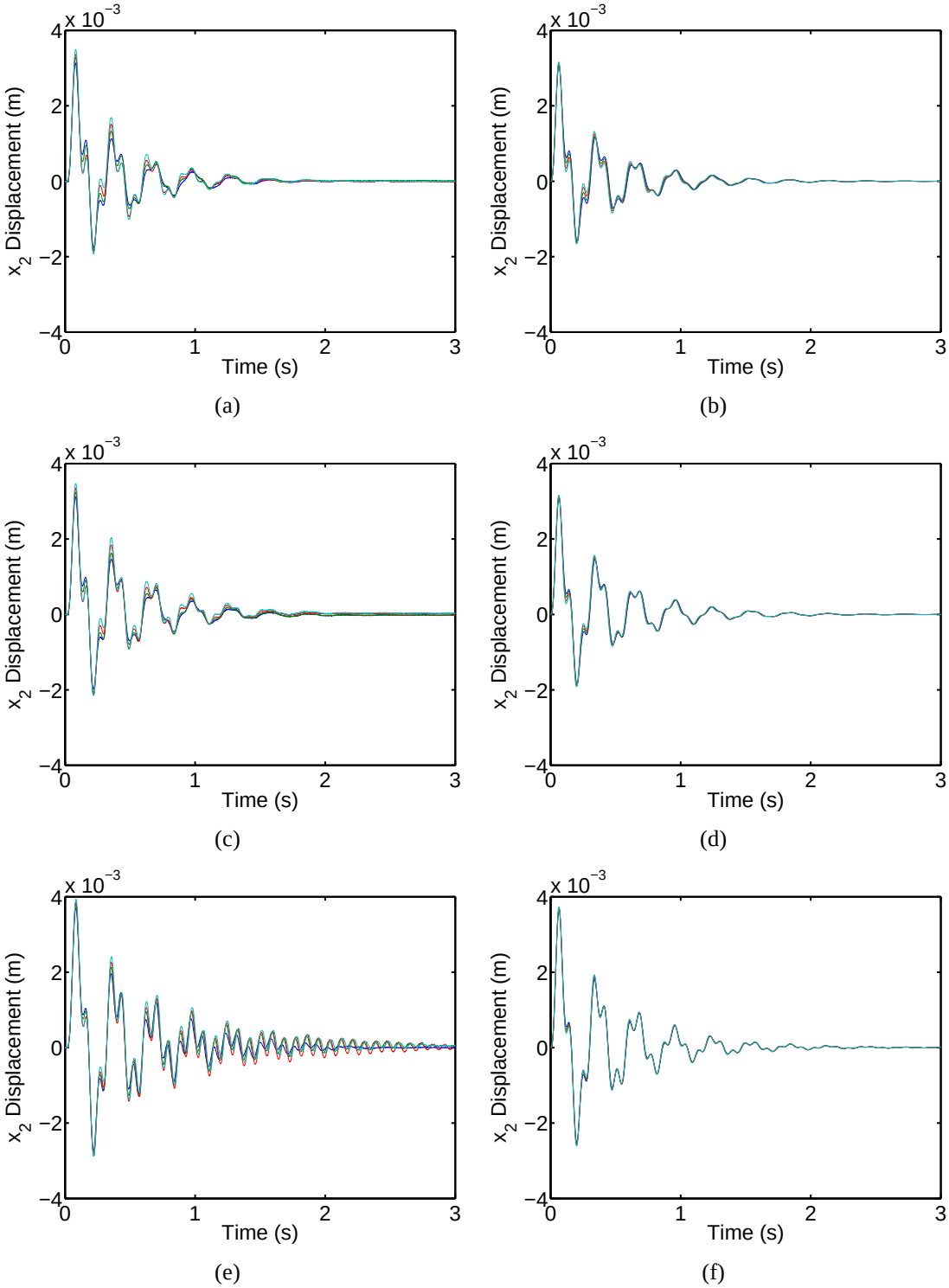


Figure 6.27: Responses for x_2. Case 4: Low Variance Weight for x_2–(a) experimental, (b) simulation. Case 5: Medium Variance Weight for x_2–(c) experimental, (d) simulation. Case 6: High Variance Weight for x_2–(e) experimental, (f) simulation. Legend: — 5 lb (2.268 kg) added, — 65 lb (29.483 kg) added, — 135 lb (61.235 kg) added, — 195 lb (88.451 kg) added

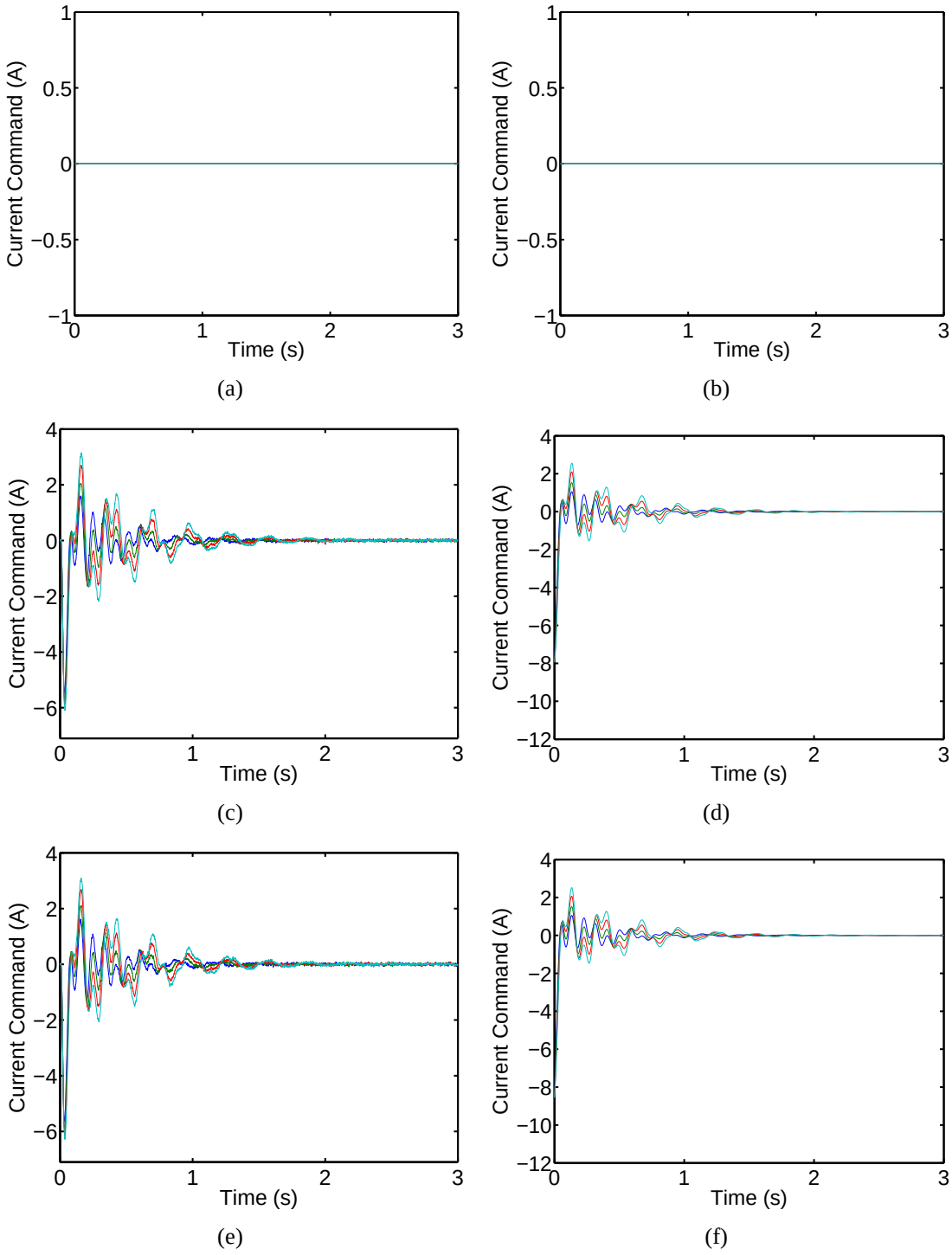


Figure 6.28: Current command (I) response. Case 1: Open Loop–(a) experimental, (b) simulation. Case 2: Standard $\mathcal{H}_2$–(c) experimental, (d) simulation. Case 3: Minimum $\mathrm{E}[\mathcal{H}_2^2]$–(e) experimental, (f) simulation. Legend: — 5 lb (2.268 kg) added, — 65 lb (29.483 kg) added, — 135 lb (61.235 kg) added, — 195 lb (88.451 kg) added

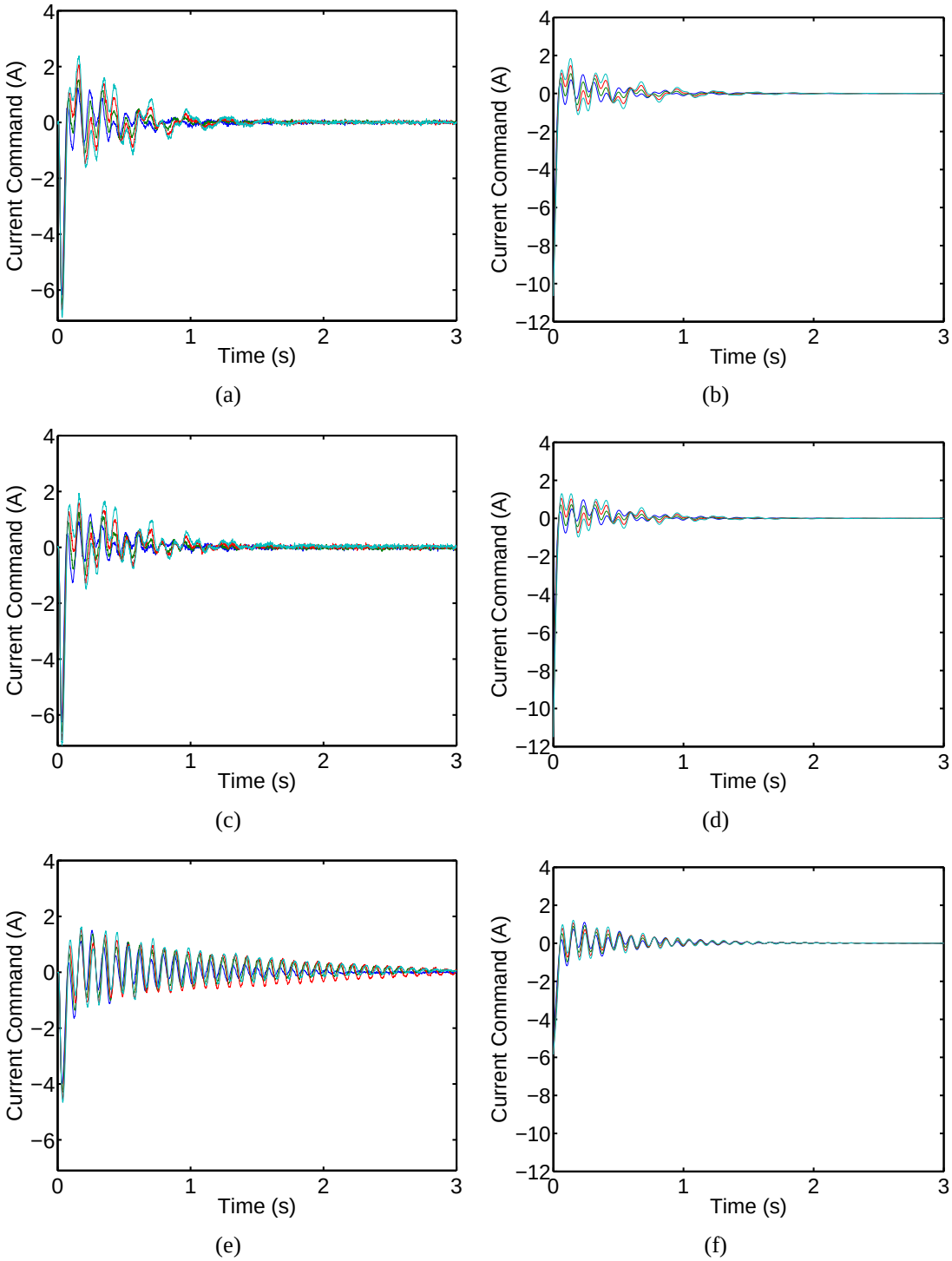


Figure 6.29: Current command (I) response. Case 4: Low Variance Weight for x_2–(a) experimental, (b) simulation. Case 5: Medium Variance Weight for x_2–(c) experimental, (d) simulation. Case 6: High Variance Weight for x_2–(e) experimental, (f) simulation. Legend: — 5 lb (2.268 kg) added, — 65 lb (29.483 kg) added, — 135 lb (61.235 kg) added, — 195 lb (88.451 kg) added

Table 6.12: Calculated experimental L_2 norms computed using trapezoidal integration. Note: $\mathrm{E}\left[|\mathrm{x}_1(\xi_1,\mathrm{t})|^2\right] = \left|\mathrm{E}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right|^2 + \left|\mathrm{std}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right|^2$, as well as analogous relationships for the other states.

	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Sys 6
$\left\|\mathrm{E}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right\|^2$	4.0009e-05	2.7976e-05	2.8331e-05	2.3807e-05	1.9047e-05	9.0055e-06
$\left\|\mathrm{std}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right\|^2$	4.1718e-05	4.3779e-07	3.9763e-07	1.5584e-07	9.8024e-08	1.4015e-07
$\left\|\mathrm{E}\left[\mathrm{x}_2(\xi_1,\mathrm{t})\right]\right\|^2$	2.7518e-06	7.3209e-07	7.1696e-07	7.7922e-07	9.3295e-07	1.6015e-06
$\left\|\mathrm{std}\left[\mathrm{x}_2(\xi_1,\mathrm{t})\right]\right\|^2$	1.2986e-06	1.9969e-08	1.7928e-08	8.0462e-09	6.3273e-09	1.9956e-08
$\left\|\mathrm{E}\left[\dot{\mathrm{x}}_1(\xi_1,\mathrm{t})\right]\right\|^2$	1.3355e-02	1.2749e-02	1.2811e-02	1.0479e-02	8.1119e-03	3.9569e-03
$\left\|\mathrm{std}\left[\dot{\mathrm{x}}_1(\xi_1,\mathrm{t})\right]\right\|^2$	1.3367e-02	2.4369e-04	2.1878e-04	9.0937e-05	4.8316e-05	4.2732e-05
$\left\|\mathrm{E}\left[\dot{\mathrm{x}}_2(\xi_1,\mathrm{t})\right]\right\|^2$	1.7279e-03	8.8176e-04	8.4857e-04	8.6982e-04	9.6725e-04	2.1226e-03
$\left\|\mathrm{std}\left[\dot{\mathrm{x}}_2(\xi_1,\mathrm{t})\right]\right\|^2$	4.1823e-04	1.5887e-05	1.4172e-05	1.0445e-05	1.0225e-05	7.2566e-05
$\left\|\mathrm{E}\left[\mathrm{u}(\xi_1,\mathrm{t})\right]\right\|^2$	0	1.3684e+00	1.3852e+00	1.3394e+00	1.2900e+00	1.0302e+00
$\left\|\mathrm{std}\left[\mathrm{u}(\xi_1,\mathrm{t})\right]\right\|^2$	0	1.1395e-01	1.0630e-01	8.2161e-02	7.1803e-02	7.6263e-02

As can be seen in this table, the L_2 norm for the variance of x_2 does in fact significantly decrease for Systems 4 and 5 compared to System 2, the nominal $\mathcal{H}_2$ controller. Additionally, this trend does not hold for System 6, the case with the maximum weighted variance for the state. Further, by comparing the experimental values in Table 6.12 with the theoretical values in Table 6.10, it is found that the norms for the square mean, representing a nominal trajectory, match well for an experimental system. However, it turns out that the variance terms are larger than expected.

Hence, it is important to look at the inherent run to run variance. To do this, ten tests, all with 100 lb (45.359 kg) added, were run. The L_2 norm for the variance trajectory is given in Table 6.13. As can be seen, the inherent run to run variance for each state and case can be considered small (by at least a factor of ten, usually a significantly larger factor) in comparison to the effects created through changing the mass.

Finally, since the idea of $\mathcal{H}_2$ control is to seek an optimal over many different norms, it was valuable to combine these L_2 norms into the $\mathcal{H}_2$ norm for all of the established cost functions. These norms are given in Table 6.14.

Table 6.13: Calculated L_2 norms for run to run experimental variation tests, calculated using trapezoidal integration. Note: $\mathrm{E}\left[|\mathrm{x}_1(\xi_1,\mathrm{t})|^2\right] = \left|\mathrm{E}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right|^2 + \left|\mathrm{std}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right|^2$, as well as analogous relationships for the other states.

	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Sys 6
$\left\|\mathrm{std}\left[\mathrm{x}_1(\xi_1,\mathrm{t})\right]\right\|^2$	7.7215e-09	1.5099e-09	1.4100e-09	4.4678e-09	3.2592e-08	6.7897e-08
$\left\|\mathrm{std}\left[\mathrm{x}_2(\xi_1,\mathrm{t})\right]\right\|^2$	6.7414e-10	3.1218e-10	2.8223e-10	9.0063e-10	2.7144e-09	3.8508e-09
$\left\|\mathrm{std}\left[\dot{\mathrm{x}}_1(\xi_1,\mathrm{t})\right]\right\|^2$	2.6256e-06	2.1021e-06	2.3815e-06	3.0162e-06	3.4846e-06	1.9871e-06
$\left\|\mathrm{std}\left[\dot{\mathrm{x}}_2(\xi_1,\mathrm{t})\right]\right\|^2$	3.4123e-07	3.3501e-07	3.1756e-07	2.7032e-07	2.7773e-07	2.2941e-06
$\left\|\mathrm{std}\left[\mathrm{u}(\xi_1,\mathrm{t})\right]\right\|^2$	0	1.3416e-03	1.4881e-03	2.5628e-03	4.9231e-03	5.9637e-03

Table 6.14: Calculated $\mathcal{H}_2^2$, induced solely by excitation into filter, using trapezoidal integration. The table shows the norm for each cost function for the open loop system and each closed loop system. A box is placed around the minimum norms for each cost function—corresponding to the closed loop system with lowest norms. These boxes include all values within 1% of the minimum.

	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Sys 6
J_2	4.4190e-04	1.3079e-04	1.3014e-04	1.3000e-04	1.3957e-04	1.9679e-04
J_3	4.8966e-04	1.3466e-04	1.3344e-04	1.3227e-04	1.4122e-04	1.9404e-04
J_4	2.9571e-03	1.7260e-04	1.6750e-04	1.4755e-04	1.5325e-04	2.3196e-04
J_5	1.3346e-02	3.3235e-04	3.1092e-04	2.1192e-04	2.0386e-04	3.9161e-04
J_6	6.5292e-02	1.1311e-03	1.0280e-03	5.3377e-04	4.5696e-04	1.1898e-03

As can be seen, the trend in $\mathcal{H}_2$ norms is reasonably similar to that expected theoretically. The one main exception is that System 6 did not perform as theoretically expected and hence did not come close to minimizing its cost function.

6.2.7 Discussion

The performance of the controllers designed with the weighted $\mathcal{H}_2$ norms is interesting. In an experimental setting, it is expected there will be deviation from the theoretical norm. This is one reason why experimental tests are run.

Additionally, several trends can be seen throughout this analysis. The first is that while the costs can follow the approximate trends expected by theory, this is not a guarantee when moving to an

experimental system. Hence, it is significant that there is a high degree of correlation between the theoretical intent and the experimental results. Further, to emphasize this point, Case 6 is important—it demonstrates that there is a point where too much can be asked of a model and thus what appears like it should happen on the model may not necessarily occur in reality. Hence, as with all methods, good judgement is important.

6.3 Variable-Valve Damper

A partial state feedback controller was designed using a plain $\mathcal{H}_2$ technique as well as one using the developed Polynomial Chaos $\mathcal{H}_2$ method. As is well known, partial feedback in $\mathcal{H}_2$ can lose some of $\mathcal{H}_2$'s guarantees—even when using observers. As such, several methods, including *Loop Transfer Recovery*, have been designed. However, while a couple of ideas have been presented in regard Polynomial Chaos type observers, they are based on different assumptions (see Section 3.3). This would get outside of the desired scope of investigating the controller since it would be difficult to discern effects of the controller from effects of the observer.

With the utilized optimization method, the same procedure can be used to design a partial state feedback controller—i.e. determine the partial set of gains that minimizes the selected quadratic cost function. As such, this section allows providing an additional experimental example as well as expanding the scope—sometimes it is desired to find controllers with a given structure and thus it is not desired or possible to use full-state feedback. One such example found in the literature is optimally determining gains for the Positive Position Feedback (PPF) method of control (i.e. see [158, 159]).

Additionally, given the close adherence to the optimal control viewpoint in the previous section, this section will take a less formal and more pragmatic approach to control design.

6.3.1 Experimental Setup

The system used in this example is based on the 2-DOF oscillator presented in the previous section. However, instead of using a distribution of masses, a distribution of damping elements between m_1 and m_2 is used. This is implemented with a variable-valve damper. Pictures of the damper and modifications to the test configuration are given in Figure 6.32.

In order to characterize the capabilities of the variable-valve damper, it was placed in a Roehrig dyno (Figure (a)). Using a 5 Hz, 0.2 in (5.08 mm) amplitude sinusoidal displacement input, force vs. velocity curves for different valve positions were obtained. This input was selected since it would create velocities and amplitudes similar to those desired during testing. The damper was only pressurized to 100 psi (690 kPa), causing the spring effect to be negligible compared to the much larger air-springs. This allowed the curve offset to be simply corrected by subtracting a single bias determined by an average force over the small interval of ±5.0 mm/s (0.20 in/s). The normalized curves showing the damping force versus velocity are plotted in Figure 6.30.

Additionally, while the curves do change slope with respect to valve angle at the origin, they obviously are not linear. Thus, in order to use the damper in a linear model, least-square lines were fit to the curves. Since the damping force should be zero when the damper is not moving, the fits were forced to travel through the origin. The resulting lines are overlaid in Figure 6.30. While only a crude model, they do admit a single damping constant. These constants are plotted as a function of the valve angle in Figure 6.31.

The approximate damping coefficient $d(\theta)$, as a function of the valve angle (in degrees), can be given as:

$$d(\theta) = 237.72e^{0.00112\theta^2+0.0617\theta}. \tag{6.20}$$

For $\theta \in [5, 20]$ degrees, the range of the approximate coefficient is $d(\theta) \in [333, 1268]$ N-s/m.

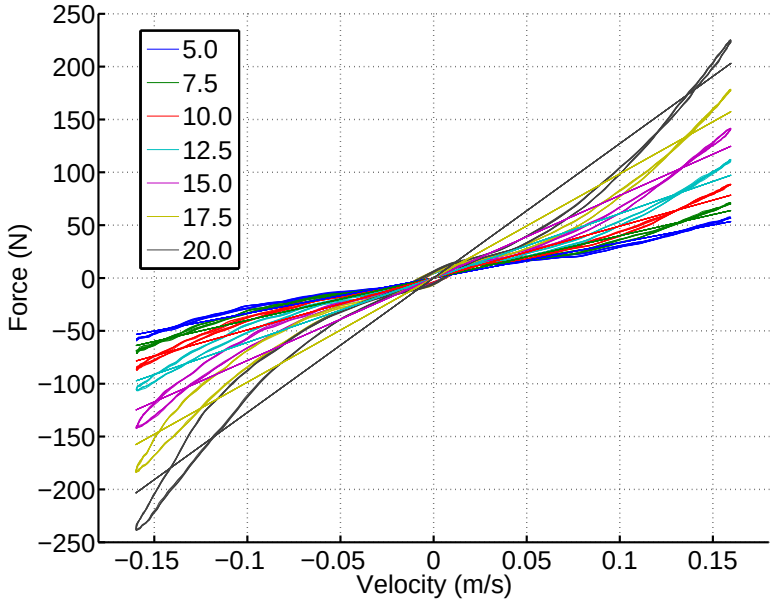


Figure 6.30: Force vs. velocity curves for the variable-valve damper. The DC-offset of 91.4 N (20.5 lb), caused by gas pressure, has been removed. The multiple curves are created by opening the valve to various angles (in degrees). Additionally, the linear least-square fits, which are required to pass through the origin, are given.

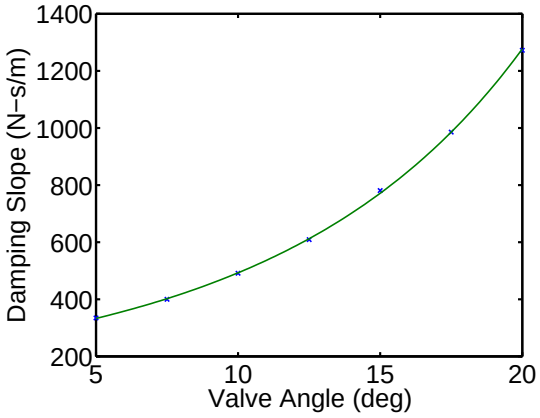
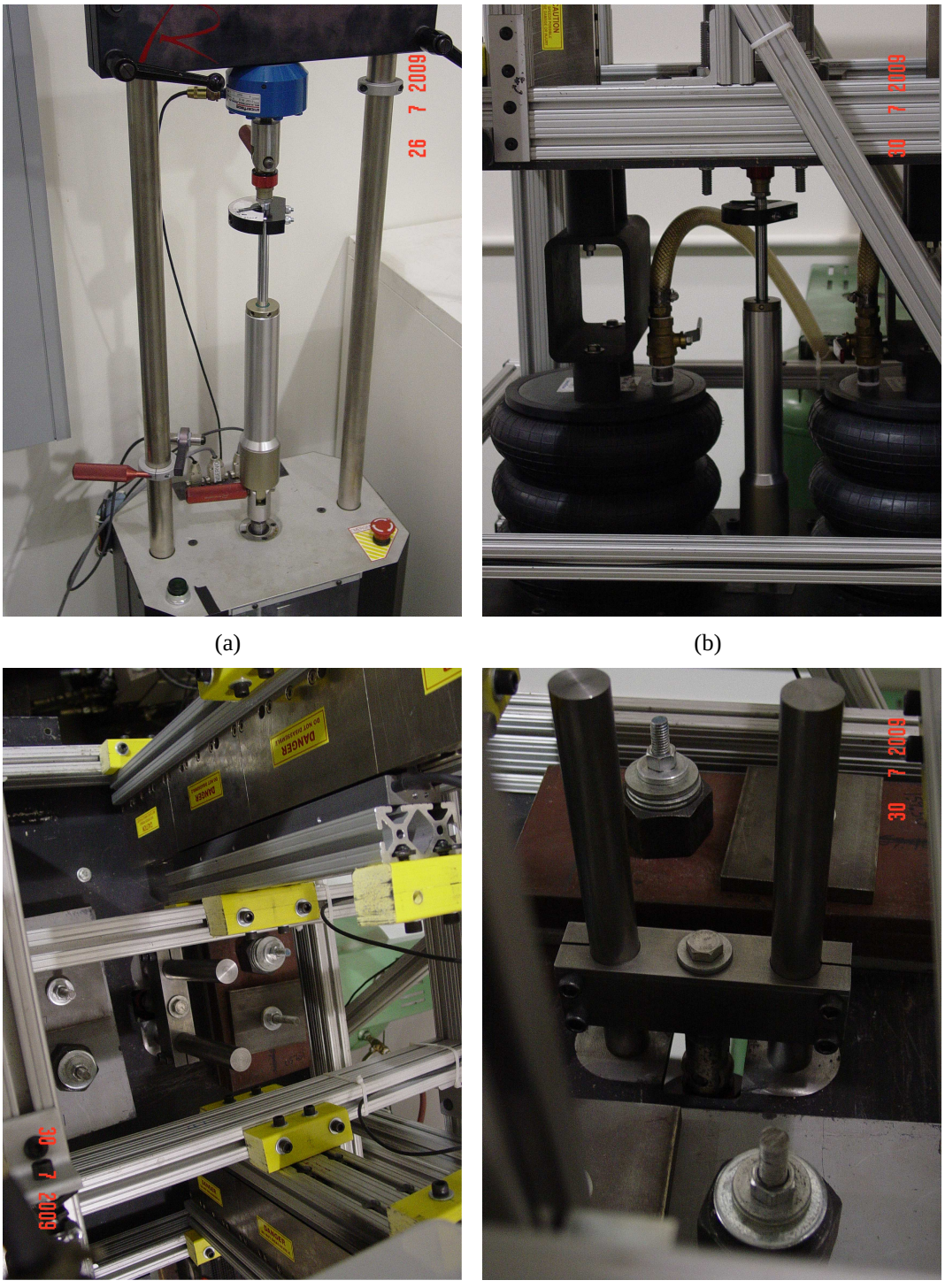


Figure 6.31: Approximate damping coefficient vs. valve angle



(a) (b)

(c) (d)

Figure 6.32: Variable-valve damping experiment pictures: (a) damper characterization in a Roehrig shock dyno, (b) damper mounted on 2-DOF oscillator, (c) mass fixed to m_1, and (d) view of damper fixture.

6.3.2 Model

The model is the same as the one presented at the beginning of Section 6.2, except the mass is now fixed such that $m_1 = 174.19$ kg and only d_1 is considered to be a function of the parameter space ξ_1. To equally weight all damping values (different than equally weighting all angles), the damping coefficient will be allowed to reside in $d_1(\xi_1) \sim u[570, 1505]$ N-s/m. This range assumes the damping from the damper and that determined in the system identification for the oscillator are additive. In terms of a PCE, the damping is expressed as $d_1(\xi_1) = 1038\phi^0(\xi_1) + 468\phi^1(\xi_1)$, with $\xi_1 \sim u[-1, 1]$.

The remaining parameters are considered to take the values given for the second mass case in Table 6.4.

It is necessary to change two matrices in the oscillator model such that

$$\mathbf{A}_{mech}(\xi_1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_1}{m_1} & \frac{k_1}{m_1} & -\frac{d_1(\xi_1)}{m_1} & \frac{d_1(\xi_1)}{m_1} \\ \frac{k_1}{m_2} & -\frac{k_1+k_2}{m_2} & \frac{d_1(\xi_1)}{m_2} & -\frac{d_1(\xi_1)+d_2}{m_2} \end{bmatrix} \tag{6.21}$$

and

$$\mathbf{B}_{u,mech}(\xi_1) = \begin{bmatrix} 0 \\ 0 \\ \frac{F_k}{m_1} \\ 0 \end{bmatrix}. \tag{6.22}$$

The filter equations are still used–with the single modification that $w_n = 5.8\pi$. However, the filter states are not used in feedback and thus it can be thought of as a form of *frequency weighting*. This heavily weights the excitation of m_1's resonance frequency and down-weights m_2's resonance

frequency. This is important for the design and experiment since there is a very low transmissibility of actuator authority at m_2's resonance frequency.

Additionally, the feedback equation is restricted to the following partial state form:

$$\mathrm{u}(t,\xi_1) = \mathbf{K}\begin{bmatrix} \mathrm{x}_1(t,\xi_1) & \mathrm{x}_2(t,\xi_1) \end{bmatrix}^T. \tag{6.23}$$

This is the form for a simple proportional compensator.

6.3.3 Design

Within the scope of the fixed proportional compensator, it was desired to select gains to try to keep both masses still. To begin, it is instructive to look at the simulated and experimental open loop responses plotted in Figures 6.33 to 6.38, Subfigures (a). The theoretical and simulated responses do in fact match reasonably well.

The problem of reducing the magnitude of the states can be thought of in terms of the deterministic $\mathcal{H}_2$ weighting matrices:

$$\begin{aligned} \mathbf{Q} &= \mathrm{diag}\begin{bmatrix} 1 & 1 \end{bmatrix} \\ \mathbf{R} &= [1e-9]\,. \end{aligned} \tag{6.24}$$

The resulting set of gains that minimize the $\mathcal{H}_2$ norm is $\mathbf{K}_1 = \begin{bmatrix} -1141 & -435.7 \end{bmatrix}$. These results were obtained using the nominal damping coefficient ($\xi = 0$). The gains were found from the global minimum. Additionally, it is interesting to note that more than one local minimum was located in this partial state feedback problem. However, the problem is simple enough to ensure that the global minimum was found.

In order to test this controller, a grid based approach is again employed to sampling the systems with different damping. The five grid points used are given in Table 6.15.

Table 6.15: The approximate damping coefficients for the used grid points and the corresponding valve angles.

ξ_1	-0.8	-0.4	0	0.4	0.8
Damping, $d_1(\xi_1)$ N-s/m	664	851	1040	1230	1410
Valve Angle, $\theta(\xi_1)$ deg	8.3	12.6	15.4	17.6	19.3

Simulated filtered impulse responses, using the design filter can be seen in Figures 6.33 to 6.35, Subfigures (b). Hence, in simulation, this controller appears to be a feasible option. The controller was then tested on the experimental platform. As can be seen in Figures 6.36 to 6.38, Subfigures (b), these closed loop systems went unstable. The cause of this problem was determined to be that the design method was overestimating the stabilizing capabilities of the damping, when in fact the damping falls below the nominal due to the range of damping as well as at low amplitudes due to non-linearity.

The expected $\mathcal{H}_2$ controller produces a very similar set of gains in this case. Thus, one way to try to solve the instability is to try to force all of the systems to act similarly. This possible concept for fixing the problem is to desensitizing the system, over the required bounds, with respect to the uncertainty of the uncertain parameter. Since the control command is a linear combination of both utilized states, it makes sense to attempt to accomplish this by more heavily weighting the variance of the control command signal. The weighting matrices become

$$\begin{aligned} \mathbf{Q}_M &= \text{diag}\begin{bmatrix} 1 & 1 \end{bmatrix}, \ \mathbf{R}_M = [1e-9] \\ \mathbf{Q}_C &= \text{diag}\begin{bmatrix} 1 & 1 \end{bmatrix}, \ \mathbf{R}_C = [1e-4]\,. \end{aligned} \tag{6.25}$$

The resulting set of gains that minimize the new $\mathcal{H}_2$ norm, using the Polynomial Chaos method, is $\mathbf{K}_2 = \begin{bmatrix} -488.1 & -312.0 \end{bmatrix}$.

As can be seen in Figures 6.33 to 6.35, Subfigures (c), this still should perform a very reasonable job of helping to suppress the motion of the masses when compared to the open loop response. When this system was implemented on the test platform, the results were not exactly as expected, but the system maintained stability and did in fact perform a reasonable job suppressing the motion. This can be seen in Figures 6.36 to 6.38, Subfigures (c).

Alternately, it is possible to attempt to stay with standard $\mathcal{H}_2$ design methods. In this case, the realization that the damping was being overestimated allowed the approach of attempting to design the controller through a system using an estimated lower bound on the damping. As in anything experimental that can have non-modeled phenomena, this approach is not guaranteed to work, but seemed reasonable to try. The same cost function in Equation (6.3.3) was used. However, the the damping coefficient in question was set to $d_1 = 301.9$, a value viewed as the lowest possible damping since it was the determined damping without the added damping element. This approach produced the gains $\mathbf{K}_3 = \begin{bmatrix} 244.2 & -282.4 \end{bmatrix}$. As can be seen by the sign change in the first element of the gain matrix, this is a substantially different control strategy than found with the Polynomial Chaos attempt. However, as can be seen in Figures 6.33 to 6.38, Subfigures (d), theoretically it should be stable over the test cases, and in fact it turned out to be when implemented on the experimental oscillator.

That said, the approach resulted in an inferior controller compared to the Polynomial Chaos method. Part of the reason for this is that the PC method can consider the whole predicted set of parameter values—as opposed to trying to guess one that is representative of the system. As such, when forcing the whole set to work, this appears to desensitize the system to uncertainty about the parameter in the vicinity of the defined bounds for the parameter. Additionally, the controller suffered from using a positive element in the gain matrix. Ideally, this should not be a problem. In practice, however, the positive feedback can combine with static friction effects to hinder the return of the steady state value to zero.

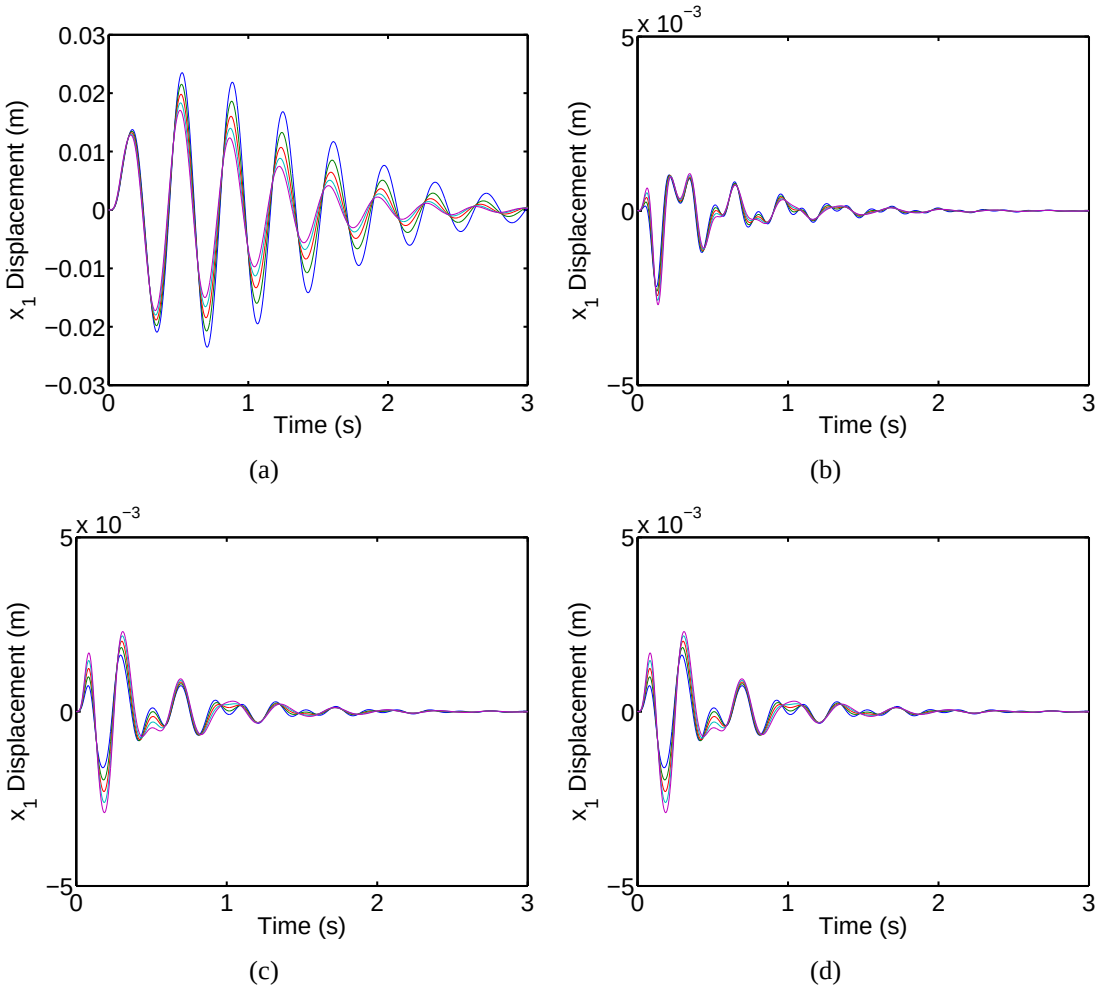


Figure 6.33: Theoretical responses for x_1. (a) Open Loop, (b) Nominal $\mathcal{H}_2$, (c) Weighted Variance for Current Command, (d) $\mathcal{H}_2$ using Lower Damping Constant. Legend: — $\xi_1 = -0.8$, — $\xi_1 = -0.4$, — $\xi_1 = 0$, — $\xi_1 = 0.4$, — $\xi_1 = 0.8$

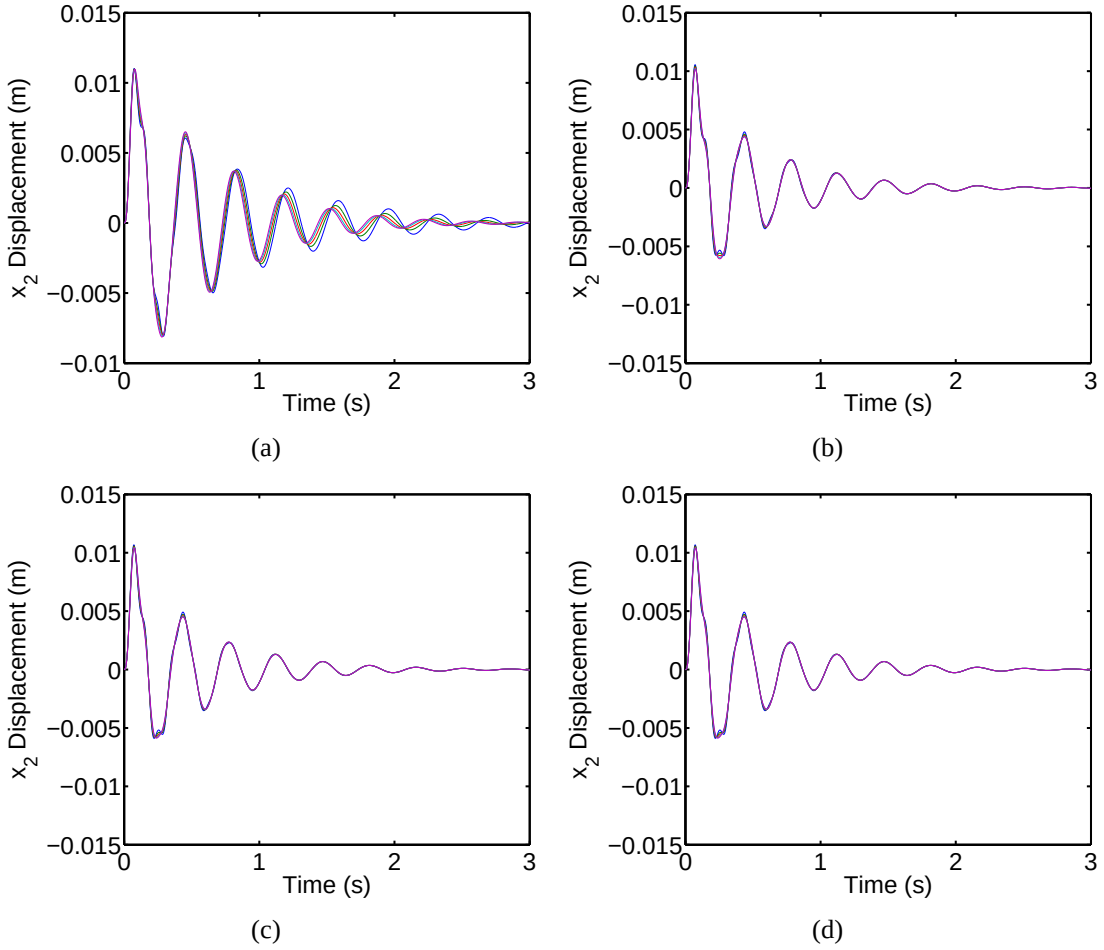


Figure 6.34: Theoretical responses for x_2. (a) Open Loop, (b) Nominal $\mathcal{H}_2$, (c) Weighted Variance for Current Command, (d) $\mathcal{H}_2$ using Lower Damping Constant. Legend: — $\xi_1 = -0.8$, — $\xi_1 = -0.4$, — $\xi_1 = 0$, — $\xi_1 = 0.4$, — $\xi_1 = 0.8$

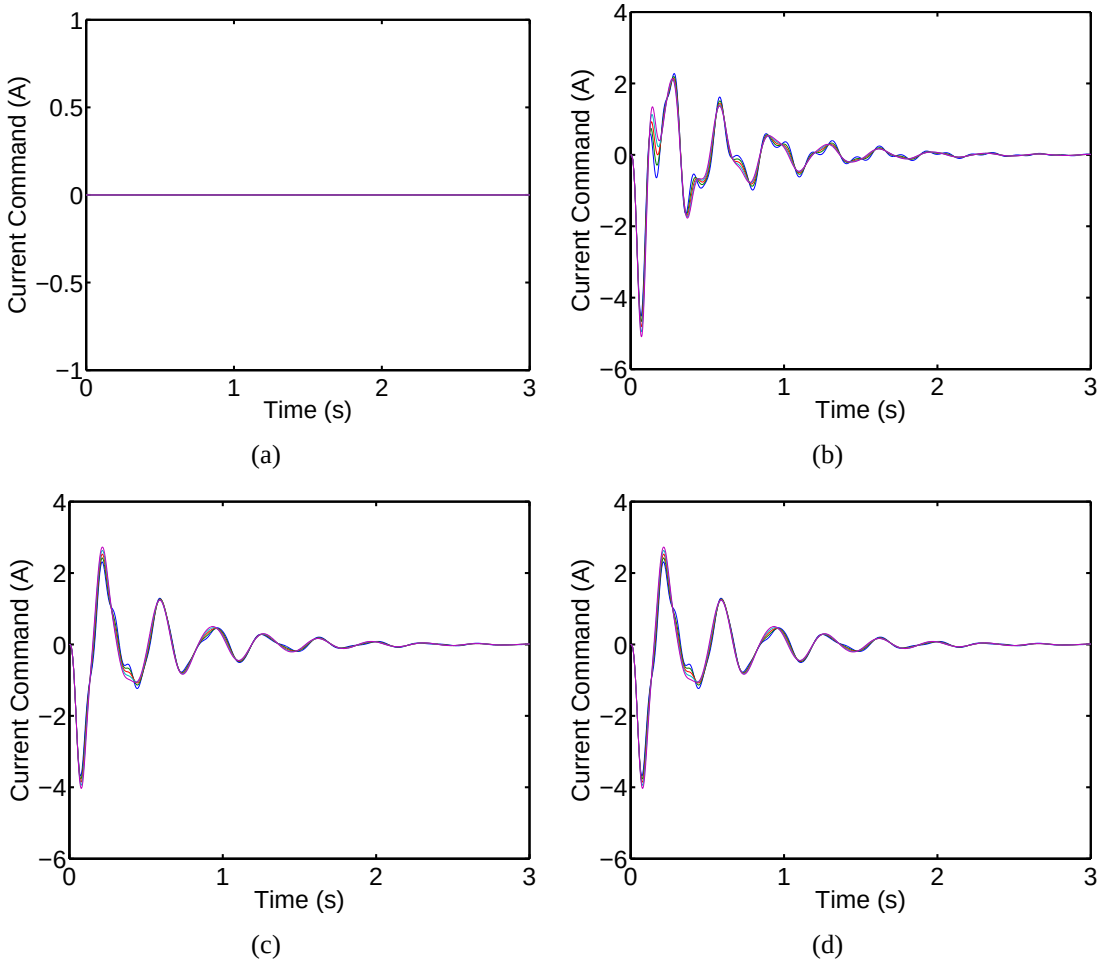


Figure 6.35: Theoretical responses for the current command, I. (a) Open Loop, (b) Nominal $\mathcal{H}_2$, (c) Weighted Variance for Current Command, (d) $\mathcal{H}_2$ using Lower Damping Constant. Legend: — $\xi_1 = -0.8$, — $\xi_1 = -0.4$, — $\xi_1 = 0$, — $\xi_1 = 0.4$, — $\xi_1 = 0.8$

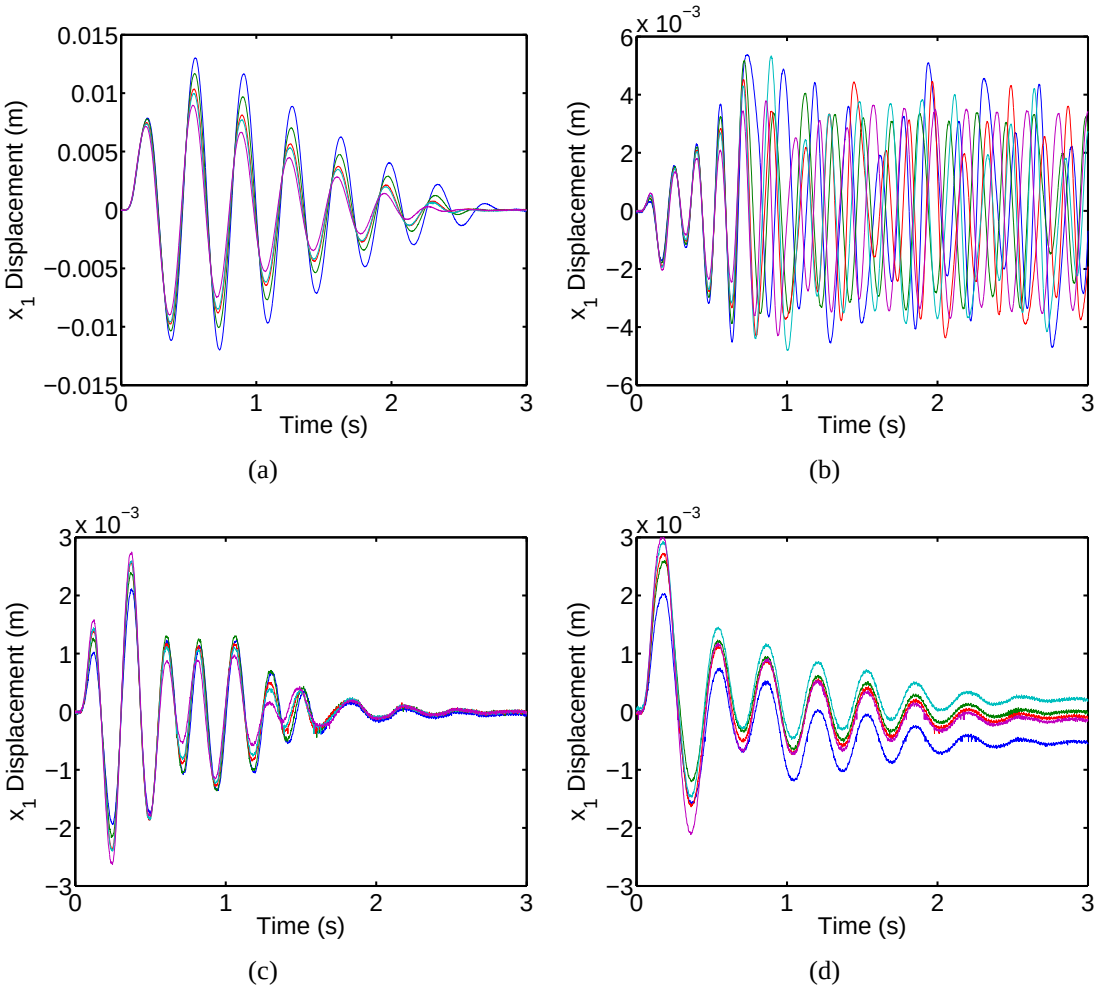


Figure 6.36: Experimental responses for x_1. (a) Open Loop, (b) Nominal $\mathcal{H}_2$, (c) Weighted Variance for Current Command, (d) $\mathcal{H}_2$ using Lower Damping Constant. Legend: — $\xi_1 = -0.8$, — $\xi_1 = -0.4$, — $\xi_1 = 0$, — $\xi_1 = 0.4$, — $\xi_1 = 0.8$

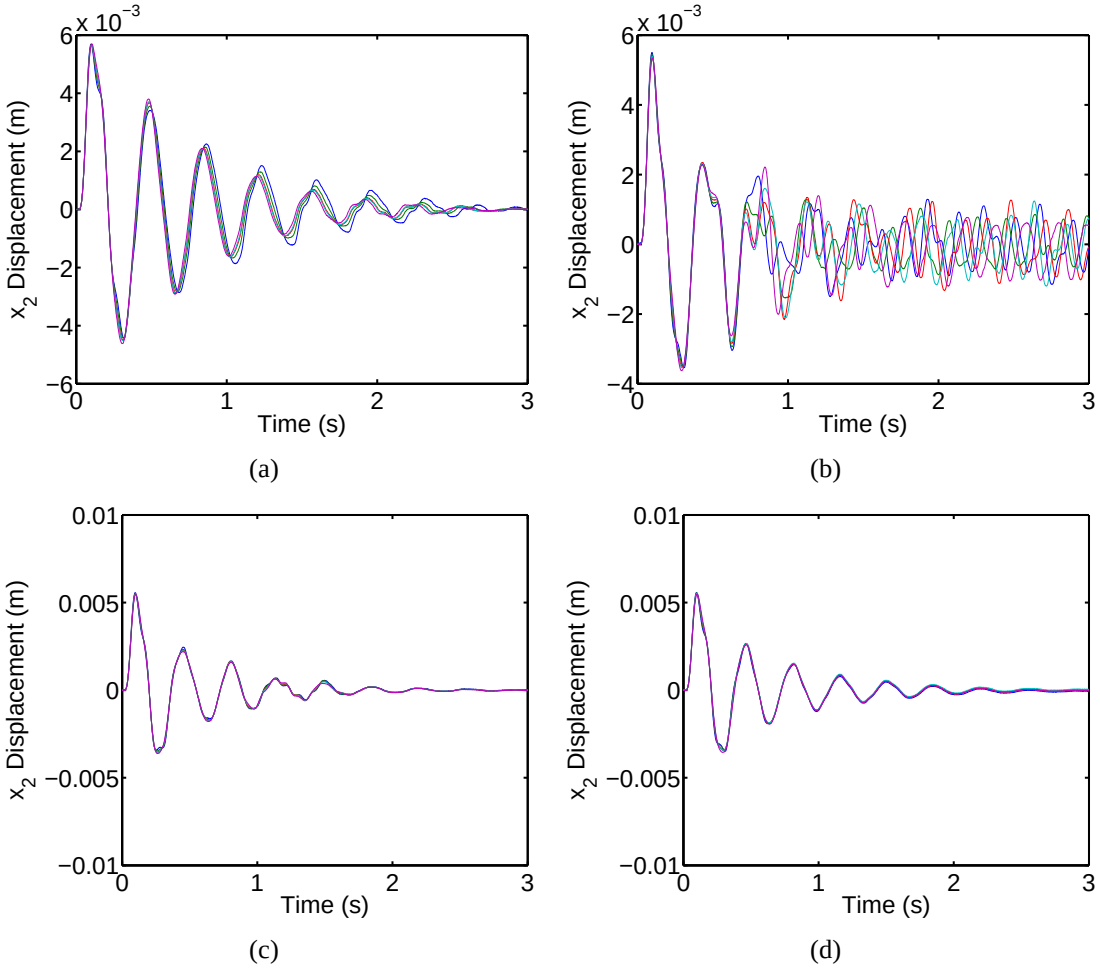


Figure 6.37: Experimental responses for x_2. (a) Open Loop, (b) Nominal $\mathcal{H}_2$, (c) Weighted Variance for Current Command, (d) $\mathcal{H}_2$ using Lower Damping Constant. Legend: — $\xi_1 = -0.8$, — $\xi_1 = -0.4$, — $\xi_1 = 0$, — $\xi_1 = 0.4$, — $\xi_1 = 0.8$

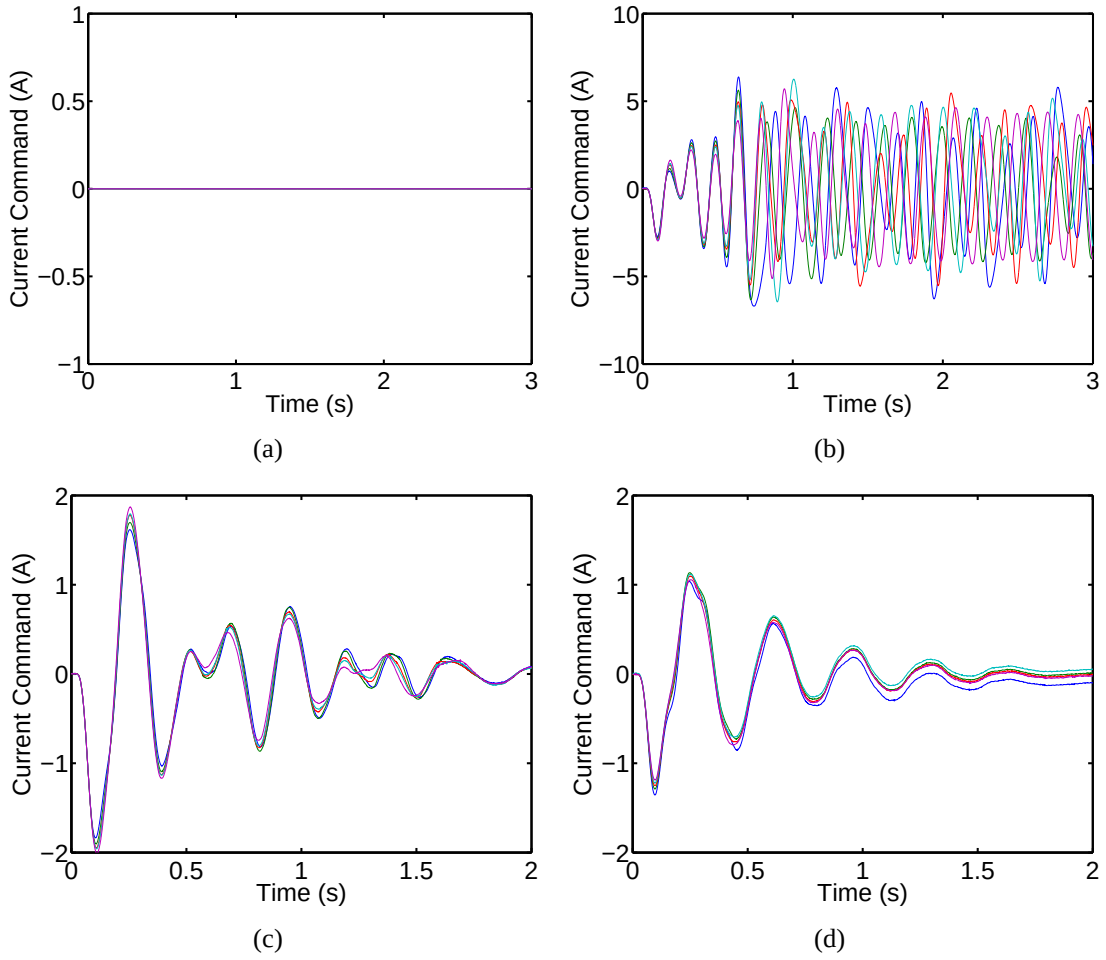


Figure 6.38: Experimental responses for I. (a) Open Loop, (b) Nominal $\mathcal{H}_2$, (c) Weighted Variance for Current Command, (d) $\mathcal{H}_2$ using Lower Damping Constant. Legend: — $\xi_1 = -0.8$, — $\xi_1 = -0.4$, — $\xi_1 = 0$, — $\xi_1 = 0.4$, — $\xi_1 = 0.8$

6.3.4 Discussion

This section brings up a point—that the Polynomial Chaos $\mathcal{H}_2$ can have value when being applied in situations where it is suboptimal. It is relatively easily used and as such it can be a flexible and valuable expansion to the standard $\mathcal{H}_2$ design process. However, as is with the case of using any control design method on models that are approximations of the system, not only is optimality not guaranteed, it is possible to produce bad controllers. While it would be possible to simply try to tune the gains on a system this simple, for more complicated systems, tools like the "tuning knobs" offered by $\mathcal{H}_2$ and the presented expansion become increasingly important to the control designer. Hence, the fact that the "tuning knobs" presented by the Polynomial Chaos extension add additional conceptually meaningful ways to define cost functions is valuable in and of itself.

Chapter 7

Conclusion

A novel method of designing a controller for a system modeled with parametric uncertainties was derived. Polynomial Chaos was introduced and then used to find a spectral approximation of the "flow" of dynamic state solutions resulting from parametric uncertainty in a state space model. For a simple probabilistic normally distributed initial condition problem, the method was compared to Monte Carlo simulation, the analytical solution, and Louisville's theorem.

For more complicated problems, exact solutions are typically difficult to find. Thus, Polynomial Chaos using Galerkin projections and finite length expansions becomes interesting. This was justified by first showing the infinite expansion converges in the appropriate weighted L_2 space. Additionally, the mean and variance were presented in a manner such that they are directly related to the L_2 norm. These were then expanded to the notions of square mean, variance, and mean square trajectories. Further, to help justify truncating the expansion to a finite length, numerical error analysis for a problem with an exact solution was given. This analysis showed an approximately exponential rate of decrease in error as the order of the expansion was increased. Additionally,

further error analysis was performed to support the use of Lyapunov equations for solving $\mathcal{H}_2$ problems with separated variance and mean square trajectories.

Building on this analysis, a control design problem was presented as a probabilistic extension of an $\mathcal{H}_2$ control design problem. It was shown that an expanded quadratic cost function could be constructed such that statistical aspects of the uncertain states were weighted. These statistics involved weighting terms relating to the means of the states and covariances of the states. Next, theoretical bounds on the H_2 norm were proposed using LMIs. Following, the $\mathcal{H}_2$ problem was treated as a minimization problem in order to converge upon optimal and robust gains. Additionally, it was noted that the presented work could be applied to a probabilistic LQR problem. Further, examples relating to uncertain one and two degree-of-freedom oscillators were presented in order to help explore concepts in the chapter.

The next chapter discussed implementation of the control design method on an experimental mechanical two degree-of-freedom oscillator. First a simple test was presented, using a Magneto-Rheologic (MR) damper, to demonstrate the notion of probabilistic systems in the experimental domain.

This was then expanded to a rigorous experimental study. Four controllers designed using the presented theory, a related standard $\mathcal{H}_2$ controller, and the open loop system were compared. This was accomplished through examining a system where a mass in the oscillator was distributed over a large range. Linear electromagnetic motors were used as the control actuators and a hydraulic cylinder for a disturbance input. The controllers were designed in such a manner as to demonstrate that the method worked to reduce the "spread" in behavior between tests, in addition to doing a good job of accomplishing the traditional $\mathcal{H}_2$ design objectives for the set. The theoretical and experimental impulse responses were presented side by side. It could be seen that the central tendency of the trajectories, related to the mean, was very consistent between the experimental and theoretical responses. In the "spread" of the trajectories, the experimental and theoretical results

did not match as well—but the trend of decreasing variance could still be seen. The one exception to this is one of the test cases was designed to be an extreme example. It showed that eventually weighting the variance too much could result in other properties being neglected enough that the controller was a bad controller. These effects were also quantified by looking at associated L_2 and $\mathcal{H}_2$ norms for the theoretical and experimental cases.

Following, a less involved test was presented. This test used a variable-valve damper to allow the introduction of a range of damping. Other than this, the experimental system was set up similarly to the one used in the mass experiment. This included using an electromagnetic actuator for the control input and the hydraulic cylinder for a disturbance. One of the main purposes was to provide contrast with the formal optimal control approach by using the controller in a more casual design procedure.

7.1 Future Work

The method investigated is very general. As such, there are many interesting directions the research could travel in:

I. Linear systems provided a very interesting place to begin the analysis of Polynomial Chaos and control. However, it is expected that parametric uncertainty could play a larger role in non-linear systems. The one caveat is that the method most likely will work best for equations with differentiable solutions. One way to establish stability and appropriateness for the the Galerkin method might be to look at finding Lyapunov functions that apply for the set of systems. Additionally, in general the Lyapunov equation method will not work. As such, simulations will be needed to evaluate time

integrals. The burden for doing this could be reduced through moving towards the collocation method.

II. Further, the collocation method could be used to split up the Lyapunov equations for linear systems for the case of minimizing the expected $\mathcal{H}_2^2$.

III. As alluded to a number of times, there are striking similarities between spectral methods for PDEs and the derived method. It'd be interesting to explore the combination for the sake of controlling PDEs. An example of this might be vibration control of beams and plates. There has been substantial research involving the application of Polynomial Chaos to PDEs—as such this appears to be fertile ground.

IV. An important item to look into is finding more applications. The fact that the method allows extra "tuning knobs," while producing the same dimension controller, makes it a prime candidate for places that $\mathcal{H}_2$ methods are already used. Such examples might include applications that are on a scale that are too small to include powerful microprocessors and there is variability in the manufacturing process. Additionally, this may be useful for designing physical systems that are based on intrinsic mechanical feedback. Also, another application might be found in relation to general experimental set-ups: many times one of the most important attributes for them is repeatability.

V. Further investigation into trajectory tracking problems would be useful. Frequently the exact trajectory is not important. Hence making the trajectories more repeatable in the face of uncertainty could provide an interesting improvement.

Works Cited

[1] W. Gautschi. *Orthogonal polynomials: computation and approximation*. Oxford University Press Inc., New York, 2004.

[2] D. Xiu and G. E. Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Society for Industrial and Applied Mathematics*, 24(2):619–644, 2002.

[3] J. Fisher and R. Bhattacharya. On stochastic LQR design and polynomial chaos. In *American Control Conference, 2008*, pages 95–100, Seattle, WA, June 2008.

[4] J. Fisher and R. Bhattacharya. Stability analysis of stochastic systems using polynomial chaos. In *American Control Conference, 2008*, pages 4250–4255, Seattle, WA, June 2008.

[5] S. M. Ross. *A first course In probability*. Prentice Hall, Upper Saddle River, NJ, 6th edition, 2006.

[6] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust statistics: the approach based on influence functions*. John Wiley & Sons, New York, 1986.

[7] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications, Inc., Mineola, NY, 2nd edition, 2000.

[8] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral methods: evolution to complex geometries and applications to fluid dynamic*. Springer, New York, 2007.

[9] D. J. Scheeres, F.-Y. Hsiao, R. S. Park, B. F. Villac, and J. M. Maruskin. Fundamental limits on spacecraft orbit uncertainty and distribution propagation. *Journal of the Astronautical Sciences*, 54(3–4):505–523, 2006.

[10] J. V. José and E. J. Saletan. *Classical dynamics*. Cambridge University Press, New York, 6th edition, 2006.

[11] D. A. Wagie and R. E. Skelton. Model reduction and controller synthesis in the presence of parameter uncertainty. *Automatica*, 22:295–308, 1986.

[12] R. K. Yedavalli and R. E. Skelton. Controller design for parameter sensitivity reduction in linear regulators. *Optimal Control Applications and Methods*, 3(3):221–240, 1982.

[13] K. Okada and R. E. Skelton. Sensitivity controller for uncertain systems. *Journal of Guidance, Control, and Dynamics*, 13(2):321–329, 1990.

[14] S. C. O. Grocott. Comparison of control techniques for robust performance on uncertain structural systems. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1994.

[15] B. R. Barmish and H. I. Kang. A survey of extreme point results for robustness of control systems. *Automatica*, 29(1):13–15, 1993.

[16] J. Ackermann, D. Kaesbauer, and R. Muench. Robust, gamma-stability analysis in a plant parameter space. *Automatica*, 27(1):75–85, 1991.

[17] C. T. Tse, K. W. Wang, C. Y. Chung, and K. M. Tsang. Parameter optimisation of robust power system stabilisers by probabilistic approach. *IEE proceedings generation, transmission and distribution*, 147(2):69–74, 2000.

[18] L. R. Ray and R. F. Stengel. A Monte Carlo approach to the analysis of control system robustness. *Automatica*, 29(1):229–236, 1993.

[19] R. Huerta-Ochoa and S. H. Johnson. Control system design for structural systems under statistical parametric uncertainty. In *SPIE Conference on Mathematics and Control in Smart Structures*, pages 219–230, Newport Beach, California, March 1999.

[20] L. G. Crespo and S. P. Kenny. Reliability-based control design for uncertain systems. *Journal of Guidance, Control, and Dynamics*, 28(4):649–658, 2005.

[21] B. T. Polyak and R. Tempo. Probabilistic robust design with linear quadratic regulators. In *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 1037–1042, Sydney, Australia, December 2000.

[22] S. Kanev, B. D. Schutter, and M. Verhaegen. An ellipsoid algorithm for probabilistic robust controller design. *Systems & Control Letters*, 49:365–375, 2003.

[23] P. Djavdan, H. J. A. F. Tulleken, M. H. Voetter, H. B. Verbruggen, and G. J. Olsder. Probabilistic robust controller design. In *Proceedings of the 27th Conference on Decision and Control*, pages 2164–2172, Tampa, FL, December 1989.

[24] M. Sternad and A. Ahlén. Robust filtering and feedforward control based on probabilistic descriptions of model errors. *Automatica*, 29(3):661–679, 1993.

[25] K. Öhrn, A. Ahlén, and M. Sternad. A probabilistic approach to multivariable robust filtering and open-loop control. *IEEE Transactions on Automatic Control*, 40(3):405–418, 1995.

[26] D. W. Apley. A cautious minimum variance controller with ARIMA disturbances. *IIE Transactions*, 36:417–432, May 2003.

[27] Y. Boers, S. Weiland, and A. A. H. Damen. Average H_2 performance design by randomized algorithms. In *Proceedings of the 37th IEEE Conference on Decision & Control*, pages 3347–3348, Tampa, FL, December 1998.

[28] K. Tsumura. Unification of modeling, estimation and controller design. In *Proceedings of 40th IEEE*, Orlando, FL, December 2001.

[29] D. Tenne and T. Singh. Robust feed-forward/feedback design for tape transport. In *AIAA Guidance, Naviagation, and Control Conference and Exhibit*, pages 1–6, Providence, RI, August 2004.

[30] D. Tenne and T. Singh. Efficient minimax control design for prescribed parameter uncertainty. *Journal of Guidance, Control and Dynamics*, 27(6):1009–1016, 2004.

[31] C. I. Marrison and R. F. Stengel. The use of random search and genetic algorithms to optimize stochastic robustness functions. In *Proceedings of the American Control Conference*, pages 1484–1489, Baltimore, Maryland, June 1994.

[32] R. F. Stengel and Q. Wang. Searching for robust minimal-order comensators. In *Proceedings of the American Control Conference*, pages 3138–3142, Philadelphia, PA, June 1998.

[33] L. G. Crespo. Probabilistic formulations to robust optimal control. In *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, pages 1–21, Palm Springs, CA, April 2004.

[34] L. G. Crespo and S. P. Kenny. Robust control design for systems with probabilistic uncertainty. Technical Report TP-2005-213531, NASA, 2005.

[35] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[36] R. H. Cameron and W. T. Martin. The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *Annals of Mathematics*, 48(2):385–392, 1947.

[37] R. Ghanem and P. D. Spanos. Polynomial chaos in stochastic finite elements. *Journal of Applied Mechanics*, 57:197–202, 1990.

[38] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: a spectral approach*. Springer-Verlag, New York, 1991.

[39] A. Sandu, C. Sandu, B. J. Chan, and M. Ahmadian. Control mechanical systems using a parameterized spectral decomposition approach. In *Proceedings of the IMECE'04: ASME International Sixth Annual Symposium on Advanced Vehicle Technologies*, Anaheim, CA, November 2004.

[40] A. Monti, F. Ponci, and T. Lovett. A polynomial chaos theory approach to the control design of a power converter. In *Power Electronics Specialists Conference. PESC 04. IEEE 35th Annual*, volume 6, pages 4809–4813, Aachen, Germany, June 2004.

[41] A. Smith, A. Monti, and F. Ponci. Robust controller using polynomial chaos theory. In *Industry Applications Conference, 41st IAS Annual Meeting, Conference Record of the 2006 IEEE*, volume 5, pages 2511–2517, Tampa, FL, October 2006.

[42] A. H. C. Smith, A. Monti, and F. Ponci. Indirect measurements via a polynomial chaos observer. *Instrumentation and Measurement, IEEE Transactions on*, 56(3):743–752, June 2007.

[43] P. Sarma, L. J. Durlofsky, and K. Aziz. Efficient closed-loop production optimization under uncertainty. In *SPE Europec/EAGE Annual Conference*, Madrid, Spain, June 2005.

[44] F. S. Hover and M. S. Triantafyllou. Application of polynomial chaos in stability and control. *Automatica*, 42(5):789–795, May 2006.

[45] F. S. Hover. Gradient dynamic optimization with Legendre chaos. *Automatica*, 44(1):135–140, January 2008.

[46] R. Ghanem. The nonlinear Gaussian spectrum of log-normal stochastic processes and variables. *Journal of Applied Mechanics*, 66:964–973, 1999.

[47] S. F. Wojtkiewicz, M. S. Eldred, Jr. Filed, R. V., A. Urbina, and J. R. Red-Horse. Uncertainty quantification in large computational engineering models. In *42nd AIAA/ASME/AHS/ASC Structure, Structural Dynamics, and Materials Conference*, Seattle, WA, April 2001.

[48] B. J. Debusschere, H. N. Najm, P. P. Pébay, O. M. Knio, R. G. Ghanem, and O. P. Le Maître. Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM J. Sci. Comput.*, 26(2):698–719, 2004.

[49] R. V. Field and M. Grigoriu. On the accuracy of the polynomial chaos approximation. *Probabilistic Engineering Mechanics*, 19(1–2):65–80, January 2004.

[50] R. Ghanem and J. Red-Horse. Orthogonal representations of stochastic processes and their propagation in mechanics. In *Decision and Control. CDC. 43rd IEEE Conference on*, volume 2, pages 1784–1787, Nassau, Bahamas, December 2004.

[51] A. A. Giunta, M. S. Eldred, L. P. Swiler, T. G. Trucano, and Jr. Wojtkiewicz, S. F. Perspectives on optimization under uncertainty: Algorithms and applications. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, August–September 2004.

[52] L. Huyse and B. H. Thacker. A framework to estimate uncertain random variables. In *45th AIAA/ASME/AHS/ASC Structure, Structural Dynamics, and Materials Conference*, Palm Springs, CA, April 2004.

[53] S. S. Isukapalli, S. Balakrishnan, and P. G. Georgopoulos. Computationally efficient uncertainty propagation and reduction using the stochastic response surface method. In *Decision and Control, CDC. 43rd IEEE Conference on*, volume 2, pages 2237–2243, Paradise Island, Bahamas, December 2004.

[54] O. P. Le Maître, O. M. Knio, H. N. Najm, and R. G. Ghanem. Uncertainty propagation using Wiener-Haar expansions. *Journal of Computational Physics*, 197(1):28–57, June 2004.

[55] D. R. Millman, P. I. King, R. C. Maple, and P. S. Beran. Prediction uncertainty propagation in a highly nonlinear system with a stochastic projection method. In *45th AIAA/ASME/AHS/ASC Structure, Structural Dynamics, and Materials Conference*, Palm Springs, CA, April 2004.

[56] C. Sandu, A. Sandu, B. J. Chan, and M. Ahmadian. Treating uncertainties in multibody dynamic systems using a polynomial chaos spectral decomposition. In *Proceedings of the IMECE'04: ASME International Sixth Annual Symposium on Advanced Vehicle Technologies*, Anaheim, CA, November 2004.

[57] B. Wu, J. Zhu, and F. N. Najm. Dynamic range estimation for nonlinear systems. In *Computer Aided Design. ICCAD-2004. IEEE/ACM International Conference on*, pages 660–667, San Jose, CA, November 2004.

[58] B. Wu, J. Zhu, and F. N. Najm. A non-parametric approach for dynamic range estimation of nonlinear systems. In *Design Automation Conference. Proceedings. 42nd*, pages 841–844, June 2005.

[59] B. Wu, J. Zhu, and F. N. Najm. Dynamic-range estimation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(9):1618–1636, September 2006.

[60] J. Yang, D. Zhang, and Z. Lu. Stochastic analysis of saturated-unsaturated flow in heterogeneous media by combining Karhunen-Loeve expansion and perturbation method. *Journal of Hydrology*, 294(1–3):18–38, July 2004.

[61] S. E. Geneser, S. Choe, R. M. Kirby, and R. S. MacLeod. The influence of stochastic organ conductivity in 2D ECG forward modeling: A stochastic finite element study. In *Engineering in Medicine and Biology Society. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 5528–5531, Shanghai, China, April 2005.

[62] S. E. Geneser, R. M. Kirby, and F. B. Sachse. Sensitivity analysis of cardiac electrophysiological models using polynomial chaos. In *Engineering in Medicine and Biology Society. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 4042–4045, Shanghai, China, September 2005.

[63] R. Ghanem, S. Masri, M. Pellissetti, and R. Wolfe. Identification and prediction of stochastic dynamical systems in a polynomial chaos basis. *Computer Methods in Applied Mechanics and Engineering*, 194(12–16):1641–1654, April 2005.

[64] D. Ghosh and R. Ghanem. A new algorithm for solving the random eigenvalue problem using polynomial chaos expansion. In *46th AIAA/ASME/AHS/ASC Structure, Structural Dynamics, and Materials Conference*, Austin, TX, April 2005.

[65] C. H. Lamarque and E. Gourdon. Nonlinear energy sink with uncertain parameters. *Journal of Computational and Nonlinear Dynamics*, 1:187–195, 2006.

[66] T. Lovett, F. Ponci, and A. Monti. A polynomial chaos approach to measurement uncertainty. In *Advanced Methods for Uncertainty Estimation in Measurement. Proceedings of the 2005 IEEE International Workshop on*, pages 33–38, Ontario, Canada, May 2005.

[67] A. Monti, F. Ponci, and T. Lovett. A polynomial chaos theory approach to uncertainty in electrical engineering. In *Intelligent Systems Application to Power Systems. Proceedings of the 13th International Conference on*, Arlington, VA, November 2005.

[68] T. E. Lovett, F. Ponci, and A. Monti. A polynomial chaos approach to measurement uncertainty. *Instrumentation and Measurement, IEEE Transactions on*, 55(3):729–736, June 2006.

[69] A. Monti, F. Ponci, T. Lovett, A. Smith, and R. Dougal. Modeling of uncertainty and applications in monitoring and control of power electronics. In *American Control Conference. Proceedings of the*, volume 3, pages 2011–2016, Minneapolis, MN, June 2005.

[70] Q. Su and K. Strunz. Stochastic circuit modelling with Hermite polynomial chaos. *Electronics Letters*, 41(21):1163–1165, October 2005.

[71] X. Wan and G. E. Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2): 617–642, November 2005.

[72] P. S. Beran, C. L. Pettit, and Daniel R. Millman. Uncertainty quantification of limit-cycle oscillations. *Journal of Computational Physics*, 217(1):217–247, September 2006.

[73] G. D'Antona, A. Monti, F. Ponci, and L. Rocc. Maximum entropy analytical solution for stochastic differential equations based on the Wiener-Askey polynomial chaos. In *Advanced Methods for Uncertainty Estimation in Measurement. AMUEM 2006. Proceedings of the 2006 IEEE International Workshop on*, pages 62–66, Trento, Italy, April 2006.

[74] G. D'Antona, A. Monti, F. Ponci, and L. Rocca. Maximum entropy multivariate analysis of uncertain dynamical systems based on the Wiener Askey polynomial chaos. *Instrumentation and Measurement, IEEE Transactions on*, 56(3):689–695, June 2007.

[75] S. Das, R. Ghanem, and J. C. Spall. Asymptotic sampling distribution for polynomial chaos representation of data: A maximum entropy and Fisher information approach. In *Decision and Control, 45th IEEE Conference on*, pages 4139–4144, San Diego, CA, December 2006.

[76] R. G. Ghanem and A. Doostan. On the construction and analysis of stochastic models: Characterization and propagation of the errors associated with limited data. *Journal of Computational Physics*, 217(1):63–81, September 2006.

[77] M. Guedri, N. Bouhaddi, and R. Majed. Reduction of the stochastic finite element models using a robust dynamic condensation method. *Journal of Sound and Vibration*, 297(1–2): 123–145, October 2006.

[78] O. M. Knio and O. P. Le Maître. Uncertainty propagation in CFD using polynomial chaos decomposition. *Fluid Dynamics Research*, 38(9):616–640, September 2006.

[79] K. D. Lepage. Estimation of acoustic propagation uncertainty through polynomial chaos expansions. In *Information Fusion, 9th International Conference on*, pages 1–5, Florence, Italy, July 2006.

[80] G. Lin, C. H. Su, and G. E. Karniadakis. Predicting shock dynamics in the presence of uncertainties. *Journal of Computational Physics*, 217(1):260–276, September 2006.

[81] G. Lin, L. Grinberg, and G. E. Karniadakis. Numerical studies of the stochastic Korteweg-de Vries equation. *Journal of Computational Physics*, 213(2):676–703, April 2006.

[82] M. Mi, J. Fan, and S. X.-D. Tan. Statistical analysis of power grid networks considering lognormal leakage current variations with spatial correlation. In *Computer Design. ICCD 2006. International Conference on*, pages 56–62, San Jose, CA, October 2006.

[83] N Mi, J. Fan, and S. X.-D. Tan. Simulation of power grid networks considering wires and lognormal leakage current variations. In *Behavioral Modeling and Simulation Workshop, Proceedings of the 2006 IEEE International*, pages 73–78, San Jose, CA, September 2006.

[84] S. B. Mulani, R. K. Kapania, and R. W. Walters. Stochastic eigenvalue problem with polynomial chaos. In *47th AIAA/ASME/AHS/ASC Structure, Structural Dynamics, and Materials Conference*, Newport, RI, May 2006.

[85] S. Sachdeva, P. B. Nair, and A. J. Keane. Comparative study of projection schemes for stochastic finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 195(19–22):2371–2392, April 2006.

[86] S. K. Sachdeva, P. B. Nair, and A. J. Keane. Hybridization of stochastic reduced basis methods with polynomial chaos expansions. *Probabilistic Engineering Mechanics*, 21(2): 182–192, April 2006.

[87] A. Sinha. Computation of the statistics of forced response of a mistuned bladed disk assembly via polynomial chaos. *Journal of Vibration and Acoustics*, 128:449–457, 2006.

[88] A. Smith, A. Monti, and F. Ponci. Indirect measurements via polynomial chaos observer. In *Advanced Methods for Uncertainty Estimation in Measurement. AMUEM 2006. Proceedings of the 2006 IEEE International Workshop on*, pages 27–32, Trento, Italy, April 2006.

[89] Q. Su and K. Strunz. Stochastic polynomial-chaos-based average model of twelve-pulse diode rectifier for aircraft applications. In *Computers in Power Electronics. COMPEL '06. IEEE Workshops on*, pages 64–68, Troy, NY, July 2006.

[90] D. M. Tartakovsky and D. Xiu. Stochastic analysis of transport in tubes with rough walls. *Journal of Computational Physics*, 217(1):248–259, September 2006.

[91] B. Velamur Asokan and N. Zabaras. A stochastic variational multiscale method for diffusion in heterogeneous random media. *Journal of Computational Physics*, 218(2):654–676, November 2006.

[92] X. Wan and G. E. Karniadakis. Long-term behavior of polynomial chaos in stochastic flow simulations. *Computer Methods in Applied Mechanics and Engineering*, 195(41–43):5582–5596, August 2006.

[93] M. M. R. Williams. Polynomial chaos functions and stochastic differential equations. *Annals of Nuclear Energy*, 33(9):774–785, June 2006.

[94] N. Agarwal and N. R. Aluru. A stochastic Lagrangian approach for geometrical uncertainties in electrostatics. *Journal of Computational Physics*, 226(1):156–179, September 2007.

[95] H. Cheng and A. Sandu. Numerical study of uncertainty quantification techniques for implicit stiff systems. In *ACMSE*, pages 367–372, Winston-Salem, NC, 2007.

[96] J. Foo, Z. Yosibash, and G. E. Karniadakis. Stochastic simulation of riser-sections with uncertain measured pressure loads and/or uncertain material properties. *Computer Methods in Applied Mechanics and Engineering*, 196(41–44):4250–4271, September 2007.

[97] B. Ganapathysubramanian and N. Zabaras. Sparse grid collocation schemes for stochastic natural convection problems. *Journal of Computational Physics*, 225(1):652–685, July 2007.

[98] S. E. Geneser, R. M. Kirby, D. Xiu, and F. B. Sachse. Stochastic Markovian modeling of electrophysiology of ion channels: Reconstruction of standard deviations in macroscopic currents. *Journal of Theoretical Biology*, 245(4):627–637, April 2007.

[99] S. Ghanmi, M. L. Bouazizi, and N. Bouhaddi. Robustness of mechanical systems against uncertainties. *Finite Elements in Analysis and Design*, 43(9):715–731, June 2007.

[100] R. Ghanem, G. Saad, and A. Doostan. Efficient solution of stochastic systems: Application to the embankment dam problem. *Structural Safety*, 29(3):238–251, July 2007.

[101] S. Hosder, R. W. Walters, and M. Balch. Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. In *48th AIAA/ASME/AHS/ASC Structure, Structural Dynamics, and Materials Conference*, Honolulu, HI, April 2007.

[102] S. Huang, S. Mahadevan, and R. Rebba. Collocation-based stochastic finite element analysis for random field problems. *Probabilistic Engineering Mechanics*, 22(2):194–205, April 2007.

[103] O. P. Le Maître and O. M. Knio. A stochastic particle-mesh scheme for uncertainty propagation in vortical flows. *Journal of Computational Physics*, 226(1):645–671, September 2007.

[104] L. Li and C. Sandu. On the impact of cargo weight, vehicle parameters, and terrain characteristics on the prediction of traction for off-road vehicles. *Journal of Terramechanics*, 44 (3):221–238, July 2007.

[105] D. Lucor, C. Enaux, H. Jourdren, and P. Sagaut. Stochastic design optimization: Application to reacting flows. *Computer Methods in Applied Mechanics and Engineering*, 196(49–52): 5047–5062, November 2007.

[106] Y. M. Marzouk, H. N. Najm, and L. A. Rahn. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics*, 224(2):560–586, June 2007.

[107] S. J. Moon, B. Nabet, N. E. Leonard, S. A. Levin, and I. G. Kevrekidis. Heterogeneous animal group models and their group-level alignment dynamics: An equation-free approach. *Journal of Theoretical Biology*, 246(1):100–112, May 2007.

[108] Z. K. Nagy and R. D. Braatz. Distributional uncertainty analysis using power series and polynomial chaos expansions. *Journal of Process Control*, 17(3):229–240, March 2007.

[109] M. F. Ngah and A. Young. Application of the spectral stochastic finite element method for performance prediction of composite structures. *Composite Structures*, 78(3):447–456, May 2007.

[110] M. Paffrath and U. Wever. Adapted polynomial chaos expansion for failure detection. *Journal of Computational Physics*, 226(1):263–281, September 2007.

[111] C. P. Rupert and C. T. Miller. An analysis of polynomial chaos approximations for modeling single-fluid-phase flow in porous medium systems. *Journal of Computational Physics*, 226 (2):2175–2205, October 2007.

[112] G. Saad, R. Ghanem, and S. Masri. Robust system identification of strongly non-linear dynamics using a polynomial chaos based sequential data assimilation technique. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Honoulu, HI, April 2007.

[113] B. Seynaeve, E. Rosseel, B. Nicolaï, and S. Vandewalle. Fourier mode analysis of multigrid methods for partial differential equations with random coefficients. *Journal of Computational Physics*, 224(1):132–149, May 2007.

[114] A. Smith, A. Monti, and F. Ponci. Uncertainty and worst case analysis for a low-pass filter using polynomial chaos theory. In *Advanced Methods for Uncertainty Estimation in Measurement. IEEE International Workshop on*, pages 59–63, Sardagna, Italy, July 2007.

[115] J. A. S. Witteveen, S. Sarkar, and H. Bijl. Modeling physical uncertainties in dynamic stall induced fluid-structure interaction of turbine blades using arbitrary polynomial chaos. *Computers & Structures*, 85(11–14):866–878, June–July 2007.

[116] D. Xiu and S. J. Sherwin. Parametric uncertainty analysis of pulse wave propagation in a model of a human arterial network. *Journal of Computational Physics*, 226(2):1385–1407, October 2007.

[117] X. F. Xu. A multiscale stochastic finite element method on elliptic problems involving uncertainties. *Computer Methods in Applied Mechanics and Engineering*, 196(25–28):2723–2736, May 2007.

[118] Y. Zou, Y. Cai, Q. Zhou, X. Hong, S. X.-D. Tan, and L. Kang. Practical implementation of stochastic parameterized model order reduction via Hermite polynomial chaos. In *Design Automation Conference. ASP-DAC '07. Asia and South Pacific*, pages 367–372, Yokohama, Japan, January 2007.

[119] N. Agarwal and N. R. Aluru. Stochastic modeling of coupled electromechanical interaction for uncertainty quantification in electrostatically actuated MEMS. *Computer Methods in Applied Mechanics and Engineering*, 197(43–44):3456–3471, August 2008.

[120] G. Blatman and Bruno Sudret. Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach. *Comptes Rendus Mécanique*, 336(6):518–523, June 2008.

[121] L. Chamoin, J. T. Oden, and S. Prudhomme. A stochastic coupling method for atomic-to-continuum Monte-Carlo simulations. *Computer Methods in Applied Mechanics and Engineering*, 197(43–44):3530–3546, August 2008.

[122] R. G. Ghanem, A. Doostan, and John Red-Horse. A probabilistic construction of model validation. *Computer Methods in Applied Mechanics and Engineering*, 197(29–32):2585–2595, May 2008.

[123] U. Hauptmanns. Comparative assessment of the dynamic behaviour of an exothermal chemical reaction including data uncertainties. *Chemical Engineering Journal*, 140(1–3):278–286, July 2008.

[124] B. Jin and J. Zou. Inversion of Robin coefficient by a spectral stochastic finite element approach. *Journal of Computational Physics*, 227(6):3282–3306, March 2008.

[125] C. R. Laing and I. G. Kevrekidis. Periodically-forced finite networks of heterogeneous globally-coupled oscillators: A low-dimensional approach. *Physica D: Nonlinear Phenomena*, 237(2):207–215, February 2008.

[126] X. Li, J. M. Wang, W Tang, and H. Wu. Stochastic analysis for crosstalk noise of coupled interconnects with process variations. In *Integrated Circuit Design and Technology and Tutorial. ICICDT 2008. IEEE International Conference on*, pages 289–292, Grenoble, France, June 2008.

[127] T. E. Lovett, A. Monti, and F. Ponci. Automatic synthesis of uncertain models for linear circuit simulation: A polynomial chaos theory approach. *Simulation Modelling Practice and Theory*, 16(7):796–816, August 2008.

[128] N. Mi, J. Fan, S. X-D. Tan, Y Cai, and X. Hong. Statistical analysis of on-chip power delivery networks considering lognormal leakage current variations with spatial correlation. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 55(7):2064–2075, August 2008.

[129] N. Mi, S. X.-D. Tan, P. Liu, J. Cui, Y. Cai, and X. Hong. Stochastic extended Krylov subspace method for variational analysis of on-chip power grid networks. In *Computer-Aided Design. ICCAD 2007. IEEE/ACM International Conference on*, pages 48–53, San Jose, CA, November 2007.

[130] A. Monti, F. Ponci, and M. Valtorta. Extending polynomial chaos to include interval analysis. In *Advanced methods for uncertainty estimation in measurement, AMUEM, IEEE international workshop on*, pages 7–11, Trento, Italy, July 2008.

[131] A. Smith, A. Monti, and F. Ponci. Confidence interval estimation using polynomial chaos theory. In *Advanced methods for uncertainty estimation in measurement, IEEE International Workshop on*, pages 12–16, Sardagna, Italy, July 2008.

[132] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, July 2008.

[133] J. A. S. Witteveen, A. Loeven, S. Sarkar, and H. Bijl. Probabilistic collocation for period-1 limit cycle oscillations. *Journal of Sound and Vibration*, 311(1–2):421–439, March 2008.

[134] N.-Z. Chen and C. Guedes Soares. Spectral stochastic finite element analysis for laminated composite plates. *Computer Methods in Applied Mechanics and Engineering*, 197:4830–4839, July 2008.

[135] J. Guilleminot, C. Soize, D. Kondo, and C. Binetruy. Theoretical framework and experimental procedure for modelling mesoscopic volume fraction stochastic fluctuations in fiber reinforced composites. *International Journal of Solids and Structures*, 45:5567–5583, October 2008.

[136] A. Nouy, A. Clément, F. Schoefs, and N. Moës. An extended stochastic finite element method for solving stochastic partial differential equations on random domains. *Computer Methods in Applied Mechanics and Engineering*, 197:4663–4682, October 2008.

[137] D. L. Wei, Z. S. Cui, and J. Chen. Uncertainty quantification using polynomial chaos expansion with points of monomial cubature rules. *Computers & Structures*, 86:2175–2108, December 2008.

[138] H. N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41(1), 2009.

[139] K. A. Loparo and X. Feng. Stability of stochastic systems. In W. W. Levine, editor, *The Control Handbook*, pages 1105–1126. CRC Press, Boca Raton, FL, 1996.

[140] L. Debnath and P. Mikusiński. *Introduction to Hilbert spaces with applications*. Academic Press, New York, 1999.

[141] A. M. Krall. *Hilbert space, boundary value problems and orthogonal polynomials*. Birkhäuser Verlag, Boston, 2002.

[142] H. K. Khalil. *Nonlinear systems*. Prentice Hall, Inc., Upper Saddle River, NJ., 3rd edition, 2002.

[143] C. Canuto and A. Quarteroni. Approximation results for orthogonal polynomials in Sobolev spaces. *Mathematics of Computation*, 38(157):67–86, 1982.

[144] G. Bernardi, C. Coppoletta and Y. Maday. Some spectral approximations of two-dimensional fourth-order problems. *Mathematics of Computation*, 59(199):63–76, 1992.

[145] J. S. Heshaven, S. Gottlieb, and D. Gottlieb. *Spetral methods for time-dependant problems*. Cambridge University Press, New York, 2007.

[146] D. Gottlieb and S. A. Orszag. *Numerical analysis of spectral methods: theory and applications*. Society for Industrial Mathematics, 1987.

[147] A. Sard. *Linear approximation*. American Mathematical Society, Providence, RI, 1963.

[148] S. Shimp. Vehicle sprung mass parameter estimation using an adaptive polynomial-chaos method. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2008.

[149] D. Lucor and G. E. Karniadakis. Adaptive generalized chaos for nonlinear random oscillators. *SIAM J. Sci. Comput*, 26:720–735, 2004.

[150] J. Ackermann. *Robust control: the parameter space approach*. Springer-Verlag, London, 2002.

[151] D. Cox. *Control design for parameter dependent aeroelastic systems*. PhD thesis, Duke University, Durham, NC, 2003.

[152] L. Lublin, S. Grocott, and M. Athans. $\mathcal{H}_2$ (LQG) and $\mathcal{H}_\infty$ control. In W. W. Levine, editor, *The Control Handbook*, pages 651–661. CRC Press, Boca Raton, FL, 1996.

[153] S. P. Boyd. *Linear matrix inequalities in system and control theory*. SIAM, Philadelphia, PA, 1994.

[154] J. Lofberg. YALMIP: A toolbox for modeling and optimization in matlab. In *Proceedings of the CACSD Conference*, Tapei, Taiwan, 2004.

[155] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[156] R. L. Ott and M. Longnecker. *An introduction to statistical methods and data analysis*. Duxbury, Thomson Learning, Pacific Grove, CA, 5th edition, 2001.

[157] J. Ljung. System identification. In W. W. Levine, editor, *The Control Handbook*, pages 1033–1054. CRC Press, Boca Raton, FL, 1996.

[158] Q. Hu and G. Ma. Manoeuvering and vibration reduction of a flexible spacecraft integrating optimal sliding mode controller and distributed piezoelectric sensors/actuators. *International Journal of Adaptive Control and Signal Processing*, 21:452–476, 2007.

[159] A. Bax and S. Poh. Optimal vibration control with modal positive position feedback. *Optimal Control Applications & Methods*, 17:141–149, 1996.

Appendix A

Matlab Implementation

This appendix contains the code used to solve the example problems in Chapter 5. The first section contains higher level scripts to implement the two problems. These problems utilize a developed Multi-Dimensional Polynomical Chaos (MDPC) Library. Section A.2 contains the libraries that work with one-dimensional basis functions. Section A.3 builds on these libraries in order to allow using the Galerkin method over multiple probabilistic dimensions. These libraries are used for the other Polynomial Chaos examples in the dissertation, as well as in tandem with Matlab's *fminunc* for optimizing the controllers used in Chapter 6.

A.1 Scripts using the MDPC Library

The first script in this section, *uncert_cont_kplot2.m* was used in the analysis of the first example in Section 5.9 and Figure 5.2. The following example used the script *uncert_contr_test3.m*. Although not required for working directly with the Polynomial Chaos libraries, the SeDuMi [155] and YALMIP [154] libraries were used to solve LMIs.

```
% Brian Templeton
% Example using MDPC Library, v0.3
% Test 1-dof problem--creates cost surface
% Created: June 18, 2008
% Last Updated: July 26, 2008

genconsts = 1; % set to 1 to generate constants, set to 0 if you already
               % generated the constants

if genconsts
    % Note that there are 2 independent random variables
    rv1 = 1;
    rv2 = 2;

    % Define multidimensional polynomial set
    ps.poly(rv1).name = 'Lege';
    %ps.poly(rv2).name = 'Lege';
    ps.poly(rv2).name = 'Jaco';
    ps.poly(rv2).a = 1; %Extra parameters needed for the Jacobi polynomials
    ps.poly(rv2).b = 1;
    % Order of 1D polynomial sets
    ps.L1D = 4;

    % Complete the structure
    ps = gen_basis_struct(ps);
end

% Going to assume that random variables for parameters are independent

% Defining parameters
% Do NOT make names substrings of other names. I.e.
% k1 and k2 together are okay
% k11 and k12 together are okay
% k1 and k11 together are NOT okay
% Do NOT include random parameters in any non-linear way.
% I.e., where k1 and k2 are functions random parameters
% k1+k2 is okay
% k1*k2 is NOT okay for now unless one of them is declared as a constant
% 1/k1 is NOT okay
%
% Note, need to be able to distinguish between constants that drop out and
% those that do not. I.e. c*k1 does not drop out. c in c+k1 only effects
% the first coefficient of the expansion
%
% The cf flag means:
% cf = 1, evaluate constant for all ic
% cf = 0, only evaluate constant when ic==0

pa.k1.val = [1 .5];      pa.k1.rv = rv1;
pa.k2.val = [1];         pa.k2.cf = 0;
pa.d1.val = [0 .5];      pa.d1.rv = rv2; %[1 .2];      pa.d1.rv = rv2;
```

```
52 pa.d2.val = [1];          pa.d2.cf = 0;
53 pa.m1.val = [1];          pa.m1.cf = 1;
54 pa.m2.val = [1];          pa.m2.cf = 1;
55 pa.one.val = [1];         pa.one.cf = 0;
56
57 %state_matrix.A = '[k1]';
58
59 % Declare state matrices
60 state_matrix.A = '[0, one; -k1/m1, -d1/m1]';
61 % state_matrix.Bwe = '[0, 0; 0, 0; 0, 0; k2/m2, d2/m2]';  % Used for
       excitation disturbance
62 state_matrix.Bwe = '[0, 0; k1/m1, d1/m1]';  % Used for excitation
       disturbance
63 % state_matrix.Bwedesign = '[one 0 0 0; 0 one 0 0; 0 0 one 0; 0 0 0 one]';
64 state_matrix.Bu = '[0; one/m1]';  % The one with the control input
65 state_matrix.C = '[one, 0; 0, one]';
66
67 % Evaluate the multidimensional polynomials
68 %rand_var_vec = [0 0];
69 %[pvmd,pv1d] = eval_md_poly(ps,rand_var_vec);
70
71 % Expand state matrices
72 e_s_m = exp_state_matrix(state_matrix,pa,ps);
73
74 %d_s_m = det_state_matrix(pa,state_matrix,pv1d);
75
76
77 % Create the Realization matrix
78 %num_states = 2;
79 %e_s_m.E = create_realiz_m(ps,num_states,rand_var_vec);
80
81 % Make it so that the matrix only needs deterministic inputs
82 %temp = zeros(ps.L_p1,1); temp(1) = 1;
83 e_s_m.Bw = e_s_m.Bwe*d_i_matrix(2,ps.L);
84 %e_s_m.Bwedesign = e_s_m.Bwe*d_i_matrix(2,ps.L);
85
86 % Forcing Inputs
87 Amp = 1;
88 omega = 4;
89 t = 0:.01:10;
90 u = [Amp*sin(omega*t); Amp*omega*cos(omega*t)];
91
92 % Open Loop System
93
94 % Polynomial Chaos
95 %pcss = ss(e_s_m.A,e_s_m.Bw,e_s_m.E,0);
96 %[Yexp,T,Xexp]=lsim(pcss,u,t);
97
98 % Closed Loop System
99
100 % Define weights for Means and Covariances
```

```
QMd = [1 1];
QCd = [0 0];

QM = diag(QMd); %[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
QC = diag(QCd); %[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
RM = [1];
RC = [.1];

% Calculate Weighting Matrices for LQR
Qlqr = create_Mean_lqr(QM,ps.L) + create_Cov_lqr(QC,ps.mddelta2);
Rlqr = create_Mean_lqr(RM,ps.L) + create_Cov_lqr(RC,ps.mddelta2);

% Output from solver
opts = sdpsettings('savesolveroutput',1);

% Expanded state matrices
A = e_s_m.A;
Bu = e_s_m.Bu;
Bw = e_s_m.Bw;
C = e_s_m.C;


%Qlqr=QM;
%Rlqr=RM;

%Ar = e_s_m.E*A*d_i_matrix(4,ps.L);
%Bur = e_s_m.E*Bu*d_i_matrix(1,ps.L);
%Bwr = e_s_m.E*Bw;
%Cr = e_s_m.E*C*d_i_matrix(4,ps.L);

%N = size(A,1);           % number of expanded states
%N2 = N/ps.L_p1;          % number of unexpanded states
%M = size(Bu,2);          % number of control inputs
%M2 = M/ps.L_p1;          % number of unexpanded control inputs
%Mw = size(Bw,2);         % number of disturbance inputs
%Mwr = size(Bwr,2);       % number of realization disturbance inputs
%P = size(C,1);           % number of measurement outputs
%P2 = P/ps.L_p1;          % number of unexpanded measurement outputs


k1v = (linspace(-3,1,50));
k2v = (linspace(-4,0,50));

H22 = zeros(length(k1v),length(k2v));

I = eye(ps.L_p1);
%I = 1;

for k1c = 1:length(k1v)
    for k2c = 1:length(k2v);
        Kexp = kron(I,[k1v(k1c), k2v(k2c)]);
```

```
        Acl = A + Bu*Kexp;
        stable_cond = (sum(real(eig(Acl))>=0) == 0);
        if stable_cond
            %X = lyap((Acl), (C.'*Qlqr*C+Kexp.'*Rlqr*Kexp));
            X = lyap((Acl).', (C.'*Qlqr*C+Kexp.'*Rlqr*Kexp));
            H22(k1c,k2c) = trace(Bw.'*X*Bw);
        else
            H22(k1c,k2c) = nan;
        end
    end
end


%title(sprintf('ps.L1D = %d',ps.L1D))
%colorbar

% Starting point for sample optimization trajectory
kp1 = .3; %2*rand(1)-.4;
kp2 = -2.5; %2*rand(1)-2;

k0 = [kp1, kp2];

%k0 = [-1,-1];

Kv(1,:) = k0;

alpha = 1;

% number of iterations
nH2=100
% simple finite difference used for Lyapunov gradient
dk = [0.001, 0.001];

for cc=1:nH2
    [grad_H2_2, ngrad_H2_2, H2_2] = del_H2(A,Bw,Bu,C,Kv(cc,:),dk,Qlqr,Rlqr)
        ;

    grad_H2_2v(cc,:) = grad_H2_2;
    ngrad_H2_2v(cc,:) = ngrad_H2_2;
    H2_2v(cc) = H2_2;
    Kv(cc+1,:) = Kv(cc,:)-alpha*ngrad_H2_2v(cc,:);
end

figure
%surf(k2v,k1v,log(H22))
plot3(Kv(1:nH2,2),Kv(1:nH2,1),log10(H2_2v),'r','LineWidth',2)
xl=xlabel('$k_2$','interpreter','latex');
yl=ylabel('$k_1$','interpreter','latex');
zl=zlabel('$log_{10} \left ( \left \|H \right\|_2^2 \right )$','interpreter
    ','latex');
pub_plot(xl,yl,zl)
```

```
%title(sprintf('k=[%6.4d, %6.4d]',Kv(20,1),Kv(20,2)))
hold on;
surf(k2v,k1v,log10(H22))
grid on;
hold off;

view(-63,31)
%shading interp
%lightangle(100,100)
%set(gcf,'Renderer','opengl')
%set(findobj(gca,'type','surface'),...
%    'FaceLighting','phong',...
%    'AmbientStrength',.3,'DiffuseStrength',.8,...
%    'SpecularStrength',.9,'SpecularExponent',25,...
%    'BackFaceLighting','unlit')
```

```
% Brian Templeton
% Example using MDPC Library, v0.3
% Test 2-dof problem--optimizes gains
% Created: June 18, 2008
% Last Updated: July 26, 2008

plot_figures = 1;

genconsts = 1; % set to 1 to generate constants, set to 0 if you already
               % generated the constants

if genconsts
    % Note that there are 2 independent random variables
    rv1 = 1;
    rv2 = 2;

    % Define multidimensional polynomial set
    ps.poly(rv1).name = 'Lege';
    %ps.poly(rv2).name = 'Lege';
    ps.poly(rv2).name = 'Jaco';
    ps.poly(rv2).a = 1; %Extra parameters needed for the Jacobi polynomials
    ps.poly(rv2).b = 1;
    % Order of 1D polynomial sets
    ps.L1D = 4;

    % Complete the structure
    ps = gen_basis_struct(ps);
end

% Going to assume that random variables for parameters are independent

% Defining parameters
% Do NOT make names substrings of other names. I.e.
% k1 and k2 together are okay
```

```
% k11 and k12 together are okay
% k1 and k11 together are NOT okay
% Do NOT include random parameters in any non-linear way.
% I.e., where k1 and k2 are functions random parameters
% k1+k2 is okay
% k1*k2 is NOT okay for now unless one of them is declared as a constant
% 1/k1 is NOT okay
%
% Note, need to be able to distinguish between constants that drop out and
% those that do not. I.e. c*k1 does not drop out. c in c+k1 only effects
% the first coefficient of the expansion
%
% The cf flag means:
% cf = 1, evaluate constant for all ic
% cf = 0, only evaluate constant when ic==0

pa.k1.val = [1 .5];      pa.k1.rv = rv1;
pa.k2.val = [1];         pa.k2.cf = 0;
pa.d1.val = [1 .5];      pa.d1.rv = rv2; %[1 .2];      pa.d1.rv = rv2;
pa.d2.val = [1];         pa.d2.cf = 0;
pa.m1.val = [1];         pa.m1.cf = 1;
pa.m2.val = [1];         pa.m2.cf = 1;
pa.one.val = [1];        pa.one.cf = 0;

%state_matrix.A = '[k1]';

% Declare state matrices
state_matrix.A = '[0, 0, one, 0; 0, 0, 0, one; -k1/m1, k1/m1, -d1/m1, d1/m1
    ; k1/m2, -(k1+k2)/m2, d1/m2, -(d1+d2)/m2]';
%state_matrix.Bwe = '[0, 0; 0, 0; 0, 0; k2/m2, d2/m2]';  % Used for
    excitation disturbance
state_matrix.Bwesim = '[0; 0; 0; one]';  % Used for excitation disturbance
state_matrix.Bwedesign = '[one 0 0 0; 0 one 0 0; 0 0 one 0; 0 0 0 one]';
state_matrix.Bu = '[0; 0; one/m1; 0]';  % The one with the control input
state_matrix.C = '[one, 0, 0, 0; 0, one, 0, 0; 0, 0, one, 0; 0, 0, 0, one]'
    ;

% Evaluate the multidimensional polynomials
rand_var_vec = [0 0];
[pvmd,pv1d] = eval_md_poly(ps,rand_var_vec);

% Expand state matrices
e_s_m = exp_state_matrix(state_matrix,pa,ps);

%d_s_m = det_state_matrix(pa,state_matrix,pv1d);


% Create the Realization matrix
num_states = 4;
e_s_m.E = create_realiz_m(ps,num_states,rand_var_vec);

```

```
% Make it so that the matrix only needs deterministic inputs
%temp = zeros(ps.L_p1,1); temp(1) = 1;
e_s_m.Bwsim = e_s_m.Bwesim*d_i_matrix(1,ps.L);
e_s_m.Bw = e_s_m.Bwedesign*d_i_matrix(4,ps.L);


% Forcing Inputs
Amp = 1;
omega = 4;
t = 0:.01:10;
u = [Amp*sin(omega*t); Amp*omega*cos(omega*t)];

% Open Loop System

% Polynomial Chaos
pcss = ss(e_s_m.A,e_s_m.Bw,e_s_m.E,0);
%[Yexp,T,Xexp]=lsim(pcss,u,t);

% Closed Loop System

% Define weights for Means and Covariances
QMd = [1 1 1 1];
QCd = [0 0 100 0];

QM = diag(QMd); %[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
QC = diag(QCd); %[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
RM = [1];
RC = [1];

% Calculate Weighting Matrices for LQR
Qlqr = create_Mean_lqr(QM,ps.L) + create_Cov_lqr(QC,ps.mddelta2);
Rlqr = create_Mean_lqr(RM,ps.L) + create_Cov_lqr(RC,ps.mddelta2);

% Output from solver
opts = sdpsettings('savesolveroutput',1);

A = e_s_m.A;
Bu = e_s_m.Bu;
Bw = e_s_m.Bw;
Bwsim = e_s_m.Bwsim;
C = e_s_m.C;

Ar = e_s_m.E*A*d_i_matrix(4,ps.L);
Bur = e_s_m.E*Bu*d_i_matrix(1,ps.L);
Bwr = e_s_m.E*Bw;
Cr = e_s_m.E*C*d_i_matrix(4,ps.L);

N = size(A,1);          % number of expanded states
N2 = N/ps.L_p1;         % number of unexpanded states
M = size(Bu,2);         % number of control inputs
M2 = M/ps.L_p1;         % number of unexpanded control inputs
```

```
Mw = size(Bw,2);          % number of disturbance inputs
Mwr = size(Bwr,2);        % number of realization disturbance inputs
P = size(C,1);            % number of measurement outputs
P2 = P/ps.L_p1;           % number of unexpanded measurement outputs

% Clear YALMIP variables
yalmip('clear');


% Reformulate as weights for LMIs
WQ = sqrtm(Qlqr)*C; %may not be quite right
WR = sqrtm(Rlqr);


% LQR-esque test for a deterministic system
% Clear YALMIP variables
yalmip('clear');

% Create LMI variables
Q = sdpvar(N2,N2,'symmetric');
Y = sdpvar(M2,N2,'full');
Z = sdpvar(Mwr,Mwr,'symmetric');
Ik = eye(ps.L_p1);


% Reformulate as weights for LMIs
WQr = sqrtm(QM)*Cr; %may not be quite right
WRr = sqrtm(RM);

% Define LMIs
H2lmi_1 = [Ar*Q+Q*Ar'+Bur*Y+Y'*Bur', [Q*WQr.', Y'*WRr'];
            [WQr*Q; WRr*Y], -eye(P2+M2)];

H2lmi_2 = [Z, Bwr';
           Bwr, Q];

% Assembly LMIs
F = [Q>0, H2lmi_1<0, H2lmi_2>0];

% Optimize LMI
[soln] = solvesdp(F,trace(Z),opts);
soln;

% Variable values
Qvr = double(Q);
Yvr = double(Y);
Zvr = double(Z);

% Gain for A+B*K
Klmir = Yvr*inv(Qvr)

```

```
185 % Closed loop system
186 Aclr = A+Bu*kron(Ik,Klmir);
187 pcsscl = ss(Aclr,Bwsim,kron(Ik,Klmir),0);
188 %[Yexpcl,T,Xexpcl] = lsim(pcsscl,u,t);
189 [Yexpcl,T,Xexpcl] = impulse(pcsscl,t);
190
191 % Find the H2 norm, squared
192 H2_2r = trace(Bw.'*lyap(Aclr.',(WQ*C).'*(WQ*C)+(WR*kron(Ik,Klmir)).'*(WR*
        kron(Ik,Klmir)))*Bw);
193
194 % Find the means and covariances
195 X_meancl1 = mean_states(Xexpcl,ps.L);
196 X_covcl1 = cov_states(Xexpcl,ps.L,ps.mddelta2);
197 U_mean1 = mean_states(Yexpcl,ps.L);
198 U_var1 = cov_states(Yexpcl,ps.L,ps.mddelta2);
199
200 if plot_figures
201 figure;
202 % Closed Loop
203 subplot(2,3,1);
204 plot(t,X_meancl1,t,U_mean1)
205 title(sprintf('Polynomial Chaos Simulation Mean Values, Closed Loop\n QM=
        diag[%1.2f, %1.2f, %1.2f, %1.2f], RM=%1.2f\n QC=diag[%1.2f, %1.2f, %1.2f
        , %1.2f], RC=%1.2f\nTrue H2 norm of realization solution = %6.2f', QMd
        (1),QMd(2),QMd(3),QMd(4),RM,QCd(1),QCd(2),QCd(3),QCd(4),RC,sqrt(H2_2r)))
        ;
206 xlabel('Time (s)')
207 ylabel('Amplitude')
208 h=legend('$mean(x_1)$','$mean(x_2)$','$mean(\dot x_1)$','$mean(\dot x_2)$',
        '$mean(F)$');
209 set(h,'interpreter','latex');
210 axis([0,10,-1,1])
211 grid on;
212 %
213 % % Variances
214 subplot(2,3,4);
215 plot(t,sqrt(X_covcl1(:,1,1)),t,sqrt(X_covcl1(:,2,2)),t,sqrt(X_covcl1(:,3,3)
        ),t,sqrt(X_covcl1(:,4,4)),t,sqrt(U_var1))
216 title(sprintf('Polynomial Chaos Simulation sqrt(Variance), Closed Loop\n QM
        =diag[%1.2f, %1.2f, %1.2f, %1.2f], RM=%1.2f\n QC=diag[%1.2f, %1.2f, %1
        .2f, %1.2f], RC=%1.2f', QMd(1),QMd(2),QMd(3),QMd(4),RM,QCd(1),QCd(2),QCd
        (3),QCd(4),RC));
217 xlabel('Time (s)')
218 ylabel('Amplitude')
219 h=legend('$var(x_1)$','$var(x_2)$','$var(\dot x_1)$','$var(\dot x_2)$','
        $var(F)$');
220 set(h,'interpreter','latex');
221 axis([0,10,0,.1])
222 grid on;
223 end
224
```

```
225
226 %%%%%%%%%%%%%%%%%%%%%%%%
227 % Local optimizations
228
229 % Find the local minimum norm
230
231 nperts = 10;
232
233
234 %initialize
235 H2_2lmr_v = zeros(nperts,1);
236 lstr(nperts).eps = 0;
237 lstr(nperts).Kloptr = zeros(size(Klmir));
238 lstr(nperts).pert = zeros(size(Klmir));
239 lstr(nperts).count = 0;
240 lstr(nperts).stable_eigs = 0;
241
242 alpha_v = [0.1, 0.1, 0.10];
243 eps_lim_v = [0.1, 0.0001, 0.00001];
244 dKc_v = [0.01, 0.001, 0.0001];
245
246 for lc=1:nperts
247 fprintf('%d\n',lc);
248
249 %pert = 4*(rand(1,4)-.5).*Klmir/2;
250
251 %Kloptr = Klmir + pert;
252 Kloptr = -10*rand(1,4);
253 Kloptr0 = Kloptr;
254
255 eps = eps_lim_v(1);
256
257 H2_2lmr = 100*H2_2r;
258
259 count = 0;
260
261 sys_eig = eig(A+Bu*kron(eye(ps.L_p1),Kloptr));
262 stable_eigs = sum(real(sys_eig)>0)==0;
263
264 if ~stable_eigs
265     fprintf('oops1\n');
266 end
267
268 dK = ones(size(Kloptr));
269
270 for cc = 1:length(alpha_v);
271
272 eps_lim = eps_lim_v(cc);
273 alpha = alpha_v(cc);
274 dKc = dKc_v(cc);
275 while (eps > eps_lim) && stable_eigs
```

```
276      count = count + 1;
277      back_up.eps = eps;
278      back_up.H2_2lmr = H2_2lmr;
279      back_up.Kloptr = Kloptr;
280
281      [pH2,npH2,H2_2lmr] = del_H2(A,Bw,Bu,C,Kloptr,dK*dKc,WQ^2,WR^2);
282      Kloptr = Kloptr-alpha*npH2; % changed this
283      eps = back_up.H2_2lmr-H2_2lmr;
284      sys_eig = eig(A+Bu*kron(eye(ps.L_p1),Kloptr));
285      stable_eigs = sum(real(sys_eig)>0)==0;
286      if ~stable_eigs
287          fprintf('oops2\n');
288          break;
289      end
290      back_up.stable_eigs = stable_eigs;
291  end
292  if eps < 0 || ~stable_eigs
293      fprintf('oops3\n');
294      eps = back_up.eps;
295      Kloptr = back_up.Kloptr;
296      H2_2lmr = back_up.H2_2lmr;
297
298      sys_eig = eig(A+Bu*kron(eye(ps.L_p1),Kloptr));
299      back_up.stable_eigs = stable_eigs;
300
301      stable_eigs = sum(real(sys_eig)>0)==0;
302  end
303  end
304  lstr(lc).eps = eps;
305  lstr(lc).Kloptr = Kloptr;
306  lstr(lc).Kloptr0 = Kloptr0;
307  lstr(lc).count = count;
308  lstr(lc).stable_eigs = back_up.stable_eigs;
309  H2_2lmr_v(lc) = H2_2lmr;
310  end
311  %eig(A+Bu*kron(eye(ps.L_p1),Kloptr))
312
313  [H2_2lmr, lc]=min(H2_2lmr_v);
314
315  fprintf('\n\n%d Local optimizations complete.\nNumber of steps: %d\nEpsilon
         : %6.2g\nEigenvalue status: %d\nBegin H2 norm: %6.4f\nEnd H2 norm:   %6
         .4f\n\n',nperts, lstr(lc).count,lstr(lc).eps,lstr(lc).stable_eigs,sqrt(
        H2_2r),sqrt(H2_2lmr));
316
317  Kloptr = lstr(lc).Kloptr;
318
319  % Closed loop system
320  Acllmlo = A+Bu*kron(Ik,Kloptr);
321  pcsscl = ss(Acllmlo,Bwsim,kron(Ik,Kloptr),0);
322  %[Yexpcl,T,Xexpcl] = lsim(pcsscl,u,t);
323  [Yexpcl,T,Xexpcl] = impulse(pcsscl,t);
```

```
324
325 % Find the means and covariances
326 X_meancl2 = mean_states(Xexpcl,ps.L);
327 X_covcl2 = cov_states(Xexpcl,ps.L,ps.mddelta2);
328 U_mean2 = mean_states(Yexpcl,ps.L);
329 U_var2 = cov_states(Yexpcl,ps.L,ps.mddelta2);
330
331 if plot_figures
332
333 % Closed Loop
334 subplot(2,3,2);
335 title(sprintf('Polynomial Chaos Simulation Mean Values, Closed Loop\n QM=
      diag[%1.2f, %1.2f, %1.2f, %1.2f], RM=%1.2f\n QC=diag[%1.2f, %1.2f, %1.2f
      , %1.2f], RC=%1.2f\nLocal minimum of the H2 norm of realization solution
       = %6.2f', QMd(1),QMd(2),QMd(3),QMd(4),RM,QCd(1),QCd(2),QCd(3),QCd(4),RC
      ,sqrt(H2_2lmr)));
336 plot(t,X_meancl2,t,U_mean2,'linewidth',2)
337 xl=xlabel('Time (s)');
338 yl=ylabel('Response');
339 %legend('State 1','State 2','State 3','State 4','Input 1')
340 h=legend('$\rm{mean(x_1)}$','$\rm{mean(x_2)}$','$\rm{mean(\dot x_1)}$','$\
      rm{mean(\dot x_2)}$','$\rm{mean(F)}$','location',[0,0,1.62,1.5]);
341 set(h,'interpreter','latex');
342 axis([0,10,-1,1])
343 grid on;
344 pub_plot(xl,yl);
345
346 %
347 % % Variances
348 subplot(2,3,5);
349 title(sprintf('Polynomial Chaos Simulation sqrt(Variance), Closed Loop\n QM
      =diag[%1.2f, %1.2f, %1.2f, %1.2f], RM=%1.2f\n QC=diag[%1.2f, %1.2f, %1
      .2f, %1.2f], RC=%1.2f', QMd(1),QMd(2),QMd(3),QMd(4),RM,QCd(1),QCd(2),QCd
      (3),QCd(4),RC));
350 plot(t,sqrt(X_covcl2(:,1,1)),t,sqrt(X_covcl2(:,2,2)),t,sqrt(X_covcl2(:,3,3)
      ),t,sqrt(X_covcl2(:,4,4)),t,sqrt(U_var2),'linewidth',2)
351 xl=xlabel('Time (s)');
352 yl=ylabel('Response');
353 %legend('State 1','State 2','State 3','State 4','Input 1')
354 h=legend('$\rm{std(x_1)}$','$\rm{std(x_2)}$','$\rm{std(\dot x_1)}$','$\rm{
      std(\dot x_2)}$','$\rm{std(F)}$','location',[0,0,1.65,1.5]);
355 set(h,'interpreter','latex');
356 axis([0,10,0,.1])
357 grid on;
358 pub_plot(xl,yl);
359
360 % Change
361 % Closed Loop
362 subplot(2,3,3);
363 plot(t,X_meancl2-X_meancl1,t,U_mean2-U_mean1)
```

```
364 title(sprintf('Change in State Means, Closed Loop\n QM=diag[%1.2f, %1.2f,
       %1.2f, %1.2f], RM=%1.2f\n QC=diag[%1.2f, %1.2f, %1.2f, %1.2f], RC=%1.2f\
       nLocal minimum of the H2 norm of realization solution = %6.2f', QMd(1),
       QMd(2),QMd(3),QMd(4),RM,QCd(1),QCd(2),QCd(3),QCd(4),RC,sqrt(H2_2lmr)));
365 xlabel('Time (s)')
366 ylabel('Response')
367 legend('State 1','State 2','State 3','State 4','Input 1')
368 axis([0,10,-1,1])
369 grid on;
370 %
371 % % Variances
372 subplot(2,3,6);
373 plot(t,sqrt(X_covcl2(:,1,1))-sqrt(X_covcl1(:,1,1)),t,sqrt(X_covcl2(:,2,2))-
       sqrt(X_covcl1(:,2,2)),t,sqrt(X_covcl2(:,3,3))-sqrt(X_covcl1(:,3,3)),t,
       sqrt(X_covcl2(:,4,4))-sqrt(X_covcl1(:,4,4)),t,sqrt(U_var2)-sqrt(U_var1))
374 title(sprintf('Change in Variance, Closed Loop\n QM=diag[%1.2f, %1.2f, %1
       .2f, %1.2f], RM=%1.2f\n QC=diag[%1.2f, %1.2f, %1.2f, %1.2f], RC=%1.2f',
       QMd(1),QMd(2),QMd(3),QMd(4),RM,QCd(1),QCd(2),QCd(3),QCd(4),RC));
375 xlabel('Time (s)')
376 ylabel('Response')
377 legend('State 1','State 2','State 3','State 4','Input 1')
378 axis([0,10,-0.05,0.05])
379 grid on;
380
381 end
```

A.2 Single-Dimension Library Files

These libraries are used for generating and processing the single dimensional-polynomials (Table 2.1). In particular, *gen_pol_set_w.m* uses the symbolic capabilities of Matlab to generate the polynomial sets. The function, *gen_kron_delta.m* can create tables of inner products. Note that these are the only sections that utilize the symbolic routines. If desired, it would obviously be possible to do the manipulation and integration in other ways. However, since the tables only needed to be calculated once, the simple implementation was preferred.

```
1 function [pT]=basis_sym2poly(T)
2 % Convert a symbolic set of polynomials to a matrix of coefficients
3 %
4 % MDPC Library, v0.2
```

```
% Brian Templeton
% Created: in 2007
% Last Updated: June 19, 2008

const=length(T);
pT = zeros(const,const);
for cc = 1:const;
    pT(cc,:) = [zeros(1,(const-cc)), sym2poly(T(cc))];
end
```

```
function [c] = frac_coeff(a,b,poly)
% used to calculate the expansions for sum(a(xi))/sum(b(xi))

T = poly.T;
bounds = poly.bounds;

a = [a,zeros(1,(length(T)-length(a)))];
b = [b,zeros(1,(length(T)-length(b)))];

for cc=1:length(T)
    c(cc)=double(int((sum(a.*T)/sum(b.*T))*T(cc),bounds(1),bounds(2))/int(T
       (cc)^2,bounds(1),bounds(2)));
end
```

```
function delta = gen_kron_delta(T, w, x, bounds, dim, dbl_or_sym)
% This function tabulates the Kronecker delta for a 1D polynomial set for
% a given dimension.
%
% The function call is of the form:
%
% delta = gen_kron_delta(T, w, x, bounds, dim, dbl_or_sym)
% Function Arguments:
%
% T: An array containing the symbolic form of the polynomial set
% w: the symbolic weighting function
% x: the symbolic variable
% bounds: an 1D array of bounds for orthogonality interval. The first entry
%   is the left bound and the second entry is the right bound.
% dim: The dimensions of Kronecker delta, for one up to four
% Optional Argument:
% doubleorsym: convert to double with 'dbl' or no argument, leave as sym
%   with 'sym'
%
% Function Returns:
% delta: dirac deltas representing inner products
%
% Could be more efficient. Symmetry is not exploited.
%
```

```
% MDPC Library, v0.2
% Brian Templeton
% Created: in 2007
% Last Updated: November 24, 2008

if nargin == 5, dbl_or_sym = 'dbl'; end

Np1 = length(T);
lbound = bounds(1);
rbound = bounds(2);

% Define Polynomials

if dbl_or_sym == 'dbl'

% pre-allocate
delta = zeros(Np1*ones(1,dim));

for a=1:Np1
    if dim==1
        %T(a)
        delta(a)=double(int(T(a)*w,x,lbound,rbound));
    else
        for b=1:Np1
            if dim==2
                if a~=b
                    delta(a,b)=0;
                else
                    delta(a,b)=double(int(T(a)*T(b)*w,x,lbound,rbound));
                end
            else
                for c=1:Np1
                    if dim==3
                        %poly=expand(T(a)*T(b)*T(c))
                        delta(a,b,c)=double(int(T(a)*T(b)*T(c)*w,x,lbound,
                            rbound));
                        %fprintf('\n%d,%d,%d,%f\n',a,b,c,delta(a,b,c));
                    else
                        for d=1:Np1
                            if dim==4
                                delta(a,b,c,d)=double(int(T(a)*T(b)*T(c)*T(
                                    d)*w,x,lbound,rbound));
                            else
                                fprintf('\nError: This function is not
                                    designed for arguments greater than
                                    order 4 or less than order 1.\n')
                            end
                        end
                    end
                end
            end
```

```
        end
    end
end

else  % probably a temporary condition

% pre-allocate
delta = sym(zeros(Np1*ones(1,dim)));

for a=1:Np1
    if dim==1
        %T(a)
        delta(a)=(int(T(a)*w,x,lbound,rbound));
    else
        for b=1:Np1
            if dim==2
                if a~=b
                    delta(a,b)=0;
                else
                    delta(a,b)=double(int(T(a)*T(b)*w,x,lbound,rbound));
                end
            else
                for c=1:Np1
                    if dim==3
                        %poly=expand(T(a)*T(b)*T(c))
                        delta(a,b,c)=(int(T(a)*T(b)*T(c)*w,x,lbound,rbound)
                            );
                        %fprintf('\n%d,%d,%d,%f\n',a,b,c,delta(a,b,c));
                    else
                        for d=1:Np1
                            if dim==4
                                delta(a,b,c,d)=(int(T(a)*T(b)*T(c)*T(d)*w,x
                                    ,lbound,rbound));
                            else
                                fprintf('\nError: This function is not
                                    designed for arguments greater than
                                    order 4 or less than order 1.\n')
                            end
                        end
                    end
                end
            end
        end
    end
end

end
```

```
function [T,w,x,bounds] = gen_pol_set_w(order, set, a, b)
% Function recursively generates an array of a symbolic set of 1D
```

```
% polynomials, up to the desired order specified by the function call
%
% [T,w,x,bounds] = gen_pol_set_w(order,set,a,b).
%
% Function Arguments:
% order: Order of the Polynomial set, i.e. because the zeroth polynomial,
%   there are order+1 polynomials returned
% set: specify the set with a string: 'Lege', Herm', 'Cheb', 'Jaco'
% Optional Arguments:
% a, b: If the Jacobi polynomials are used, these parameters should be
%   specified
% a: If the Chebyshev polynomials are used, this argument should be used
%   to distinguish between the first kind and the second kind with a 1 or 2
%
% Function Outputs:
% T: An array of symbolic polynomials
% w: A symbolic weighting equation
% x: The symbolic variable
% bounds: an 1D array of bounds for orthogonality interval. The first entry
%   is the left bound and the second entry is the right bound.
%
% MDPC Library, v0.2
% Brian Templeton
% Created: in 2007
% Last Updated: July 29, 2009

if order < 0
    fprintf('\n ERROR: The variable "order" passed to function gen_pol_set
       is less than zero.');
    T = [];
    return
elseif mod(order,1) ~= 0
    fprintf('\n ERROR: The variable "order" passed to function gen_pol_set
       is not an integer.');
    T = [];
    return
end

syms x;

% Define Polynomials

if order == 0;
    T(1) = 1;
else
    % Updated July 12, 2009 to fix bug when generating Jacobi with order=1;
    if strcmp(set(1:4), 'Jaco')
        T(2) = simplify(1/2*(2*(a+1)+(a+b+2)*(x-1)));
    else  % all others, the 1th (starting with 0th) term is 'x'
        T(2) = x;
    end
```

```
52      T(1) = 1;
53      for n = 1:(order-1)
54          ni = n+1;  %adjusts for matlab starting arrays at index 1
55          if strcmp(set(1:4), 'Lege')
56              % recurrence formula for Legendre polynomials
57              T(ni+1) = simplify((2*n+1)/(n+1)*x*T(ni)-n/(n+1)*T(ni-1));
58          elseif strcmp(set(1:4), 'Cheb')
59              if n==1 && a==2
60                  T(2) = 2*x;
61              end
62              % recurrence formula for Chebyshev polynomials
63              T(ni+1) = simplify(2*x*T(ni)-T(ni-1));
64          elseif strcmp(set(1:4), 'Herm')
65              % recurrence formula for Hermite polynomials
66              T(ni+1) = simplify(x*T(ni)-n*T(ni-1));
67          elseif strcmp(set(1:4), 'Jaco')
68              % recurrence formula for Jacobi Polynomials, parameters 'a' and
69              % 'b' are arguments for the function
70              T(ni+1) = simplify((2*n+a+b+1)*((2*n+a+b+2)*(2*n+a+b)*x+a^2-b
                    ^2)/(2*(n+1)*(n+a+b+1)*(2*n+a+b))*T(ni) - 2*(n+a)*(n+b)*(2*n
                    +a+b+2)/(2*(n+1)*(n+a+b+1)*(2*n+a+b))*T(ni-1));
71          end
72      end
73  end
74
75  if strcmp(set(1:4), 'Lege')
76      w = 0*x+1/2; % make it symbolic
77      bounds = [-1,1];
78  elseif strcmp(set(1:4), 'Cheb')
79      if a == 1
80          w = 1/sqrt(1-x^2);  % need to be normalized
81      elseif a == 2;
82          w = sqrt(1-x^2);  % need to be normalized
83      end
84      bounds = [-1, 1];
85  elseif strcmp(set(1:4), 'Herm')
86      w = exp(-(x^2/2))/sqrt(2*pi);
87      bounds = [-inf,inf];
88  elseif strcmp(set(1:4), 'Jaco')
89      w = (1-x)^a*(1+x)^b*gamma(a+b+2)/(gamma(a+1)*gamma(b+1)*2^(a+b+1));
90      bounds = [-1,1];
91  end
```

A.3 Multi-Dimensional Library Files

The libraries in this section allow for a convenient way to implement Galerkin projections on a state space system with parameterized matrices, using the form derived in Chapter 5. This includes functions for parsing parameterized matrix expressions; easily constructing structures for multi-dimensional polynomials (Section 2.7), associated weights, relevant inner products, and a map back to the constituent single-dimensional polynomial sets; generating "$\mathbf{F}^i$" matrices (Equation (5.17)); and calculating mean and variance/covariance trajectories (Equation (2.23)).

```
function cX = cov_states(X,L,mddelta2)
% Find the covariances of a polynomial chaos simulation. This function
% assumes a specific arrangement of the expanded states.
%
% Make sure that progressing along a column is progressing through time!
%
% function cX = cov_states(X,L,mddelta2)
%
% Function Arguments:
% X: matrix for simulation of expanded states
% L: Order of the Polynomial Chaos expansion
% mddelta2: a vector of inner products for the multi-dimensional (or single
%    dimensional) basis functions.
%
% Function Outputs:
% cX: a matrix containing the covariances of the states. Note that the
%    symmetric portion is not calculated, so part of it contains zeros.
%
% MDPC Library, v0.3
% Brian Templeton
% Created: June 23, 2008
% Last Updated: July 17, 2008

[rows,cols] = size(X);

num_states = cols/(L+1);
state_ind = num_states:num_states:(cols-num_states);
%mir = mil + L;
%mil = mil + 1; % the zeroth index is left off for all states

% initialize
cX = zeros(rows,num_states,num_states);

```

```
% loop through states
for cc = 1:num_states
    % loop through states again (triangular grid so that states aren't
    % repeated)
    for dd = 1:cc
        cX(:,cc,dd) = (X(:,(state_ind+cc)).*X(:,(state_ind+dd)))*mddelta2
            (2:(L+1));
    end
end
```

```
function [basis_map, L] = create_basis_map(r,L1D)
% This function creates a 1:1 mapping for multidimensional basis functions
% in terms of 1D basis functions. It is constructed according to the Askey-
% scheme.
%
% [basis_map, L] = create_basis_map(r,L1D)
%
% Function Arguments:
% r: number of 1D basis functions
% L1D: maximum order of 1D basis functions
%
% Function Outputs:
% basis_map: A 2D matrix arranged such that each row has the indices for
%    each of the 1D basis functions
% L: The order of the multi-dimensional basis functions
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 19, 2008
% Last Updated: June 19, 2008

% The order of the basis functions
L = factorial(r+L1D)/(factorial(r)*factorial(L1D))-1;

% initialize the variable to save the map
basis_map = zeros(L, r);

% a temporary vector used in generating the map
temp = zeros(1, r);

for ic = 2:(L+1)
    if sum(temp) < L1D
        temp(r) = temp(r) + 1;
    else
        for dc = (r):-1:1
            if temp(dc) ~= 0
                temp(dc) = 0;
                temp(dc-1) = temp(dc-1)+1;
                break;
            end
```

```
        end
    end
    basis_map(ic,:) = temp;
end
```

```
function [E] = create_realiz_m(ps,num_states,rand_var_vec)
% This function creates a 'Realization' matrix for a Polynomial Chaos
% System
%
% function [E] = create_realiz_m(ps,num_states,rand_var_vec)
%
% Function Arguments
% ps: a structure containing information about the polynomial sets
% num_states: the number of states
% rand_var_vec: a vector containing the values of the random parameters for
%   the realizations
%
% Function Outputs
% E: Realization matrix
%
% MDPC Library, v0.3
% Brian Templeton
% Created: June 23, 2008
% Last Updated: July 17, 2008

% Evaluate the multidimensional polynomial
pvmd = eval_md_poly(ps,rand_var_vec);

temp = zeros(ps.L_p1,1); temp(1) = 1;
E = kron(pvmd.',eye(num_states));
```

```
function d_s_m = det_state_matrix(pa,matrix_s,pv1d)
% This function will evaluate passed matrix strings in order to create a
% deterministic realizations for the matrices.
%
% function d_s_m = det_matrix_s(pa,matrix_s,pv1d)
%
% Function Arguments
% pa: a structure containing information about the parameters
% matrix_s: passed matrix strings with parameters
% pv1d: an array containing evaluated 1D polynoials
%
% Function Outputs
% d_s_m: the deterministic realizations of the matrices
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 23, 2008
```

```
% Last Updated: June 23, 2008

% Find the values of the parameters
par_names = fieldnames(pa);
for cc = 1:length(fieldnames(pa));
    pname = cell2mat(par_names(cc));
    % put into variable for processing without parsing
    eval(sprintf('par_data = pa.%s;',pname));
    if isfield(par_data,'rv')
        if ~isempty(par_data.rv)
            rv = par_data.rv;
            lp = length(par_data.val);
            par_real = par_data.val*pv1d(rv,(1:lp)).';
        else
            par_real = par_data.val(1);
        end
    else
        par_real = par_data.val(1);
    end
    eval(sprintf('%s = par_real;',pname));
end

% Get the matrix names from the passed structure
matrix_names = fieldnames(matrix_s);

% Number of matrices passed in structure
num_matrix = length(matrix_names);

for cc = 1:num_matrix
    % Find the string for the matrix name in the struct
    matrix_name = cell2mat(matrix_names(cc));
    % Evaluate the deterministic realization of the matrices
    eval(sprintf('d_s_m.%s = eval(matrix_s.%s);', matrix_name, matrix_name)
        );
end
```

```
function [pv1d] = eval_1d_polys(ps,rand_var_vec)
% This function evaluates the 1d polynomial sets for a passed vector of
% values
%
% function [pv1d] = eval_1d_polys(ps,rand_var_vec)
%
% Function Arguments
% ps: contains information about the polynomial sets
% rand_var_vec: a 'realization' vector at which the polynomials sets are
%    evaluated
%
% Function Outputs
% pv1d = a matrix with the evaluated polynomial sets along rows, with the
%    rows indexed by random variable
```

```
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 23, 2008
% Last Updated: April 11, 2009

% Wrong could have correlated random variables!
%if length(rand_var_vec) ~= ps.r
%    error('An inconsistant number of random variables was passed.');
%end

% initialize
pv1d = zeros(ps.r,ps.L1D_p1);

for cc = 1:ps.r
    x = rand_var_vec(cc);  % obtain the value for the random variable
    pv1d(cc,:) = eval(ps.poly(cc).T);   % evaluate the symbolic polynomials
end
```

```
function [pvmd,pv1d] = eval_md_poly(ps,rand_var_vec)
% This function evaluates the multi-variable polynomial sets for a
% passed vector of values
%
% function [pvmd,pv1d] = eval_md_poly(ps,rand_var_vec)
%
% Function Arguments
% ps: contains information about the polynomial sets
% rand_var_vec: a 'realization' vector at which the polynomials sets are
%    evaluated
%
% Function Outputs
% pvmd: an array containing the evaluated multi-variable polynomial set
% pv1d: a matrix with the evaluated polynomial sets along rows, with the
%    rows indexed by random variable
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 23, 2008
% Last Updated: June 23, 2008

% initialize
pvmd = ones(ps.L_p1,1);

% evaluate the 1D polynomials
pv1d = eval_1d_polys(ps,rand_var_vec);

% loop along the indices of the multi-dimensional basis function
for cc = 1:ps.L_p1
    % go through every single dimensional basis function
    for dd = 1:ps.r
```

```
        pvmd(cc) = pvmd(cc)*pv1d(dd,ps.md_map_p1(cc,dd));
    end
end
```

```
function [e_s_m] = exp_state_matrix(ue_sm,pa,ps)
% This is a function to expand passed state matrices into expanded form.
%
% function [e_s_m] = exp_state_matrix(ue_sm,pa,ps)
%
% Function Arguments
% ue_sm: A structure containing the unexpanded state matrices in string
%    form, including parameter names
% pa: A structure containing the data for parameters
% ps: A structure containing the data for the multidimensional polynomial
%     chaos basis functions
%
% Function Outputs
% e_s_m: A structure containing the expanded state matrices
%
% MDPC Library, v0.3
% Brian Templeton
% Created: June 20, 2008
% Last Updated: July 17, 2008

% Find the number of possible random variables
num_rv = length(ps.poly);

% Pad the length of the parameter arrays of pa with zeros
pa = process_pa(ps,pa,num_rv,ps.L_p1);

% Get the matrix names from the passed structure
matrix_names = fieldnames(ue_sm);

% Number of matrices passed in structure
num_matrix = length(matrix_names);

for cc = 1:num_matrix
    % Find the string for the matrix name in the struct
    matrix_name = cell2mat(matrix_names(cc));
    % copy string for processing to a new struct
    eval(sprintf('pr_st.%s.ms = ue_sm.%s;', matrix_name, matrix_name));
    % parse the matrix for processing--loading into an array of structures
    % so that many following statements do not need to be parsed
    [M_str(cc).M_s_exp, M_str(cc).dim] = parse_pce_matrix(eval(sprintf('
        ue_sm.%s',matrix_name)),pa);
    % Copy parsed string so that it can be outputted
    % eval(sprintf('e_s_m.raw.%s.parse_s = M_str(cc).M_s_exp;',matrix_name)
        );
    % eval(sprintf('e_s_m.raw.%s.dim = M_str(cc).dim;',matrix_name));

```

```
    % initialize matrices
    M_str(cc).M_exp = zeros(M_str(cc).dim*ps.L_p1);
end


for ic = 0:ps.L
    F_ic = mddelta3_slice(ps,ic);
    for cc = 1:num_matrix
        M_str(cc).M_ic = ppcem2m(M_str(cc).M_s_exp,pa,ic);  % looks like it
            could be simplified to eliminate a function
        M_str(cc).M_exp = M_str(cc).M_exp + kron(F_ic,M_str(cc).M_ic);
    end
end

% Convert back to a structure for outputting
for cc = 1:num_matrix
    matrix_name = cell2mat(matrix_names(cc));
    eval(sprintf('e_s_m.%s = M_str(cc).M_exp;',matrix_name));
end
```

```
function [ps] = gen_basis_struct(ps)
% Function to find constants associated with defined basis functions and
% then add them to the passed structure
%
% function [ps] = gen_basis_struct(ps)
%
% Function Arguments:
% ps: A structure defined as follows-
%    ps.poly(n).name: every n-th entry defines a four letter abreviation
%       for a polynomial set, options are 'Jaco','Lege','Herm', and 'Cheb'
%    ps.poly(n).a: required for Jacobi polynomials, 'a' argument
%       Also required for Chebyshev polynomials, a=1 specifies first kind,
%       a=2 specifies second kind
%    ps.poly(n).b: required for Jacobi polynomials, 'b' argument
%    ps.L1D: Maximum order for each of the 1D polynomial sets, starting at
%       zero.
%
% Function Output:
% ps: A structure that includes everything in the inputted structure and:
%    ps.poly(n).T: array of symbolic polynomials of specified set
%    ps.poly(n).w: symbolic weighting function for specified set
%    ps.poly(n).bounds: bounds for specified set
%    ps.poly(n).delta2: 1D matrix of inner products for specified set
%    ps.poly(n).delta3: 3D matrix for specified set
%    ps.delta2: 1D matrix of inner products for multidimensional set
%    ps.L: Order of multidimensional set
%    ps.L_p1: Order of multidimensional set, plus 1
%    ps.L1D_p1: ps.L1D, plus 1
%    ps.md_map: An 1:1 mapping of new indices to 1D basis function indices
%       The rows are the new indices for the multidimensional set
```

```
%       The n-th column of each of the rows contains the indice for the old
%       n-th 1D set
%    ps.md_map_p1: Same as ps.md_map, except all of the values have been
%       increased by one such that the index notation matches Matlab's
%       array notation--i.e. indices start at 1 instead of 0.
%    ps.r: number of dimensions in multidimension basis function
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 19, 2008
% Last Updated: June 19, 2008

% Number of dimensions in multidimensional basis function
ps.r = length(ps.poly);

% Generate the multidimensional 1:1 map to the multidimensional basis
% functions
[ps.md_map, ps.L] = create_basis_map(ps.r, ps.L1D);

% Add one so that polynomials indices match with Matlab's array indices
% i.e. p1 = 'plus 1'
ps.md_map_p1 = ps.md_map + 1;
ps.L_p1 = ps.L + 1;
ps.L1D_p1 = ps.L1D + 1;

% If there are not optional arguments declared, make blank optional args
% This could be broken in future Matlab implimentations, but doesn't seem
% likely.
if ~isfield(ps.poly(1),'a')
    ps.poly(1).a = [];
end
if ~isfield(ps.poly(1),'b')
    ps.poly(1).b = [];
end

% initialize the variable
% The memory impact of storing a multi-dimensional dirac delta of two
% indices, mddelta2 is small, since it is a 1D array (orthogonality),
% size L
% However, the 3 indice version could be huge, size L^3, and so should not
% be stored
% Parts can be generated on demand from the much smaller LD1^3 arrays for
% the individual 1D basis functions
ps.mddelta2 = ones(ps.L_p1,1);

for pc = 1:ps.r
    [ps.poly(pc).T,ps.poly(pc).w,x,ps.poly(pc).bounds] = gen_pol_set_w(
       ps.L1D, ps.poly(pc).name, ps.poly(pc).a, ps.poly(pc).b);
    ps.poly(pc).delta3 = gen_kron_delta(ps.poly(pc).T, ps.poly(pc).w, x,
       ps.poly(pc).bounds, 3, 'dbl');
    ps.poly(pc).delta2 = diag(ps.poly(pc).delta3(:,:,1));
```

```
    ps.mddelta2 = ps.mddelta2.*ps.poly(pc).delta2(ps.md_map_p1(:,pc));
end
```

```
function [F] = kron_matrix(delta3sl,delta2)
% Generate a special matrix of inner products.
% Specify inner products
%
% function [F] = kron_matrix(delta3sl,delta2)
% Function Arguments:
% delta3sl: A 2D matrix slice of 3D matrix delta3
% delta2: An 1D column vector of inner products for basis functions
%
% Function Outputs:
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 18, 2008
% Last Updated: June 19, 2008

F = delta3sl./krons(ones(1,length(delta2)),delta2);
```

```
function [kslice] = mddelta3_slice(ps,kc)
% Generate a 2D matrix, with the third index kc, of the 3D delta3 matrix
% for a muldimensional set.
%
% function [kslice] = mddelta3_slice(ps,kc)
%
% Function Arguments:
% ps: The structure for the multidimensional polynomial
% kc: The third indice--held constant for the slice
%
% Function Outputs
% kslice: The 2D slice the 3D array for the multidimensional delta3 matrix
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 19, 2008
% Last Updated: June 20, 2008

kslice = ones(ps.L_p1);

for ic = 1:ps.L_p1
    for jc = 1:ps.L_p1
        for pc = 1:ps.r
            kslice(ic,jc) = kslice(ic,jc)*ps.poly(pc).delta3(ps.md_map_p1(
                ic,pc),ps.md_map_p1(jc,pc),ps.md_map_p1(kc+1,pc));
        end
    end
```

```
end

kslice = kslice./kron(ones(1,ps.L_p1),ps.mddelta2);
```

```
function mX = mean_states(X,L)
% Find the means of a polynomial chaos simulation. This function assumes
% a specific arrangement of the expanded states.
%
% Make sure that progressing along a column is progressing through time!
%
% function mX = mean_states(X,L)
%
% Function Arguments:
% X: matrix for simulation of expanded states
% L: Order of the Polynomial Chaos expansion
%
% Function Outputs:
% mX: a matrix containing the mean of the states
%
% MDPC Library, v0.3
% Brian Templeton
% Created: June 23, 2008
% Last Updated: July 17, 2008

% Numper of expanded states
Nexpstates = size(X,2);

L_p1 = L+1;

% Find the correct columns containing the mean information
mindex = 1:(Nexpstates/L_p1);

% Return the correct columns
mX = X(:,mindex);
```

```
function [A_parse, A_dim] = parse_pce_matrix(A,pa)
% Function to parse a string containing matrix information such that
% Polynomial Chaos random parameter coefficients can be properly filled
% into the matrix when there are multiple random variables.
%
% [A_parse, A_dim] = parse_pce_matrix(A,pa)
%
% Function Arguments:
% A: a string used to define a matrix in matlab notation. Can include
%    parameter names pa.parameter_name by using 'parameter_name' as an
%    entry.
% pa: parameter structure of a special form
%
```

```
% Function Outputs:
% A_parse: The parsed form of 'A' such that 'eval' can be used to evaluate
%    the matrix.
% A_dim: The dimensions of the matrix defined by the string.
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 19, 2008
% Last Updated: June 20, 2008


% copy string
A_parse = A;
A_parse2 = A;

rv_count = 0;
par_names = fieldnames(pa);
for cc = 1:length(par_names)
   var_string = strcat('pa.', cell2mat(par_names(cc)));
   par_string = strcat(var_string,'.val');
   if isfield(eval(var_string), 'rv')
       var_vr_val = eval(strcat(var_string, '.rv'));
       if ~isempty(var_vr_val) && var_vr_val > 0
           % At this point, the variable should be an expansion
           % par_string = strcat(par_string,'(rv_ind(',int2str(var_vr_val)
              ,'))');
           par_string = strcat(par_string,'(ic_p1)');
           rv_count = rv_count+1;
       elseif isfield(eval(var_string), 'cf')
           var_cf_val = eval(strcat(var_string, '.cf'));
           if var_cf_val == 0
               % only let the parameter be non-zero when ic==0
               par_string = strcat(par_string,'*(ic==0)');
           end
       end
   end
   A_parse = strrep(A_parse,par_names(cc),par_string);

   % Used to find the dimensions of the array. All variables replaced with
   % a number so that the string can be evaluated and the size found
   A_parse2 = strrep(A_parse2,par_names(cc),'1');
end

A_parse = cell2mat(A_parse);
A_dim = size(eval(cell2mat(A_parse2)));
```

```
function Aic = ppcem2m(A_parse,pa,ic)
% This is a function to evaluate a parsed string for a matrix
%
% function Aic = ppcem2m(A_parse,pa,ps,ic)
```

```
%
% Function Arguments
% A_parse: A string of parsed matrix information
% pa: A structure containing the information about parameters
% ic: This is an indice that corresponds to the numbering of the
%    set of multidimensional basis functions.
%
% Function Outputs
% Aic: The evaluated expanded matrix, at index ic.
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 19, 2008
% Last Updated: June 23, 2008


% Note: pa and rv_ind are used! They are contained in the evaluated string,
% A_parse

% Change the index to match Matlab's index notation
ic_p1 = ic + 1;

% Grab the correct column from the 1:1 map so that the correct 1D parameter
% expansion values correspond to the correct location in the
% multidimensional expansion
% rv_ind = ps.md_map_p1(ic_p1,:);

% Evaluate the string containing the matrix information
Aic = eval(A_parse);
```

```
function pa_new = process_pa(ps,pa,num_rv,L_p1)
% This is a function to process the parameter structure and check it for
% errors. If no errors are found, the structure is returned, with its
% arrays correctly padded with zeros.
%
% pa_new = process_pa(pa,num_rv,L_p1)
%
% Function Arguments
% ps: A structure containing information about the muldimensional
%    polynomial set
% pa: A structure containing information about the random parameters
% num_rv: The number of independent random variables
% L_p1: The order of the multivariable expansion, plus 1
%
% Function Outputs
% pa_new: The processed parameter structures
%
% MDPC Library, v0.2
% Brian Templeton
% Created: June 20, 2008
```

```
% Last Updated: June 23, 2008

% Get the variable names in the structure
var_names = fieldnames(pa);

for cc = 1:length(var_names)
    % Variable string
    vn = cell2mat(var_names(cc));
    % If the variable for the random variable is not declare, declare it as
    % an empty element. This avoids errors.
    if ~isfield(eval(strcat('pa.',vn)),'rv')
        eval(strcat('pa.',vn,'.rv=[];'));
    end
    % If the variable for the values of the expansion for the random
    % variable are not declared, declare them as an empty element. This
    % avoids errors.
    if ~isfield(eval(strcat('pa.',vn)),'val')
        eval(strcat('pa.',vn,'.val=[];'));
    end
    % Contents of rv part of structure
    rv_val = eval(strcat('pa.',vn,'.rv'));
    % Length of val array
    val = eval(strcat('pa.',vn,'.val'));
    val_length = length(val);
    % Check if there are values for the parameter. If not, give a warning
    % that it is not a valid parameter and don't use it.
    if isempty(val_length)
        warning('MDPC:emptyVal','pa.%s.val is empty and thus pa.%s will not
            be processed',rv,rv);
    else
        % If there is no random variable associated with an expansion and
        % the expansion is not a single scalar, give a warning and only use
        % the first value of the array.
        if isempty(rv_val)
            if val_length > 1
                warning('MDPC:extraValEnt','pa.%s is not associated with a
                    random variable and so only pa.%s.val(1) will be used.',
                    vn,vn);
                eval(strcat('pa.',vn,'.val = pa.',vn,'.val(1);'));
            end
        else
            % Check to make sure the random variable has a valid index.
            % I.e. it must match up with the bounds of the
            % single-dimensional random variables
            if rv_val < 1 | rv_val > num_rv
                error('pa.%s.rv has been declared outside of the bounds of
                    1:%d for the random variables.', vn,num_rv);
            % Everything seems to be correct, make sure the array is the
            % correct length
            else
                % pad the array
```

```
68                  new_val = zeros(1,L_p1);
69                  for dd = 1:L_p1
70                      cmparray = ones(1,(ps.r-1));
71                      if isequal(cmparray, ps.md_map_p1(dd,[1:(rv_val-1),(
                            rv_val+1):ps.r]))
72                          mv_ind = ps.md_map_p1(dd,rv_val);
73                          if mv_ind <= val_length
74                              new_val(dd) = val(mv_ind);
75                          end
76                      end
77                  end
78                  eval(strcat('pa.',vn,'.val = new_val;'));
79              end
80          end
81          % Copy the parameter to a new structure. This allows getting rid of
82          % invalid parameters.
83          eval(strcat('pa_new.',vn,'=pa.',vn,';'));
84      end
85  end
```