ABSTRACT

Experimental Investigation of a Time Scales-Based Stability Criterion over Finite Time Horizons

Matthew D. Mosley, M.S.E.C.E

Mentor: Ian A. Gravagne, Ph.D.

Feedback control systems that employ large area networks or other unpredictable or unreliable communications protocols between sensors, actuators, and controllers may experience non-uniform sampling characteristics. Previous work by Poulsen, et. al. [1] gives a criterion for exponential stability of non-uniformly discretized feedback control systems, assuming sample periods drawn from a known statistical distribution. However, the given stability theorem assumes an infinite time horizon. This work therefore examines the exponential stability criterion experimentally, over a finite time horizon, on a 2nd-order servo mechanism as well as a system with higher-order dynamics.

Experimental Investigation of a Time Scales-Based Stability Criterion over Finite Time Horizons

by

Matthew D. Mosley, B.S.E.C.E

A Thesis

Approved by the Department of Electrical and Computer Engineering

____

Kwang Y. Lee, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of

Masters of Science in Electrical and Computer Engineering

Approved by the Thesis Committee

____

Ian A. Gravagne, Ph.D., Chairperson

____

John M. Davis, Ph.D.

____

Scott Koziol, Ph.D.

Accepted by the Graduate School

May 2017

____

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

Background

1.1 Motivation and Outline

When a linear, time invariant dynamical system on continuous time is digitally controlled via sample-and-hold with equally spaced samples, the equivalent discrete-time dynamical system is also linear and time invariant. However, if the underlying sample-and-hold time base is non-uniform the discretized system coefficients are sample-time dependent. Despite this additional layer of complexity, several stability results are given in the literature [1, 3–7], including a sufficient condition that ensures "exponential stability almost surely" if the sample times are drawn from distributions of known mean and variance. The limitation of these results is that their conclusions depend on the assumption of an infinite time horizon. Practical control designs must achieve good results over finite time horizons.

This thesis is organized as follows:[1]

Chapter One provides the necessary background material in control theory, stability theory and dynamic equations on time scales (DETS). We present a stability criterion that we later use in Chapters Two and Four to analyze the behavior of closed loop control systems on non-uniform time domains.

Chapter Two examines how an experimental 2nd-order servo system (Figure 2.1) behaves under state feedback control with stochastically generated sample times over a 15-second time horizon. We then compare the experimental results with the stability criterion.

Chapter Three is a derivation of the solution to the Euler-Bernoulli partial differential equation. This chapter builds physical and mathematical intuition for the

[1] Large portions of this work are taken from papers [8, 9] the author submitted to the ASME DSCC 2015 conference and the ASEE GSW 2016 conference.

model identified in Chapter Four.

Chapter Four presents the results of an experiment similar to the one described in Chapter Two. This experiment, however, uses a flexible beam instead of a rigid arm, making the dynamics more difficult for analysis.

1.2 Control Theory

Control theory is an interdisciplinary field of engineering and mathematics that describes the behavior of dynamical systems and there is a large body of literature on the subject [2, 10, 11]. In general, a system can be described using the following differential equation

$$\dot{x}(t) = f(x, t, u),$$

where x, t and u are the state, time, and input of the system, respectively and where $\dot{x} = \frac{\mathrm{d}x}{\mathrm{d}t}$. If f is linear, the following equation describes the dynamics of a single-input system and is known as the state space model,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \tag{1.1}$$

where $x : \mathbb{R} \to \mathbb{R}^n$, $A : \mathbb{R} \to \mathbb{R}^{n\times n}$ and $B : \mathbb{R} \to \mathbb{R}^{n\times m}$. The vector $x(t)$ represents the state and A describes the internal dynamics of the system. Notice the time dependence of the A and B matrices. In general, the dynamics of a system will depend on t. A time-invariant system is one whose A and B coefficients do not depend explicitly on time; that is, given an input $u(t)$ and an output $y(t)$, a time shifted input $u(t + \Delta t)$ will produce a time shifted output $y(t + \Delta t)$. In other words, time shift does not change the dynamics of a time-invariant system. The state space representation of an LTI system is as follows:

$$\dot{x}(t) = Ax(t) + Bu(t). \tag{1.2}$$

In reality, most systems are non-linear and a linear approximation must be determined for the region of operation [12]. Once a system has been modeled, we can use control theory techniques to force the system to produce a desired output, provided certain conditions are met. A system's control signal is often generated by a computer, meaning it changes at uniform, discrete times rather than continuously. In this case, we use the following difference equation [10] to approximate the system dynamics

$$\frac{x(nT+T)-x(nT)}{T} = Ax(nT) + Bu(nT), \tag{1.3}$$

where T is the time step size and n is a positive integer. Note that A and B in (1.3) are not the same as A and B in (1.2). But (1.3) is only applicable in cases where the control signal is applied at constant intervals. Of interest in this work are systems which produce (or sample) signals at irregular intervals. For example, unlike QNX and certain Linux distributions, Microsoft Windows is not a real time operating system, meaning that for many applications it cannot reliably produce a control signal with the minimum required latency to stabilize certain systems [6, 13]. Systems such as these can be modeled by a more general class of equations called dynamic equations on time scales. In fact, (1.2) and (1.3) are both special cases of this more general class. Time scales will be discussed more later.

To make the context of this paper more clear, we should distinguish between "hard" and "soft" real time systems. Hard real time means that the consequences of a missed timing deadline are unacceptable and thus, missing a timing deadline is considered a system failure. Examples of hard real time systems are pace-makers, air traffic control systems, etc. We are more concerned with soft real time systems; that is, systems whose timing requirements are not so strict as to preclude missing "some" deadlines. Failure to meet a limited number of timing deadlines results in a degradation of service quality, but not total failure. We are interested in how missing timing deadlines affects the *stability* of such a system, which basically means how "well behaved" the system is.

1.3 Stability Theory

In an LTI system on continuous time, stability can easily be determined by checking if the eigenvalues of the system matrix A of (1.2) are in the left half of the complex plane. But for nonlinear and time-variant systems, pole-placement arguments are no longer valid; some time-variant systems have constant poles in the right half plane and are stable, while other systems may have poles completely contained in the left half plane and yet be unstable. Furthermore, when considering an LTI system, stability is a trait of the system itself. But in the non-LTI case, a system may be stable at some points in the state space and yet unstable at others [2]. The concept of stability is much more complex when dealing with non-LTI systems. Here, we will discuss stability intuitively and then present a few more formal stability definitions.

Any point where the state will remain constant ($\dot{x} = 0$) is known as an equilibrium point. Points A - G in Figure 1.1 are all equilibrium points. These points can be classified as stable, neutral, or unstable. The ball in Figure 1.1 is currently at point C where it is free to roll in either direction. Points between B and D are considered neutrally stable because any small disturbance between those points will cause the ball to roll to a new position and stay there. In contrast, points A and F are called unstable equilibrium points because any slight perturbation will cause the ball to roll down the slope. Points E and G are stable points; if the ball is disturbed at one of these points, it will eventually return, provided the disturbance was not too great. Note that for linear systems, the origin is always an equilibrium point [2].

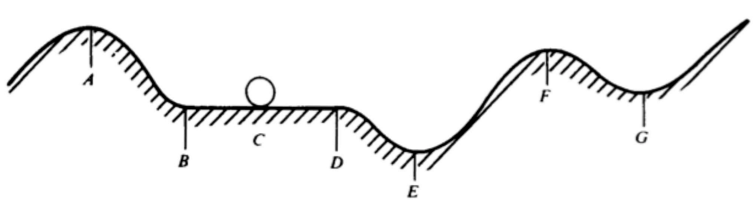


Figure 1.1. Various equilibrium points [2]

The following definitions [2, 12] assume that the equilibrium point is at the origin. If not, the equilibrium point can be translated to the origin through a change of coordinates.

Definition 1.1 (**Lyapunov Stability**). The origin is a stable equilibrium point if for any given value $\epsilon > 0$, there exists a number $\delta(\epsilon, t_0) > 0$ such that if $\|x(t_0)\| < \delta$, then the resultant motion $x(t)$ satisfies $\|(x(t)\| < \epsilon$ for all $t > t_0$.

This type of stability is also known as "start close, stay close" stability.

Definition 1.2 (**Asymptotic stability**). The origin is an asymptotically stable equilibrium point if it is Lyapunov stable and in addition, there exists a number $\delta'(t_0) > 0$ such that whenever $\|x(t_0)\| < \delta'(t_0)$ the resultant motion will satisfy $\lim_{t\to\infty} \|x(t)\| = 0$.

Definition 1.3 (**Exponential stability**). The origin is an exponentially stable equilibrium point if it is asymptotically stable and in addition, there exists $\alpha < 0$ and $C > 0$ such that $\|x(t)\| < C \exp(\alpha t)$ for all t.

When state feedback is used to control a system, this means that the state influences the control input so that the system input signal will be proportional to the discrepancy between the desired state and the current state. In many cases the desired state is identically zero, so we can express the input as $u(t) = -Kx(t)$, where K is commonly referred to as the gain matrix. This equation is called the *control law.* The state equation can now be rewritten as

$$\dot{x}(t) = [A - BK]x(t). \tag{1.4}$$

1.4 Dynamic Equations on Time Scales

We now turn our attention to DETS. Originally introduced in 1988 by Stefan Hilger in his Ph.D. dissertation [14], DETS is a branch of mathematics that allows for powerful analysis on systems whose domains are of non-uniform step size. Up until now, the systems we have presented have been either continuous or uniformly discrete. The theory of DETS allows for much more general analysis. It can describe systems that operate on a time scale of any general non-uniform time domain, be it continuous, discrete, multiples of $\mathbb{Z}$ (denoted by $h\mathbb{Z}$), non-uniform, or even stochastic [1]. A time scale, denoted by $\mathbb{T}$, is defined as any arbitrary nonempty closed subset of the real numbers [15]. Examples of time scales include $\mathbb{R}$, $h\mathbb{Z}$, $\mathbb{P}_{\alpha T}$ and $\mathbb{T}$.

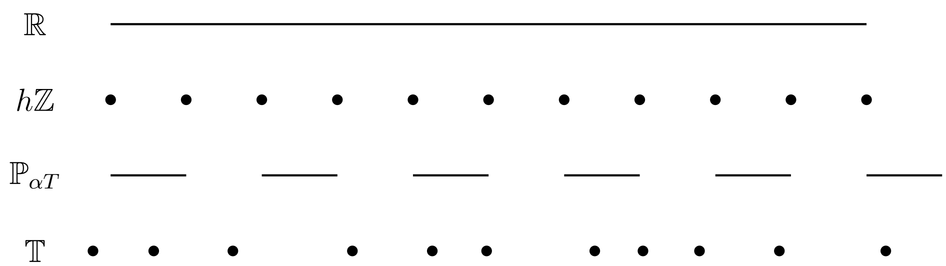


Figure 1.2. Examples of Time Scales

Given $t \in \mathbb{T}$, the successor of t is denoted by $\sigma(t)$ and is known as the forward jump operator. It is defined as

$$\sigma(t) := \inf \{x \in \mathbb{T} : s > t\}. \tag{1.5}$$

For example,

$$\sigma(1) = 2, \tag{1.6}$$

$$\sigma(2) = 3, \tag{1.7}$$

$$\vdots \tag{1.8}$$

$$\sigma(t) = t + 1. \tag{1.9}$$

Likewise the backward jump operator, $\rho(t)$, gives the previous value of t and is defined as

$$\rho(t) := \sup \{s \in \mathbb{T} : s < t\}. \tag{1.10}$$

The graininess function (step size) of a domain is defined as

$$\mu(t) := \sigma(t) - t. \tag{1.11}$$

In the cases of $\mathbb{T} = \mathbb{Z}$ and $\mathbb{T} = \mathbb{R}$, the graininess function simplifies to $\mu(t) \equiv 1$ and $\mu(t) \equiv 0$, respectively.

A modified definition of the derivative is required for time scales calculus:

$$f^{\Delta}(t) = \frac{f(t + \mu(t)) - f(t)}{\mu(t)}. \tag{1.12}$$

Notice that in the limit as $\mu(t)$ approaches zero, this is equivalent to the derivative on $\mathbb{R}$. The derivative on $\mathbb{Z}$ is found simply by using $\mu(t) \equiv 1$:

$$f^{\Delta}(t) = \frac{f(t+1) - f(t)}{1} = f(t+1) - f(t) = \Delta f(t), \tag{1.13}$$

where Δ is the usual forward difference operator.

We will now derive the product rule for time scales and then show that it is consistent with that of $\mathbb{R}$ and $\mathbb{Z}$. By definition of the derivative, we have

$$(f(t)g(t))^{\Delta} = \frac{f(t + \mu(t))g(t + \mu(t)) - f(t)g(t)}{\mu(t)}. \tag{1.14}$$

Adding and subtracting $f(t + \mu(t))g(t)$ gives us

$$(f(t)g(t))^{\Delta} = \frac{f(t + \mu(t))g(t + \mu(t)) - f(t + \mu(t))g(t) + f(t + \mu(t))g(t) - f(t)g(t)}{\mu(t)}, \tag{1.15}$$

which can be rewritten as

$$(f(t)g(t))^{\Delta} = \frac{f(t + \mu(t))\,[g(t + \mu(t)) - g(t)]}{\mu(t)} + \frac{g(t)[f(t + \mu(t)) - f(t)]}{\mu(t)}, \tag{1.16}$$

which simplifies to

$$(f(t)g(t))^{\Delta} = f(t + \mu(t))g^{\Delta}(t) + g(t)f^{\Delta}(t) = f(\sigma(t))g^{\Delta}(t) + g(t)f^{\Delta}(t). \tag{1.17}$$

Note that

$$f(\sigma(t))g^{\Delta}(t) + g(t)f^{\Delta}(t) = g(\sigma(t))f^{\Delta}(t) + f(t)g^{\Delta}(t). \tag{1.18}$$

The Hilger complex plane is important for analyzing regions of stability for systems operating on non-uniform time domains, and it comprises the following sets:

Definition 1.4. For $h > 0$, we define the following:

$$\text{The Hilger complex numbers: } \mathbb{C}_h := \left\{ z \in \mathbb{C} : z \neq -\frac{1}{h} \right\} \tag{1.19}$$

$$\text{The Hilger real axis: } \mathbb{R}_h := \left\{ z \in \mathbb{C}_h : z \in \mathbb{R} \text{ and } z > -\frac{1}{h} \right\} \tag{1.20}$$

$$\text{The Hilger alternating axis: } \mathbb{A}_h := \left\{ z \in \mathbb{C}_h : z \in \mathbb{R} \text{ and } z < -\frac{1}{h} \right\} \tag{1.21}$$

$$\text{The Hilger imaginary circle: } \mathbb{I}_h := \left\{ z \in \mathbb{C}_h : \left| z + \frac{1}{h} \right| = \frac{1}{h} \right\} \tag{1.22}$$

$$\tag{1.23}$$

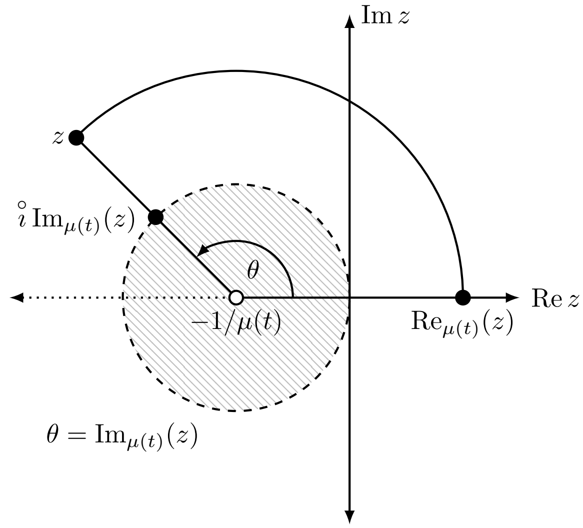


Figure 1.3. The Hilger complex plane

Definition 1.5. Let $h > 0$ and $z \in \mathbb{C}_h$. Define the Hilger real part of z as

$$\operatorname{Re}_h(z) := \frac{|zh+1| - 1}{h}. \tag{1.24}$$

and the Hilger imaginary part of z as

$$\operatorname{Im}_h(z) := \frac{\arg(zh+1)}{h}. \tag{1.25}$$

Definition 1.6. For $h > 0$, let $\mathbb{Z}_h$ be the strip

$$\mathbb{Z}_h := \left\{ z \in \mathbb{C} : -\frac{\pi}{h} < \operatorname{Im}(z) \leq \frac{\pi}{h} \right\}. \tag{1.26}$$

We can then define the cylinder transformation $\xi_h : \mathbb{C}_h \to \mathbb{Z}_h$ by

$$\xi_h(z) = \frac{1}{h}\mathrm{Log}(1 + zh). \tag{1.27}$$

The cylinder transform is a mapping from the Hilger complex plane to the strip $\mathbb{Z}_h$. Points that lie within the Hilger circle are mapped into the left half of $\mathbb{Z}_h$.

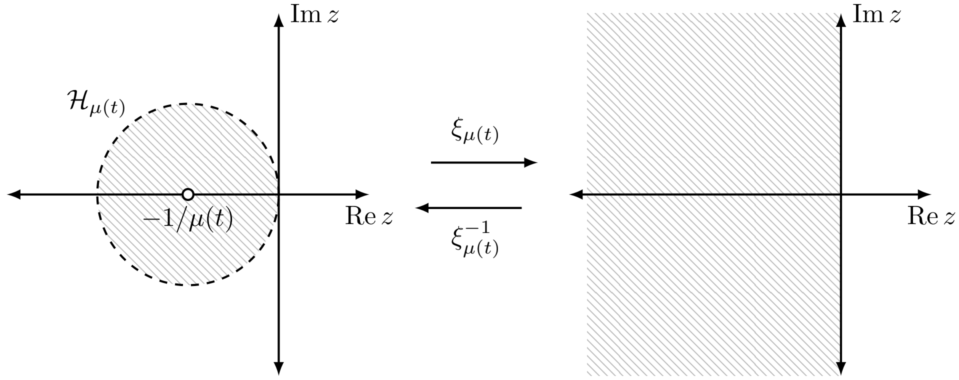


Figure 1.4. Cylinder Transform Mapping

1.5 Stability on Time Scales

We will now discuss recent contributions made to the theory of DETS. A fundamental question of DETS is what region of the complex plane results in stability of

$$x^{\Delta}(t) = \lambda x(t), \qquad t \in \mathbb{T},\ \lambda \in \mathbb{C}. \tag{1.28}$$

We will denote that region by

$$\mathcal{S}(\mathbb{T}) := \{\lambda \in \mathbb{C} \,|\, 1.28 \text{ is exponentially stable}\}. \tag{1.29}$$

Pötsche *et al.* have shown [3] that (1.28) is exponentially stable if $\lambda \in \mathcal{S}(\mathbb{T})$ where

$$\mathcal{S}(\mathbb{T}) = \{\lambda \in \mathbb{C} \,|\, \limsup_{T\to\infty} \frac{1}{T - t_0} \int_{t_o}^{T} \lim_{s \searrow \mu(t)} \frac{\ln|1 + s\lambda|}{s} \Delta t < 0\}. \tag{1.30}$$

In essence, (1.30) states that exponential stability requires that the average value of the cylinder transformation of the system eigenvalues must have a negative Hilger real part. This region is sufficient and necessary for exponential stability, but it is difficult to calculate in general.

As Gard *et al.* [16] have noted, an easily calculable subset of the region of exponential stability can be described as

$$\mathcal{H}_{min} := \left\{ z \in \mathbb{C}_{\mu_{max}} : |z + \frac{1}{\mu_{max}}| < \frac{1}{\mu_{max}} \right\}, \tag{1.31}$$

which follows directly from considering that the Hilger circle corresponding to the longest μ will be the smallest. This region is very conservative, however, and while it is sufficient for exponential stability, it is much smaller than necessary. To calculate $\mathcal{S}(\mathbb{T})$, Davis *et al.* have reduced (1.30) to the following more tractable inequality [17], which can handle discrete time scales and a finite number of graininesses:

$$\prod_{k=1}^{N} |1 + \mu_k \lambda|^{d_k} < 1, \tag{1.32}$$

where d_k is the weight corresponding to graininess μ_k. For example, if the time scale is periodic and has graininess

$$\{1, 2, 2, 1, 2, 2, \ldots\}, \tag{1.33}$$

then the graininesses are $\mu_1 = 1$ with $d_1 = 1$ and $\mu = 2$ with $d_2 = 2$. The region described by (1.32) can be regarded as a weighted geometric mean of the individual Hilger circles [17]. To see this, take the natural logarithm of (1.32). Using the rules of logarithms, we then have

$$\sum_{k=1}^{N} d_k \ln |1 + \mu_k \lambda| < 0. \tag{1.34}$$

Multiplying by μ_k/μ_k gives us

$$\sum_{k=1}^{N} d_k \mu_k \frac{\ln |1 + \mu_k \lambda|}{\mu_k} = \sum_{k=1}^{N} d_k \mu_k \xi_{\mu_k} < 0. \tag{1.35}$$

So criterion (1.32) is logically equivalent to a requirement that the average value of the cylinder transformation ξ_{μ_k} for each graininess, μ_k, has a negative Hilger real part. Poulsen has generalized [1] the criterion from (1.30) to one that applies to the stochastic μ-varying dynamic equation:

$$x^{\Delta} = \lambda(\mu(t))x. \tag{1.36}$$

By stochastic, we mean that the time scale is not known in advance. Therefore the criterion now contains a probability density function $f(\mu)$. And because the time scale is stochastic, we have to qualify what we mean by stability. We now say exponential stability *almost surely* because satisfaction of the criterion ensures stability with a probability equal to one. The criterion is as follows:

$$\int_0^T f(\mu) \ln |1 + \mu\lambda(\mu)| \mathrm{d}\mu < 0, \tag{1.37}$$

where f is the probability density function of the graininess μ and T is the upper limit of f. Note the similarity to (1.34). Indeed, the only differences are that

(1) the weight d_k is now a probability density function,

(2) λ may now depend on μ,

(3) and the summation is now an integral because we are assuming there are an infinite number (i.e. a continuous distribution) of graininesses.

We will use this criterion later in Chapters Two and Four to determine the stability of experimental systems for various probability density functions. It is important to note that this criterion is not associated with a static region of stability when λ varies according to μ.

Poulsen has also found the largest Hilger circle that will fit into $\mathcal{S}(\mathbb{T})$ [7]. The circle has curvature

$$\delta = \frac{E[\mu^2]}{E[u]} \tag{1.38}$$

in the stochastic case and

$$\delta = \limsup_{\tau \to \infty} \frac{\int_{t_o}^{\tau} \mu(s)\Delta(s)}{\tau - t_0},$$

in the deterministic case. Pole placement within this region, $\mathcal{H}_\delta$, is sufficient to ensure mean square exponential stability (MSES). It is known as the osculating circle because it is the largest circle that shares curvature with $\mathcal{S}(\mathbb{T})$.

For a concrete example of the regions of stability discussed in this chapter, consider a system $x^{\Delta}(t) = \lambda x(t)$, $t \in \mathbb{T}$ operating on a stochastic time scale $\mathbb{T}$ where the graininess is distributed according to a die roll. That is, let

$$\Pr(\mu_k = x) = \frac{1}{6}, \quad x \in \{1,2,3,4,5,6\} \tag{1.39}$$

where $\Pr(\mu_k = x)$ is the probability that the k^{th} graininess will be x seconds. We can generate the time scale by repeatedly rolling the die. For example, if a 3 is rolled then the next graininess is 3 seconds. Assuming we have a system operating on this time scale, the discussed stability regions can be found and are shown in Figure 1.5.

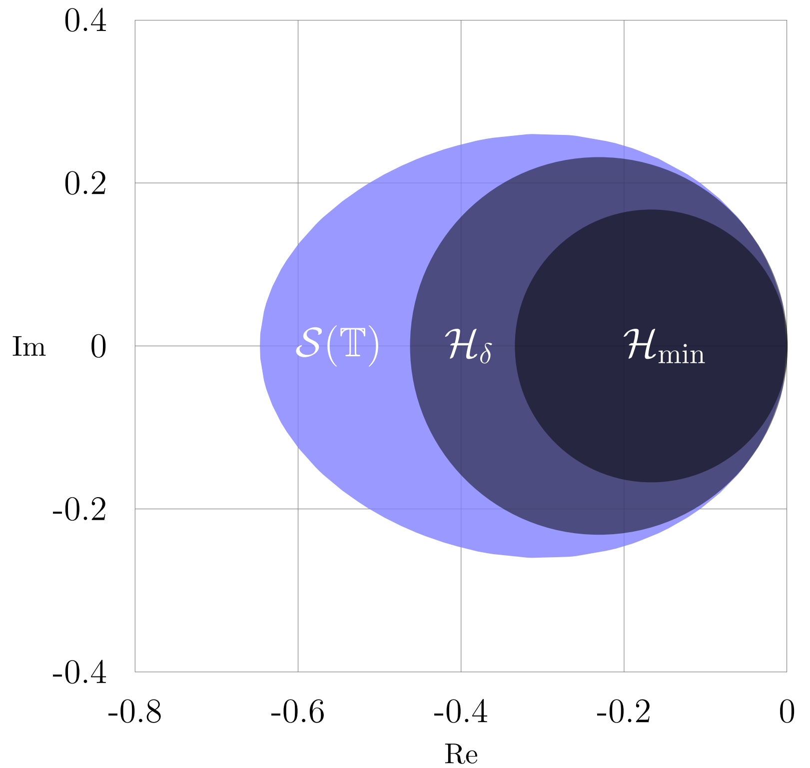


Figure 1.5. Regions of stability for the stochastic time scale

CHAPTER TWO

SRV-02 Experiment

This chapter published as: Matthew Mosley, Ian Gravagne, Dylan Poulsen, and John Davis. Experimental investigation of a time scales-based stability criterion over finite time horizons. In Proceedings of the ASME 2015 Dynamic Systems and Control Conference. American Society of Mechanical Engineers, ASME, October 2015.

In this chapter we examine how an experimental 2nd-order servo system (Figure 2.1) behaves under state feedback control with stochastically generated sample times over a 15-second time horizon. We then compare the experimental results with the stability criterion.

2.1 Overview of Experiment

The system is a Quanser Rotary Servo Base Unit (SRV02) connected to a PC running Windows 7 and QUARC® version 2.4, connected via an amplifier, a breakout board, and a Quanser Q4 card. Operation is as follows:

(1) The simulation is launched from Simulink.

(2) At $t = 1$, the servo begins moving the arm to a specified angle, using state feedback which is sampled on some time scale.

(3) The servo arm either

  (a) settles to the specified angle or

  (b) continues to oscillate for the duration of the experiment.

Intuitively we can predict that as the graininess is increased, the system will be less likely to stabilize. Eventually the graininess will increase to the point where the system arm will spin out of control without settling to the desired position. We ran two sets of experiments. The first was with constant graininess; the second was with graininess distributed according to beta and gamma distributions.

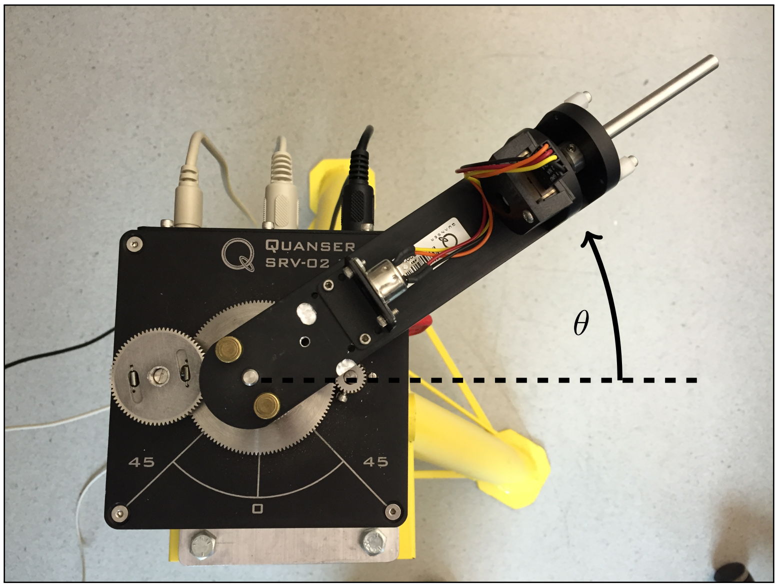


Figure 2.1. SRV-02

2.2 System Identification

To estimate the dynamics of this system on continuous time, we used MATLAB and Simulink to find the parameters of the following first order transfer function

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{b}{s+a}, \tag{2.1}$$

where $\dot{\theta}(s)$ is the angular velocity of the arm, and $V(s)$ is the input motor voltage.

We found the parameters to be $a = 6.598$ and $b = 11.37$. The system was then transformed into the following state space representation, with the states chosen as $x_1(t) = \theta(t)$ and $x_2(t) = \dot{\theta}(t)$.

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{2.2}$$

$$y(t) = Cx(t) \tag{2.3}$$

Figure 2.2. The amplifier

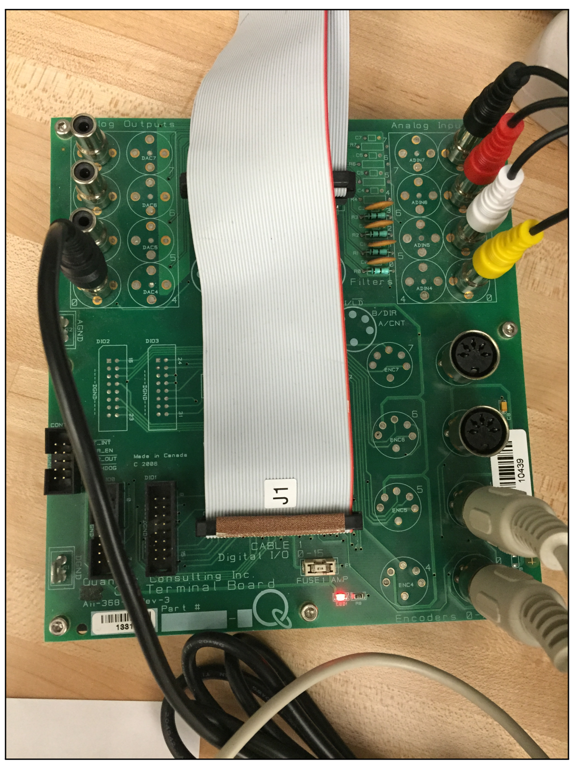


Figure 2.3. The breakout-board

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix}, \ B = \begin{bmatrix} 0 \\ b \end{bmatrix} \tag{2.4}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{2.5}$$

and $t \in \mathbb{R}$. Figure 2.4 shows the open loop step response of the system.

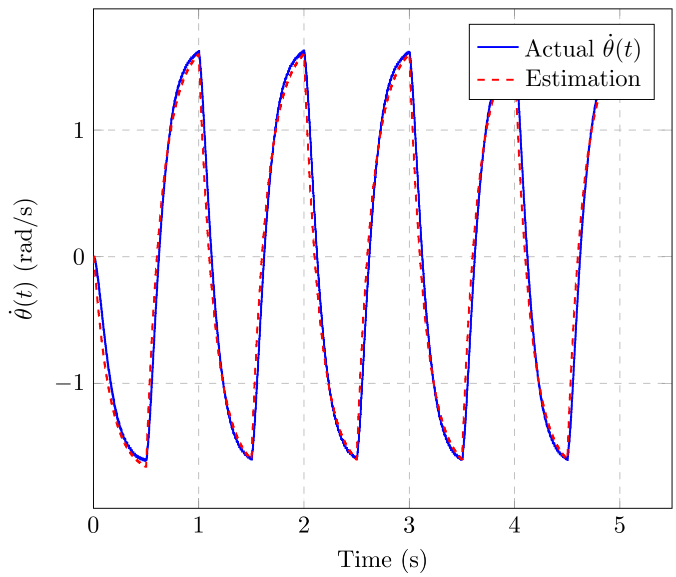


Figure 2.4. The open loop step response of the system in (2.2), simulated and actual.

2.3 Discretization

We want to investigate the stability of this system when the control signal graininess is distributed according to a known probability density function, including the case where the variance is zero.

First, we discretize the system onto a time scale, $\mathbb{T}$, using the control law

$$u(t) = Kx(t),$$

where $K = [-3.5 \ \ 0]$. As in [18], we define the discretized system as

$$x^{\Delta}(t) := [\mathcal{A}(\mu(t)) + \mathcal{B}(\mu(t))K]x(t), \ t \in \mathbb{T}, \tag{2.6}$$

where

$$\mathcal{A}(\mu(t)) = \text{expc}\,(\mu(t)A)A, \tag{2.7}$$

$$\mathcal{B}(\mu(t)) = \text{expc}\,(\mu(t)A)B, \tag{2.8}$$

where expc is the *exponent cardinal* [18] and is defined as

$$\text{expc}\,(X) := I + \frac{1}{2}X + \frac{1}{6}X^2 + ... + \frac{1}{n!}X^{n-1}, \tag{2.9}$$

which can be rewritten in the following closed form

$$\text{expc}\,(X) := (\exp\,(X) - I)X^{-1}, \tag{2.10}$$

provided that X^{-1} exists. Since A is singular, we appeal to the Cayley-Hamilton theorem to evaluate $\text{expc}\,(\mu(t)A)$ [2].

Theorem 2.1 (Cayley-Hamilton Theorem)**.** *Every matrix satisfies its own characteristic equation [2].*

Since $\text{expc}\,(z)$ is analytic, it can be expressed as a power series and its terms regrouped such that

$$\text{expc}\,(z) = \Delta(z)\sum_{k=0}^{\infty}\beta_k z^k + R(z), \tag{2.11}$$

where $\Delta(z)$ is the characteristic equation and R is a polynomial of degree less than or equal to $n-1$. Substituting $\mu(t)A$ for z and using that $\Delta(\mu(t)A) = 0$, we have

$$\text{expc}\,(\mu(t)A) = R(\mu(t)A) = \alpha_0 + \alpha_1(t)\mu(t)A. \tag{2.12}$$

We can solve for the coefficients α_0 and α_1 by using

$$\text{expc}\,(\mu(t)\lambda_i) = R(\mu(t)\lambda_i) = \alpha_0 + \alpha_1(t)\mu(t)\lambda_i, \tag{2.13}$$

where λ_i are the eigenvalues of the system matrix A. We can now compute $\mathcal{A}$ and $\mathcal{B}$ for the discretized system. Note that in general, the coefficient $\alpha_1(t)$ is a function of time.

When μ is constant, the discretized system loses its time-dependency:

$$\mathcal{A} = \operatorname{expc}(\mu A) A \tag{2.14}$$

$$\mathcal{B} = \operatorname{expc}(\mu A) B \tag{2.15}$$

Experiments were conducted on the SRV-02 servo at constant graininess. At a relatively low constant graininess, the system displays some overshoot (Figure 2.5), but the system is stable[1] since the poles are contained within the Hilger circle [15]. As the graininess increases, the complex poles approach the real axis and eventually converge, after which one pole leaves the Hilger circle, resulting in system instability (Figure 2.6). The maximum graininess for which the system is stable is 630 ms, where one of the system poles lies on the edge of the Hilger circle.

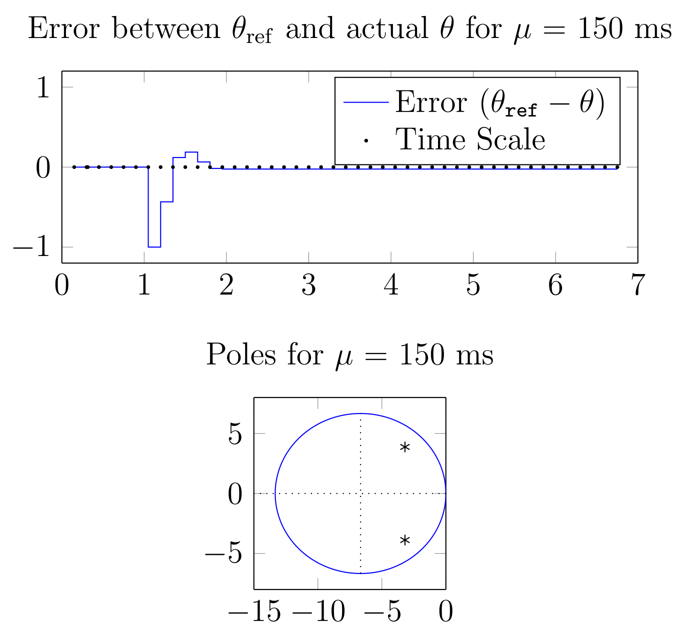


Figure 2.5. At $\mu = 150$ ms, there is little oscillation and the response is stable since the poles are inside the Hilger circle. As μ increases, the poles move nearer to the edge of the circle.

[1] We recognize that the mathematical definition of stability would require observation for infinite time. But since this is an experiment, we use a finite time horizon of 15s, after which we make judgments about stability of the system response.

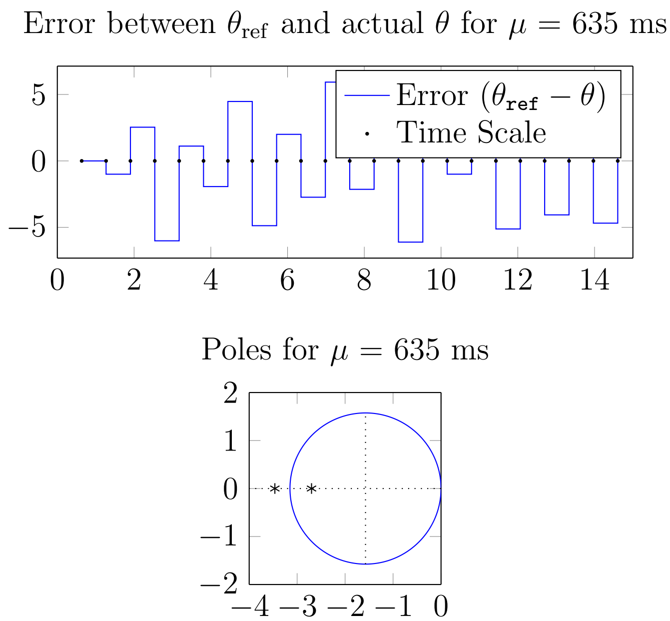


Figure 2.6. At $\mu = 635$ ms, one pole is outside the Hilger circle, so the system is now unstable (the response oscillates back and forth throughout the entire time horizon).

2.5 Time Scales with Non-Uniform Graininess

In order to operate the system on a non-uniform discrete time scale, we placed the Simulink system model (Figure 2.7) inside of a triggered subsystem block (Figure 2.8). The "From Workspace" block in Figure 2.8 contains a timeseries object that stores a sequence of numbers that represent points in time at which the subsystem is triggered. The grainsinesses of this time scale are generated by standard MATLAB functions for beta and gamma distributions. This timeseries object that contains the time scale is created before the system is run. This way, the statistics of the time scale can be checked and compared to the desired statistics before system operation.

As mentioned before, we cannot appeal to a static region of stability if we allow the graininess $\mu(t)$ to be time-dependent. But if $\mu(t)$ is a random variable from a known continuous statistical distribution, we can make use of the criterion mentioned in Chapter One to determine whether a distribution with a given mean and variance will stabilize the closed loop system. Restated here, the criterion says that the equilibrium

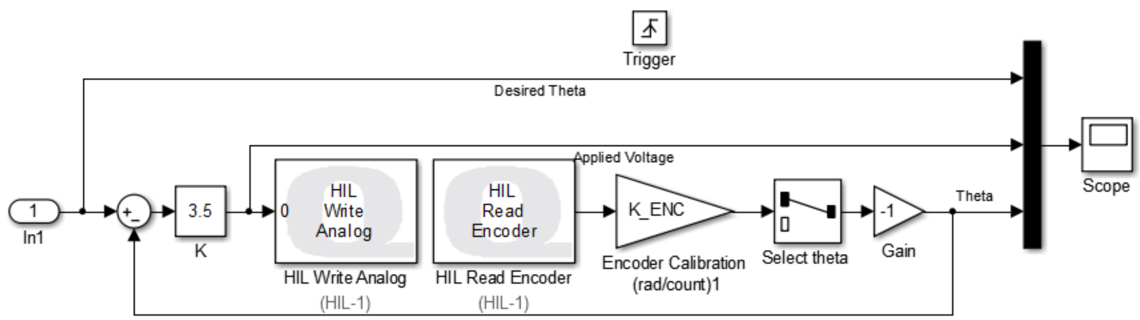


Figure 2.7. Simulink block diagram subsystem

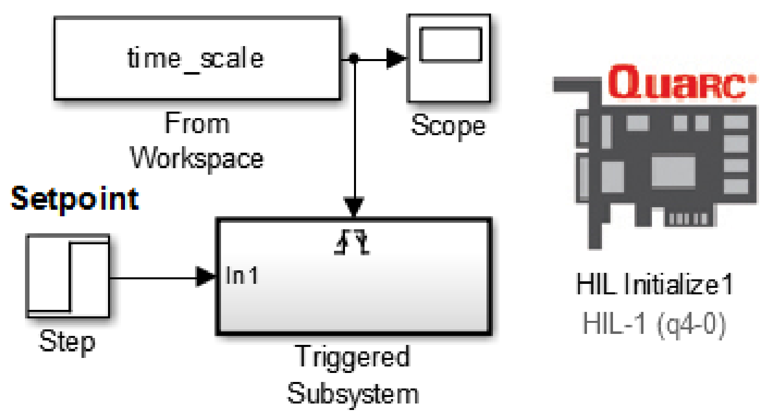


Figure 2.8. The "Triggered Subsystem" block contains the model shown in Figure 2.7. The "time_scale" block contains an object representing the points in time at which the subsystem is triggered.

of (2.6) will be exponentially stable if

$$\max_i \int_0^T f(\mu) \log |1 + \mu\lambda_i(\mu)| \mathrm{d}\mu < 0, \tag{2.16}$$

where $f(\mu)$ is the probability density function by which μ is distributed, and $\lambda_i(\mu)$ is the i_{th} eigenvalue of the system matrix, $[\mathcal{A}(\mu) + \mathcal{B}(\mu)K]$. Note that $T = 1$ for beta distributions and $T = \infty$ for gamma distributions. We use an integral here instead of a summation, even though we have a finite number of graininesses. For our purposes this is justifiable because the number of possible graininesses is high enough so that their distribution can be approximated continuously. In this work, we will investigate time scales with $\mu(t)$ drawn from beta and gamma distributions. The criterion of (2.16) only ensures "exponential stability almost surely"[1]. This is a relatively weak stability criterion and does not guarantee that the error will settle immediately.

Mathematica was used to compute the left hand side of (2.16) for beta and gamma distributions, where the mean was held at 400 ms and the variance was varied from 0 to 0.070 s^2. The values of the LHS of (2.16) are displayed in Figure 2.9. As can be seen from this plot, systems whose μ's fall on beta and gamma distributions become unstable at variances of 0.052 s^2 and 0.063 s^2, respectively. However, since we are only looking at a finite time horizon, this does not imply that every experimental trial below these critical values will exhibit convergent behavior. Likewise, not every trial with variance greater than these values will destabilize.

To investigate exponential stability on a time scale experimentally, we

(1) Initialize 100 randomly generated time scales for a given distribution with a given mean and variance.

(2) Apply a step input to the system and observe the step response error for each time scale.

(3) Record the proportion of failures vs. successes.[2]

The results of this experiment are summarized in Figure 2.10. The success rate for a beta distribution with a variance of 0.052 s^2 (the critical value) is about 80%. In [19], it is shown that the SRV-02 system is guaranteed mean square exponential stability up until a variance of 0.020 s^2. Mean square exponential stability is a much stronger stability criterion based on Lyapunov theory, and therefore it is an indicator of greater performance. Indeed, Figure 2.10 indicates that 98% of the simulations succeed when the variance is 0.020 s^2. See Figures 2.11 and 2.12 for a comparison between the experimental results and the output of the stability criterion.

The plots in Appendix A result from applying a step input to the system for five different time scales, in which the graininess is distributed according to a beta distribution with a mean of 400 ms and a given variance. When the variance is

[2] An individual trial is deemed a success when the last three samples of the time scale have less than 5% error.

$0.01\ s^2$, the responses are consistently stable. Increasing the variance to $0.08\ s^2$ results in approximately half the responses being stable, whereas only about 20% of step responses are stable at a variance of $0.12\ s^2$.

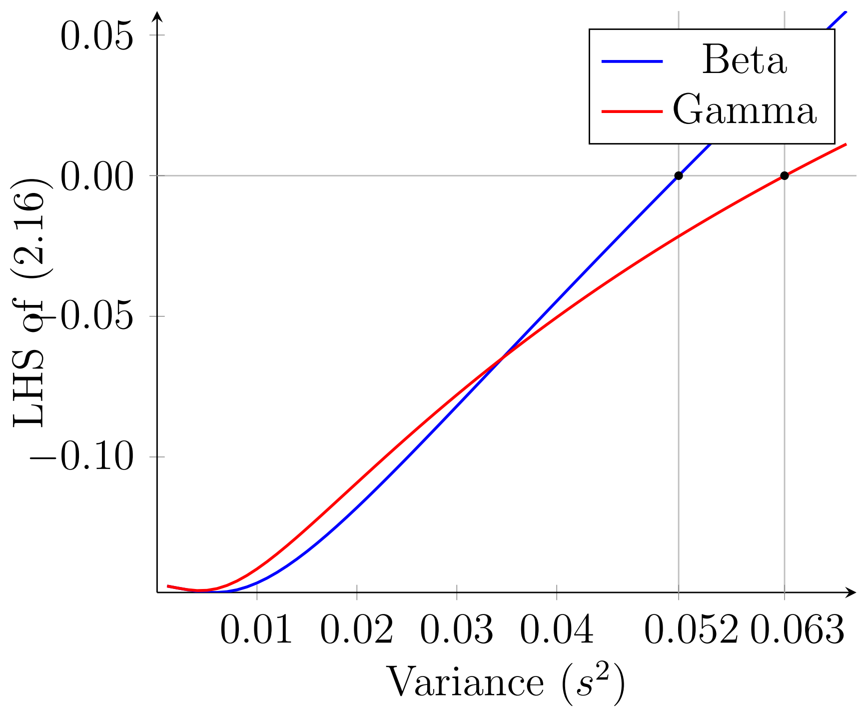


Figure 2.9. The left hand side of (2.16) for beta and gamma distributions

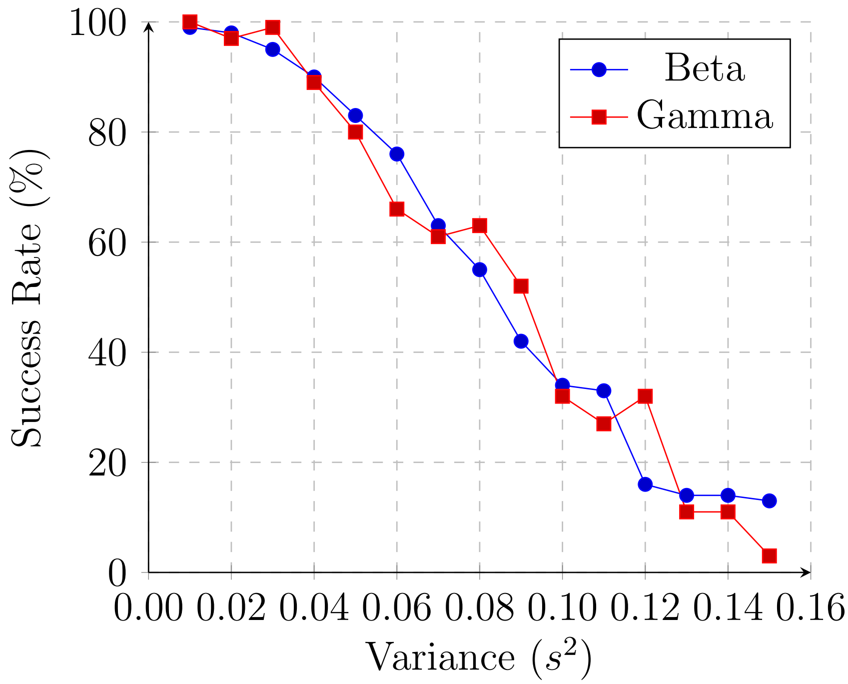


Figure 2.10. 15-second stabilization success rates (percentage of trials that stabilized) for the closed-loop servo system on beta and gamma distributions with mean graininess of 400 ms.

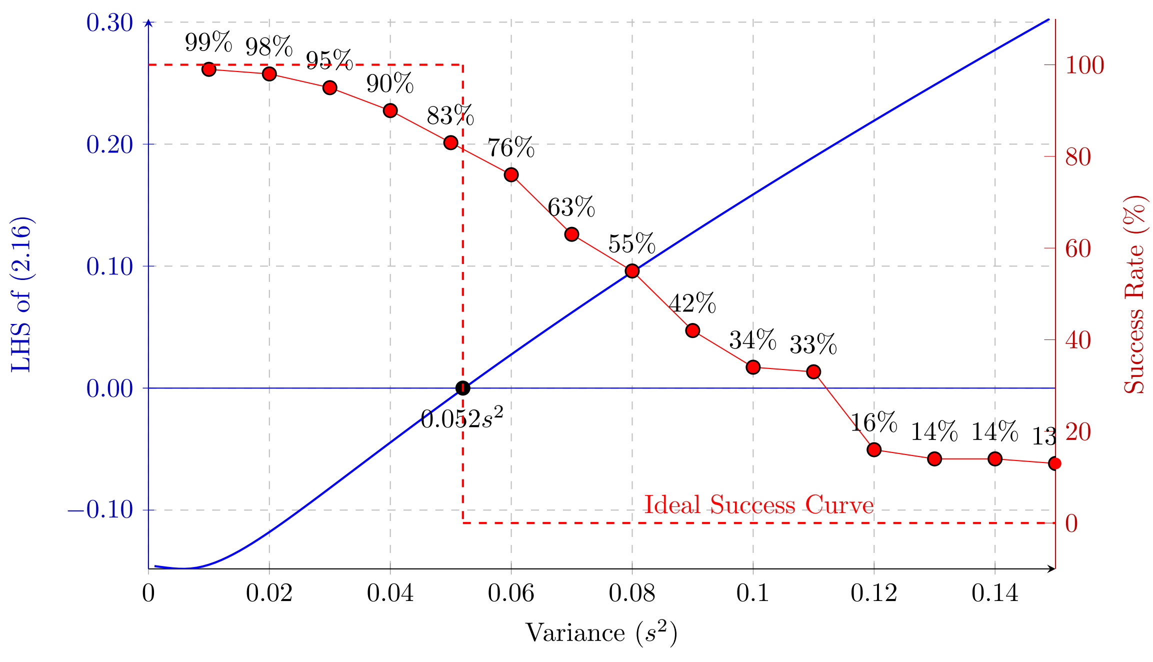


Figure 2.11. Comparison of left hand side of (2.16) and experimental results for the beta distribution.

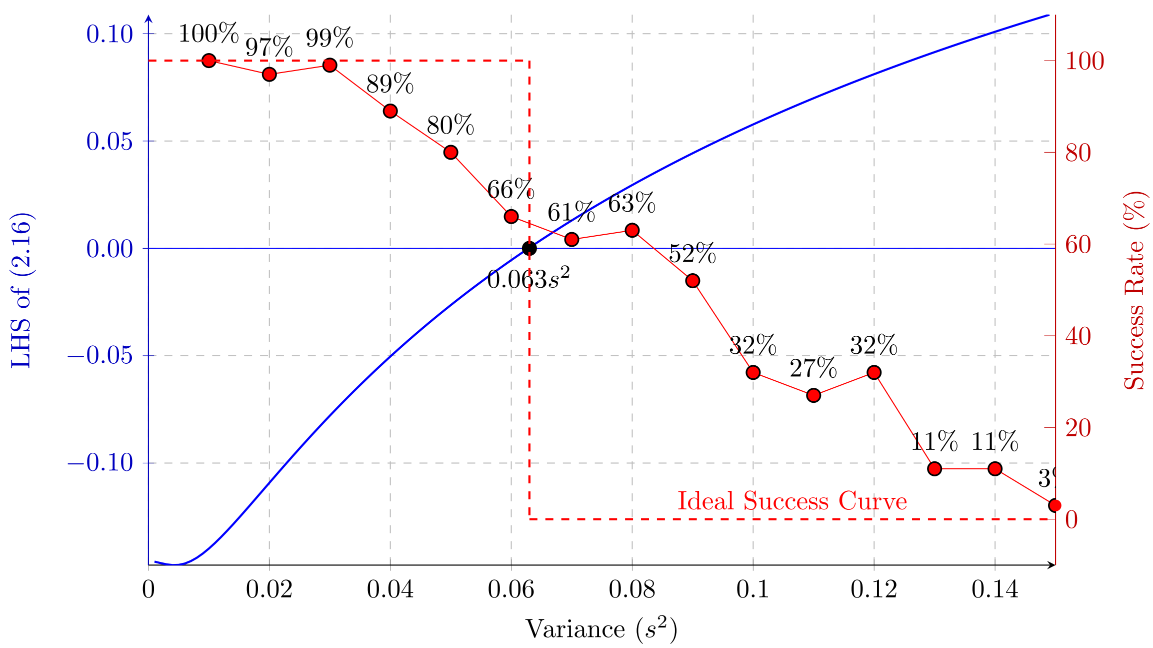


Figure 2.12. Comparison of left hand side of (2.16) and experimental results for the gamma distribution.

CHAPTER THREE

Euler-Bernoulli Model Solution

This chapter published as: Matthew D Mosley, Ian A Gravagne, and John M Davis. A solution of the euler- bernoulli flexible rotating arm. In Proceedings of the American Society for Engineering Education Gulf Southwest Conference (ASEE GSW). American Society for Engineering Education, ASEE, March 2016.

The planar one-degree-of-freedom flexible arm is a canonical problem for students and researchers investigating novel feedback control algorithms, as well as PDE numerical and theoretical solution methods. The purpose of this chapter is twofold: first, it serves to build intuition for the model identified in Chapter Four. Second, this chapter shows that modeling the flexible arm can serve as a pedagogical tool for undergraduate PDE instructors. Generally, researchers have approached the problem via either energy methods (e.g. Hamilton's principle [20]) which are generally intractable to undergraduates, or by eigenfunction expansion. Eigenfunction expansion is an approach tractable to undergraduates; however, researchers traditionally do not handle the nonhomogeneous boundary conditions in manner consistent with typical undergraduate instruction. Here, we outline the solution in a way that parallels typical PDE instruction regarding nonhomogeneous boundary conditions, namely, to convert the nonhomogeneous boundary problem into a homogeneous boundary problem by adding an appropriate (nonhomogeneous) forcing term to the field equation. This idea is motivated by Duhamel's principle and can often be seen in textbooks, illustrated using the heat equation [21].

3.1 Problem Setup

Following [22], the Euler-Bernoulli model is used to model the dynamics of the flexible beam. This model yields the following partial differential equation and boundary conditions:

$$y^{(4)}(x,t) + \frac{\rho}{EI}\ddot{y}(x,t) = 0, \qquad t > 0,\ 0 < x < L, \tag{3.1a}$$

$$y(0,t) = 0, \qquad t > 0, \tag{3.1b}$$

$$y''(L,t) = 0, \qquad t > 0, \tag{3.1c}$$

$$J\ddot{y}'(0,t) - EIy''(0,t) = \tau(t), \quad t > 0, \tag{3.1d}$$

$$m\ddot{y}(L,t) - EIy'''(L,t) = 0 \qquad t > 0, \tag{3.1e}$$

where $y(x,t)$ is the deflection of the beam and the other parameters are listed in Table 4.1. Boundary condition (3.1d) is non-homogeneous, so we assume the solution can be decomposed [21] into the following two parts:

$$y(x,t) = w(x,t) + v(x,t) \tag{3.2}$$

where $w(x,t)$ is the solution to a second boundary-value-problem (BVP), and $v(x,t)$ is a function that we introduce in order to homogenize the boundary conditions of that BVP. Substituting (3.2) into (3.1) yields the following:

$$w^{(4)}(x,t) + \frac{\rho}{EI}\ddot{w}(x,t) = -v^{(4)}(x,t) - \frac{\rho}{EI}\ddot{v}(x,t), \qquad t > 0,\ 0 < x < L, \tag{3.3a}$$

$$w(0,t) + v(0,t) = 0, \qquad t > 0, \tag{3.3b}$$

$$w''(L,t) + v''(L,t) = 0, \qquad t > 0, \tag{3.3c}$$

$$J\left(\ddot{w}'(0,t) + \ddot{v}'(0,t)\right) - EI\left(w''(0,t) + v''(0,t)\right) = \tau(t), \quad t > 0, \tag{3.3d}$$

$$m\left(\ddot{w}(L,t) + \ddot{v}(L,t)\right) - EI\left(w'''(L,t) + v'''(L,t)\right) = 0, \qquad t > 0, \tag{3.3e}$$

If we define

$$f(x,t) := -v^{(4)}(x,t) - \frac{\rho}{EI}\ddot{v}(x,t) \tag{3.4}$$

and choose the following conditions for $v(x,t)$:

$$v(0,t) = 0 \tag{3.5a}$$

$$v''(L,t) = 0 \tag{3.5b}$$

$$J\ddot{v}'(0,t) - EIv''(0,t) = \tau(t) \tag{3.5c}$$

$$m\ddot{v}'(L,t) - EIv'''(L,t) = 0 \tag{3.5d}$$

then the boundary value problem in (3.3) is transformed into

$$w^{(4)}(x,t) + \frac{\rho}{EI}\ddot{w}(x,t) = f(x,t), \quad 0 < x < L,\ t > 0, \tag{3.6a}$$

$$w(0,t) = 0, \qquad t > 0, \tag{3.6b}$$

$$w''(L,t) = 0, \qquad t > 0, \tag{3.6c}$$

$$J\ddot{w}'(0,t) - EIw''(0,t) = 0, \qquad t > 0, \tag{3.6d}$$

$$m\ddot{w}(L,t) - EIw'''(L,t) = 0, \qquad t > 0, \tag{3.6e}$$

which is a partial differential equation with homogeneous boundary conditions, but with a forcing function $f(x,t)$. Thus, the non-homogeneity is transferred from the boundary conditions to the field equation.

3.2 The Forcing Function

The forcing function is some polynomial in x, scaled by the hub torque $\tau(t)$. To calculate $f(x,t)$, we first find $v(x,t)$. The only restriction on $v(x,t)$ is that it must satisfy the boundary conditions in (3.5). We assume that $v(x,t)$ is separable:

$$v(x,t) = g(x)\tau(t). \tag{3.7}$$

where $\tau(t)$ is the hub torque and we choose $g(x)$ as

$$g(x) = c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0, \tag{3.8}$$

where the coefficients c_n are found by translating the conditions in (3.5) into the following conditions for g:

$$g(0) = 0 \tag{3.9a}$$

$$g''(L) = 0 \tag{3.9b}$$

$$Jg'(0)\ddot{\tau}(t) - \left[EIg''(0) + 1\right]\tau(t) = 0 \tag{3.9c}$$

$$mg(L)\ddot{\tau}(t) - EIg'''(L)\tau(t) = 0 \tag{3.9d}$$

From (3.9c) and (3.9d), we also have that

$$g'(0) = 0 \tag{3.10a}$$

$$EIg''(0) + 1 = 0 \tag{3.10b}$$

$$g(L) = 0 \tag{3.10c}$$

$$g'''(L) = 0 \tag{3.10d}$$

Normally this would not be mathematically justifiable, but from a physical perspective we know that we have arbitrary control of τ and $\ddot{\tau}$. Therefore there is no *fixed* relationship between them and so it follows that the coefficients in (3.9c) and (3.9d) must be 0. Application of these boundary conditions yields a solution for $g(x)$, plotted in Figure 3.1.

3.3 The Eigenfunctions

The method of eigenfunction expansion begins by finding the eigenfunctions for the unforced (homogeneous) system [23], i.e. equation (6a) with $f(x,t) = 0$:

$$w^{(4)}(x,t) + \frac{\rho}{EI}\ddot{w}(x,t) = 0. \tag{3.11}$$

We assume that the solution $w(x,t)$ is separable and has the form

$$w(x,t) = \sum_{n=0}^{\infty} X_n(x)T_n(t). \tag{3.12}$$

Substituting this into the PDE for a particular n and rearranging gives

$$\frac{EIX_n^{(4)}(x)}{\rho X_n(x)} = -\frac{\ddot{T}_n(t)}{T_n(t)} = \omega_n^2. \tag{3.13}$$

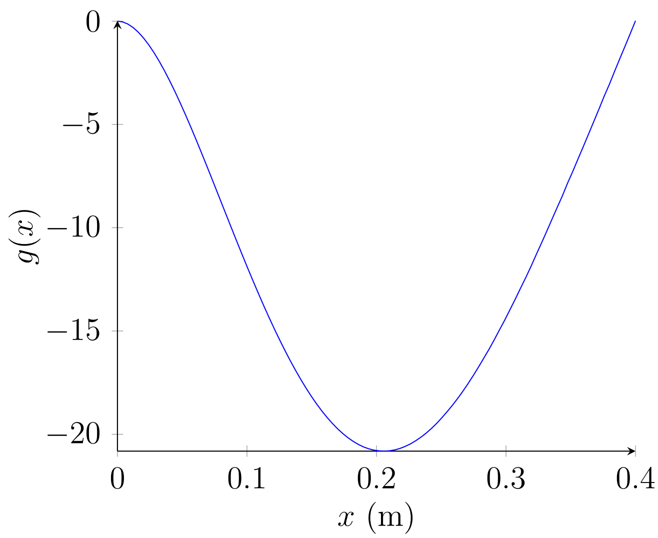


Figure 3.1. The function $g(x)$ represents a distributed forcing function that would displace the beam into the same shape as the application of a boundary hub torque.

which yields the following ODE:

$$X_n^{(4)} - \beta_n^4 X_n = 0, \tag{3.14}$$

where

$$\beta_n^4 = \frac{\rho\omega_n^2}{EI}. \tag{3.15}$$

We assume that the eigenfunctions $X_n(x)$ have the following form

$$X_n(x) = a_n \cos\beta_n x + b_n \sin\beta_n x + c_n \cosh\beta_n x + d_n \sinh\beta_n x \tag{3.16}$$

and we translate the boundary conditions from (3.6) into the following:

$$X(0) = 0 \tag{3.17a}$$

$$X''(L) = 0 \tag{3.17b}$$

$$J\omega^2 X'(0) + EIX''(0) = 0 \tag{3.17c}$$

$$m\omega^2 X(L) + EIX'''(L) = 0 \tag{3.17d}$$

Taking the first three derivatives of (3.16) and using (3.17) yields a system of four equations, represented by the following matrix equation:

$$\mathbf{Z}(\beta)\begin{bmatrix} a_n \\ b_n \\ c_n \\ d_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.18}$$

where

$$\mathbf{Z}(\beta) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ -\cos\beta L & -\sin\beta L & \cosh\beta L & \sinh\beta L \\ -1 & J\beta^3/\rho & 1 & J\beta^3/\rho \\ \sin\beta L + \frac{m}{\rho}\beta\cos\beta L & -\cos\beta L + \frac{m}{\rho}\beta\sin\beta L & \sinh\beta L + \frac{m}{\rho}\beta\cosh\beta L & \cosh\beta L + \frac{m}{\rho}\beta\sinh\beta L \end{bmatrix} \tag{3.19}$$

In order to solve for a_n, b_n, c_n and d_n, we find β for which $\mathbf{Z}(\beta_n)$ is singular. Ignoring the trivial solution, we assume a_n, b_n, c_n, d_n non-zero. For each β_n, the coefficients a_n, b_n, c_n, d_n can be computed numerically by finding the null-space of $\mathbf{Z}(\beta_n)$.

3.3.1 When $\beta = 0$

When $\beta = 0$, we have a different form for $X_0(x)$. Since we have that

$$X_0^{(4)} = 0, \tag{3.20}$$

we assume that $X_0(x)$ is a 3rd order polynomial.

$$X_0(x) = ax^3 + bx^2 + cx + d. \tag{3.21}$$

Again, we use the boundary conditions in (3.17) (with $\omega = 0$) and get that $a = b = d = 0$ and c is a free variable. Without loss of generality we will let $c = 1$. Then the eigenfunction for $\beta = 0$ is

$$X_0(x) = x. \tag{3.22}$$

3.4 Solution Via Orthogonality

Henceforth, we assume that $f(x,t) \neq 0$, so it is no longer true that $\ddot{T} + \omega^2 T = 0$. From [22] we have the following orthogonality condition on the eigenfunctions:

$$\int_0^L \rho X_r(x) X_n(x) \mathrm{d}x + m X_r(L) X_n(L) + J X_r'(0) X_n'(0) = \begin{cases} 0 & n \neq r \\ M_r & n = r \end{cases} \tag{3.23}$$

and it follows from (3.12) that

$$\int_0^L w(x,t) \rho X_r(x) = T_r(t) M_r - \sum_{n=0}^{\infty} T_n(t) m X_r(L) X_n(L) - \sum_{n=0}^{\infty} T_n(t) J X_r'(0) X_n'(0) \tag{3.24}$$

Recall that the field equation is

$$w^{(4)}(x,t) + \frac{\rho}{EI} \ddot{w}(x,t) = f(x,t). \tag{3.25}$$

Multiplying each side by $X_r(x)$ and integrating gives us

$$\int_0^L X_r(x)[w^{(4)}(x,t) + \frac{\rho}{EI} \ddot{w}(x,t)] \mathrm{d}x = \int_0^L X_r(x) f(x,t) \mathrm{d}x. \tag{3.26}$$

Rewrite (3.26) as

$$\int_0^L EI X_r(x) w^{(4)}(x,t) \mathrm{d}x + \int_0^L \rho X_r(x) \ddot{w}(x,t) \mathrm{d}x = \int_0^L EI X_r(x) f(x,t) \mathrm{d}x. \tag{3.27}$$

Using integration by parts four times on the first term and using boundary conditions from (3.6) and (3.17), we get

$$\begin{aligned} m X_r(L) \ddot{w}(L,t) + J X_r(0) \ddot{w}'(0,t) + J\omega^2 X_r'(0) w'(0,t) + m\omega^2 X(L) w(L,t) + \\ \int_0^L EI X_r^{(4)} w(x,t) \mathrm{d}x + \int_0^L \rho X_r(x) \ddot{w}(x,t) \mathrm{d}x = \int_0^L EI X_r(x) f(x,t) \mathrm{d}x \end{aligned} \tag{3.28}$$

where we have from (3.24) that

$$\int_0^L EI X_r^{(4)} w(x,t) \mathrm{d}x = \omega_r^2 \int_0^L \rho X_r w(x,t) \mathrm{d}x \tag{3.29}$$

$$= \omega_r^2 \left[T_r M_r - \sum_{n=0}^{\infty} T_n m X_r(L) X_n(L) - \sum_{n=0}^{\infty} T_n J X_r'(0) X_n'(0) \right] \tag{3.30}$$

and

$$\int_0^L \rho X_r(x)\ddot{w}(x,t)\mathrm{d}x = \frac{\mathrm{d}^2}{\mathrm{d}t^2}\int_0^L \rho X_r w(x,t)\mathrm{d}x \tag{3.31}$$

$$= \ddot{T}_r M_r - \sum_{n=0}^{\infty}\ddot{T}_n m X_r(L)X_n(L) - \sum_{n=0}^{\infty}\ddot{T}_n J X_r'(0)X_n'(0) \tag{3.32}$$

We see that substituting (3.12), (3.30) and (3.32) into (3.28) results in

$$\left(\ddot{T}_r(t) + \omega^2 T_r\right) M_r = \int_0^L EIX_r(x)f(x,t)\mathrm{d}x. \tag{3.33}$$

Now, substituting in (3.4) and (3.7) gives us

$$\left(\ddot{T}_r(t) + \omega^2 T_r\right) M_r = -\tau(t)EI\int_0^L X_r(x)g^{(4)}(x)\mathrm{d}x - \ddot{\tau}(t)\rho\int_0^L X_r(x)g(x)\mathrm{d}x. \tag{3.34}$$

If we perform integration by parts four times on

$$EI\int_0^L X_r(x)g^{(4)}(x)\mathrm{d}x, \tag{3.35}$$

using the boundary conditions from (3.9) and (3.17), we get

$$\omega_r^2\int_0^L \rho X_r(x)g(x)\mathrm{d}x - X_r'(0). \tag{3.36}$$

If we then define

$$A_r = \int_0^L \rho X_r(x)g(x)\mathrm{d}x, \tag{3.37}$$

we can simplify (3.34) to the following:

$$\left(\ddot{T}_r(t) + \omega_r^2 T_r(t)\right) M_r = -\tau(t)\left(\omega_r^2 A_r - X_r'(0)\right) - \ddot{\tau}(t)A_r. \tag{3.38}$$

Taking the Laplace transform and rearranging terms gives us

$$T_r(s) = -\tau(s)\left(\frac{A_r}{M_r} - \frac{X_r'(0)}{M_r(s^2+\omega_r^2)}\right). \tag{3.39}$$

Making use of (3.12) and (3.2) gives us

$$y(x,s) = g(x)\tau(s) - \tau(s)\sum_{r=0}^{\infty}X_r(x)\left(\frac{A_r}{M_r} - \frac{X_r'(0)}{M_r(s^2+\omega_r^2)}\right) \tag{3.40}$$

Proceeding formally, if we expand $g(x)$ using the eigenfunctions $X_r(x)$ as its basis, we can define g_r such that

$$g(x) = \sum_{r=0}^{\infty} g_r X_r(x). \tag{3.41}$$

Then

$$g_r = \frac{\int_0^L \rho X_r(x) g(x) \mathrm{d}x}{\int_0^L \rho X_r^2(x) \mathrm{d}x + m X_r^2(L) + J\left(X_r'(0)\right)^2} = \frac{A_r}{M_r} \tag{3.42}$$

Using (3.42), equation (3.40) can now be written as

$$y(x,s) = \tau(s)\left(g(x) - \sum_{r=0}^{\infty} g_r X_r(x)\right) + \tau(s) \sum_{r=0}^{\infty} \frac{X_r(x) X_r'(0)}{M_r(s^2+\omega_r^2)} \tag{3.43}$$

and (3.41) implies that

$$\frac{y(x,s)}{\tau(s)} = \sum_{r=0}^{\infty} \frac{X_r(x) X_r'(0)}{M_r(s^2+\omega_r^2)} \tag{3.44}$$

Since $X_0(x) = x$,

$$\frac{y(x,s)}{\tau(s)} = \frac{x}{M_0 s^2} + \sum_{r=1}^{\infty} \frac{X_r(x) X_r'(0)}{M_r(s^2+\omega_r^2)}. \tag{3.45}$$

If we take the spatial derivative of (3.45) and evaluate it at zero, then we have a transfer function that relates the motor torque to the motor hub angle:

$$\frac{y'(0,s)}{\tau(s)} = \frac{1}{M_0 s^2} + \sum_{r=1}^{\infty} \frac{(X_r'(0))^2}{M_r(s^2+\omega_r^2)}. \tag{3.46}$$

Similarly, if we take the 2nd-derivative of (3.45) with respect to t and evaluate it at L, then we have a transfer function that relates the motor torque to the acceleration experienced by the tip of the beam:

$$\frac{\ddot{y}(L,s)}{\tau(s)} = \frac{L}{M_0} + s^2 \sum_{r=1}^{\infty} \frac{X_r(L) X_r'(0)}{M_r(s^2+\omega_r^2)}. \tag{3.47}$$

Figures 3.2 and 3.3 display the locations of the first several poles and zeros for the transfer functions in (3.46) and (3.47), respectively, using the parameters in Table 4.1.

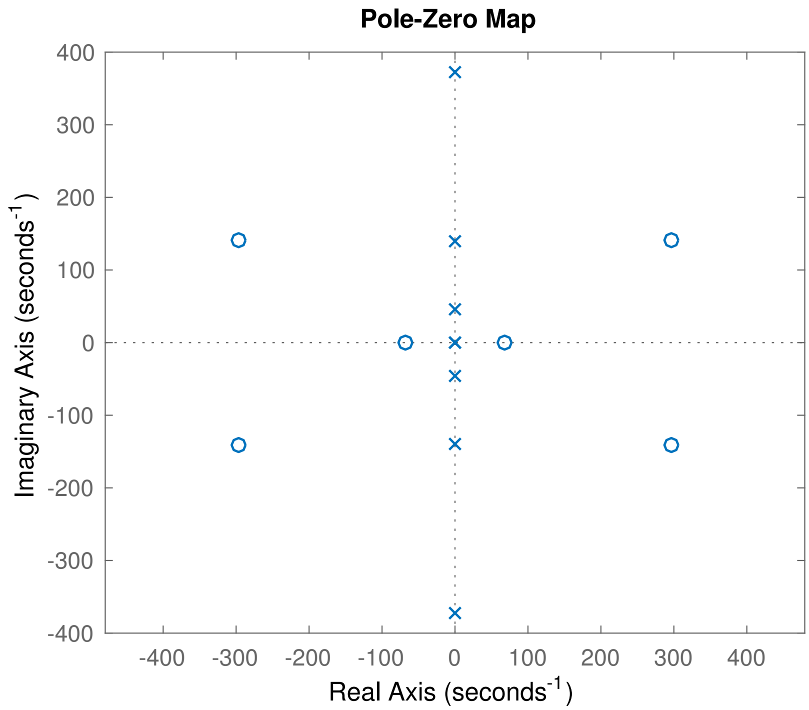


Figure 3.2. Poles and zeros for the transfer function in (3.46).

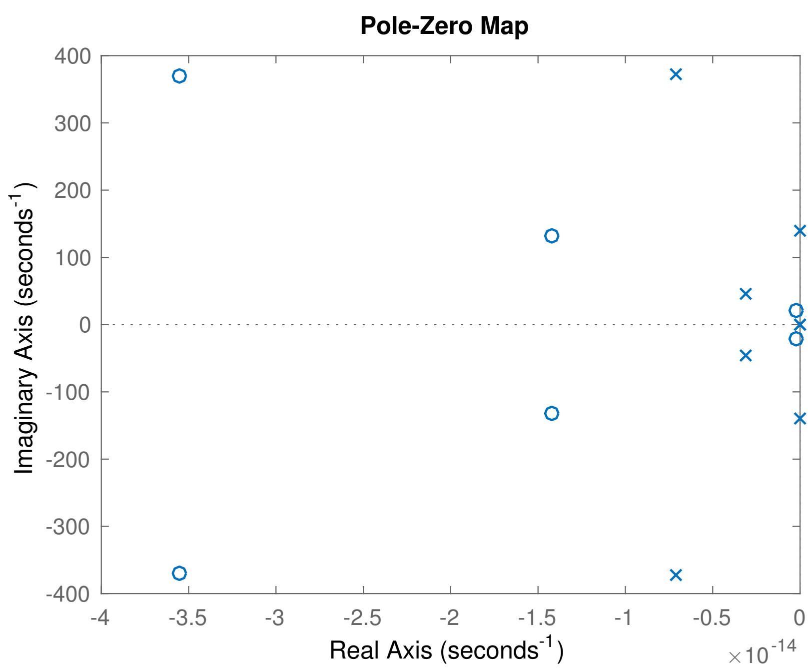


Figure 3.3. Poles and zeros for the transfer function in (3.47).

CHAPTER FOUR

Flexible Arm Experiment

In this chapter we present results from a stability experiment for a flexible beam system. This experiment is very similar to that from Chapter 2, but this system has much more interesting dynamics due to higher order vibrational modes. We then compare the results to the stability criterion and note that they are in agreement.

4.1 Description of System

The system comprises a thin flexible aluminum yardstick (which we refer to as the beam) clamped to the end of a DC motor, shown in Figures 4.1 and 4.2. The input to the system is the motor torque and the output is the hub angle. Since torque is related to current by the relationship

$$\tau(t) = k_t I(t), \tag{4.1}$$

where $\tau(t)$ is the torque, k_t is the motor torque constant and $I(t)$ is the motor current, the input torque is controlled directly by adjusting the amount of current applied. This was accomplished using the transconductance amplifier shown in Figure 4.3. See Table 4.1 for the physical parameters of the beam.

Table 4.1. Physical parameters of the flexible beam system.

Parameter	Value
Total Length L (m)	0.94
Height (m)	0.029
Thickness (m)	0.003
Mass of End Effector m (g)	1.27
Mass Linear Density ρ (kg/m)	0.1553
Young's Modulus E (N/m^2)	6.895×10^{10}
Cross-Sectional Area Moment of Inertia I (m^4)	6.525×10^{-11}
Motor Hub Moment of Inertia J (kg-m^2)	0.01

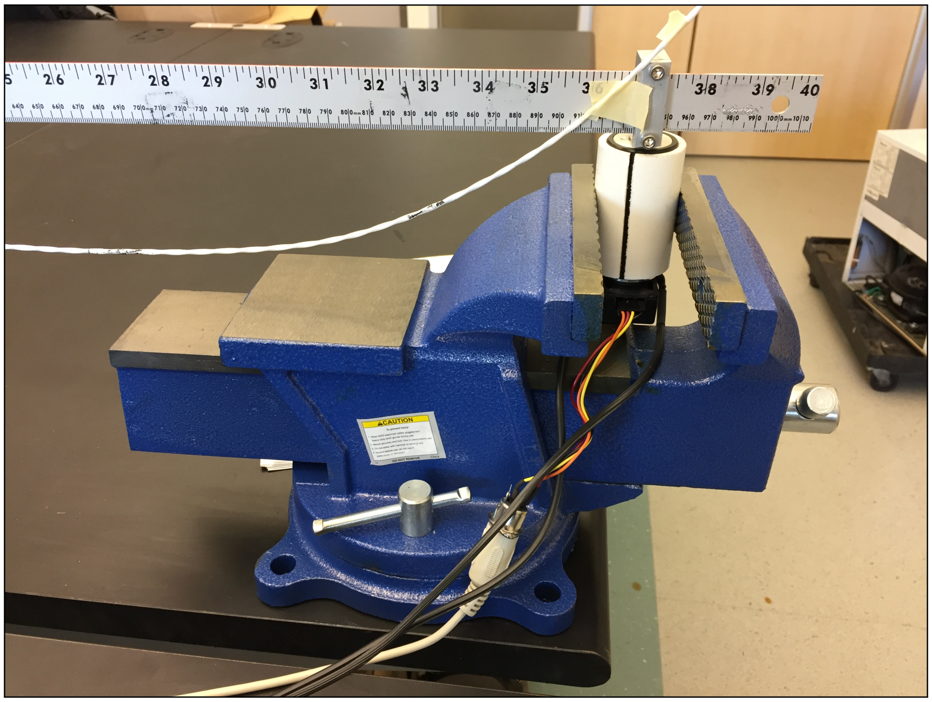


Figure 4.1. The flexible arm system

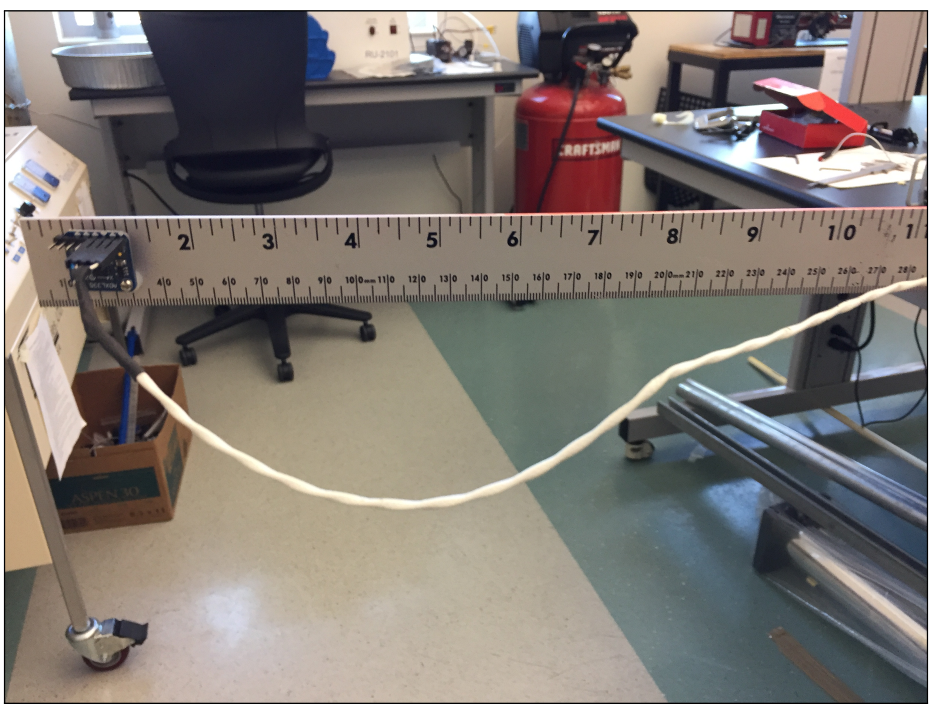


Figure 4.2. Accelerometer attached to tip

Figure 4.3. The transconductance amplifier

The feedback configuration for the system is shown in Figure 4.4. In addition to the motor hub angle, the tip velocity is also fed back to the input in order to more effectively damp out the tip vibrations.

4.2 System Identification

The system was identified using standard frequency response techniques. Two transfer functions were estimated. The first is

$$H_1(s) := \frac{\theta(s)}{\tau(s)}, \tag{4.2}$$

which relates the input motor torque to the hub angle of the motor. Figure 4.5 shows the measured frequency response data and the frequency response of the estimated transfer function. Figure 4.6 is a plot of the poles and zeros of this estimated system. A list of the poles and corresponding damping ratios and frequencies is shown in Table 4.2.

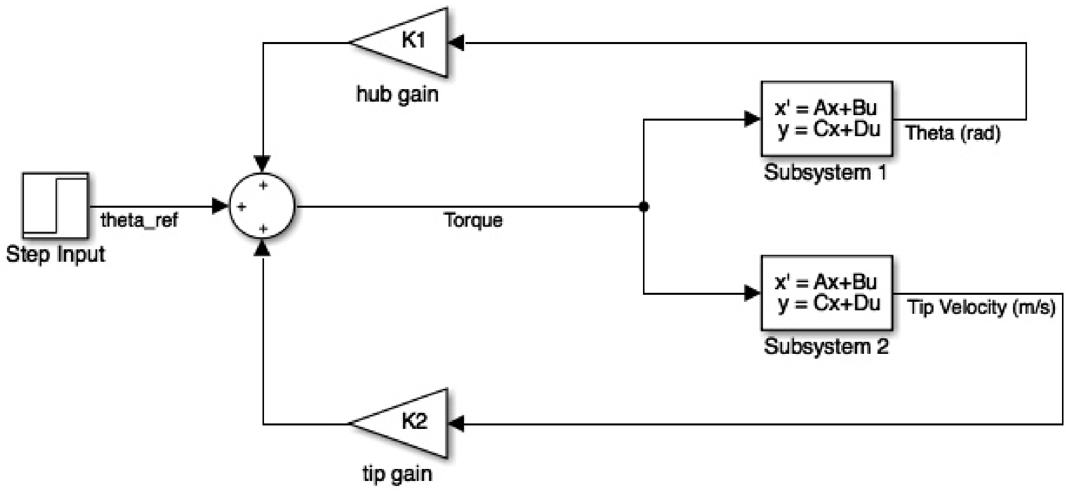


Figure 4.4. Block diagram of the closed loop system

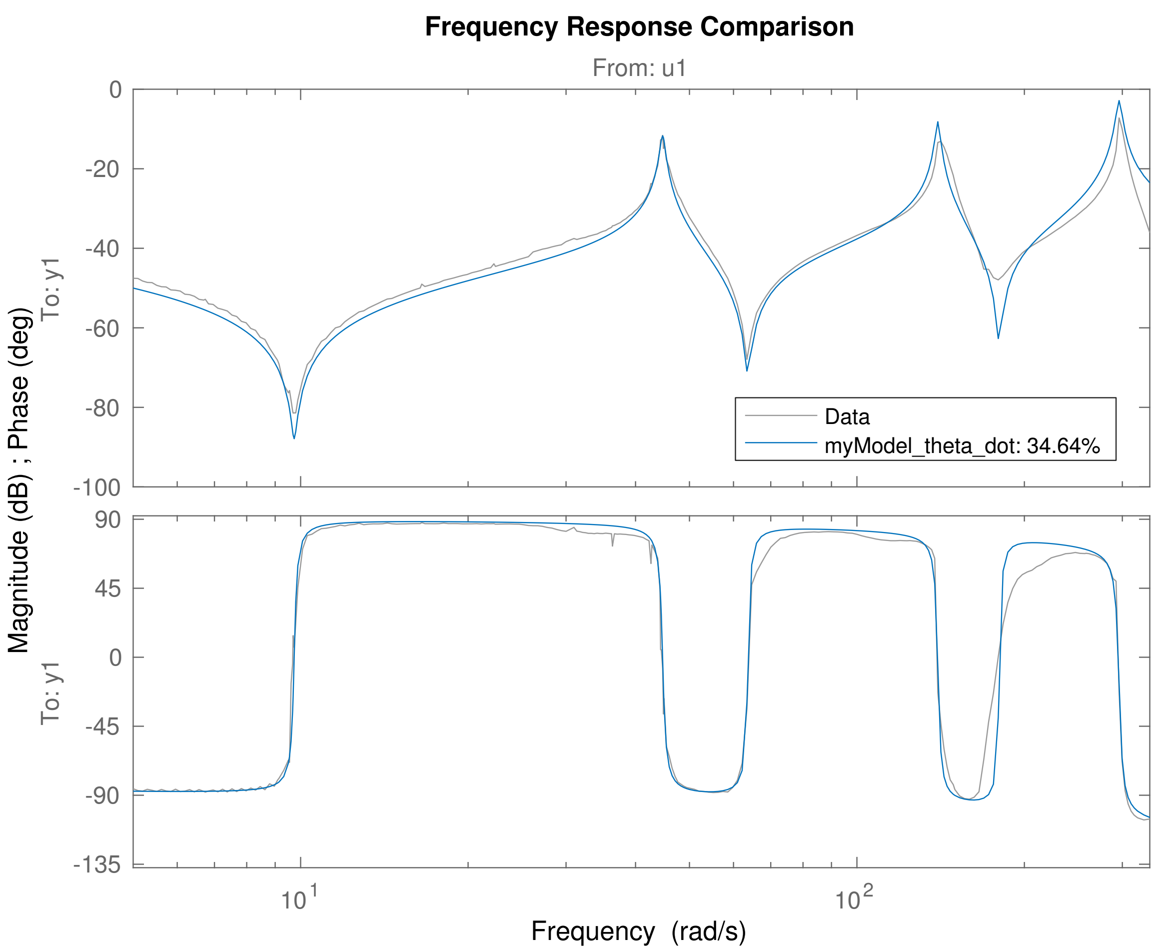


Figure 4.5. Frequency response for $H_1(s)$

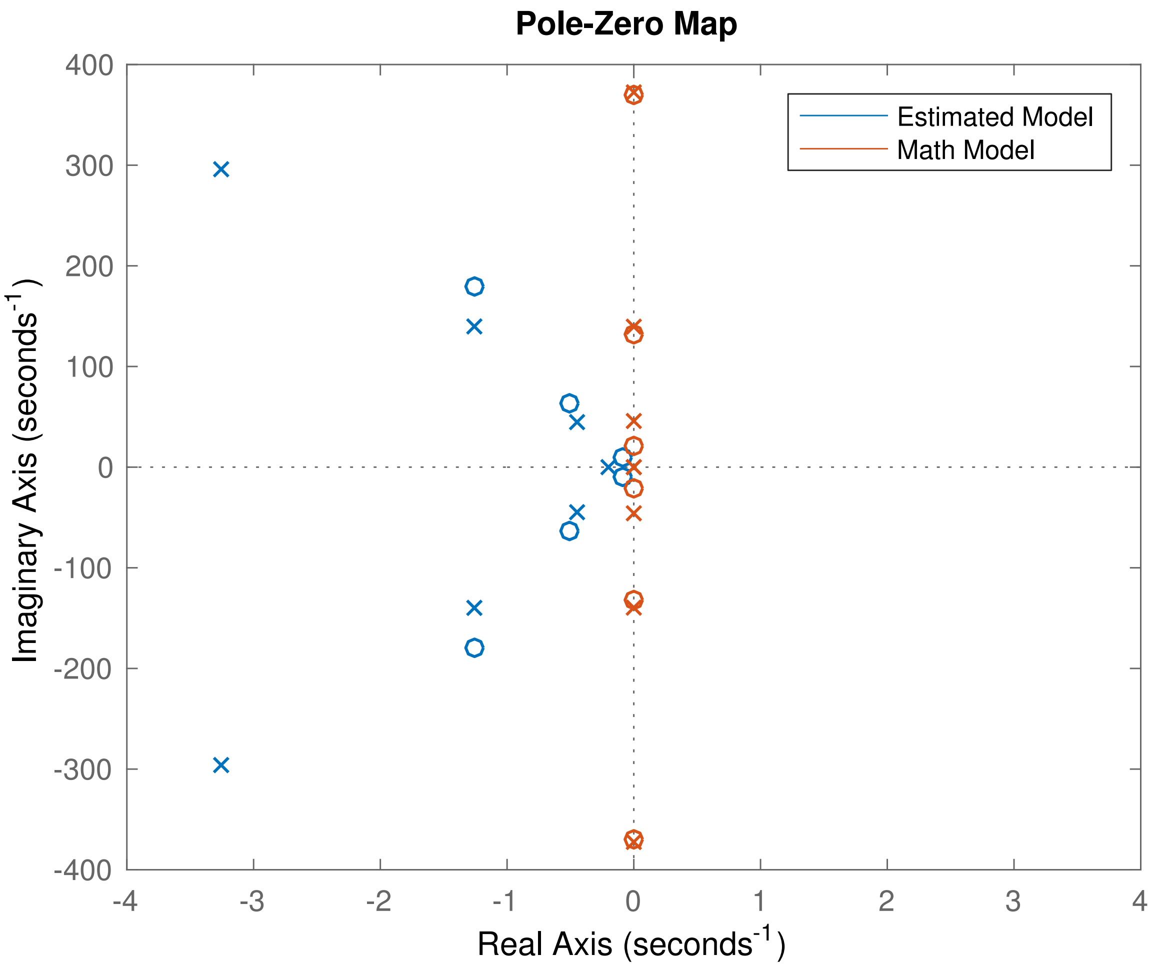


Figure 4.6. The poles and zeros for $H_1(s)$. The poles and zeros from the system derived in Chapter 3 are also presented for comparison. Although both models have the alternating pole-zero pattern along the imaginary axis, the model estimated from the frequency response data exhibits some damping (Note the scale of the x-axis is one-hundredth the scale of the y-axis).

Table 4.2. Frequency information for $H_1(s)$

Poles	Damping Ratio	Frequency (rad/s)
0	N/A	N/A
-0.200	N/A	N/A
-0.448 $\pm$ j44.7	0.01	44.7
-1.26 $\pm$ j140	0.009	140
-3.26 $\pm$ j296	0.011	296

Table 4.3. Frequency information for $H_2(s)$

Poles	Damping Ratio	Frequency (rad/s)
-0.589 ± j44.7	0.0132	44.7
-3.92 ± j142	0.0276	142
-4.85 ± j295	0.0165	295

The second transfer function is

$$H_2(s) := \frac{\alpha(s)}{\tau(s)}, \tag{4.3}$$

which relates the input motor torque to the acceleration experienced by the tip of the flexible beam. Figure 4.7 shows the frequency response data for this system, and Figure 4.8 is a plot of the poles and zeros for the estimated transfer function. A list of the poles and corresponding damping ratios and frequencies is shown in Table 4.3.

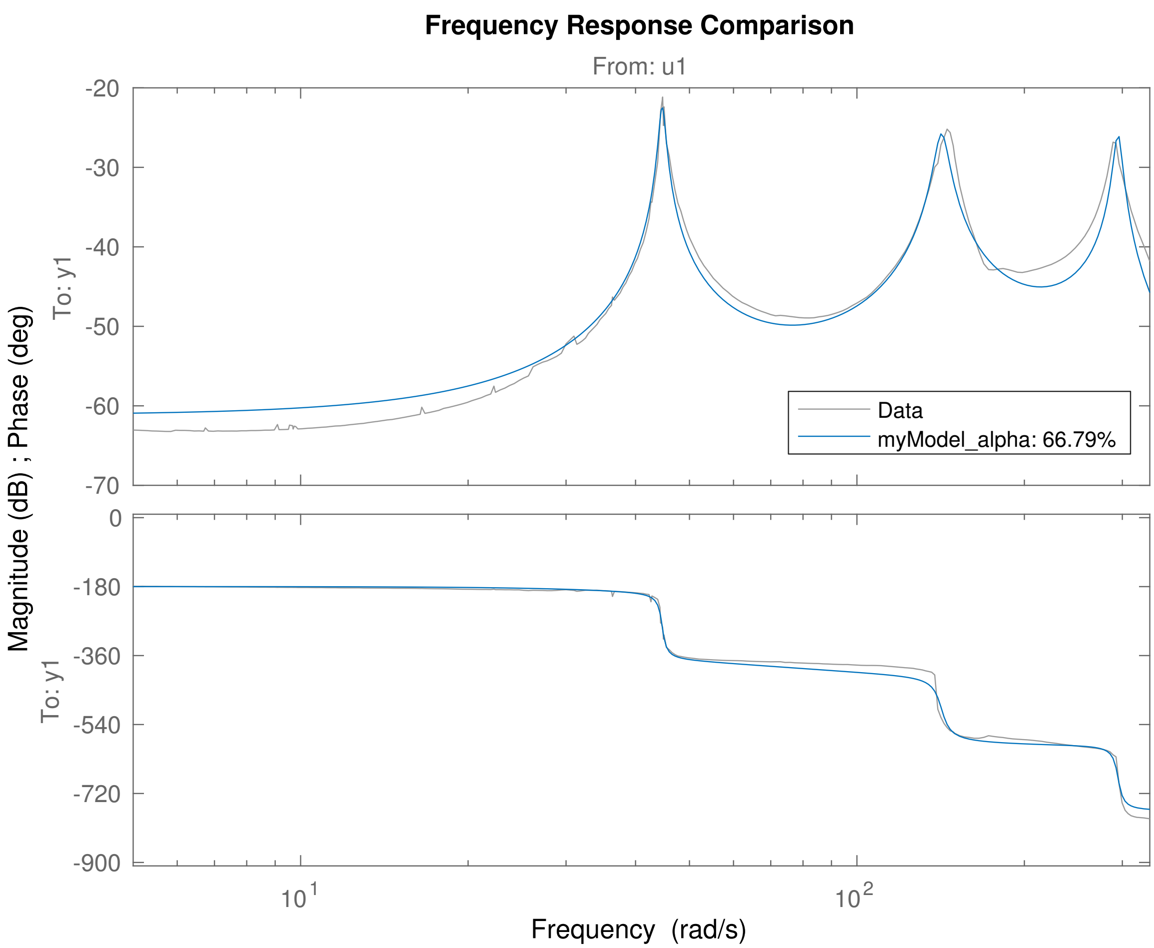


Figure 4.7. Frequency response for $H_2(s)$

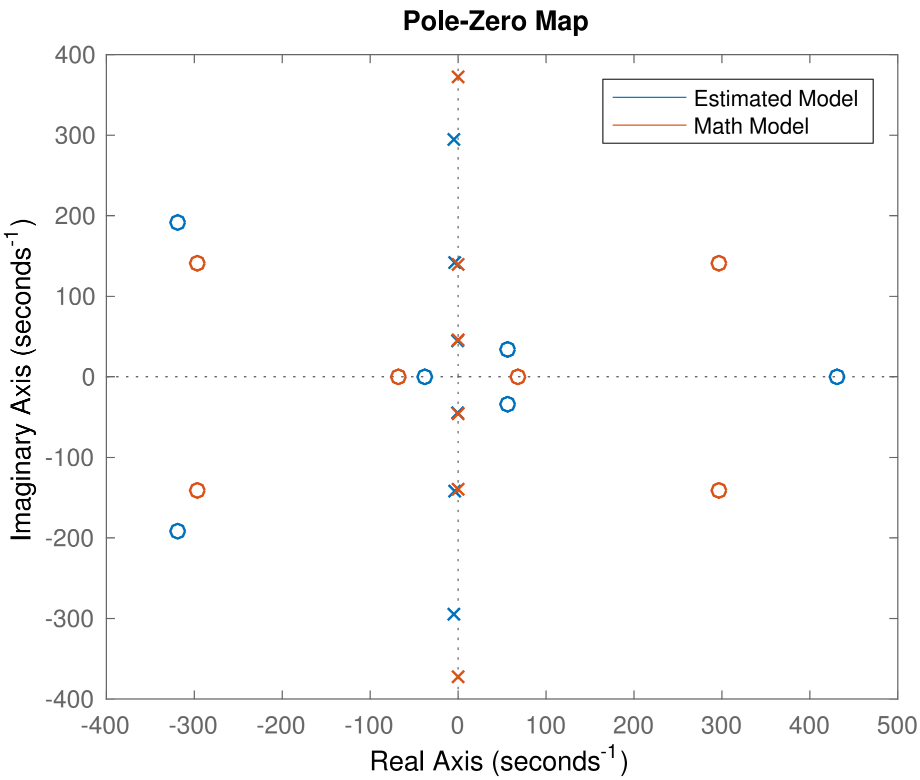


Figure 4.8. Poles and zeros for $H_2(s)$. The main discrepancy between the estimated and mathematical models is in the placement of the zeros.

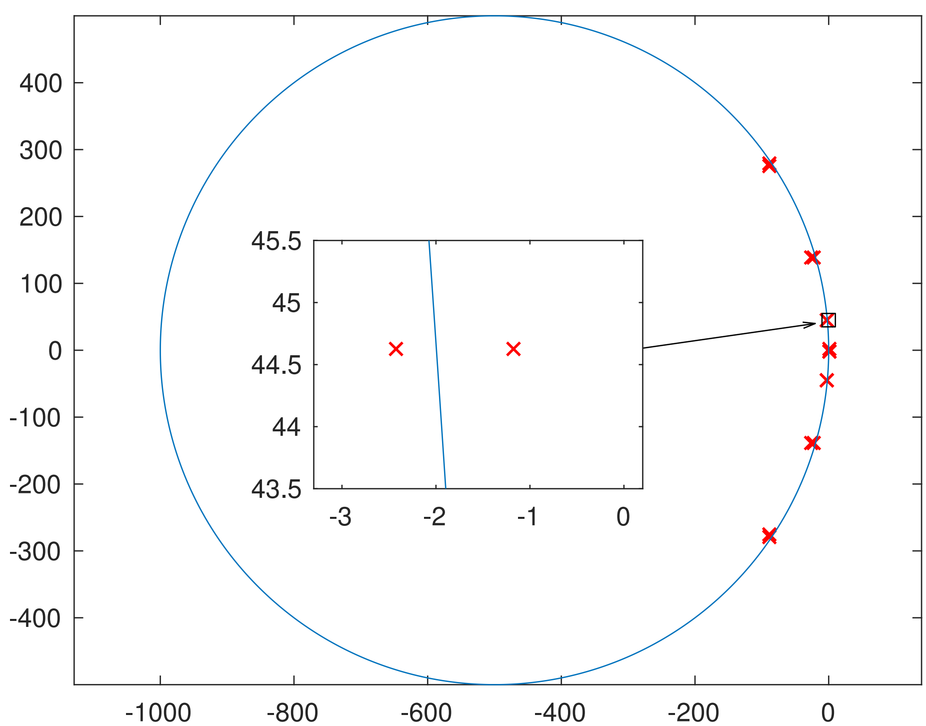


Figure 4.9. Poles for the closed-loop system discretized at 2ms. One complex conjugate pair is located slightly outside the Hilger circle, resulting in an unstable system.

4.3 Time Scale Experiments

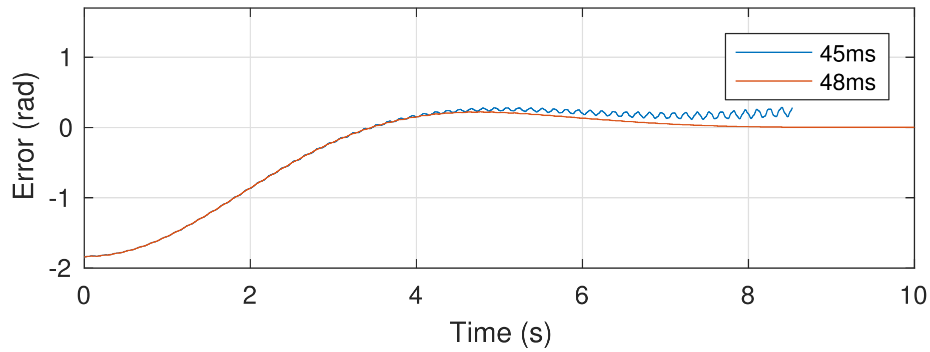
In order to discretize the closed loop system, the two transfer functions H_1 and H_2 were converted to a state space representation using MATLAB. The state matrices were then discretized using the same procedure described in Section 2.3. Stability experiments were then performed for the flexible arm system, using both uniform and non-uniform time scales.

4.3.1 Uniform Time Scales

For this portion of the experiment, only uniform discrete time scales were used. The results of using four different graininesses are described here. The same gain values are used in each case:

$$K_1 = -100, \quad K_2 = 1500. \tag{4.4}$$

The step responses for the first two discretized systems and their corresponding pole plots are shown in Figure 4.10. These two responses are fairly similar except that the system running on a 45 ms sampling time is oscillating back and forth at the frequency corresponding to the single pole outside the Hilger circle. The system discretized on 48 ms has all its poles contained within the corresponding Hilger circle, so the 48 ms step response is stable. What is interesting to note here is that increasing the graininess actually results in a more stable system. This may seem counterintuitive since the system described in Chapter 2 was stable for low graininesses and unstable for higher graininesses. The reason for this is simple: The feedback signal is formed from the tip velocity and hub angle, both of which are oscillating at some vibrational mode of the beam. So for small graininesses, it is not surprising that this feedback signal excites the system. Increasing the graininess of the time scale has the side-effect of sampling the feedback signal at a lower rate. As the graininess is increased further, eventually the feedback signal becomes aliased into a signal with lower frequency components that do not excite any of the system modes.



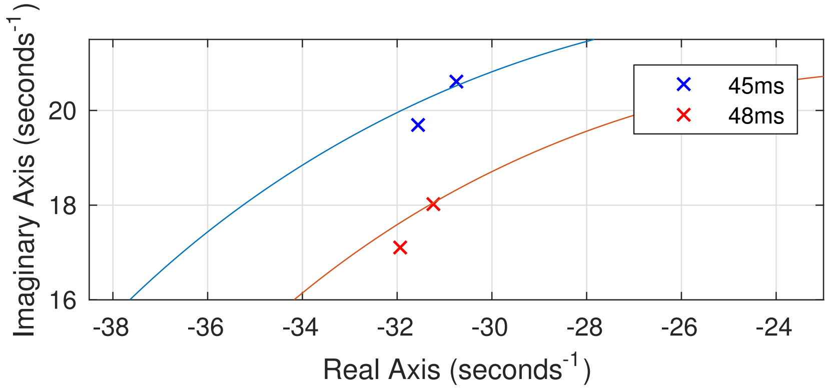


Figure 4.10. The system responses (top) and pole plots with corresponding Hilger circles (bottom).

Figure 4.11 shows the next two step responses and their corresponding pole locations. In this pair of responses, the system is stable at 1.28 s, but unstable when the graininess is increased further to 1.30 s. Here the instability is simply a result of an insufficient control rate. On comparing the two responses associated with 45 ms and 1.30 s, we note that they are both unstable, but the second oscillates at a much lower frequency. This is because the pole outside the Hilger circle for the 45 ms system corresponds to a frequency of about 6 Hz, whereas the unstable pole for the 1.30 s discretized system corresponds to a frequency of approximately 1 Hz.

System stability can also be determined analytically by appealing directly to the cylinder transform (1.27), without plotting eigenvalues in the Hilger complex plane. The system is stable if and only if the cylinder transforms of all eigenvalues are negative. Figure 4.12 shows the cylinder transforms of five eigenvalues of the flexible arm system as functions of the graininess. These five eigenvalues were chosen because they are the only ones that have cylinder transforms that are greater than zero for

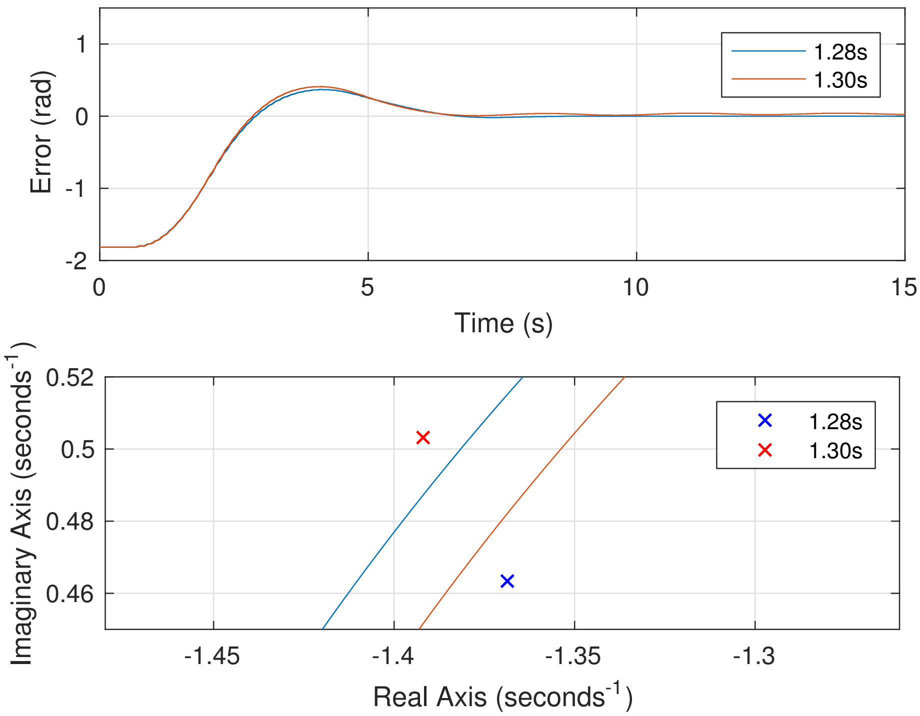


Figure 4.11. The system responses (top) and pole plots with corresponding Hilger circles (bottom).

some μ. Thus for stability analysis purposes, these are the only interesting ones. The transform associated with the third eigenvalue, $\lambda_3(\mu)$ is greater than zero for the small interval $[0\text{ s}, 0.047\text{ s}]$, indicating that the system is unstable for these sampling rates. This transform is also greater than zero for the interval [1.291 s, 1.335 s] indicating instability for those sampling rates as well. This is in agreement with the results shown in Figure 4.10 and Figure 4.11.

4.3.2 Non-Uniform Time Scales

To determine whether the system discretized on non-uniform time scales is stable, we appeal to the criterion in (1.37), reproduced here for convenience:

$$\max_i \int_0^T f(\mu) \ln |1 + \mu\lambda_i(\mu)| \mathrm{d}\mu < 0. \tag{4.5}$$

Let us examine (4.5) more closely. The integrand of (4.5) is a probability density function (PDF), f, multiplied by the cylinder transform of the i^{th} eigenvalue of the discretized system matrix. This integrand can be thought of as a weighted sum of

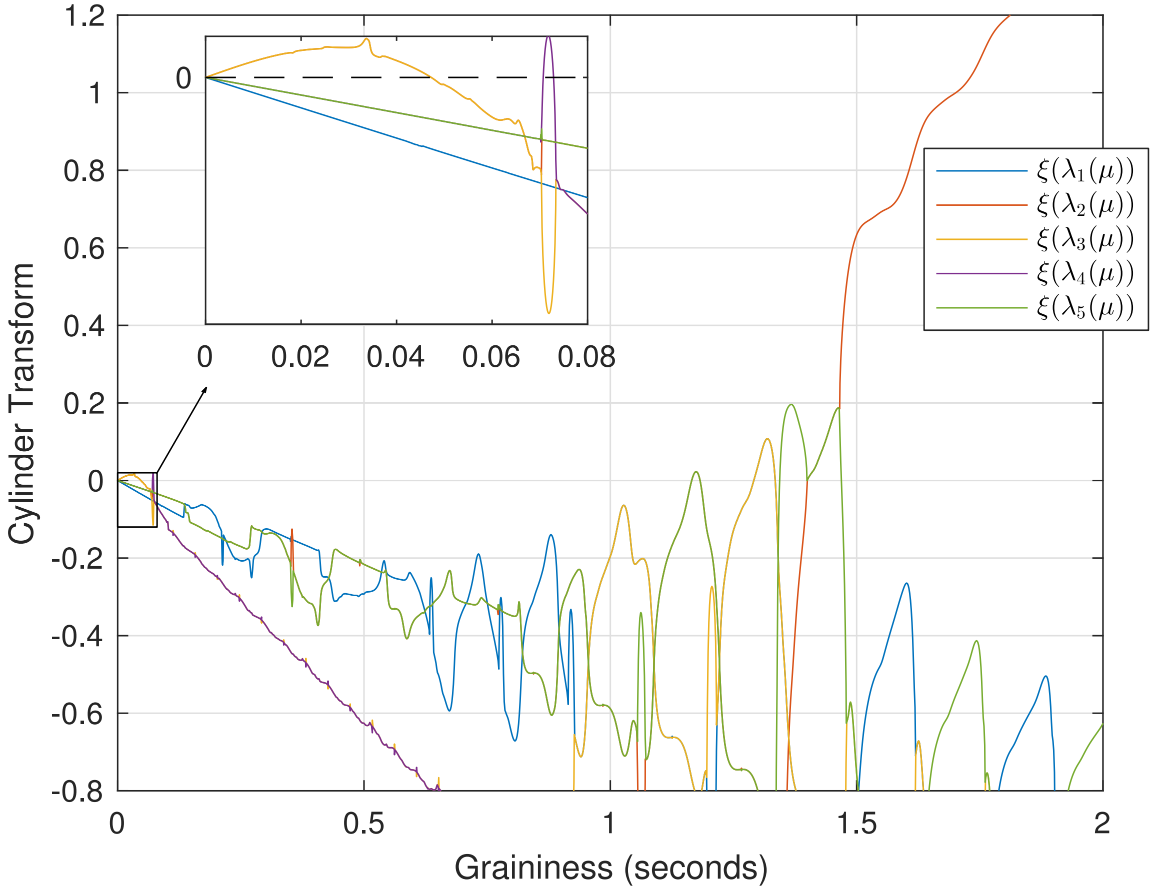


Figure 4.12. Output of the cylinder transformation for five eigenvalues as functions of μ. On the interval [0,0.047], $C(\lambda_3(\mu))$ is greater than zero and thus the system will be unstable at this sampling rate.

the output of the cylinder transform, where the weighting factor is determined by the PDF. So if the PDF has an expected value for which the cylinder transform greater than 0, and the variance is exceedingly small, then it is likely that the integral will also be greater than zero. This intuition is confirmed by Figure 4.14, which traces the output of the stability criterion for a mean of 0.030 s as the variance is increased from 4×10^{-4} s^2 to 14×10^{-4} s^2.

4.3.2.1 *Experimental Results* A stability experiment similar to the one presented in Chapter 2 was then performed on the flexible arm system. This time however, the experiment was carried out only on time scales whose grainninesses were distributed according to gamma distributions. The mean was held constant at 0.030 ms and

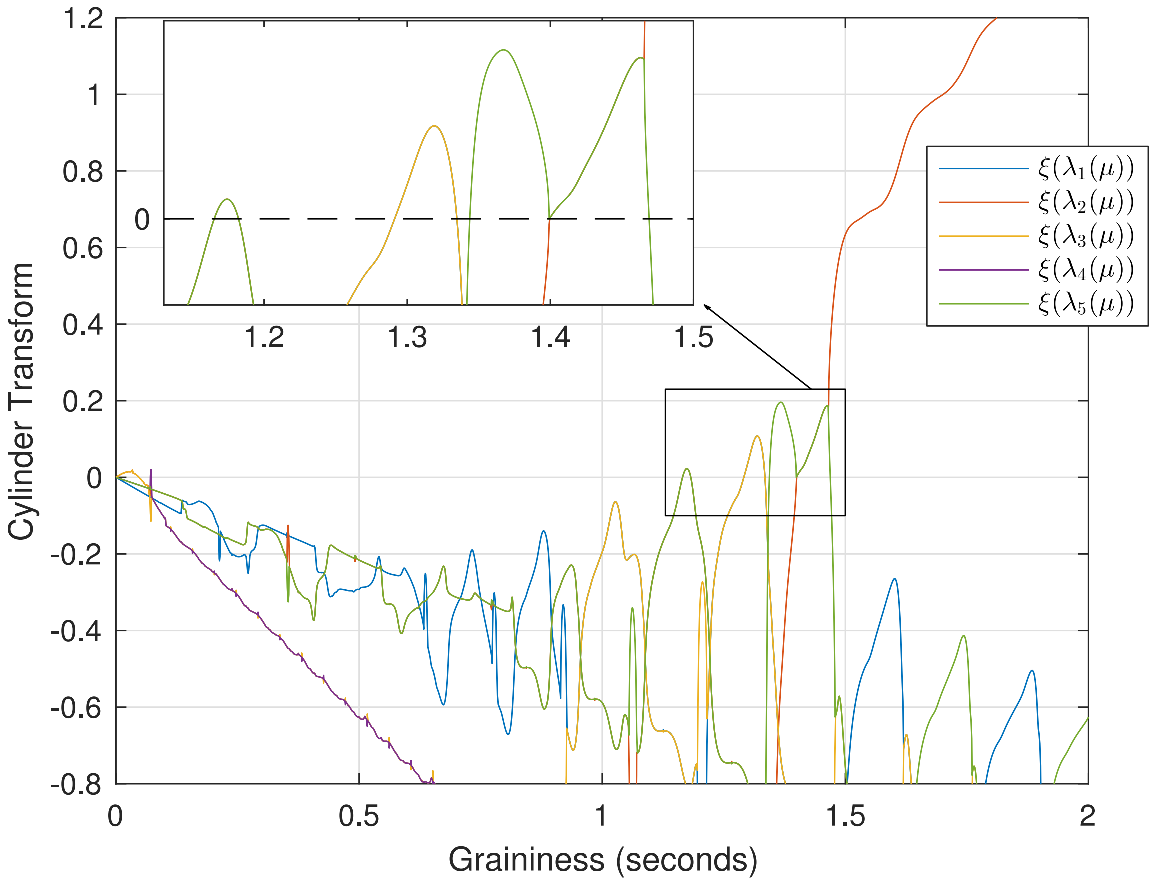


Figure 4.13. Output of the cylinder transformation for five eigenvalues as functions of μ. There are ranges of μ that produce instability at [1.165, 1.183], [1.291, 1.335] and [1.344, ∞]

.

the variance took on values from 5×10^{-4} s^2 to 10.3×10^{-4} s^2. The percentage of stable responses was recorded for each variance. The results of this experiment are summarized in Figure 4.14. Again, the result here is counterintuitive because it demonstrates that a system with more variance in the sampling rate can sometimes be more stable than one with little to no variance. The reasoning is similar to the above discussion: with little to no variance, the sinusoidal feedback signal excites the vibrational modes of the beam. But when variance is introduced to the sampling rate, the resulting sampled control signal no longer contains those excitatory frequencies, and thus the response is stable.

A second experiment was performed at a higher mean of 1.2 s with the variance ranging from 1×10^{-8} s^2 to 0.5 s^2. The results are shown in Figure 4.15 and they indicate that the responses do tend to be less stable as variance is increased. However, due to the nature of performing experiments on finite time horizons, many responses that should be stable are misclassified as unstable. This is likely due to the relatively short experiment time being insufficient for the response to stabilize.

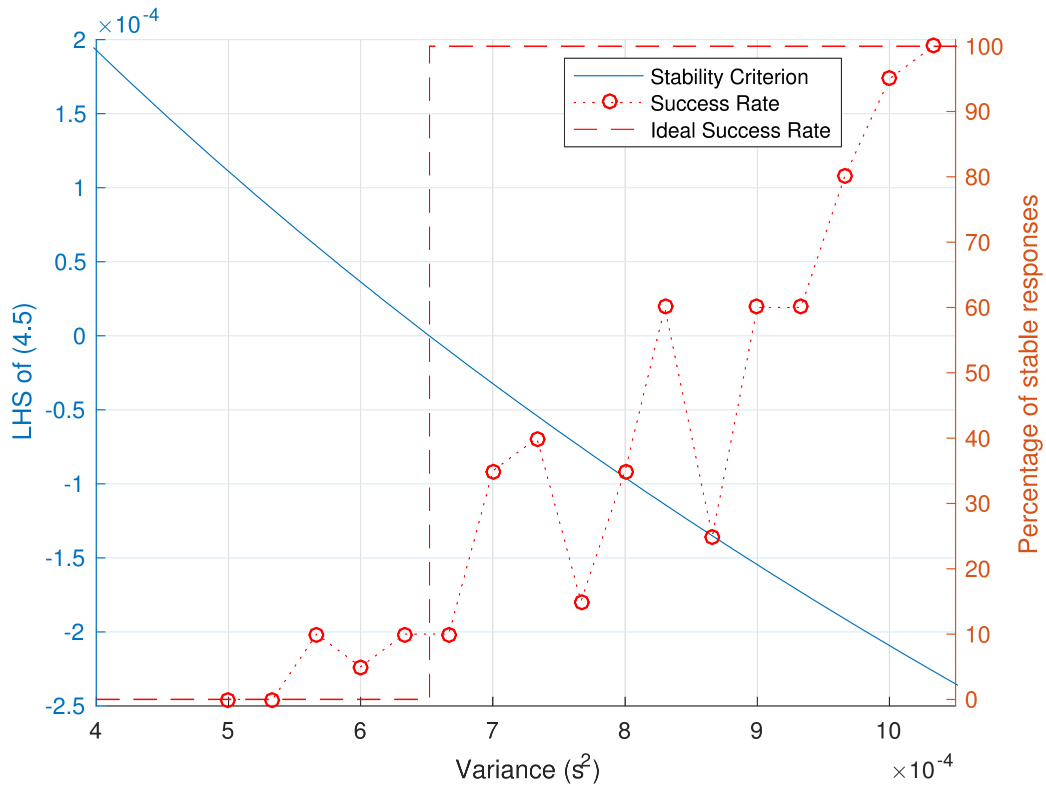


Figure 4.14. Comparison of criterion output with experimental results. Black dotted line is ideal success rate (Ideally, 100% of trials should be stable for variances greater than 6.52×10^{-4}s^2 and 0% should be stable for variances less than 6.52×10^{-4}s^2).

4.3.2.2 *Simulated Results* The primary reason that the actual success rate does not track the ideal success rate more closely is the finite time horizon constraint. To see the effect that this constraint has on the results, we now show the results of executing simulations on much longer time horizons than in the previous experiment. Figure 4.16 shows the results of simulating the system with a mean step size of 30

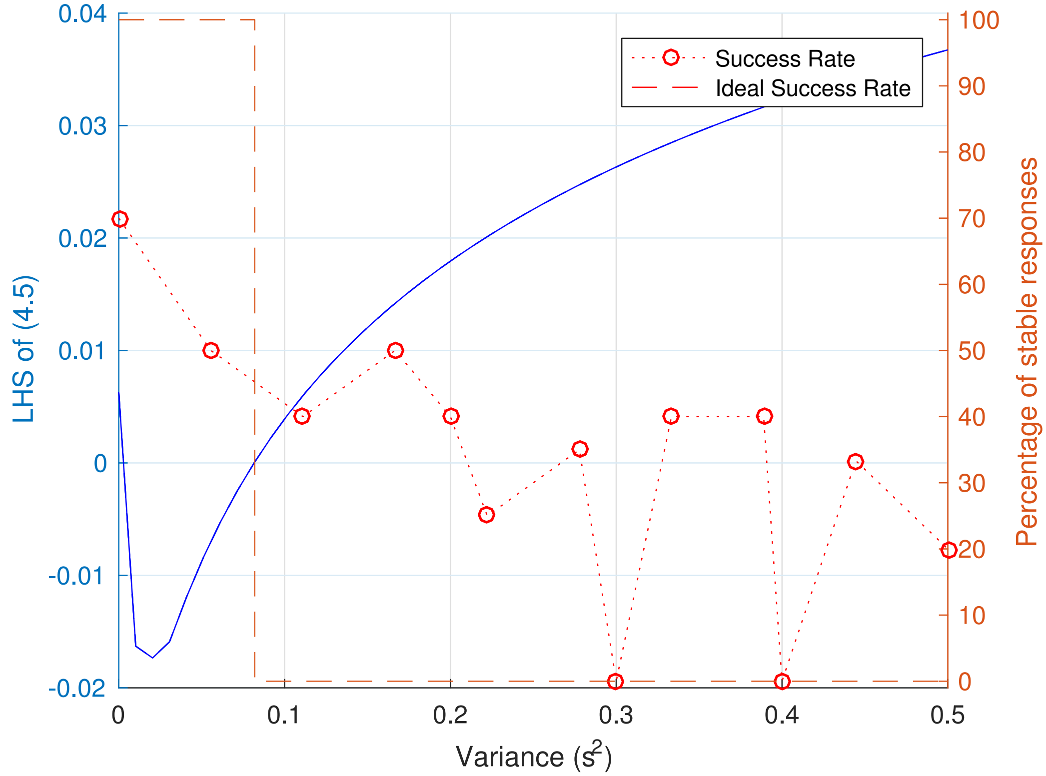


Figure 4.15. Comparison of criterion output with experimental results. The data doesn't line up quite as nicely as before because there are many responses misclassified as unstable since they don't settle during the finite time horizon.

ms with time horizons of 50 s and 500 s. Figure 4.17 shows the same thing but with a mean step size of 1.2 s.

4.4 Conclusions

This thesis presents experimental results from two different closed loop feedback control systems with sampling times distributed according to known statistical distributions. For both systems, the results illustrate that the "exponential stability almost surely" criterion of [1] is a reasonable predictor of stabilizing behavior over finite time horizons; however, the results also indicate that time scales with variance and mean near the limit of the criterion may not reliably stabilize the system, yielding unacceptable performance in some cases. One possible way to ensure better performance is to design for a more conservative stability criterion, such as "mean square exponential stability"[19].

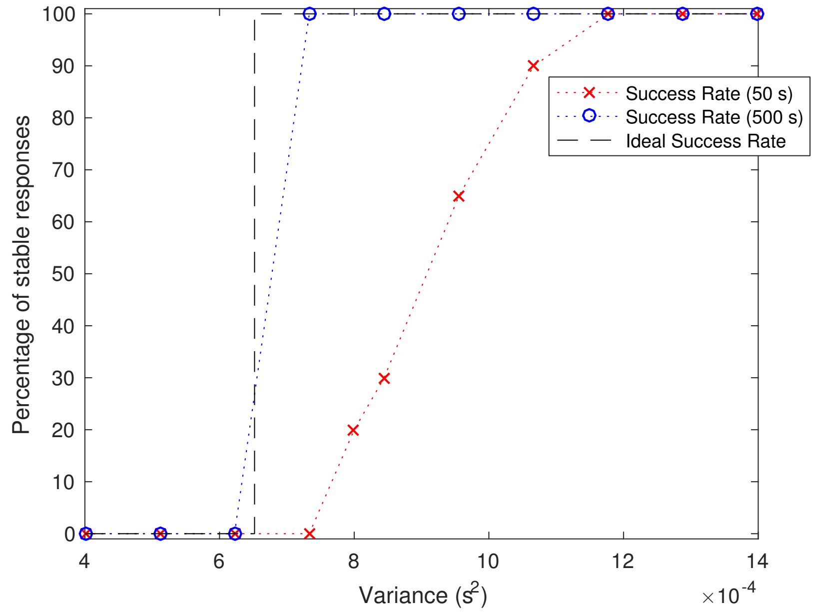


Figure 4.16. Results of simulated responses, using a 50 s time horizon and a 500 s time horizon with a mean step size of 30 ms. As the time horizon is increased, there are less "false negatives" since the responses have sufficient time to settle to a steady state.

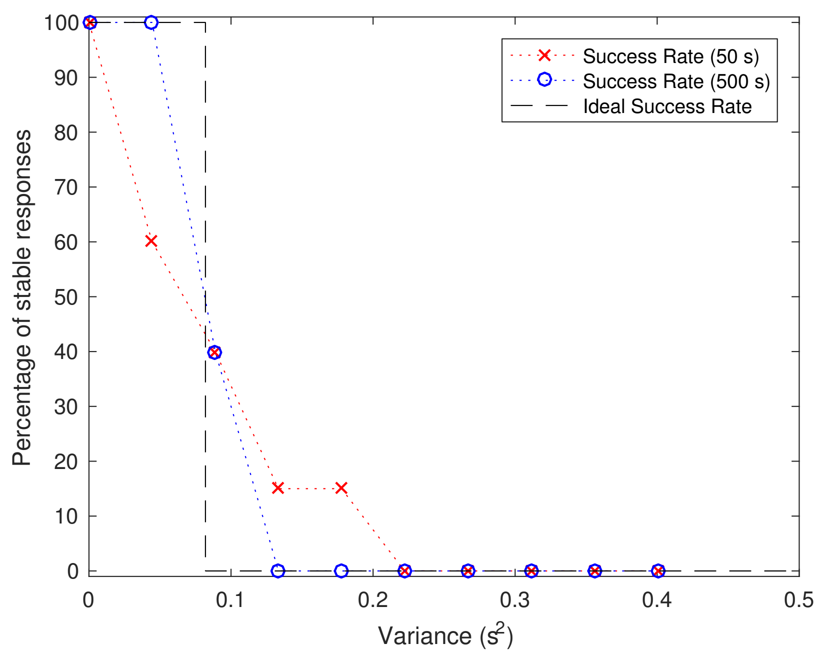


Figure 4.17. Same as Figure 4.16 but with mean step size of 1.2 s.

APPENDIX A

System Responses for SRV-02 Experiment

The following plots show the system response error for beta distributions with a mean of 400 ms.

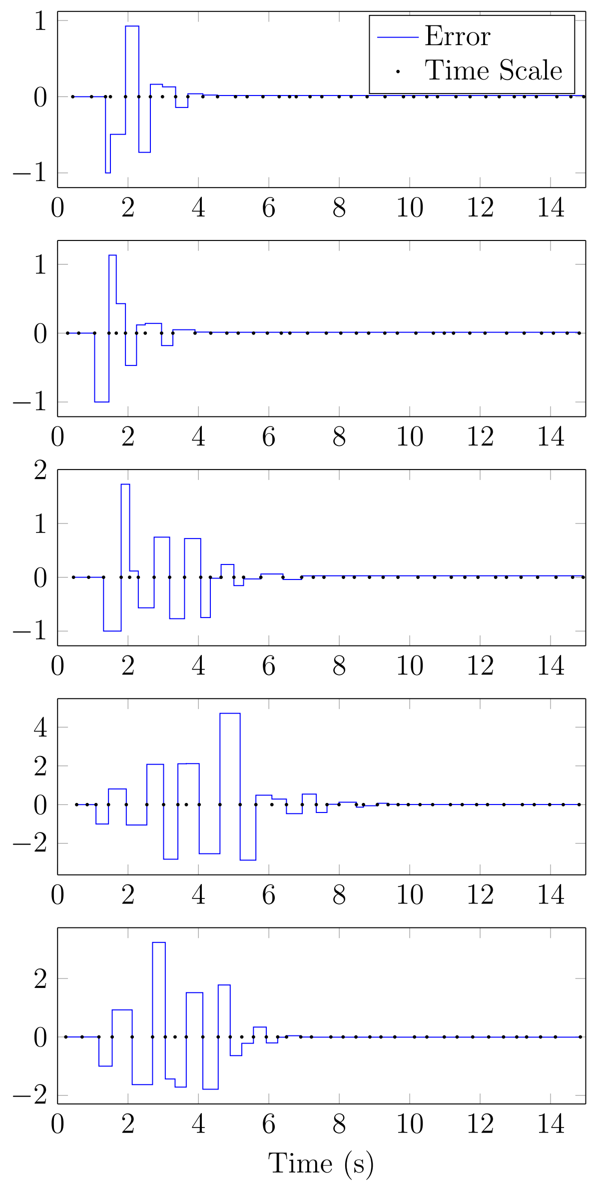


Figure A.1. The system responses have a 99% success rate when the time scale has a variance of 0.01 s^2

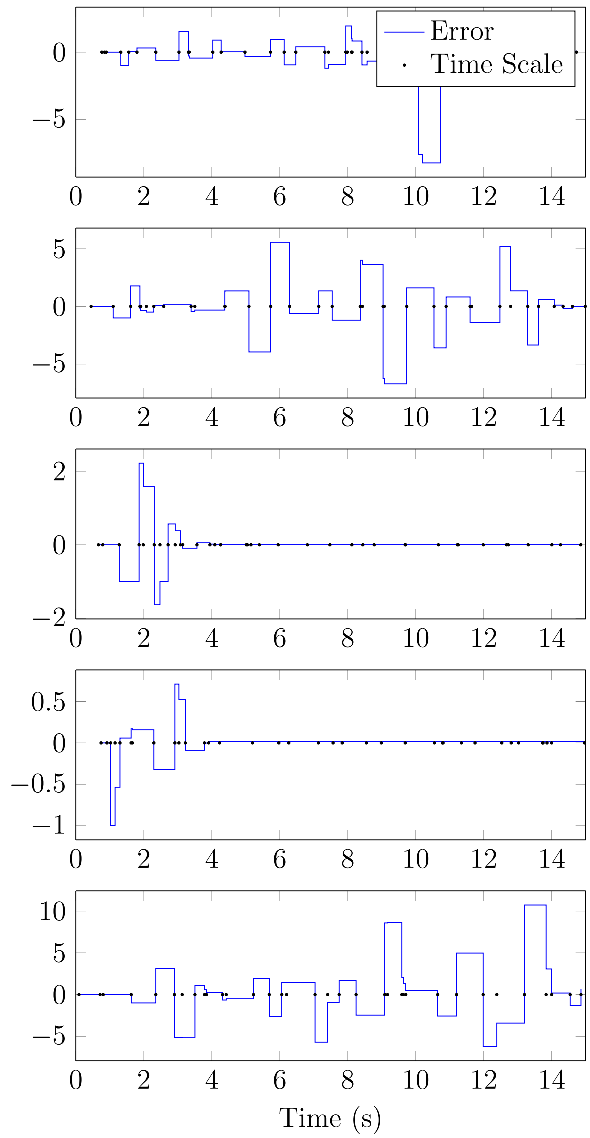


Figure A.2. The system responses have a 55% success rate when the time scale has a variance of 0.08 s^2

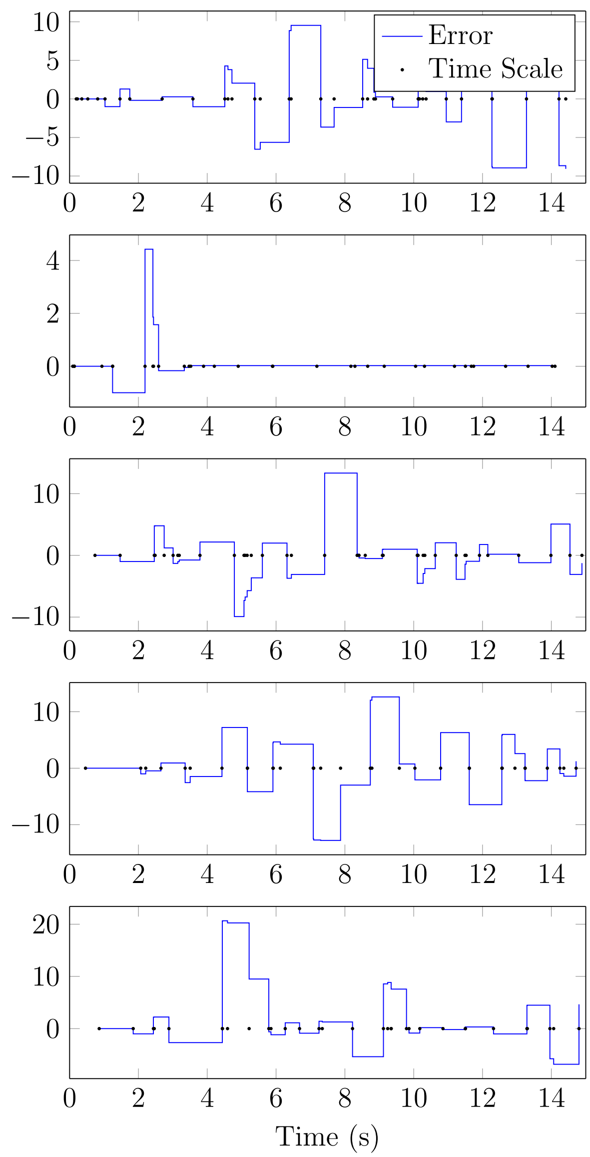


Figure A.3. The system responses have a 16% success rate when the time scale has a variance of 0.12 s^2

APPENDIX B

MATLAB Code

Listing B.1. system_identification.m

```
% Get data from scope and put it in data objects

close all;
t = Data(:,1); v = Data(:,2); theta = Data(:,3); theta_dot = Data(:,4);

t = circshift(t,2); t(1) = 0; t(2) = 0.002; % force t to start at time 0
Ts = t(2)-t(1); % graininess in ms according to model configuration parameters

% put data in iddata objects
datat = iddata(theta,v,Ts);
datat.InputName = 'Voltage'; datat.OutputName = 'Theta';
datat.InputUnit = 'Volts'; datat.OutputUnit = 'rads';
datatd = iddata(theta_dot,v,Ts);
datatd.InputName = 'Voltage'; datatd.OutputName = 'Theta Dot';
datatd.InputUnit = 'Volts'; datatd.OutputUnit = 'rads per second';

% estimate new 1st order transfer function for theta_dot.
tf_theta_dot = tfest(datatd,1);
simplot(tf_theta_dot, datat, datatd, t);

% Compare original transfer function output using new input data
B = 1.2;
C = 1.7;
a = 3.881; b = 9.475;
tf_theta_dot = tf([B*b],[1 C*a]);
theta_dot_est = simplot(tf_theta_dot, datat, datatd, t);

% print results of system identification to csv
Mat = [t theta_dot theta_dot_est];
dlmwrite('system_ident.csv', Mat);
```

Listing B.2. simplot.m

```
function theta_dot_est = simplot( tf_theta_dot, datat, datatd, t )
% This function does a linear simulation for the provided transfer
% functions and then superimposes the measured data

% multiply denominator by s to get tf_theta
    [num,den] = tfdata(tf_theta_dot); den{1} = [den{1} 0];
    tf_theta = tf(num,den);

    figure(1); clf

    theta_dot_est = lsim(tf_theta_dot, datatd.InputData, t); % simulate tf_theta_dot
```

```
    theta_dot_est = circshift(theta_dot_est,2); theta_dot_est(1:2)=0;
    plot(t, theta_dot_est, t, datatd.OutputData);
    legend('simulated␣theta␣dot','actual␣theta␣dot');

end
```

Listing B.3. simulate_model.m

```
clear; close all;
my_setup(); % load necessary constants for servo operation
tstart = tic();
Mean = 0.400; dist = 'B'; totalSims = 5;

simtime = 15; assignin('base', 'simtime', simtime);

fileID = fopen('test.txt','w');
fprintf(fileID,'%7s␣%18s\n', 'Var(s)', 'Percent␣Success');
fclose(fileID);

set_param(0, 'CacheFolder', fullfile('C:','Quanser','QuaRC_simulations','timescale_
set_param(0, 'CodeGenFolder', fullfile('C:','Quanser','QuaRC_simulations','timescal

count = 1; % counter for storing results into vector
for Variance = 0.120:0.010:0.120
    success = 0; fail = 0;
    for simNumber = 1:totalSims

        init_time_scale(Mean, Variance, dist, simtime);

        disp('Building␣model...')
        slbuild('servosim') % build model
        set_param(gcs,'SimulationCommand','connect'); % connect to target
        set_param('servosim','SimulationCommand', 'start') % start the simulation
        fprintf('Starting␣simulation␣No.␣%.0f\n', simNumber);

        pause(simtime+3); % wait for simulation to finish

        % Get data from scope and compute error
        t{simNumber} = Data(:,1); desired_theta = Data(:,2); v = Data(:,3); theta =
        e{simNumber} = theta - desired_theta;

        plot_data();

        if ( abs(e{simNumber}(end-2:end)) < 0.05 )  % check if error for past 3 sam
            title('Success')
            success = success + 1;
            fprintf('Success!\n');
        else
            title('Fail');
            fail = fail + 1;
            fprintf('Fail!\n');
        end
        fprintf('Percentage␣of␣stable␣simulations:␣%.2f\n', 100*success/simNumber)
    end
```

```
    csvData = padcat(zeros(38,1), t{1}, e{1}, t{2}, e{2}, t{3}, e{3}, t{4}, e{4}, t

    dlmwrite('plot_data_120.csv', csvData); % write plot data to a csv file
    results(count,:) = [Variance, 100*success/totalSims];
    fileID = fopen('test.txt','a');
    fprintf(fileID,'%7.3f␣%12.2f\n', results(count,:));
    fclose(fileID);
    count = count + 1;
end
fprintf('Time␣elapsed␣is␣%.2f␣minutes\n', toc(tstart)/60);

figure(2); clf;
plot(results(:,1),results(:,2))
```

Listing B.4. init_time_scale.m

```
function time_scale = init_time_scale(M, V, dist, simtime)
%INIT_TIME_SCALE This function initializes a new time scale for a given mean and
%variance
% d is the number of time scale points
% choose from constant graininess, uniform distribution, beta distribution,
% and gamma distribution
    disp('Initializing␣new␣time␣scale...')
    n = simtime*3; % set number of time scale points to 3 * simulation time (just t
    switch dist
      case 'C' % constant graininess
        mu = M*ones(1,n);

      case 'U' % uniform distribution
        a = M - sqrt(3*V);
        b = M + sqrt(3*V);
        mu = (b-a).*rand(n,1) + a;
        assert(a>=0, 'Minimum␣graininess␣cannot␣be␣negative') % assert that minimum

      case 'B' % beta distribution
               % get shape parameters
        a = ((1-M)/V-1/M)*M^2;
        b = a * (1/M - 1);
        mu = betarnd(a,b,n,1);

        % check if mu has mean and variance close enough to the desired
        % mean and variance
        while abs(mean(mu) - M) >  0.001 || abs(var(mu) - V) > 0.001
            mu = betarnd(a,b,n,1);
        end

      case 'G' % gamma distribution (random number = gamma (k, theta))
               % mean = k*theta, var = k*theta^2
        k = M^2/V;
        theta = V/M;
        mu = gamrnd(k, theta, n, 1);

        while abs(mean(mu) - M) >  0.001 || abs(var(mu) - V) > 0.001
            mu = gamrnd(k, theta, n, 1);
```

```
        end

      otherwise
        error('Please␣select␣a␣valid␣distribution␣for␣discretization.');
    end

    time_scale = timeseries('Time␣Scale'); % initialize timeseries object

    time(1) = mu(1);
    data(1) = -1;
    for j = 2:n
        time(j) = time(j-1) + mu(j); % create time scale points from graininesses
        data(j) = (-1)^j; % make every other element positive for sake of trigger
    end

    time_scale.Time = time';
    time_scale.Data = data';
    time_scale.UserData = mu; % store the graininess in userdata
    assignin('base', 'time_scale', time_scale); % assign in workspace so that simul
end
```

Listing B.5. plot_data.m

```
disp('Plotting␣data...')
% set to 1 if uniform graininess or 0 if distribution

opt = 0;

% plot with time scale points shown on x axis
% get plot number
n = floor((simNumber-1) / 5) + 1;
figure(n);

if (opt)
    subplot(2,1,1);
    Ts = t(2)-t(1); % get the step size ( only if uniform graininess)
else
    p = mod(simNumber-1,5);
    subplot(5,1,p+1); % plot data in jth subplot
end

stairs(t{simNumber},e{simNumber});
hold on
plot(t{simNumber}, 0, 'k.') % plot the time scale

%     if (simNumber == 1)
%         legend('Reference Theta','Theta','Control Signal');
%         if opt
%             str = sprintf('Discretized System Response for mu = %.f ms', Ts*1000)
%         else
%             str = sprintf('Discretized System Response for mean = %.f ms and var 
%         end
%         title(str);
%     end

if(opt)
    plot_hilger(); % plot hilger circle and poles
end
```

Listing B.6. plot_hilger.m

```
% plot hilger circle and poles
subplot(2,1,2)
circle(-1/Ts,0,1/Ts); hold on;

a = 6.598; b = 11.37;% from the tf identification
k = 3.5; % gain

poles = discretize(a,b,k,Ts); % get the pole locations for the discretized system

r = real(poles);
im = imag(poles);
plot(r(1),im(1),'k*',r(2),im(2), 'k*') % plot the poles

plot([-1/Ts -1/Ts], [-1/Ts 1/Ts], 'k:'); % plot the vertical line
```

```
axis('square'); x = xlim; % get the x axis limit
plot([0 x(1)], [0 0], 'k:'); % plot the x-axis
str = sprintf('Poles␣for␣mu␣=␣%.f␣ms', Ts*1000);
title(str)
FigHandle = figure(1);
set(FigHandle, 'Position', [100, 100, 670, 520]);
```

Listing B.7. circle.m

```
function circle(x,y,r)
%x and y are the coordinates of the center of the circle
%r is the radius of the circle
%0.01 is the angle step, bigger values will draw the circle faster but
%you might notice imperfections (not very smooth)
    ang=0:0.01:2*pi;
    xp=r*cos(ang);
    yp=r*sin(ang);
    plot(x+xp,y+yp);
end
```

Listing B.8. discretize.m

```
function poles = discretize(a,b,k,mu)
%DISCRETIZE find the poles of the discretized system
%   This function takes as argument the transfer function representation of
%   the system, the gain, and the graininess. It discretizes the system and
%   then returns the poles of that discretized system
% chosen for states x1 = theta and x2 = theta dot
    A = [0 1; 0 -a]; B = [0 b]'; C = [1 0]; D = [0];
    sys = ss(A,B,C,D); % our open loop system
    K = [k 0];
    Bn = [0 0]';
    % Now to discretize on constant graininess
    % To find A and B we make use of the Cayley-Hamilton Theorem
    % Since expc(mu*A) = R(mu*A) = a1 + a2*x, then we can solve for a1 and a2
    % in the following system of equations:
    % -- expc(eval1) = alph0 + alph1*eval1
    % -- expc(eval2) = alph0 + alph1*eval2
    alph0 = 1;
    tmp = (exp(-a*mu)-1)/(-a*mu) - 1;
    alph1 = tmp/ -a;
    expc = alph0*eye(2) + alph1*A;
    sA = expc*A;
    sB = expc*B; % Discretized B matrix
    sysd = ss(sA-sB*K,Bn,C,D); % our discretized system with A = sA-sB*K
    poles = vpa(pole(sysd)); % poles *should* be in the LHP
end
```

BIBLIOGRAPHY

[1] Dylan R Poulsen, John M Davis, Ian A Gravagne, and RJ Marks. On the stability of mu-varying dynamic equations on stochastically generated time scales. In *System Theory (SSST), 2012 44th Southeastern Symposium on*, pages 18–23. IEEE, 2012.

[2] William L Brogan. *Modern Control Theory*. Pearson Education India, 1985.

[3] Christian Pötzsche, Stefan Siegmund, and Fabian Wirth. A spectral characterization of exponential stability for linear time-invariant systems on time scales. *Discrete Contin. Dyn. Syst.*, 9:1223–1241, 2003.

[4] Dylan R Poulsen, John M Davis, and Ian A Gravagne. Observer based feedback controllers on stochastic time scales. *IEEE Proc. SSST*, pages 104–107, 2013.

[5] Dylan R Poulsen, John M Davis, and Ian A Gravagne. Stochastic time scales: quadratic lyapunov functions and probabilistic regions of stability. In *IEEE Proceedings of the 45th Meeting of the Southeastern Symposium on System Theory, Baylor University*, pages 98–103, 2013.

[6] Dylan R Poulsen, John M Davis, and Ian A Gravagne. Is deterministic real time control always necessary? a time scales perspective. In *ASME 2014 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2014.

[7] Dylan R Poulsen. *Stability and Control on Stochastic Time Scales*. Ph.d. dissertation, Baylor University, 2015.

[8] Matthew D Mosley, Ian A Gravagne, and John M Davis. A solution of the euler-bernoulli flexible rotating arm. In *Proceedings of the American Society for Engineering Education Gulf Southwest Conference (ASEE GSW)*, number 132. American Society for Engineering Education, ASEE, March 2016.

[9] Matthew D Mosley, Ian A Gravagne, Dylan R Poulsen, and John M Davis. Experimental investigation of a time scales-based stability criterion over finite time horizons. In *Proceedings of the ASME 2015 Dynamic Systems and Control Conference* Ame [24].

[10] Richard C Dorf and Robert H Bishop. *Modern control systems*. Pearson (Addison-Wesley), 12 edition, 1998.

[11] Lotfi Asker Zadeh and Charles A Deoser. *Linear system theory*. Robert E. Krieger Publishing Company, 1976.

[12] Morris W Hirsch, Stephen Smale, and Robert L Devaney. *Differential equations, dynamical systems, and an introduction to chaos*, volume 60. Academic press, 2013.

[13] Wael F Asaad and Emad N Eskandar. Achieving behavioral control with millisecond resolution in a high-level programming environment. *Journal of neuroscience methods*, 173(2):235–240, 2008.

[14] Stefan Hilger. *Ein maßkettenkalkül mit anwendung auf zentrumsmannigfaltigkeiten*. PhD thesis, 1989.

[15] Martin Bohner and Allan Peterson. *Dynamic equations on time scales: An introduction with applications*. Springer Science & Business Media, 2001.

[16] T Gard and J Hoffacker. Asymptotic behavior of natural growth on time scales. *Dynamic Systems and Applications*, 12(1/2):131–148, 2003.

[17] John M Davis, Ian A Gravagne, RJ Marks, and Billy J Jackson. Regions of exponential stability for lti systems on nonuniform discrete domains. In *System Theory (SSST), 2011 IEEE 43rd Southeastern Symposium on*, pages 37–42. IEEE, 2011.

[18] Ben Allen. Experimental investigation of a time scales linear feedback control theorem. Master's thesis, Baylor University, 2007.

[19] Dylan R Poulsen, Ian A Gravagne, and John M Davis. Numeric and analytic investigations of mean-square exponential stability for stochastically timed systems. In *Proceedings of the ASME 2015 Dynamic Systems and Control Conference* Ame [24].

[20] Christopher D Rahn. *Mechatronic control of distributed noise and vibration*. Springer, 2001.

[21] John M. Davis. *Introduction to Applied Partial Differential Equations*, chapter 2, pages 67–68. W. H. Freeman and Company, 2012.

[22] Chung-Feng Jeffrey Kuo and Lin Shu-Chyuarn. Modal analysis and control of a rotating euler-bernoulli beam part i: Control system analysis and controller design. *Mathematical and computer modelling*, 27(5):75–92, 1998.

[23] Stanley J Farlow. *Partial differential equations for scientists and engineers*. Courier Corporation, 2012.

[24] *Proceedings of the ASME 2015 Dynamic Systems and Control Conference*, October 2015. American Society of Mechanical Engineers, ASME.