ABSTRACT

Exploration of Stability Between Power Grids on Time Scales of Different Variance

James Worsham

Director: Ian Gravagne, Ph.D

Differential equations are useful tools for evaluating the theoretical long-term behavior of systems, particularly control systems. But in real world applications, unlike continuous differential equations, controllers are often unable to react continuously or immediately. For example, the dynamics of power flow between local power grids in Southern California and the Pacific Northwest are maintained by controllers located midway between these areas that use the internet to relay information. The feedback of these controllers is subject to both nonuniform spacing and time delay. Using the theory of Time Scale calculus, as well as the Stability-Almost-Surely Criteria, I will evaluate the stability of power dynamics on different time scales.

APPROVED BY DIRECTOR OF HONORS THESIS:

_____

Dr. Ian Gravagne, Department of Electrical Engineering

APPROVED BY THE HONORS PROGRAM:

_____

Dr. Elizabeth Corey, Director

DATE: _____

EXPLORATION OF STABILITY BETWEEN POWER GRIDS ON TIME SCALES OF DIFFERENT VARIANCE

A Thesis Submitted to the Faculty of

Baylor University

In Partial Fulfillment of the Requirements for the

Honors Program

By

James Worsham

Waco, Texas

May, 2019

TABLE OF CONTENTS

CHAPTER ONE

Motivating Theory

Before addressing power in dynamics in detail, it is necessary to define some terminology and basic theory for the mathematics that motivate the problem. These include an overview of time scale theory and dynamic equations and an explanation of the mathematical stability criterion known as stability-almost-surely.

Basic Timescale Theory

Using the theories of calculus, mathematicians can work with and solve differential equations, which relate a function to its derivaties. For example, consider the canonical example:

$$y'(x) - 2y(x) = 0 \tag{1}$$

This can be solved as such: Let $y(x) = Ce^{2x}$, where C is an unknown constant.

$$y' = 2Ce^{2x}$$

$$2Ce^{2x} - 2Ce^{2x} = 0$$

The value of C can be determined only in the presence of initial conditions that fix its value. For the preceding example, suppose it is known that $y(0) = 1$. Then, substituting the solution,

$$y(0) = Ce^0 = C = 1$$

These results can be useful in physics applications, particularly for modeling systems with continuous time. However, real world hardware is often limited in taking measurements continuously. For example, a voltage monitoring system might only be

able to take measurements once a second. A second branch of mathematics, difference equations, deals with change as it occurs discretely.

While differential equations are applications of functions, difference equations are applications of sequences, usually of real numbers. Given a sequence $\{a_t\}_{t=1}^{\infty}$, the difference is defined as such:

$$\Delta a_t = a_{t+1} - a_t \tag{2}$$

The second difference, and subsequent ones are also defined:

$$\Delta^2 a_t = \Delta a_{t+1} - \Delta a_t \tag{3}$$

Which is, equivalently,

$$\Delta^2 a_t = a_{t+2} - 2a_{t+1} + a_t$$

With such a definition in mind, difference equations can be solved just as differential equations can. For example, the difference equation analogous to (1) is as such:

$$\Delta a_t - 2a_t = 0 \tag{4}$$

Substituting the terms for Δa_t yields the result:

$$a_{t+1} - a_t - 2a_t = 0$$

$$a_{t+1} = 3a_t \tag{5}$$

Therefore, the relation between terms of $\{a_t\}$ is established, but without at least one fixed term of the sequence, it cannot be determined. Suppose it is known that $a_1 = 1$.

Then $a_2 = 3a_1 = 3$, $a_3 = 3a_2 = 9$, and we can thus determine the terms of $\{a_n\}$:

$$\{a_t\} = \{1, 3, 9, 27, ...\}$$

And thus,

$$a_t = 3^{t-1} \tag{6}$$

Such results, while useful, are again limited in their scope by the fact that the interval between terms is fixed and must be the same in all cases. The similarity of the solutions of Equations (4) and (1) suggests a connection between the two techniques. The unification of the two fields is known as Dynamic Equations on Time Scales (DETS). "Dynamic Equations" is a general term that includes difference equations and differential equations.

While differential equations occur on $\mathbb{R}$, the real line, and difference equations occur on $\mathbb{Z}$, the set of integers, or a similar set with equidistant subsequent elements, dynamic equations occur on $\mathbb{T}$, a time-scale which can be any closed subset of $\mathbb{R}$. Closed, in this sense, means that if $\mathbb{T}$ includes an interval on $\mathbb{R}$ it must include the endpoints of this interval. For example, $\mathbb{T}$ cannot contain the interval $(0, 1)$ unless it also contains 0 and 1.

For any point $t \in T$, the value $\sigma(t)$, the forward jump operator, is defined as the least upper bound of all values in $\mathbb{T}$ to the right of (i.e. greater than) t. For the case where $\mathbb{T} = \mathbb{R}$, then, $\sigma(t) = t$, and in the case where $\mathbb{T} = \mathbb{Z}$, $\sigma(t) = t + 1$. The value $\rho(t)$, the backward jump operator, is defined as the greatest lower bound of all values in $\mathbb{T}$ to the left of t. In the preceding cases, $\rho(t) = t$ and $\rho(t) = t - 1$, respectively.

We define the graininess of $\mathbb{T}$ at t as $\mu(t) = \sigma(t) - t$. In the preceding examples,

for $\mathbb{T} = \mathbb{R}$,

$$\mu(t) = \sigma(t) - t = t - t = 0 \tag{7}$$

For $\mathbb{T} = \mathbb{Z}$,

$$\mu(t) = \sigma(t) - t = (t+1) - t = 1 \tag{8}$$

We can define a generalized, delta derivative $f^{\Delta}(t)$ as such: if a function is differentiable at a point, then for every $\epsilon < 0$ there is an associated neighborhood $N_r(t)$ such that

$$|f(\sigma(t)) - f(s) - f^{\Delta}(t)(\sigma(t) - s)| \leq \epsilon|\sigma(t) - s| \tag{9}$$

for all s in $N_r(t)$. Back to our canonical examples, for $\mathbb{T} = \mathbb{R}$,

$$|f(t) - f(s) - f^{\Delta}(t)(t - s)| \leq \epsilon|t - s|.$$

Assuming $s < t$,

$$\left|\frac{f(t) - f(s)}{t - s} - f^{\Delta}(t)\right| \leq \epsilon \tag{10}$$

which is equivalent to the typical derivative. For the $s > t$ case,

$$\left|\frac{f(t) - f(s)}{s - t} + f^{\Delta}(t)\right| \leq \epsilon$$

$$\left|f^{\Delta}(t) - \frac{f(t) - f(s)}{t - s}\right| \leq \epsilon,$$

which is an equivalent formulation. Thus, for $\mathbb{T} = \mathbb{R}$, $f^{\Delta}(t) = f'(t)$.

For the case where $\mathbb{T} = \mathbb{Z}$, applying the definition, (9) becomes

$$|f(t+1) - f(s) - f^{\Delta}(t)(t + 1 - s)| \leq \epsilon|t + 1 - s|.$$

Taking the limit as $s \to t$ by evaluating s at t since the time scale is discrete, the

right side is set to 0.

$$|f(t+1) - f(t) - f^{\Delta}(t)| = 0$$

$$f^{\Delta}(t) = f(t+1) - f(t) = \Delta f(t) \tag{11}$$

Therefore, the delta derivative is equivalent to the difference operator for $\mathbb{T} = \mathbb{Z}$.

Stability

Consider a generalized version of (1) with 2 replaced by an arbitrary constant $\lambda \in \mathbb{C}$:

$$y'(x) - \lambda y(x) = 0 \tag{12}$$

Such an equation has, as before, a solution given by $y(x) = Ce^{\lambda x}$. Note the behavior of this solution: for $\lambda = \alpha + \beta\imath$ and $\alpha < 0$,

$$\lim_{x \to \infty} y(x) = \lim_{x \to \infty} Ce^{\alpha x} e^{\beta \imath x} = 0 \tag{13}$$

since $e^{\beta \imath x}$ is bounded and $e^{\alpha x}$ approaches 0. For the case where $\alpha = 0$ (barring the trivial case where $\lambda = 0$ resulting in the solution being any constant function), the equation is controlled by $e^{\beta \imath x}$ and therefore remains in a circle of radius C indefinitely. Finally, for the case where $\alpha > 0$, the equation is again controlled by $e^{\alpha x}$, which increases indefinitely.

Depending on how stability is defined, different parameters will result in stability or instability. For example, were a system modeled by a dynamic equation such as (12), if stability is defined as boundedness of solutions, then any λ with $\alpha \leq 0$ results in a solution bounded by C. If it is defined by solutions approaching 0, then the strict inequality $\alpha < 0$ is needed for stability.

This example also allows for a local definition of stability that results in global

behavior. For (12), if $|y(x)|$ decreases monotonically as x increases, then the solution is stable based on our second criteria. However, this definition is not sufficient because not every system will behave ideally. The norm of the solution to a dynamic equation may or may not decrease monotonically, depending on the time scale.

As an example, consider the dynamic equation analogous to (12) on a discrete time scale:

$$y^{\Delta}(t) - \lambda y(t) = 0 \tag{14}$$

$$T = \{0, 1, 2, 5, 6, 7, 10, 11, 12, ...\}$$

For this time scale, the graininess is defined as follows:

$$\mu(T) = \{1, 1, 3, 1, 1, 3, ...\}$$

For a discrete time scale, the delta derivative takes the form $\frac{y(\sigma(t))-y(t)}{\mu(t)}$. Substituting this into (14) gives:

$$\frac{y(\sigma(t)) - y(t)}{\mu(t)} = \lambda y(t)$$

$$y(\sigma(t)) = \lambda y(t)\mu(t) + y(t) = (\lambda\mu(t) + 1)y(t)$$

Therefore, if the initial condition $y(t_0)$ is known, for our particular time scale, we can solve for $y(t_1)$ and subsequent values of y(t) as such:

$$y(t_1) = (1 + \lambda\mu(t_0))y(t_0) = (1 + \lambda)y(t_0)$$

$$y(t_2) = (1 + \lambda\mu(t_1))y(t_1) = (1 + \lambda\mu(1))(1 + \lambda)y(t_0) = (1 + \lambda)(1 + \lambda)y(t_0)$$

$$y(t_3) = (1+\lambda\mu(t_2))y(t_2) = (1+\lambda\mu(2))(1+\lambda)(1+\lambda)y(t_0) = (1+3\lambda)(1+\lambda)(1+\lambda)y(t_0)$$

Applying the absolute value to the terms of the solution (for the sake of seeing if

the solution converges to 0), the $3k$th term is thus found as such:

$$|y(t_{3k})| = |(1+3\lambda)^k| * |(1+\lambda)^{2k}| * |y(t_0)|$$

We rewrite this product, generally, as such:

$$|y(t_{3n})| = \prod_{k=1}^{n} |(1+\mu_k\lambda)^{d_k}| * |y(t_0)|, \tag{15}$$

where μ_k is the graininess and d_k is the associated frequency for each value of k. We are seeking the stability of the system, and this will only occur if the right hand term (sans the initial condition) is less than 1. So our condition for stability is as such:

$$\prod |(1+\mu_k\lambda)^{d_k}| < 1 \tag{16}$$

Taking the logarithm of this condition, we get the sum:

$$\log(\prod |(1+\mu_k\lambda)^{d_k}|) = \sum d_k \log|(1+\mu_k\lambda)| < 0 = \log(1) \tag{17}$$

Notice that as long as d_k remains proportionally the same for all values of k, the inequality will still hold. Then we can replace d_k with the values of a probability distribution function of the associated graininess values. In the case where d_k is continuous, we replace the sum with an integral:

$$\int_0^\infty f(\mu) \log|(1+\mu\lambda)| d\mu < 0, \tag{18}$$

where $f(\mu)$ is the continuous PDF (probability distribution function. Thus, the conditions for convergence of a first-degree dynamic equation on a discrete timescale can be expressed as either the sum (17) or the integral (18) depending on the associated PDF of graininess frequency.

In the case of higher-order dynamic equations, we express them as a matrix equation $y^{\Delta} = Ay$, and rather than using the euclidean norm, we use the norm of the eigenvalue λ_i of A that maximizes $|1 + \mu_k \lambda_i|$ and the stability conditions become

$$\sum d_k \log |(1 + \mu_k \lambda_i)| < 0 \tag{19}$$

and

$$\int_0^{\infty} f(\mu) \log |(1 + \mu \lambda_i)| d\mu < 0 \tag{20}$$

Finally, in the case of a higher-order dynamic equation in which the eigenvalues vary with μ, the conditions vary accordingly, becoming

$$\sum d_k \log |(1 + \mu_k \lambda_i(\mu))| < 0 \tag{21}$$

and

$$\int_0^{\infty} f(\mu) \log |(1 + \mu \lambda_i(\mu))| d\mu < 0 \tag{22}$$

The outcome of this criterion for stability is that the eigenvalues that make the system stable (for the invariant case) are not all of those in the left-half complex plane but in some region of $\mathbb{C}$. For the sums themselves, the presence of some positive terms is possible as long as the frequency remains low. In other words, the value inside the log may be greater than 1 for certain values of μ as long as they don't occur too often.

The primary limitation of this criterion for stability, henceforth denoted as "stability-almost-surely," is that it does not account for performance, i.e. the norm can reach very high values before converging to 0 after a long time [7].

CHAPTER TWO

Background of the Problem and Early Results

Maintaining stability in power systems is critically important, particularly between geographic power areas. As the flow of electric power oscillates and is subject to variation due to environmental factors, stability is constantly imposed with systems that measure the flow of power and apply damping to prevent feedback that could result in a voltage spike or another condition that can cause the failure of electrical components [2]. An additional issue that can arise when attempting to impose stability is the presence of time delay between measurements and the feedback of control systems [3].

In light of the problem of time delay, the flow of electricity between regional power grids, such as those of the northwest and southwest regions of the U.S., must be monitored to maintain stability. But even if the power is monitored, the transmission of such data between where it is measured (either the source power grids or a point between them) is subject to the delay of whatever transmission medium (e.g. internet) is used. Research has already been performed on what the criteria for stability of a two-area power system on a continuous time scale subject to constant delay are (how much delay is catastrophic, and for how strong of a feedback signal).

Linear Dynamics

The linear dynamics for the idealized two-area power system illustrated by Figure 1 are expressed as

$$\dot{x}(t) = Ax(t) + B_D P_D + B_L P_L \tag{23}$$

In this case, $x(t)$ is the matrix $[\Delta\delta(t)^T \ \Delta\omega(t)^T]^T$, where $\Delta\delta(t)$ is the vector of the rotor angles and $\Delta\omega(t)$ is the vector of the angular velocities of the two idealized generators that would represent the power grids in question. A is a matrix of torques that synchronize the two systems, P_D is the matrix of power injections, which are used to control the system, and P_L is the matrix of aggregated loads, which are used to model disturbances. Finally, B_D and B_L reflect physical properties of the system.

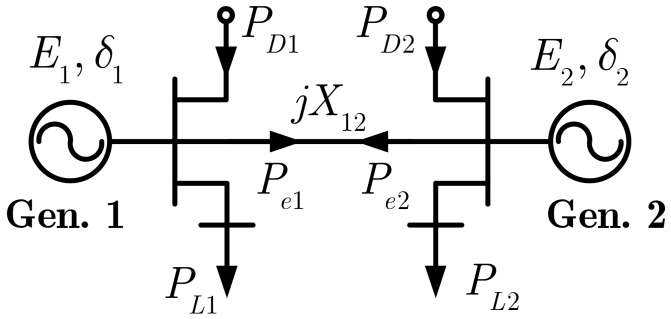

Figure 1: A two-area power system represented with one generator in each area, from [4].

We define $P_D = [P_{D1} \ P_{D2}]^T$ as such:

$$P_{D1}(t) = -k_d(\Delta\omega_1(t) - \Delta\omega_2(t)) = -P_{D2}(t) \tag{24}$$

The value of k_d, controller gain, can be chosen. When the additional variable of time delay is introduced, these dynamics become

$$P_{D1}(t) = -k_d(\Delta\omega_1(t-\tau_1) - \Delta\omega_2(t-\tau_2)) = -P_{D2}(t) \tag{25}$$

In this case, τ_1 represents the time delay in measurement from the area of injection, and τ_2 is the delay in measurement from an area between the two power grids. The research performed thus far utilizes of one of two controllers, an HVDC (High-Voltage Direct Current) transmission line or a pair of energy storage devices. In either case,

the dynamics of P_D are given as defined in (25). P_L is defined differently for the two controllers, but generally result in stability for the same parameters.

The results of experimental simulations indicate that for smaller, equivalent time delays (i.e. $\tau_1 = \tau_2 = 0.4$), increasing values of k_d increase damping and result in stability more quickly. However, an increase of τ_1 and τ_2 to 0.6 results in stability for lower values of k_d and instability for higher values. For asymmetric time delays, as long as one delay value is very low, stability is maintained, but the threshold at which it is maintained becomes lower for larger values of k_d: for $k_d = 0.25$, stability is maintained for at least one time delay less than 0.2, but for $k_d = 1$, stability is only maintained for at least one time delay less than 0.05.

What is still unclear is whether these results will hold for a different timescale. The simulations used thus far assume a continuous feedback system, but actual control systems can only update a limited number of times a second, depending on the processing speed of the control system. In addition, counter to the simulations performed thus far, the time delay is not constant but subject to variation depending on usage and other factors.

Stochastic Timescale Results

Incorporating a stochastic timescale into the system allows for a more accurate prediction of the time delays that result in stability. The dynamics described previously were built into a Simulink model with the stability criteria defined in [4] on a continuous timescale, with controller gain $k_d = 4$, resulting in the region of stability described by Figure 2 (stability indicated by the yellow area):

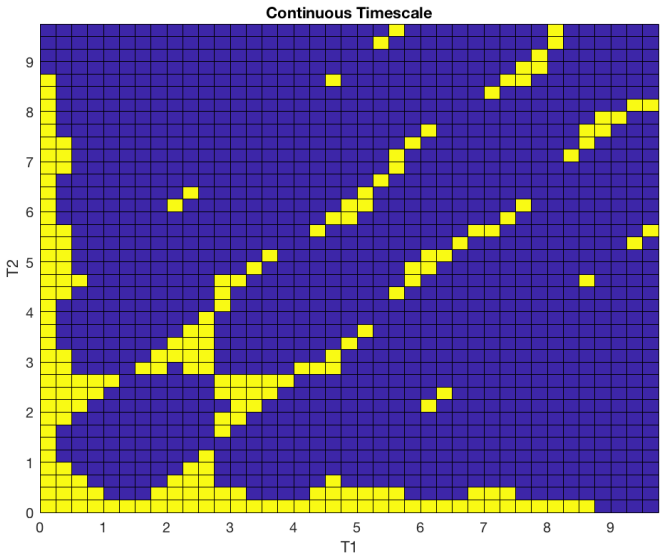

Figure 2: Region of stability for continuous timescale, $k_d = 4$

This is in line with the region found in [4]. The expectation is that introducing a stochastic timescale will result in a small region of stability due to decreased efficiency of the control system. Such control systems use internet data links to deliver data. The delays between packets on such networks have historically been assumed to be Pascal distributed [8], and the real world control systems operate at a frequency of 60 Hz. Therefore, as Pascal-distributed time scales correspond to exponentially distributed time intervals, an exponential distribution with mean $\mu = \frac{1}{60}$ was chosen. The region of stability for this delay distribution is seen in Figure 3.

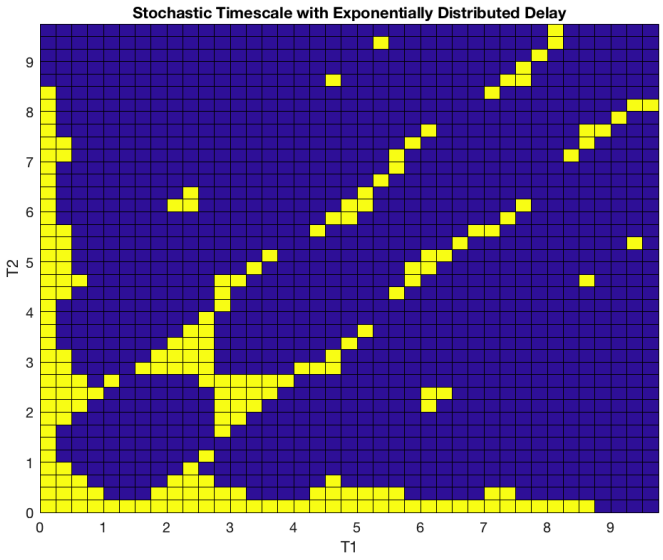

Figure 3: Stability region for stochastic timescale with spacing described by exponential distribution, $\mu = 1/60$, and $k_d = 4$

While such distributions have been used historically, recent research indicates that packet delay has higher variance than previously expected. As such, a distribution with higher variance may provide a more accurate region of stability [9]. The distribution with higher variance chosen was a Gamma distribution with $k = \frac{1}{2}$, $\theta = \frac{1}{30}$. This distribution still has the desired mean of $\frac{1}{60}$ but has a variance of $k\theta^2 = (\frac{1}{2})(\frac{1}{30})^2 = \frac{1}{1800}$, twice as much as the exponential distribution: $(\frac{1}{60})^2 = \frac{1}{3600}$. The probability distribution functions are illustrated in Figure 4.

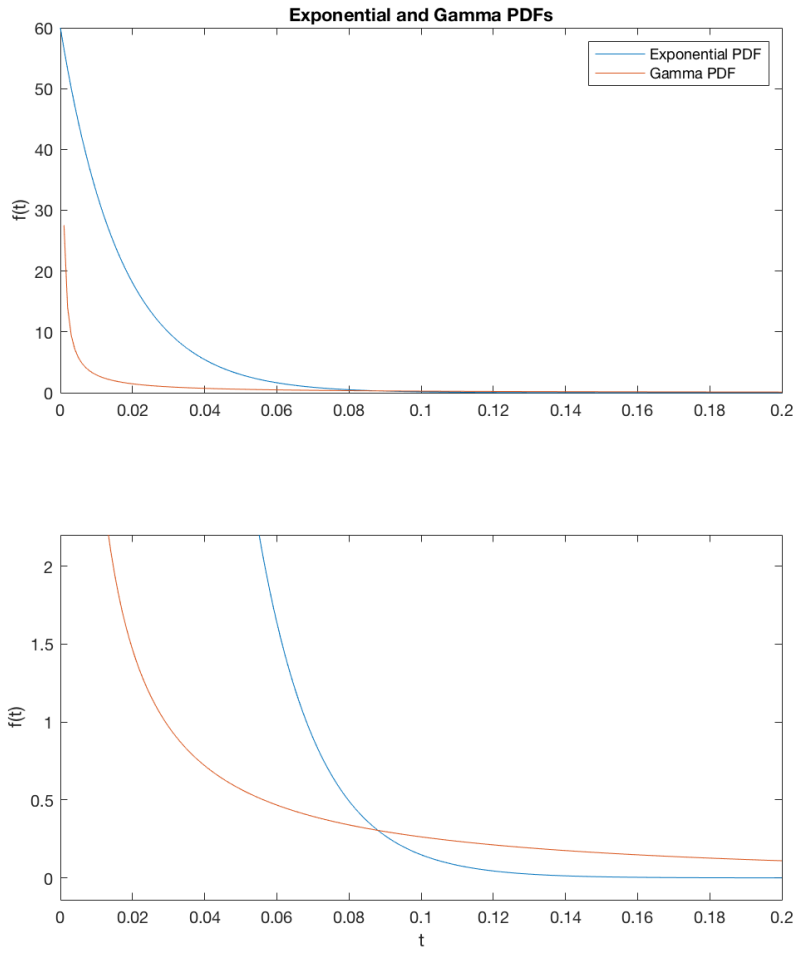

Figure 4: Graphs of the exponential and gamma probability distribution functions chosen. The second graph illustrates the heavier tail and higher variance of the gamma distribution.

The region of stability for the gamma distribution is seen in Figure 5.

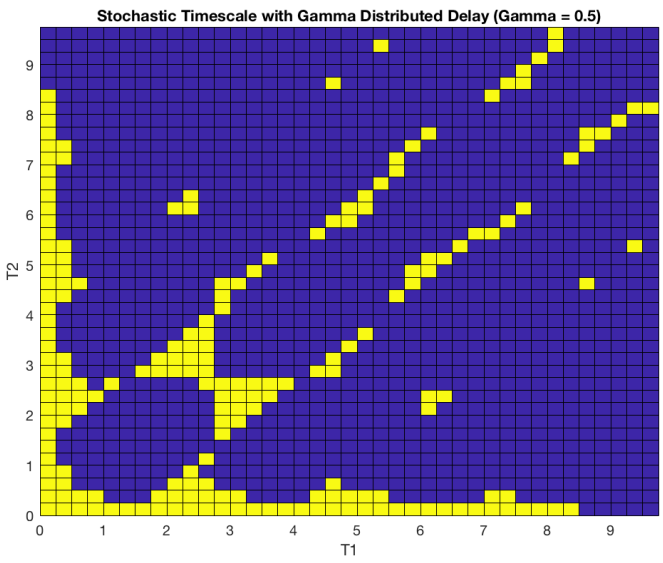

Figure 5: Stability region for stochastic timescale with spacing described by gamma distribution, $\theta = 1/30$, $k = 1/2$ and $k_d = 4$

While these results are illuminating in terms of stability's dependence on the stochastic timescale, a complete picture of packet delays is harder to capture, as the simulations thus far assume constant real-world packet delays on a stochastic timescale, while in reality, time delays themselves change over time. For example, a set of delay parameters that result in instability over a period of 30 seconds might not produce that result if only attained for a moment. Therefore, further research into variable time delays is needed.

CHAPTER THREE

Linearization and Mathematical Stability Analysis

To be able to evaluate stability mathematically using stability-almost-surely rather than with the engineering criteria defined in [4], we must convert the Simulink model used thus far into a linearized matrix differential equation.

Linearization Results

The work done in [5] shows that this equation has the form

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{26}$$

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

where A is a 4x4 matrix of the form

$$A = \begin{bmatrix} \mathbf{0} & \Omega I \\ -(2H)^{-1}T_s & -(2H)^{-1}(D + R^{-1}) \end{bmatrix}$$

B is a 4x2 matrix of the form

$$B = \begin{bmatrix} \mathbf{0} \\ (2H)^{-1} \end{bmatrix}$$

and K is a 2x4 matrix of the form

$$K = \begin{bmatrix} 0 & 0 & -k_d & k_d \\ 0 & 0 & k_d & -k_d \end{bmatrix}$$

The value Ω, as well as the 2 by 2 diagonal matrices D, R, and H are determined by parameters of the system, and $x(t)$ is the state vector containing the phase angle

and angular velocity of the two idealized matrices. Using linearization tools provided by Matlab, we found A and B to be defined as:

$$A = \begin{bmatrix} 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \\ -14 & 14 & -0.02 & 0 \\ 14 & -14 & 0 & -0.02 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$$

This is enough to evaluate stability on a continuous timescale, but to be able to evaluate on a stochastic timescale, we must define a generalized dynamic equation based on the graininess of the timescale. As a system of differential equations, according to [6], the solution to $x(t)$ can be written as such:

$$x(t) = e^{A(t-t_k)}x(t_k) + \int_{t_k}^{t} e^{A(t-\tau)}Bu(\tau)d\tau \tag{27}$$

where t_k is a point in our timescale. Due to the physical constraints of our problem, the system's input is a sample-and-hold type of signal, so $u(t) = u_k$ for $t \in [t_k, t_{k+1})$. Then (27) becomes

$$x(t) = e^{A(t-t_k)}x(t_k) + \int_{t_k}^{t} e^{A(t-\tau)}d\tau Bu_k$$

$$x(t) = e^{A(t-t_k)}x(t_k) + A^{-1}(e^{A(t-t_k)} - I)Bu_k$$

Now, returning to the original definition of $x^{\Delta}(t)$

$$x^{\Delta}(t) = \frac{x(\sigma(t)) - x(t)}{\mu(t)} \tag{28}$$

Evaluating $x(\sigma(t))$ in (28) results in

$$x^{\Delta}(t) = \frac{e^{A(\sigma(t)-t)}x(t) + A^{-1}(e^{A(\sigma(t)-t)} - I)Bu(t) - x(t)}{\mu(t)}$$

And since $\mu(t) = \sigma(t) - t$, then

$$x^{\Delta}(t) = \frac{e^{A\mu(t)}x(t) + A^{-1}(e^{A\mu(t)} - I)Bu(t) - x(t)}{\mu(t)}$$

Due to our control paradigm, $u(t)$ can be replaced with $Kx(t)$.

$$x^{\Delta}(t) = \frac{e^{A\mu(t)}x(t) + A^{-1}(e^{A\mu(t)} - I)BKx(t) - x(t)}{\mu(t)}$$

Factoring out $x(t)$ and rearranging, we get

$$x^{\Delta}(t) = \left[\frac{e^{A\mu(t)} - I}{\mu(t)}\right]\big[I + A^{-1}BK\big]x(t) = \left[\frac{e^{A\mu(t)} - I}{\mu(t)}\right]A^{-1}\big[A + BK\big]x(t)$$

We further simplify by replacing the first two terms with the *expc* function of $\mu(t)A$, which is a convergent power series for the matrix exponential. Then $x^{\Delta}(t)$ becomes

$$x^{\Delta}(t) = \mathcal{A}(\mu(t))x(t)$$

where

$$\mathcal{A}(\mu(t)) = expc(\mu(t)A)A + expc(\mu(t)A)B$$

and

$$expc(\mu(t)A) = \sum_{n=0}^{\infty} \frac{A^n}{(n+1)!}$$

Since *expc* is a convergent power series for the matrix exponential function, and it converges quickly, only the first 5 terms of the series are used in our analysis for computational ease.

In this case, the stability criterion becomes negativity of the integral

$$\int_0^{\infty} f(\mu) \log_{max\lambda} |1 + \lambda(\mathcal{A}(\mu))| du \tag{29}$$

Now, rather than μ being dependent on time, it is a given graininess value with an associated probability, given by $f(\mu)$, the PDF of the probability distribution of μ. In addition, $\lambda(\mathcal{A}(\mu))$ is the set of eigenvalues of $\mathcal{A}(\mu)$. According to [5], the linearized system has four eigenvalues including one at the origin and three in the left-half complex plane. While our stability theory normally calls for using a controller to move all eigenvalues to the left to achieve stability, in the case of our system, the state variables of rotor angle are continuously increasing since they are not measured periodically but linearly. Since the generator angles will continue to increase, the associated eigenvalue remains 0. Therefore, this eigenvalue is ignored, and the one remaining that maximizes the given sum is chosen for analysis.

With this stability criteria, it is possible for us to evaluate Stability-Almost-Surely for different probability distributions and K-values.

Stability-Almost-Surely Results

As mentioned previously, various timescales and associated probability distributions can be tested to evaluate stability of the linearized system. One of the simplest of these is an exponential distribution with mean $\frac{1}{60}$. For this distribution, the values of (29) associated with the magnitude of the gain variable k_d are seen in Figure 6.

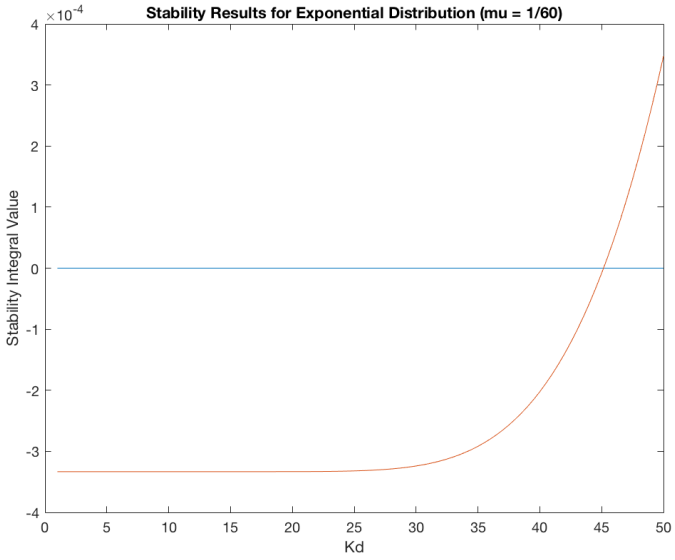

Figure 6: Stability-almost-surely analysis results for different values of k_d with exponentially distributed spacing, $\mu = 1/60$. Negative values indicate stability.

The system remains stable for relatively high values of k_d assuming this distribution. Stability results for the values about the root are seen in Figure 7.

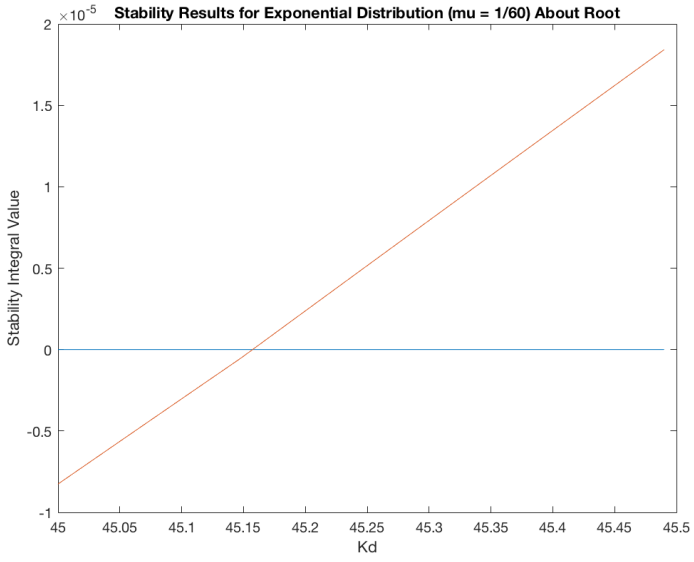

Figure 7: Analysis results of Figure 6 about the root.

However, as variance is suspected to be higher than accounted for by the exponen-

tial distribution, results were also evaluated for a Gamma distribution with $K = \frac{1}{2}$. The scale parameter for Gamma distributions is always chosen so that the mean remains $\frac{1}{60}$. These results are seen in Figure 8.

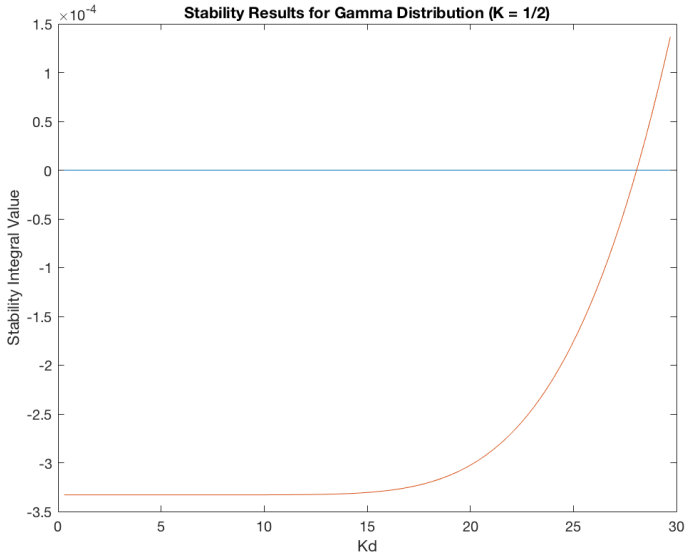

Figure 8: Analysis results with gamma distributed spacing, $\theta = 1/30$, $k = 1/2$.

This system becomes unstable for lower K values than the exponentially distributed case. Stability results for the values about the root are seen in Figure 9.

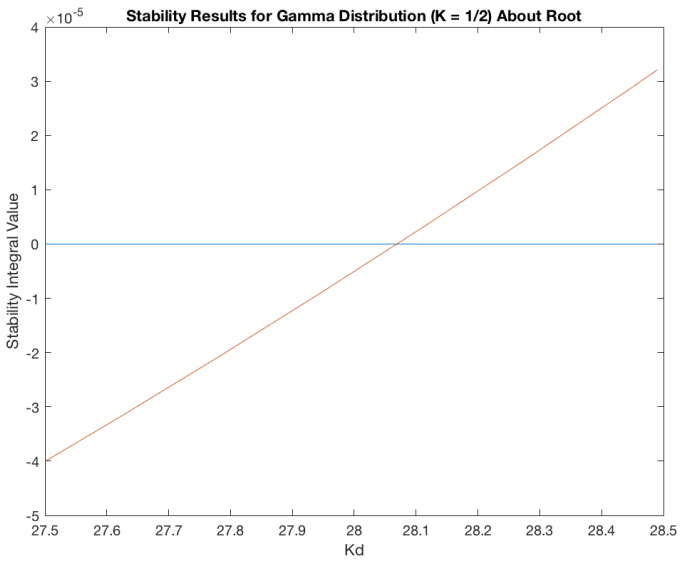

Figure 9: Analysis results of Figure 8 about the root.

The difference between when instability is reached for the Gamma and exponential distribution raises questions about the stability in terms of the variance of the actual timescale. In addition, a heavier-tailed distribution will necessarily increase the value of the stability integral. Graphs of the stability integral in terms of k_d for various Gamma distributions of differing variance (all with mean $\frac{1}{60}$) are seen in Figure 10.

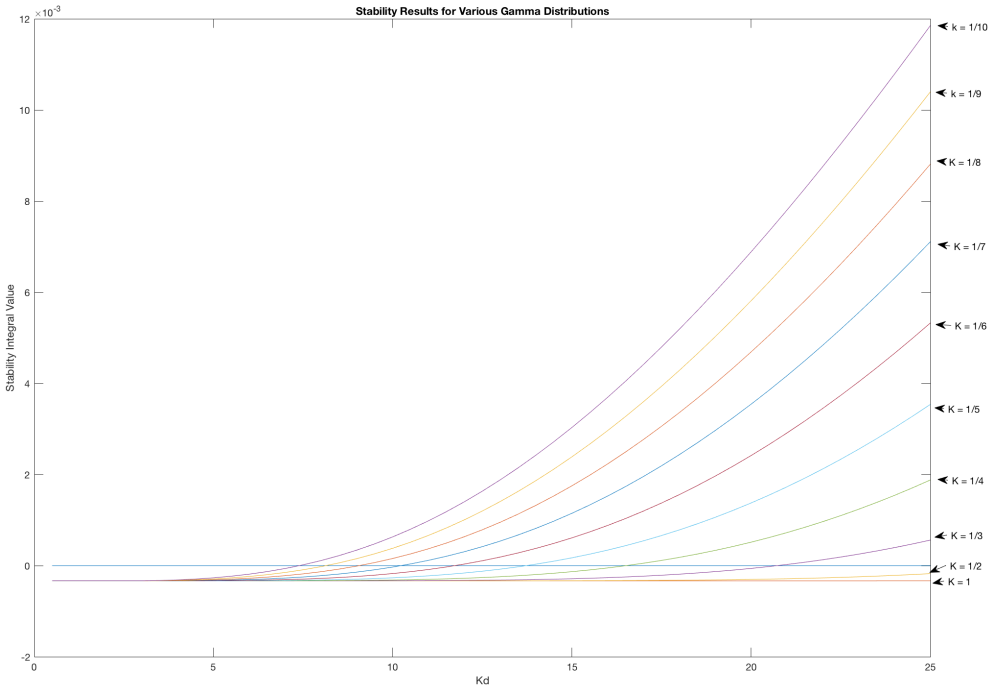

Figure 10: Analysis results of various gamma distributions, with k and corresponding θ values such that each has mean $\mu = 1/60$ but differing variance.

CHAPTER FOUR

Comparing Simulation to Stability-Almost-Surely Results

To examine the behavior of the system in relation to the stability results found, we can examine the transient response of the previous simulation with different parameters. The exponential distribution is predicted to be stable for $k_d = 40$, and unstable for $k_d = 50$. Figure 11 illustrates transient response curves of the difference in generator rotor angular velocity for these two parameter sets, with no constant linear delay.

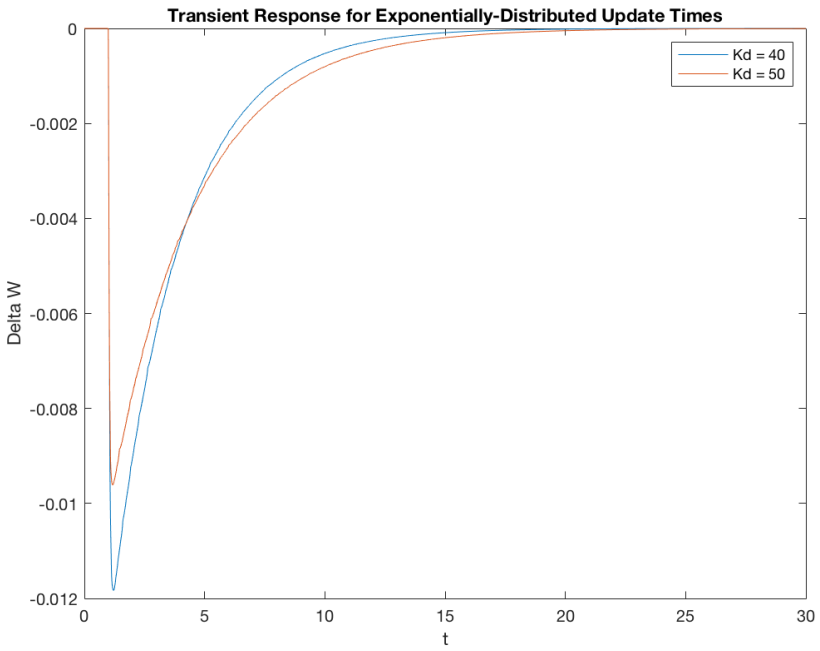

Figure 11: Transient response of exponentially-distributed-spacing system with k_d values predicting stability and instability.

The difference in stability behavior might be reflected in the differing asymptotic behavior between the two curves. In the same vein, the gamma distribution with $K = 0.5$ is predicted to be stable for $k_d = 25$, and unstable for $k_d = 30$. Figure 12

illustrates transient response curves of the difference in angular velocity, again with no constant linear delay:

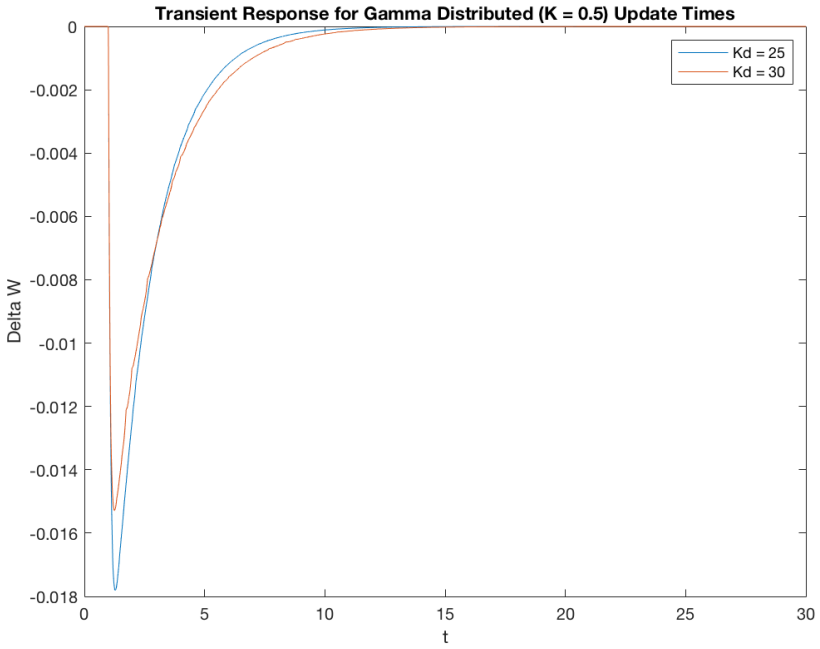

Figure 12: Response of gamma-distributed-spacing system with k_d values predicting stability and instability.

We also expect the region of stability with constant time delays by the metric described in [4] to differ about values of k_d that nearly result in instability. Below is the region of stability for exponentially and gamma-distributed ($k = \frac{1}{2}$) update times, with varying k_d values. The values $k_d = 40$ and $k_d = 28$ are values on the threshold of stability-almost-surely:

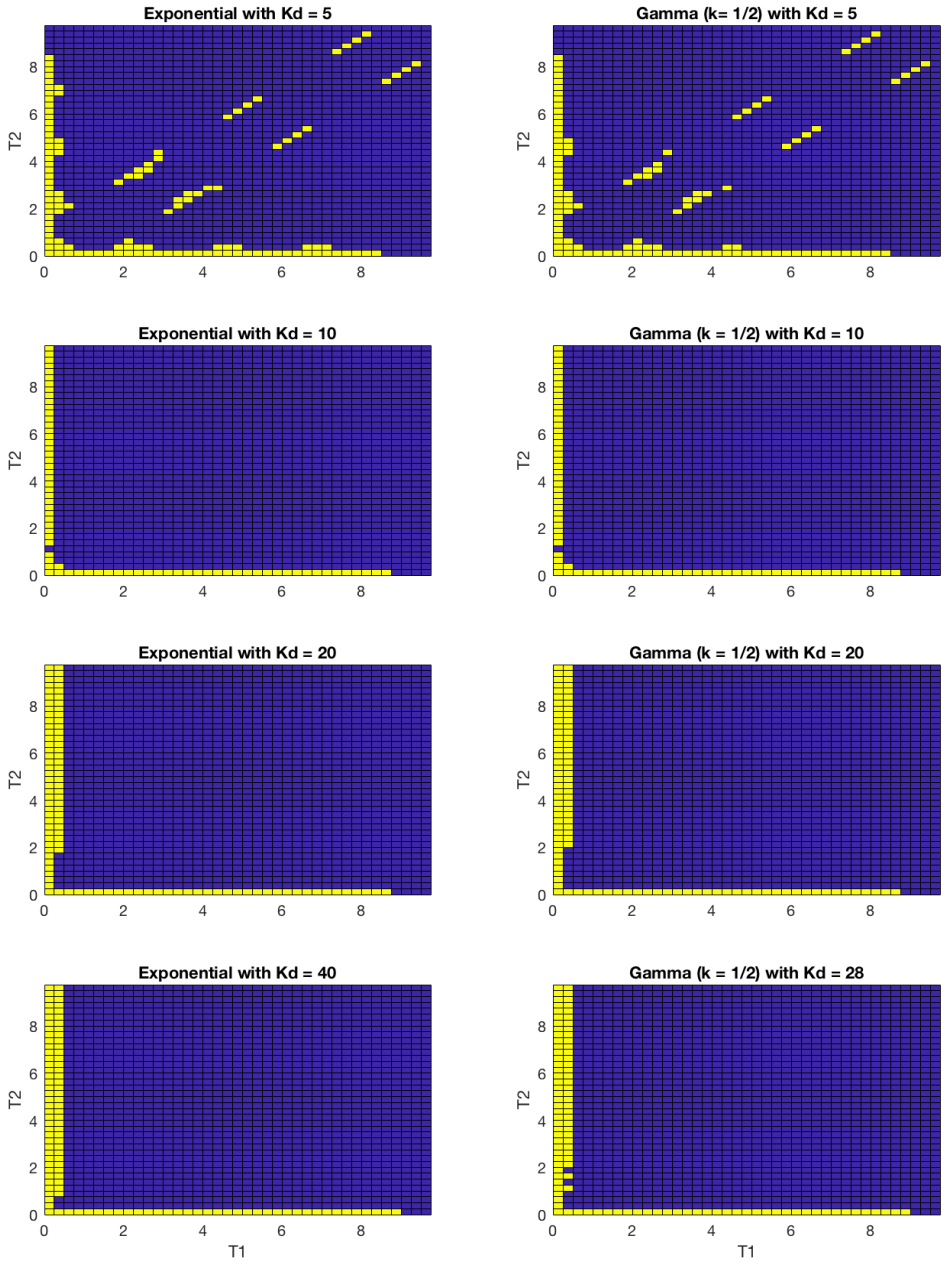

Figure 13: Stability regions of exponentially ($\mu = 1/60$) and gamma ($k = 1/2$, $\theta = 1/30$) distributed spacing systems for various k_d values.

The stability region under the criterion defined in [4] is shown not to change significantly as larger values of k_d are used, regardless of expected stability-almost-surely. A possible shortcoming of the criterion is the fact that dangerous peaks in the transient response between 5 and 10 seconds, as well as divergent transient response that occurs after $t = 30$ seconds will still be evaluated as stable. Further research can be performed into the cause of the stability regions observed, in addition to the behavior of the system for very high values of k_d that are expected to result in instability.

A still-unanswered question that the research performed thus far has pointed to is the effect random packet delays have on control systems. While control systems updated according to a timescale described by a probability distribution function can approximate this behavior, outliers such as controller feedback signals that arrive out of order are not accounted for. Other opportunities for further research include seeking greater understanding of the connection between the transient response of the system and parameters believed to result or not result in stability-almost-surely. Finally, a probability distribution function that describes packet delays more accurately than those posited in this paper might also give a better idea of the nature and consequences of stability-almost-surely and system performance.

APPENDIX

Matlab Code

```
%Generates plots of stability analysis for one or more PDF's of update
%spacing given the gain value Kd

count1 = 1;                         %Counter for outer loop
Kinc = 0.5;                         %Increment for gain value K
Dim1 = 50;                          %Number of Kd values to evaluate
Dim2 = 10;                          %Number of variability values d to evaluate
data = zeros(Dim1,Dim2+1);          %Initialize data matrix
Kd = 0.5;                           %Initialize gain Kd
d = 1;                              %Counter for inner loop
%Run stability analysis for each d value, given the Kd value, for all Kd's
while (count1 <= Dim1)
    while (d <= Dim2)
    data(count1,1+d) = stability_output_gam(Kd, 1/d);
    d = d + 1;
    end
    d = 1;
    data(count1,1) = Kd;
    Kd = Kd + Kinc;
    count1 = count1 + 1;
end

%Plot stability results for each d value
plot(data(:,1),zeros(Dim1,1)); hold on
while (d <= Dim2)
plot(data(:,1),data(:,d+1))
d = d + 1;
end

%Function returns stability-almost-surely result for given Kd gain and d
%variability values
function y = stability_output_gam(Kd, d)

A = [   0   0   0.2   0;        %State matrix
        0   0   0     0.2;
```

```
        -14  14   -.02   0;
        14  -14   0      -.02];

Bd = [   0    0 ;                %Control matrix
         0    0 ;
         .2   0 ;
         0    .2];

K =    [0    0    -Kd Kd;        %Gain matrix
         0    0    Kd -Kd];

%Initial value for integral sum (0 delay impossible)
mu = 0.001;
mu_step = 0.001;                 %Interval size
int_limit = 8;                   %Interval limit (chosen for unlikeliness)
gam_sum = 0;                     %Aggregate sum variable initialized to 0

%Approximate integral calculated based on stability-almost-surely
while mu <= int_limit
    expc_mu_A = eye(4) + (mu*A)/2 + ((mu*A)^2)/6 +
    ((mu*A)^3)/24 + ((mu*A)^4)/120;

    script_A_mu = expc_mu_A*A + expc_mu_A*Bd*K;
    eigenvalues = eig(script_A_mu);
    lambda = sort(eigenvalues,'ComparisonMethod','abs');
    lambda(1,:) = [];
    prb = log(max(abs(1+lambda*mu)));
    gam_sum = gam_sum + gampdf(mu,1*d,1/(60*d))*prb*mu_step;
    mu = mu + mu_step;
end
y = gam_sum;                     %Return stability analysis resultant value

end
```

stab_analysis_almost_surely_heatmap.m

```
% Sandia National Laboratories
% Felipe Wilches-Bernal (Org. 8813)

clear all, close all, clc

t1 = 0;
```

```
t2 = 0;


% ************************************************************************
%                     Defining system parameters
% ************************************************************************

% XT = 0.30;
% d12 = 0.3047;
% V1 = 1; % assumed
% V2 = 1; % assumed
%
% Ks = V1*V2*cos(d12)/XT;


T = 14;

Rp = inf; % has to be high for the evalues to be imaginary

M1 = 5; % Different from Kundur book page 599
D1 = 0.1; % 1;
R1 = Rp;


% Symmetric
M2 = M1;
D2 = D1;
R2 = R1;


% Control
Kd = 5;

% Delay to the feedback signal
T1 = 0.;
T2 = 0.;

%T1 = inT1;
%T2 = inT2;

% ************************************************************************
```

```
%                    Define Disturbances
% ************************************************************************
tstep = 1;
t = linspace(0,15,1000);

delP1l = zeros(size(t));
delP1l(t>tstep) = 1;
delP2l = zeros(size(t));

delPl = [delP1l; delP2l];

delP_c{1} = delPl;

delP2r = zeros(size(t));
delP2r(t>tstep) = 1;
delP1r = zeros(size(t));

delPr = [delP1r; delP2r];

delP_c{2} = delPr;

ix_dPL = 1;
delPL1 = delP_c{ix_dPL}(1,:);
delPL2 = delP_c{ix_dPL}(2,:);


t = t.';
delPL1 = delPL1.';
delPL2 = delPL2.';


% ************************************************************************
%                    Running Simulations
% ************************************************************************

Dim = 40;                       %Dimension of square stability matrix
DelayInc = 0.25;                %Increment of linear delay

count1 = 1;
count2 = 1;
data = ones(Dim^2,3)*-1;        %Initialize data matrix
```

```
89
90
91 %Generate a matrix indicating stability or instability based on input
92 %linear delay
93 while (count1 <= Dim)
94     T1 = 0;
95     count2 = 1;
96     while (count2 <= Dim)
97         data((count1-1)*Dim+count2,1) = T1;
98         data((count1-1)*Dim+count2,2) = T2;
99         data((count1-1)*Dim+count2,3) = run_sims;
100        T1 = T1 + DelayInc;
101        count2 = count2 + 1;
102    end
103    T2 = T2 + DelayInc;
104    count1 = count1 + 1;
105 end
106
107 disp(data);
108
109 %Create mesh from matrix
110 x = data(:,1);
111 y = data(:,2);
112 z = data(:,3);
113
114 %Plot matrix to indicate stability or instability
115 xi = unique(x) ; yi = unique(y) ;
116 [X,Y] = meshgrid(xi,yi) ;
117 Z = reshape(z,size(X)) ;
118 figure
119 pcolor(X,Y,Z);
120
121 %Function runs simulation from simulink file and determine stability given
122 %exponentially distributed update time gaps of controller
123 function y = run_sims
124 sim('ESt_2A_stochastic')
125 ts_h = tsim.Time.';
126 W1_h = delW1_w.Data.';
127 W2_h = delW2_w.Data.';
128 W12_h = W1_h - W2_h;
129
```

```
vec1 = W12_h(10001:20001);
vec2 = W12_h(50001:60001);

val1 = max(abs(vec1));
val2 = max(abs(vec2));

if val1 > val2
    y = 1;
else
    y = 0;
end

end

%Function runs simulation from simulink file and determine stability given
%continuously updated controller
function y = run_sims_simple
sim('ESt_2A_simple')
ts_h = tsim.Time.';
W1_h = delW1_w.Data.';
W2_h = delW2_w.Data.';
W12_h = W1_h - W2_h;

vec1 = W12_h(1001:2001);
vec2 = W12_h(5001:6001);

val1 = max(abs(vec1));
val2 = max(abs(vec2));

if val1 > val2
    y = 1;
else
    y = 0;
end


end

%Function runs simulation from simulink file and determine stability given
%gamma distributed (d = 0.5) update time gaps of controller
function y = run_sims_gamma
```

```
sim('ESt_2A_stochastic_gamma')
ts_h = tsim.Time.';
W1_h = delW1_w.Data.';
W2_h = delW2_w.Data.';
W12_h = W1_h - W2_h;

vec1 = W12_h(10001:20001);
vec2 = W12_h(50001:60001);

val1 = max(abs(vec1));
val2 = max(abs(vec2));

if val1 > val2
    y = 1;
else
    y = 0;
end

end
```

Run_Simulink_Models_simple_v1_stabmatrix.m

Bibliography

1. M. Bohner, A. Peterson, *Dynamic Equations on Time Scales: An Introduction with Applications*, Birkhäuser Boston, 2001

2. David A. Copp; Felipe Wilches-Bernal; David A. Schoenwald; Imre Gyuk, Power system damping control via power injections from distributed energy storage, *International Symposium on Power Electronics, Electrical Drives, Automation and Motion* (2018)

3. Felipe Wilches-Bernal; David A. Schoenwald; Rui Fan; Marcelo Elizondo; Harold Kirkham, Analysis of the effect of communication latencies on HVDC-based damping control, *IEEE/PES Transmission and Distribution Conference and Exposition* (2018)

4. David A. Copp, Felipe Wilches-Bernal, Ian Gravagne, David A. Schoenwald, Time-domain analysis of power system stability with damping control and asymmetric feedback delays, *North American Power Symposium* (2017)

5. Felipe Wilches-Bernal, David A. Copp, Ian Gravagne, Stability criteria for power systems with damping control and asymmetric feedback delays, *North American Power Symposium* (2018)

6. Benjamin T. Allen, Experimental investigation of a time scales linear feedback control theorem, Masters Thesis, 2007

7. Dylan R. Poulsen, Stability and Control on Stochastic Time Scales, Dissertation, 2015

8. A.M. Sukhov, N. Kuznetsova, What type of distribution for packet delay in a global network should be used in the control theory?, *arXiv e-prints* (2009)

9. X. Zhu, J. Yu, J. Doyle, Heavy-tailed distributions, generalised source coding and optimal web layout design, *California Institue of Technology* (2000)