ISSN 0280-5316
ISRN LUTFD2/TFRT--5819--SE

# Delay-dependent Stability of Genetic Regulatory Networks

Johan Ugander

Department of Automatic Control
Lund University
July 2008

**Lund University**
**Department of Automatic Control**
**Box 118**
**SE-221 00 Lund Sweden**

*Document name*
MASTER THESIS

*Date of issue*
July 2008

*Document Number*
ISRN LUTFD2/TFRT--5819--SE

*Author(s)*
Johan Ugander

*Supervisor*
Richard Murray California Institute of Technology, USA
Anders Rantzer Automatic Control, Lund(Examiner)

*Sponsoring organization*

*Title and subtitle*
Delay-dependent Stability of Genetic Regulatory Networks (Fördröjning och stabilitet i genetiska nätverk)

*Abstract*
Genetic regulatory networks are biochemical reaction systems, consisting of a network of interacting genes and associated proteins. The dynamics of genetic regulatory networks contain many complex facets that require careful consideration during the modeling process. The classical modeling approach involves studying systems of ordinary differential equations (ODEs) that model biochemical reactions in a deterministic, continuous, and instantaneous fashion.

In reality, the dynamics of these systems are stochastic, discrete, and widely delayed. The first two complications are often successfully addressed by modeling regulatory networks using the Gillespie stochastic simulation algorithm (SSA), while the delayed behavior of biochemical events such as transcription and translation are often ignored due to their mathematically difficult nature. We develop techniques based on delay-differential equations (DDEs) and the delayed Gillespie SSA to study the effects of delays, in both continuous deterministic and discrete stochastic settings. Our analysis applies techniques from Floquet theory and advanced numerical analysis within the context of delay-differential equations, and we are able to derive stability sensitivities for biochemical switches and oscillators across the constituent pathways, showing which pathways in the regulatory networks improve or worsen the stability of the system attractors. These delay sensitivities can be far from trivial, and we offer a computational framework validated across multiple levels of modeling fidelity.

This work suggests that delays may play an important and previously overlooked role in providing robust dynamical behavior for certain genetic regulatory networks, and perhaps more importantly, may offer an accessible tuning parameter for robust bioengineering

*Keywords*

*Classification system and/or index terms (if any)*

*Supplementary bibliographical information*

| | |
|---|---|
| *ISSN and key title*<br>0280-5316 | *ISBN* |

| | | |
|---|---|---|
| *Language*<br>English | *Number of pages*<br>45 | *Recipient's notes* |
| *Security classification* | | |

http://www.control.lth.se/publications/

# Contents

# 1. Introduction

The discovery of the genetic code, initiated by the discovery of the structure of DNA by Watson and Crick in 1953, fundamentally changed the nature of biological science. With the identification of genes came the understanding that biological systems were dominated by vast information processing systems, the complexity of which have only now begun to be understood. Concurrently, the nascent field of systems engineering was being developed to tackle the growing challenge of organizing complex artificial systems emerging in the early days of spacecraft design and computer architecture. These two fields have now merged in the form of the interdisciplinary field of *systems biology*, where the tools of systems engineering are now helping to organize our understanding of biological systems, with remarkable success.

By viewing cells as modular systems of systems, systems biology helps to analyze incredibly complex biological problems, and often this understanding has helped reveal even more (and often quite subtle) complexity. Systems biology has helped to raise fundamental questions of architecture, such as the ’highly optimized tolerance’ and ‘robust yet fragile’ conceptual frameworks of system design [Carlson and Doyle, 1999; Carlson and Doyle, 2002]. Historically, one of the most successful applications of the systems engineering perspective has been the analysis of genetic regulatory networks [Jacob and Monod, 1961; Glass, 1975].

Genetic regulatory networks consist of interacting genes and proteins that are distributed throughout a cell and interact in a network of activation and repression. Each gene codes for a specific protein, many of which go on to activate or repress other genes in the organism by binding to the genes as *transcription factors*. *Activation* of a gene occurs when a protein binds to the DNA region near the gene and facilitates the initiation of transcription by RNA Polymerase (RNAP) molecules, while *repression* of a gene occurs when a protein binds to the DNA region near the gene and inhibits this initiation.

Transcription by RNAP results in the production of messenger RNA (mRNA) macromolecules. These mRNA molecules are then freely translated by ribosomes, which bind to the mRNA and systematically translate the genetic code to assemble specific proteins. These proteins then often serve as the very transcription factors that return to activate or repress genes, while mRNA serves an intermediate role in the dynamics [Alberts *et al.*, 2003].

In the context of systems and control, these networks then exhibit familiar feed-

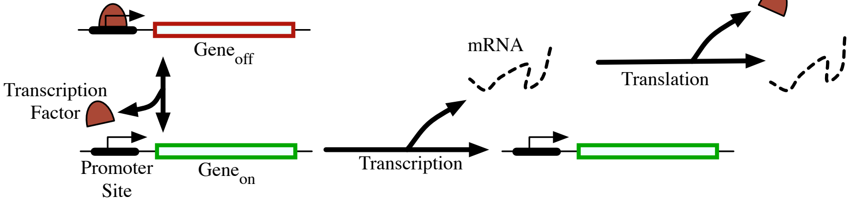

**Figure 1.1** The central dynamics of molecular biology. Regions of DNA called genes are transcribed to form mRNA, which are then translated to form proteins. Some proteins then act as transcription factors, regulating gene expression by binding to the promoter site as either an activator or repressor, while other proteins go on to perform a wide range of functions with the cell. In this figure, repression is shown.

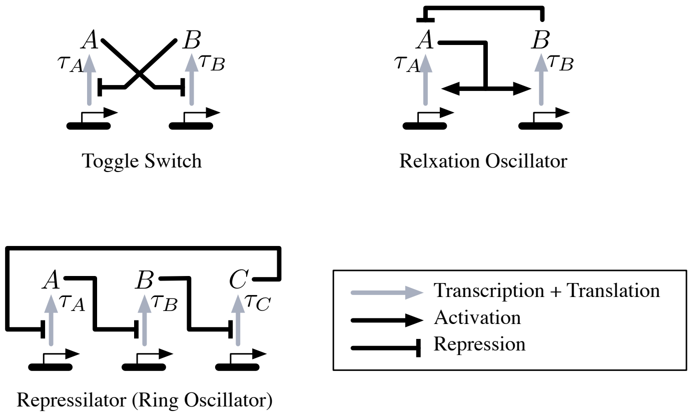

**Figure 1.2** Three basic genetic regulatory networks that will be explained and examined in this work. Clockwise from top left, the bistable toggle switch, the ring oscillator, and the hysteresis-based oscillator are shown. Introducing delays to the transcriptional pathways can drastically effect the stability of the system models.

back dynamics [Alon, 2007]. Auto-repression, whereby a gene expresses a protein that then represses its own gene, and thereby down-regulates its own production, is a classic form of feedback moderation in genetic regulatory networks. Auto-activation, whereby a gene expresses a protein that activates its own gene, can be used to amplify a response to an initial input signal. These two forms of autoregulation together occur more than 40 times in the *E. coli* genetic regulatory network [Alon, 2007]. During the past decade, principles of control theory have helped to illuminate these complex interaction networks that are so common in engineering.

Beyond simple autoregulation, common 'motifs' in networks have been identified across a wide range of organism genomes [Milo *et al.*, 2002]. These network motifs appear as small, task-specific modules embedded in a larger context. For example, feedforward motifs [Mangan and Alon, 2003] in different arrangements appear to be widely used by cells to accelerate or delay signaling. Network motifs enable a modular architecture for biological signaling, and the emergence of this modularity can be explained directly by evolutionary forces [Kashtan and Alon, 2005].

Ultimately one of the grandest challenge in cellular information processing is noise [Kaern *et al.*, 2005]. The cellular signaling environment is dominated by brownian motion, and at low molecular concentrations, the inherent randomness of the biochemical reactions tends to dominate the cellular dynamics. While some processes intentionally utilize stochasticity to achieve phenotypic heterogeneity [Raser and O'Shea, 2005], in most tasks this variability is highly undesirable. To tackle this challenge, principles of control theory have been utilized to engineering stability into genetic regulatory networks [Becskei and Serrano, 2000].

This emerging field of bioengineering stability is the central subject of this thesis. Where previous work has focused on reaction network topologies [Hasty *et al.*, 2002], this thesis develops principles for tuning *delays* in genetic regulatory networks as a means towards engineering stability. In this work we examine the role of delay in both stochastic and deterministic models of genetic regulatory networks, developing a theory for delay sensitivity and tuning.

In [Ugander *et al.*, 2007], the consequences of delay upon stability were discussed as part of an analysis of one specific oscillatory network, a toggle oscillator, which was delayed using extended transcriptional pathway cascades, a means of delay that is prohibitively complex for practical bioengineering. In this thesis the study of delay is generalized, and the possibility of embedding delays directly into the genome is studied, by a much more practical mechanism. It is our hope that this work will contribute a useful design principle for controlling noise in genetic regulatory networks, and hopefully play a role in advancing the rapidly developing and increasingly exciting field of bioengineering.

# 2. Stochastic Simulation

In this chapter, we examine detailed, stochastic models of genetic regulatory networks, using a discrete, probabilistic framework true to the discrete, probabilistic dynamics of biochemical systems. While the inherent complexity of such modeling limits the extent to which solid conclusions can be drawn, our goal with this analysis is merely qualitative insight, while a more tractable delay-differential equation (DDE) formulation presented in the next chapter will offer precise metrics for stability analysis. We examine both stable and oscillatory regulatory networks, and attempt to analyze the delay- dependent stability of their attractors. For the biochemical oscillators, we present computational results that demonstrate the delay-dependent stability of the limit cycles underlying the stochastic dynamics, introducing delay tuning as a novel means of bioengineering stability in genetic regulatory networks.

## 2.1 The Chemical Master Equation

The Chemical Master Equation (CME) is a system of ordinary differential equations describing the time evolution of the probability distribution of a system across possible concentration states. In biochemical systems, reaction events constitute state transitions between concentration states, in what amounts to a large continuous-time Markov process. The CME modeling approach is based on reactions within a confined and well-mixed vessel, and assumes that collisions are frequent, while collisions with proper energy and orientation (leading to reactions) are relatively infrequent. Furthermore, it is only strictly correct for reactions between at most two reactants [Van Kampen, 2007]. The basic CME also assumes that reactions are instantaneous, and not delayed, requiring a modification which we will return to later. In what is a fully Markovian framework, instantaneous reactions (collisions) are modeled as Poisson distributed events. For biochemical systems, the set of possible discrete molecular concentration arrangements define the set of possible states, typically the semi-infinite discrete set $\mathbb{Z}_+^n$.

Let us introduce a basic system of gene expression as an example. When modeling the expression of a particular mRNA and protein pair, the state would be given by the count of molecules in the system, $\mathbf{x} = (\#mRNA, \#protein) \in \mathbb{Z}_+^2$. The number of copies of 'Gene' is assumed to be constant, and is therefore not part of the state. The dynamics of the system would be described by the following four reactions,

$$\begin{aligned} Gene &\xrightarrow{k_1} Gene + mRNA \\ mRNA &\xrightarrow{k_2} mRNA + Protein \\ mRNA &\xrightarrow{k_3} \emptyset \\ Protein &\xrightarrow{k_4} \emptyset, \end{aligned} \tag{2.1}$$

where $k_i$ are the reaction rates. The state space and transitions for this system are summarized in Figure 2.1 below.

It is important to note that the transition propensities are non-uniform, as the birth process for protein is a first order reaction dependent upon the number of mRNA molecules in the system. Likewise, the propensities of the decay reactions are dependent upon the concentrations of the decaying molecules as well. In this simple example, the reactions are all simple birth and death reactions, but more complicated

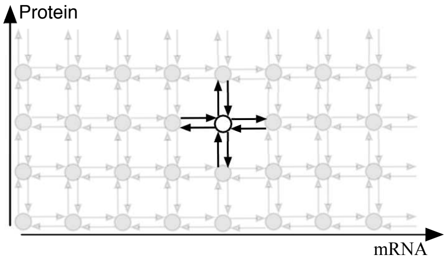

**Figure 2.1** The semi-infinite state-space lattice for a basic mRNA-protein system.

stoichiometries can lead to a complicated mesh of transition arrows. Furthermore, the state space of systems with many species becomes a high-dimensional lattice, where 30 or more species is not uncommon for basic systems. The transition pathways are however sparse for almost all systems. With further emphasis, it is important to understand that this is not merely a uniform random-walk, but that the state-dependent propensities, which for second-order reactions are nonlinear, lead to a highly non-trivial energy landscape that in turn governs the time evolving probability density function of the system.

Generally, the Chemical Master Equation is written as

$$\frac{d}{dt}P_i(t) = \sum_{j\in I} a_{ij}P_j(t), \qquad \forall i \in I, \quad t > 0, \tag{2.2}$$

$$P_i(0) = P_i^0, \qquad \forall i \in I, \tag{2.3}$$

where (2.3) is the initial condition, $P_j(t)$ is the probability of being in state $j$ at time $t$, $I$ is the countable index set of the state space, and $a_{ij}$ is the transition propensity from state $j$ to state $i$. For first-order reactions, the propensity is given by the product of the reactant concentration and the reaction rate $k_i$, in appropriate units. For the simple example introduced above, the CME becomes, with the obvious indexing:

$$\begin{aligned}\frac{d}{dt}P_{(n,m)}(t) &= -a_{(n,m)}P_{(n,m)}(t) + \\ &\quad a_{(n-1,m)}P_{(n-1,m)}(t) + a_{(n+1,m)}P_{(n+1,m)}(t) + \\ &\quad a_{(n,m-1)}P_{(n,m-1)}(t) + a_{(n,m+1)}P_{(n,m+1)}(t), \qquad \forall n,m \in \mathbb{Z}_+\end{aligned} \tag{2.4}$$

$$\begin{aligned}\frac{d}{dt}P_{(n,m)}(t) &= -\left(k_1 + k_3n + k_2n + k_4m\right)P_{(n,m)} + \\ &\quad k_1P_{(n-1,m)}(t) + k_3(n+1)P_{(n+1,m)}(t) + \\ &\quad k_2nP_{(n,m-1)}(t) + k_4(m+1)P_{(n,m+1)}(t), \qquad \forall n,m \in \mathbb{Z}_+\end{aligned} \tag{2.5}$$

where $k_i$ are the reaction rates from (2.1), in the appropriate units. Minor, straightforward modifications are required at the boundaries $m = 0$ and $n = 0$.

Notice that this is merely an infinite, sparse, linear system of differential equations, and for systems with modestly compact support, meaning that only a manageably few number of states at any time have probability over a given small threshold, the state space can be truncated and the time-evolving probability distribution can be found with impressive precision by the method of Finite State Project (FSP), see [Munsky and Khammash, 2006]. The FSP method allows for the application of elegant linear systems theory, but is however only an efficient approximation for smaller systems or for studying equilibrium dynamics of at most medium-sized systems, and we will not employ it here. Protein concentrations, which can easily number in the thousands within a given cell, quickly lead to a prohibitively large state space. It is also worth briefly mentioning that hybrid solvers have been developed, that combine discrete, stochastic and continuous, deterministic modeling, for systems with multiscale dynamics [Haseltine and Rawlings, 2002; Tian and Burrage, 2004], though these solvers have yet to be rigorously analyzed for correctness.

As a further comment, the CME is in fact a discrete-state-space analog to the Fokker-Planck Equation, where the state space is instead continuous, and a high-dimensional PDE formulation results. This connection has been exploited to perform discrete-to-continuous approximations, where the CME is approximately solved as a Fokker-Planck formulation applying numerical methods for PDEs, see [Sjöberg *et al.*, 2008] for details.

The preferred method for analyzing probabilistic models of biochemical systems is Monte Carlo simulation, by applying what is commonly called the Gillespie Stochastic Simulation Algorithm (Gillespie SSA) [Gillespie, 1976; Gillespie, 1977]. In 1976, Gillespie introduced two efficient methods for sampling trajectories from the Chemical Master Equation, known as the Direct Method (DM) and First Reaction Method (FRM). Various improved methods have been derived from these two methods, the most notable being the Next Reaction Method (NRM) by Gibson and Bruck [Gibson and Bruck, 2000]. However, recent analysis has shown that the Next Reaction Method is not a universal improvement, and further refinements have been proposed by Petzold and colleagues [Cao *et al.*, 2004; Li and Petzold, 2006], leading to the Optimized Direct Method (ODM) and Logarithmic Direct Method (LDM). In this work, we will limit ourselves to the original Direct Method by Gillespie, and ignore the debate regarding implementational details.

The Gillespie algorithm entails a straightforward simulation of one realization from the stochastic biochemical system, randomly simulating one reaction at a time. The algorithm below is presented for $S$ species and $N$ reactions. We also introduce the notation $a_n$ to denote the propensity of reaction $n$, and $a_{\sum n} := \sum_{k=1}^{n} a_k$ to denote partial sums across the reaction propensity vector. The terms $R_n$ and $P_n$ refer to the state changes for the reactants being removed and products being added during reaction $n$. These merit separate notations in preparation for the delayed Gillespie algorithm yet to come.

**Algorithm 2.1** The Gillespie Direct Method

```
Require: Initial Concentration $X = (X_1, \ldots, X_S)$, $t_{end}$.
1: time $t = 0$
2: while $t < t_{end}$ do
3:     Compute propensities $a_n$, $n = 1, \ldots, N$, at current state $X$.
4:     Generate uniform random numbers $u_1, u_2 \in [0,1]$.
5:     Compute time to next reaction, $\Delta t = -\ln(u_1/a_{\sum N})$.
6:     Find reaction channel $n$, s.t. $a_{\sum n-1} < u_2 a_{\sum N} < a_{\sum n}$.
7:     Update State $X$ according to $R_n$ & $P_n$.
8:     Update $t = t + \Delta t$
9: end while
```

Here we see that step 5 generates the time to the next reaction from an exponential distribution with the rate parameter given by the cumulative probability of any reaction occurring, and step 6 subsequently selects a reaction by their relative probabilities of occurrence. The implementational improvements mentioned earlier are mostly concerned with minimizing recalculation of the propensities in step 3 and the

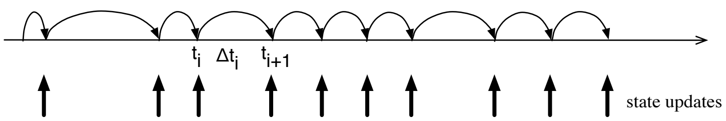

**Figure 2.2** Reaction events during the Gillespie SSA simulation, which occur with exponentially distributed time intervals, with a propensity taken by the overall propensity of any reaction occurring.

sums in steps 5 and 6, in particular by using custom data structures to efficiently find the reaction channel in step 6.

The Gillespie SSA results in a single trajectory drawn from the time-evolving probability density function, and to fully understand the system, an ensemble of trajectories is typically studied. In Figure 2.3, an ensemble of 100 simulated trajectories from the basic mRNA-protein expression system discussed above are presented (showing only the protein expression), and such an ensemble provides a confident overview of the nature and variability of the system dynamics, here entering a steady-state equilibrium between production and decay.

The Gillespie SSA approach has been remarkably successful at characterizing the stochastic dynamics of biochemical reaction networks, despite its computational complexity, see [McAdams and Arkin, 1997] for an early success story. Yet a serious short-coming of the CME approach, as mentioned earlier, is its inability to model reaction delays, since the approach assumes simple, instantaneous reactions, and not abstract macro-reactions, such as $mRNA \rightarrow mRNA + Protein$, which in truth is far from instantaneous. As computational resources have grown, simulations have been performed that model the movement of individual RNA polymerase (RNAP) molecules along DNA (as well as ribosomes along mRNA) with base-pair resolution [Kosuri *et al.*, 2007; Kosuri, 2007]. This movement results in a cascade of Poisson distributed events, thus incorporating delays within the Markovian framework and respecting the delay time implicit in transcription and translation that is otherwise ignored by the Gillespie approach involving abstract species. With gene lengths in the thousands of base-pairs, this unfortunately requires a thousand-fold increase in computational effort, by replacing each birth reaction with a lengthy cascade of reactions. This approach also introduces the possibility/challenge of modeling the collision dynamics of RNAP along the DNA, the mechanics of which have only begun to

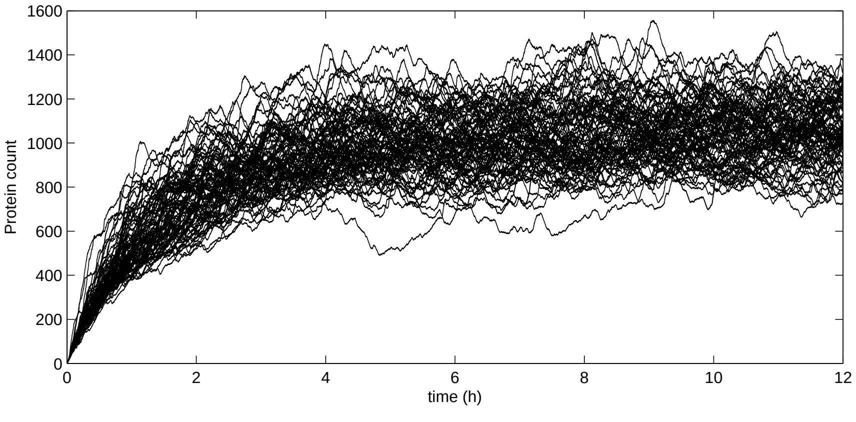

**Figure 2.3** An ensemble of 100 trajectories for the basic mRNA-protein expression model, simulated using the GIllespie stochastic simulation algorithm.

emerge [Greive and von Hippel, 2005; Herbert *et al.*, 2006; Tolic-Norrelykke *et al.*, 2004; Epshtein and Nudler, 2003], and much is still unknown. Disregarding RNAP collisions, the cascade of reactions used in base-pair resolution modeling is well approximated by delays, which brings us to the next section.

## 2.2 The Delayed Chemical Master Equation

Using delays to approximate reaction cascades is drastically more efficient than base-pair resolution modeling, but this comes at the cost of breaking the Markov property of the system. With delayed reactions, there are now reactions occurring “off-stage” with delayed results, and these events are not part of the state description, but certainly effect the future of the system. In contrast, base-pair resolution modeling is still Markovian, as the state of the system incorporates the location of RNAP along the DNA, and all events are still memoryless.

The delay times used to approximate the reaction cascades can be expected to be gamma distributed, owing to the property that the sum of i.d.d. exponential stochastic variables is gamma distributed, a property that is noted in [Gibson and Bruck, 2000]. This is reality not true, since physical principles of causality require RNA polymerase molecules to leave the DNA in the order that they attach. The matter then becomes entangled in the aforementioned unresolved question of RNAP collision dynamics, about which little is known. For this reason, in the absence of knowledge, the delay times are modeled as constant for the set gene length, at an average rate of RNAP transcription and translation of 45 bp/s [Phillips *et al.*, 2008].

A delayed variant of the Chemical Master Equation can be formulated, with $M$ delay timescales, as:

$$\frac{d}{dt}P_i(t) \quad = \quad \sum_{j\in I} a_{ij}P_j(t) + \sum_{m=1}^{M}\left(\sum_{j\in I} a_{ijm}P_j(t-\tau_m)\right). \tag{2.6}$$

This is in fact a delay-differential equation (DDE), with similarities to the problems we will be studying in the next chapter, but we will do not analyze the delayed CME as a DDE, as its size makes it difficult to approach directly, and again the Monte Carlo approach pioneered by Gillespie will be essential. The delayed Gillespie algorithm requires only basic modifications in order to simulate trajectories from the delayed CME. For the delayed Gillespie algorithm to be meaningful, we need to more precisely define what it is we are modeling.

Let us again consider the example of basic gene expression, but now respect the time delays inherent to transcription and translation. With this in mind, a more truthful model of gene expression has $M = 4$ delay timescales (for a single gene),

$$\begin{aligned} Gene(t) &\xrightarrow{k_1} Gene(t+\tau_{ps}) + mRNA(t+\tau_1) && (2.7)\\ mRNA(t) &\xrightarrow{k_2} mRNA(t+\tau_{rbs}) + Protein(t+\tau_2) && (2.8)\\ mRNA(t) &\xrightarrow{k_3} \emptyset && (2.9)\\ Protein(t) &\xrightarrow{k_4} \emptyset. && (2.10) \end{aligned}$$

where $\tau_{ps}$ is the time it takes the RNAP to clear the promoter site where new RNAP can then bind, about 1 second, and $\tau_1$ is the time until the ribosome binding site is fully transcribed, making the mRNA free for translational initiation. Typically the

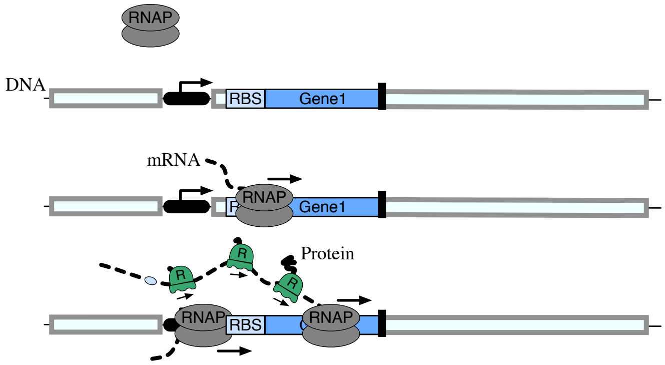

**Figure 2.4** Transcription and translation. Transcription initiates when an RNAP molecule binds to the *promoter site* ahead of a gene, shown here as a black arrow. It then begins transcribing the gene, producing mRNA, until it reaches a stop sequence at the end of the gene. Once the ribosome binding site (RBS) has been transcribed, ribosomes (green, marked R) bind the the nascent mRNA transcript and begin translation and the production of Protein. When transcription is completed, the mRNA molecule remains active for translation until decay, by a process not shown here. Protein molecules remain active in their function until decay, also by an process not shown here. Both transcription and translation occur at approximately 45 $bp/s$ [Phillips *et al.*, 2008].

ribosome binding site occurs nearly immediately after the promoter site, resulting in $\tau_1$ being on the order of 5 seconds.

This idea that mRNA is active already after RBS-transcription relies on the architecture of prokaryotes, whereby transcription and translation occur in the same vessel, in the absence of the nuclear membrane found in eukaryotes. A schematic illustration of the events of transcription and translation can be found in Figure 2.4.

In translation, $\tau_{rbs}$ is the time it takes the *ribosome* to clear the ribosome binding site (rbs), analogous to the time it takes the RNAP to clear the promoter site ($\tau_{ps}$) in transcription, and is also about 1 second. The delay $\tau_2$ is the time it takes the protein to be translated in full. By example, for the 1080 base-pair *lacI* gene commonly used in synthetic biology, this amounts to 24 seconds. The $\tau_2$ delay time can also be thought to include the time it takes the protein to become biochemically ‘active’, an activation that may entail potentially complex protein folding, or require post-translational modification in various ways, and $\tau_2$ may therefore be drastically longer. Such activation is however not considered in this study, but it should be kept in mind as a potential window for delay engineering.

The primary window for delay engineering in this study is $\tau_1$, the delay time until the ribosome binding site is transcribed. In prokaryotes, the ribosome binding site can in fact be located a long distance away from the start of the mRNA transcript, and it is not uncommon that multiple genes occupy the same mRNA transcript, transcribed behind the same promoter site. Thus we present our novel delay engineering approach, whereby it is fully possible to engineer a delay into a genetic network by either inserting a lengthy block of ‘junk’ DNA ahead of the RBS, or by placing another ‘dummy’ gene upstream of the relevant gene within the same mRNA transcript. The latter approach can be seen as favorable, seeing as the neutrality of a randomly chosen sequence of ‘junk’ DNA may be difficult to guarantee, with the possibility of hairpins and other secondary structures introducing unwanted effects that would not be caused by a properly chosen stable ‘dummy’ gene. A diagram of these delay methods in shown in Figure 2.5, where an idea for eukaryotic delay engineering is also briefly presented.

Let us now present the delayed Gillespie algorithm. At first glance, it may appear

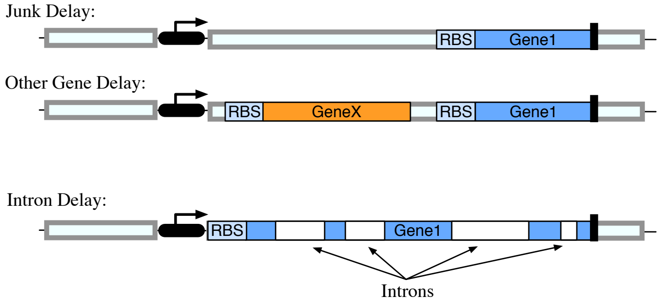

**Figure 2.5** Delay methods. In order to delay the release of an active protein, we propose two methods for delay engineering in prokaryotes. These methods entail modifying the DNA surrounding the gene by inserting either a ‘junk’ delay, or an intermediate ‘dummy’ gene ahead of the gene targeted for delay. For eukaryotes, it is possible to view intron/exon splicing as a means of delay, and lengthy introns can achieve a similar delay effect. With this approach, introns are revealed as more than ‘junk’ DNA, serving to tune the temporal dynamics of the genetic regulatory network.

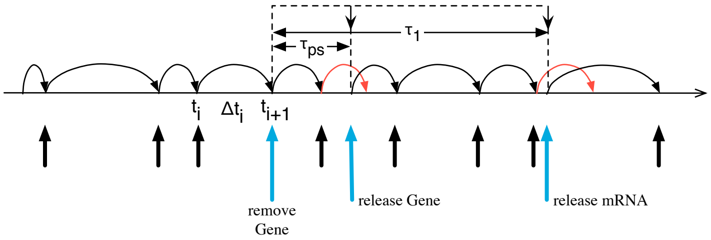

**Figure 2.6** Reaction events during the delayed Gillespie SSA simulation. Diagram borrowed from [Barrio *et al.*, 2006] (supplementary text).

significantly more complicated that the nominal Gillespie algorithm. The differences are however straight-forward. For non-delayed reactions, everything is the same. For delayed reactions, the reactants $R_n$ are removed in the same fashion, while the products $P_{n1}, \ldots, P_{nk}$ are sorted into the simulators *delay queue*, which is easily maintained as a linked list data structure. The possibility for multiple product release times is no major complication, and as we saw above, the biochemical reactions we will be modeling require two product release events. The product events are assumed to be ordered, with $\tau_{n1} < \ldots < \tau_{nk}$. The event time for the most immanent delay product is maintained in the variable $t_{d1}$, and when the simulation crosses this event time, the most immanent delay event occurs as an interruption. The relevant products are introduced, and the simulation restarts from this event time. The memoryless property of the exponential distribution makes this possible. A diagram of the sequence of reaction events during a simulation can be seen in Figure 2.6.

One of first formal attempts at constructing a delayed formulation of the Gillespie algorithm came in the work of [Bratsun *et al.*, 2005], but they unfortunately present a flawed algorithm. In the work of [Barrio *et al.*, 2006] and most carefully [Cai, 2007], the mistakes are resolved, and agreement with the delayed Chemical Master Equation is shown. In the original algorithm due to Bratsun et al., line 15 of the delayed Gillespie algorithm (as it is presented here) is incorrectly restricted to only occurring when the selected reaction is non-delayed, which neglects to include the time it takes a delayed event to randomly occur, prior to the start of the delay wait. For a detailed analysis, see [Cai, 2007].

**Algorithm 2.2** The Delayed Gillespie Direct Method

```
Require: Initial Concentration $X = (X_1, \ldots, X_S)$, $t_{end}$.
time $t = 0$
while $t < t_{end}$ do
    Compute propensities $a_n$, $n = 1, \ldots, N$, at current state $X$.
    Generate uniform random numbers $u_1, u_2 \in [0, 1]$.
    Compute time to next reaction, $\Delta t = -\ln(u_1 / a_{\sum N})$.
    if $t + \Delta t < t_{d1}$ then
        Find reaction channel $n$, s.t. $a_{\sum n-1} < u_2 a_{\sum N} < a_{\sum n}$.
        if (Reaction $m$ is a delayed reaction) then
            Update State $X$ according to $R_n$.
            Place $P_{n1}, \ldots, P_{nk}$ into delay queue at $t + \tau_{n1}, \ldots, t + \tau_{nk}$.
            $t_{d1} = \min(t_{d1}, t + \tau_{n1})$.
        else (Reaction $n$ is not delayed reaction)
            Update State $X$ according to $R_n$ & $P_n$.
        end if
        Update $t = t + \Delta t$
    else (Run delayed reaction scheduled at $t_{d1}$)
        Update State $X$ according to $P_{d1}$.
        Update $t = t_{d1}$.
        Pop delay queue ($t_{d1} = t_{d2}$, $etc.$).
    end if
end while
```

## 2.3 Delay-dependent stability

Our goal is to analyze the delay-dependent stability of attractors for biochemical systems of interacting genes and proteins, also known as genetic regulatory networks. In a stochastic setting, this amounts to studying the variance of the attractor distributions, as sampled across an ensemble of trajectories from delayed Gillespie stochastic simulations. Because such simulations are so computationally intensive, we have chosen to present simulation ensemble distributions at only a sparse grid of delay parameters for the networks we study. This should lead the reader to develop a satisfactory intuition, and we provide a more rigorous analysis using delay-differential equations and Floquet theory in the next chapter.

This crude approach is due to the fact that methods of sensitivity analysis for stochastic CME models are currently only in their infancy. There have been limited advances regarding sensitivity analysis for stationary systems, which try to develop computationally tractable correlates of sensitivity [Gunawan *et al.*, 2005; Cao and Petzold, 2006] with some success. In a very elegant application of measure theory for non-delayed systems, Girsanov measure transforms [Plyasunov and Arkin, 2007] can be used to very quickly compute the sensitivity of various functionals with respect to reaction rate parameters. This method is however not applicable to delayed systems, and especially not to delay parameters.

We apply our analysis to three genetic regulatory networks, (1) the Gardner toggle switch, (2) the Elowitz repressilator, and (3) the Barkai-Leibler relaxation oscillator. The stochastic dynamics of the toggle switch is shown to primarily exhibit robust indifference to delay manipulations, while the stability of the limit cycles underlying the stochastic oscillators is shown to be highly tunable through delay engineering.

## 2.4 Analysis: Toggle Switch

First, we study the delay-dependent stability of the Gardner toggle switch genetic regulatory network [Gardner *et al.*, 2000]. The toggle switch is one of a few ‘classic’ synthetic genetic regulatory networks, and along with the repressilator, one of the few to have been built and studied *in vivo* in *E. coli*, where its functionality has been validated. The toggle switch is a bistable network where one of two protein concentrations can be stably expressed, resulting in a phase space with two stable fixed points, and effectively offering a digital *bit* of memory storage. We wish to examine the stability of these stable fixed points as delay is introduced into the network. By examining the steady-state distribution surrounding the stable attractors (when one of the concentrations is ‘high’), we can study the effects of delay upon stability.

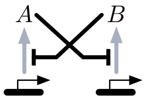

**Figure 2.7** A schematic diagram of the Gardner toggle switch genetic regulatory network.

A schematic diagram of the toggle switch regulatory network is shown in Figure 2.7, while a more complete view of the reactions that make up the full network is presented in Table 2.4. The repression mechanism used in this model features two operator sites, effectively serving as a combinatorial OR gate [Alon, 2007]. This design choice was made to strengthen the repression activity and minimize attractor crossover, where stochastic fluctuations induce a ‘flipping’ of the toggle switch. The repression mechanism and overall parameterization is taken from the original model of the repressilator [Elowitz and Leibler, 2000] analyzed next.

Figure 2.8 shows histograms over the $(P_A \times P_B)$ phase space surrounding the stable fixed point, at four different delay parameterizations. First, we consider the nominally delayed model ($\tau_{1A} = \tau_{1B} = 5$ seconds, top left), which incorporates the delays inherent to transcription and translation, but the system has not been synthetically delayed in any way. This model is then compared to models with considerable (200

| Transcription & Translation | Promoter region | Decay |
|---|---|---|
| $GeneA_0(t) \xrightarrow{k_1} GeneA_0(t+\tau_{ps}) + M_A(t+\tau_{1A})$ | $GeneA_0 + P_B \xrightarrow{k_3} GeneA_1$ | $P_A \xrightarrow{k_6} \emptyset$ |
| $GeneA_1(t) \xrightarrow{\varepsilon \cdot k_1} GeneA_1(t+\tau_{ps}) + M_A(t+\tau_{1A})$ | $GeneA_1 \xrightarrow{k_4} GeneA_0 + P_B$ | $M_A \xrightarrow{k_7} \emptyset$ |
| $GeneA_2(t) \xrightarrow{\varepsilon \cdot k_1} GeneA_2(t+\tau_{ps}) + M_A(t+\tau_{1A})$ | $GeneA_1 + P_B \xrightarrow{k_3} GeneA_2$ | |
| $M_A(t) \xrightarrow{k_2} M_A(t+\tau_{rbs}) + P_A(t+\tau_{2A})$ | $GeneA_2 \xrightarrow{k_5} GeneA_1 + P_B$ | |
| $GeneB_0(t) \xrightarrow{k_1} GeneB_0(t+\tau_{ps}) + M_B(t+\tau_{1B})$ | $GeneB_0 + P_A \xrightarrow{k_3} GeneB_1$ | $P_B \xrightarrow{k_6} \emptyset$ |
| $GeneB_1(t) \xrightarrow{\varepsilon \cdot k_1} GeneB_1(t+\tau_{ps}) + M_B(t+\tau_{1B})$ | $GeneB_1 \xrightarrow{k_4} GeneB_0 + P_A$ | $M_B \xrightarrow{k_7} \emptyset$ |
| $GeneB_2(t) \xrightarrow{\varepsilon \cdot k_1} GeneB_2(t+\tau_{ps}) + M_B(t+\tau_{1B})$ | $GeneB_1 + P_A \xrightarrow{k_3} GeneB_2$ | |
| $M_B(t) \xrightarrow{k_2} M_B(t+\tau_{rbs}) + P_B(t+\tau_{2B})$ | $GeneB_2 \xrightarrow{k_5} GeneB_1 + P_A$ | |

**Table 2.1** The complete set of reactions behind the delayed toggle switch model. The delays are confined to the transcriptional and translational reactions, and timing arguments have been omitted from the instantaneous reactions. The subscript notation for the genes refers to how many proteins are bound to the gene as transcription factors: $GeneX_0$ is an actively transcribing gene, while $GeneX_1$ and $GeneX_2$ are both repressed, with some leakage transcription included in the model ($\varepsilon = 10^{-3}$). The delays $\tau_{1A}$ and $\tau_{1B}$ are explicitly different, to emphasize the subsequent independent investigation. The translation delays $\tau_{2A}, \tau_{2B}$ assume generic 1000 *bp* genes. For reaction rate parameters, see [Elowitz and Leibler, 2000].

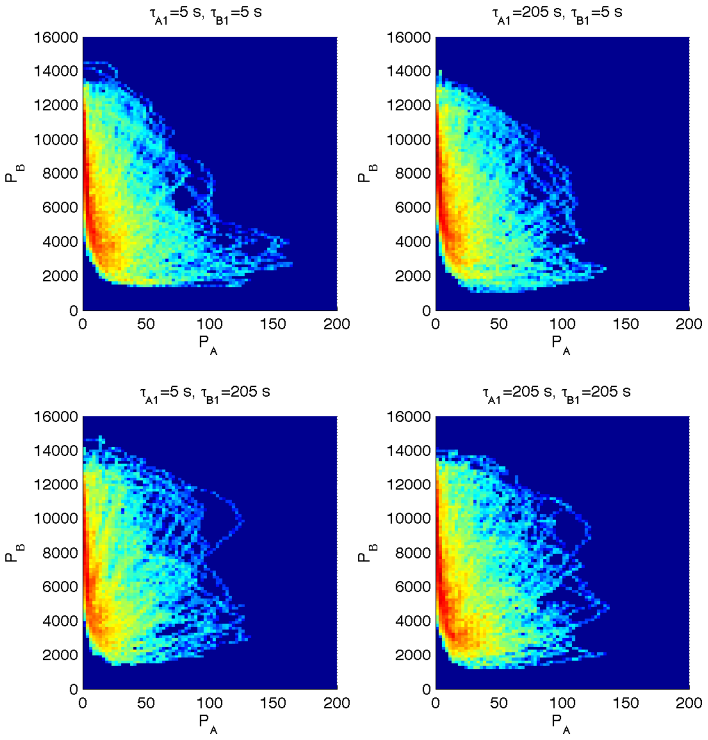

**Figure 2.8** Histograms across the projected phase space at four different delay parameterizations, when the toggle switch resides in the ($P_A$ =Off, $P_B$ =On) basin of attraction. Examining 300 simulated hours of data, we see little change in the steady-state distribution. The histogram colormap is logarithmic in scale, seeing as the system spends the majority of its time near $P_A = 0$. Notice the difference in scale between $P_A$ and $P_B$.

second) transcriptional delays incorporated in either Gene A, Gene B, or both.

Comparing the histograms shows that the effect of synthetic delays (on either transcriptional pathway) upon the stability of the toggle switch is difficult to identify conclusively, and appears to be negligible or nonexistent. The DDE analysis performed later will corroborate this finding, though in fact show that the stability of the system is slightly reduced for extremely long delays, when $\tau > 500$ seconds. This direction of change agrees with the traditional understanding of delays within control and dynamical systems, that retarded feedback mechanisms result in decreased stability for equilibrium points.

From our stochastic model analysis, we conclude that the toggle switch does not exhibit any significant sensitivity to delay. Our analysis becomes considerably more interesting when we turn our attention to oscillatory networks in the following two sections.

## 2.5 Analysis: Repressilator

Next, we study the delay-dependent stability of the Elowitz ‘repressilator’, an oscillator consisting of three genes expressing proteins that repress each other in a ring [Elowitz and Leibler, 2000]. As opposed to the functional toggle switch, when the repressilator is studied *in vivo* in *E. coli*, its oscillations are evident but disappointingly far from regular. This irregularity is a central motivation for this thesis, and

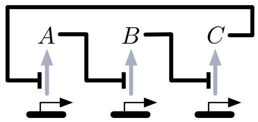

**Figure 2.9** A schematic diagram of the Elowitz repressilator genetic regulatory network, featuring three genes that express proteins that act as transcription factors, repressing each other in a ring.

engineering stability into synthetic regulatory networks such as the repressilator can be considered a leading challenged in the emerging discipline of synthetic biology. With the investigation of delay-dependent stability presented here, novel engineering principles emerge that may be used to increase the stability of the oscillator.

A schematic diagram of the regulatory network is shown in Figure 2.9. The diagram offers an effective overview of the regulatory network, while once again a more extensive documentation of the reaction network is shown in Table 2.5. Like the toggle switch, the repression mechanism used in this model features combinatorial OR gates. This approach is consistent with the stochastic model (without delays) used in the original repressilator paper [Elowitz and Leibler, 2000]. Combinatorial operators were chosen to obtain cooperatively in repression analogous to the ODE modeling approach in the same paper, which is the basis for our DDE model seen later.

First we will study the nominally delayed model. A comparison of the published, non-delayed model and the nominally delayed model can be seen below in Figure 2.10, presented as trajectories in the three-dimensional state space $(P_A \times P_B \times P_C)$. Under a general principle of ergodicity, we have chosen to study one trajectory sample for a ‘long time’, rather than an ensemble of trajectories. We see in the figure that the non-delayed model exhibits a notably different energy landscape inferable from the phase-space trajectories, compared to the more correct, nominally delayed model. In the nominally delayed model, we observe more densely concentrated behavior,

| Transcription & Translation | Promoter region | Decay |
|---|---|---|
| $GeneA_0(t) \xrightarrow{k_1} GeneA_0(t+\tau_{ps}) + M_A(t+\tau_1)$ | $GeneA_0 + P_C \xrightarrow{k_3} GeneA_1$ | $P_A \xrightarrow{k_6} \emptyset$ |
| $GeneA_1(t) \xrightarrow{\varepsilon \cdot k_1} GeneA_1(t+\tau_{ps}) + M_A(t+\tau_1)$ | $GeneA_1 \xrightarrow{k_4} GeneA_0 + P_C$ | $M_A \xrightarrow{k_7} \emptyset$ |
| $GeneA_2(t) \xrightarrow{\varepsilon \cdot k_1} GeneA_2(t+\tau_{ps}) + M_A(t+\tau_1)$ | $GeneA_1 + P_C \xrightarrow{k_3} GeneA_2$ | |
| $M_A(t) \xrightarrow{k_2} M_A(t+\tau_{rbs}) + P_A(t+\tau_{2A})$ | $GeneA_2 \xrightarrow{k_5} GeneA_1 + P_C$ | |
| $GeneB_0(t) \xrightarrow{k_1} GeneB_0(t+\tau_{ps}) + M_B(t+\tau_1)$ | $GeneB_0 + P_A \xrightarrow{k_3} GeneB_1$ | $P_B \xrightarrow{k_6} \emptyset$ |
| $GeneB_1(t) \xrightarrow{\varepsilon \cdot k_1} GeneB_1(t+\tau_{ps}) + M_B(t+\tau_1)$ | $GeneB_1 \xrightarrow{k_4} GeneB_0 + P_A$ | $M_B \xrightarrow{k_7} \emptyset$ |
| $GeneB_2(t) \xrightarrow{\varepsilon \cdot k_1} GeneB_2(t+\tau_{ps}) + M_B(t+\tau_1)$ | $GeneB_1 + P_A \xrightarrow{k_3} GeneB_2$ | |
| $M_B(t) \xrightarrow{k_2} M_B(t+\tau_{rbs}) + P_B(t+\tau_{2B})$ | $GeneB_2 \xrightarrow{k_5} GeneB_1 + P_A$ | |
| $GeneC_0(t) \xrightarrow{k_1} GeneC_0(t+\tau_{ps}) + M_C(t+\tau_1)$ | $GeneC_0 + P_B \xrightarrow{k_3} GeneC_1$ | $P_C \xrightarrow{k_6} \emptyset$ |
| $GeneC_1(t) \xrightarrow{\varepsilon \cdot k_1} GeneC_1(t+\tau_{ps}) + M_C(t+\tau_1)$ | $GeneC_1 \xrightarrow{k_4} GeneC_0 + P_B$ | $M_C \xrightarrow{k_7} \emptyset$ |
| $GeneC_2(t) \xrightarrow{\varepsilon \cdot k_1} GeneC_2(t+\tau_{ps}) + M_C(t+\tau_1)$ | $GeneC_1 + P_B \xrightarrow{k_3} GeneC_2$ | |
| $M_C(t) \xrightarrow{k_2} M_C(t+\tau_{rbs}) + P_C(t+\tau_{2C})$ | $GeneC_2 \xrightarrow{k_5} GeneC_1 + P_B$ | |

**Table 2.2** The complete set of reactions behind the delayed Elowitz repressilator model. The translation delays $(\tau_{2X})$ are given by the lengths of the genes from the original design, (1080, 675, 858) base pairs at 45 $bp/s$ results in $(\tau_{2A}, \tau_{2B}, \tau_{2C}) = (24, 15, 19)$ seconds. All other delays are the same as for the toggle switch. For reaction rate parameters, see [Elowitz and Leibler, 2000].

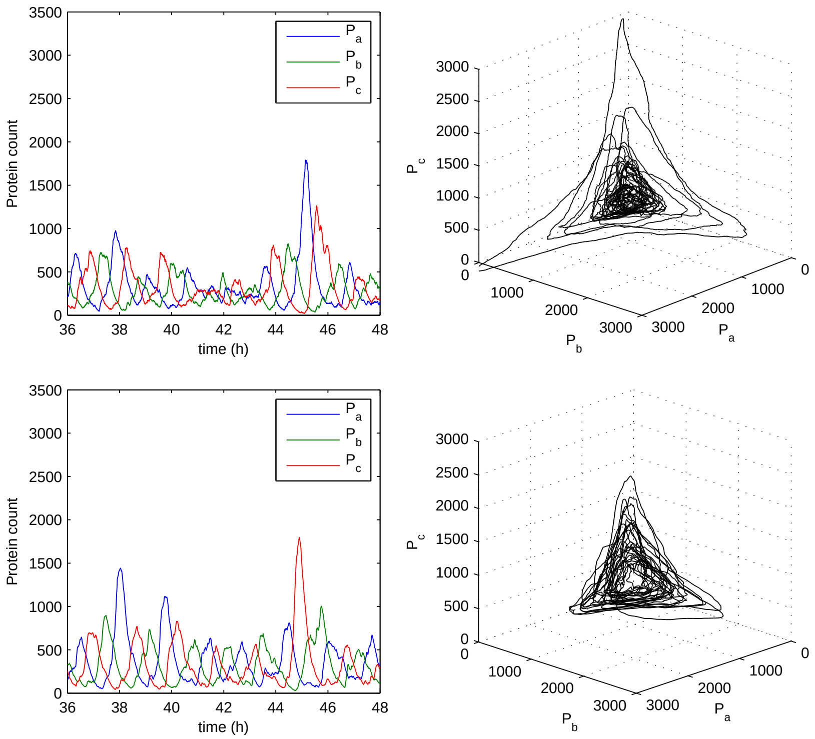

**Figure 2.10** A comparison between the (above) non-delayed and (below) nominally delayed ( $\tau_1 = 5$ s) stochastic models of the repressilator. To the left, we see 12 hours of time series data for the protein concentrations in the system, after the system has entered steady-state. To the right, we see a phase space ($P_A \times P_B \times P_C$) showing 48 hours of simulated behavior. From the phase space diagrams it is evident that the model incorporating delay features a significantly more attractive limit cycle, with fewer extreme oscillations.

with fewer extremal oscillations, and we also see evidence of a stronger repelling force away from the equilibrium point at the center of the oscillator. Qualitatively speaking, we see that the delayed model oscillates along a more stable limit cycle. Does this improvement continue when we introduce synthetic delays?

When we modify the delay time $\tau_1$ for the three genes, as suggested earlier and diagrammed in Figure 2.5, we see that the trend continues. Figure 2.11 shows histograms across the phase space, projected into the plane normal to the equilibrium axis $(P_A = P_B = P_C)$, at 4 different delay parameterizations. First the nominal delay of $\tau_1 = 5$ seconds, and then with increasing delays of 105, 205, and 305 seconds, assuming ‘junk’ of ‘other gene’ regions corresponding to 100, 200, and 300 seconds of delay.

In Figure 2.11, the normalized stationary autocorrelation functions for $P_A(t)$ are also included at each delay configuration. These functions clearly show how the autocorrelation peaks at integer multiples of the oscillation period become much stronger as delays are introduced. Without synthetic delay, the autocorrelation is washed out by oscillator phase noise, muddling the peaks. The strengthening of these peaks offers a good quantification of oscillator stability improvement for the repressilator stochastic model, but unfortunately, a similar analysis for the relaxation oscillator in the next section will say almost nothing at all. Therefore we can not rely on these autocorrelation peaks as a measure of oscillator stability.

For the repressilator, increasing the delay on all three constituent genes appears to increase the stability of the limit cycle attractor at the core of the stochastic dynam-

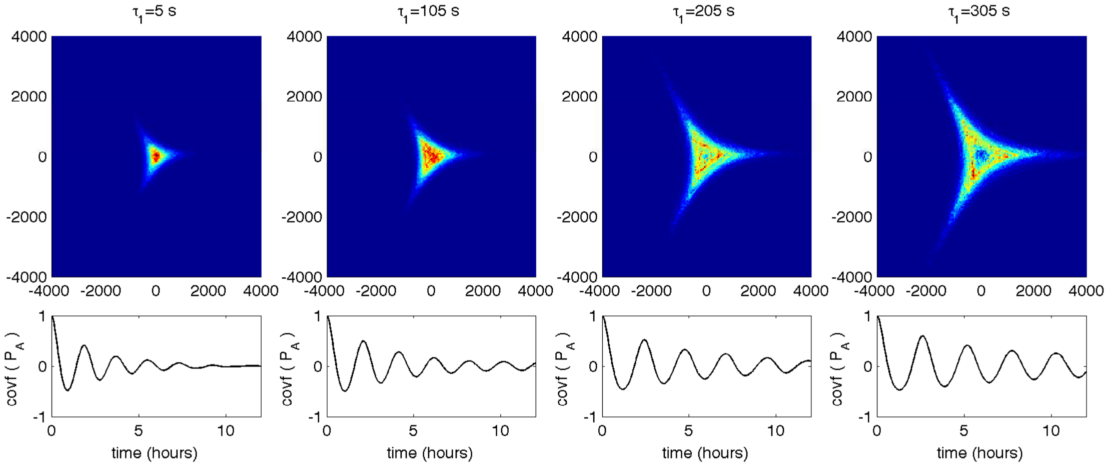

**Figure 2.11** Histograms across the projected phase space at four different delay parameterizations, along with the autocorrelation function of $P_A(t)$ at that parameterization. The delay is increased on all three genes. Examining 1440 simulated hours of data, we see that as delay is introduced, a 'hole' in the limit cycle becomes evident, caused by increased repelling forces away from the equilibrium point at the center of the oscillator. The autocorrelation functions clearly show how the autocorrelation between cycles of the oscillator are attenuated by delay, overpowering the phase noise that muddles the nominal oscillator.

ics. Based on simulation data not shown here, this improvement is in fact the result of identical contributions from all three individual genes, and while it breaks the symmetry of the oscillator limit cycle, it is fully possible to delay only one gene and see an improvement in the attractor stability. But inquiries such as this require prohibitively large simulation efforts to investigate fully, since a computationally massive simulation must be run at each parameter configuration. In depth investigations are much more feasible using the computationally tractable DDE stability analysis that follows in the next chapter.

## 2.6 Analysis: Relaxation oscillator

Lastly, we study the delay-dependent stability of the Barkai-Leibler relaxation oscillator. The relaxation oscillator was developed by Barkai and Leibler [Barkai and Leibler, 2000; Vilar *et al.*, 2002] based on a straightforward hysteretic construction, whereby a primary gene activates itself and an associate gene, which in turn represses the primary gene in a delayed fashion or on a slower timescale. Transcriptional regulation is used for activation, and a hypothetical direct protein-protein interaction between proteins A and B is used to form a decaying protein C, see [Barkai and Leibler, 2000] for details. A schematic diagram of the network is shown in Figure 2.12, and the details of the reaction network are show in Table 2.6. In contrast to the toggle

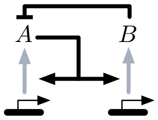

**Figure 2.12** A schematic diagram of the Barkai-Leibler relaxation oscillator genetic regulatory network. The protein C is now shown in the diagram, but serves as an intermediate species along the decay pathway when protein B represses A directly.

| Transcription & Translation | Promoter region | Decay |
|---|---|---|
| $GeneA_0(t) \xrightarrow{\varepsilon\cdot k_1} GeneA_0(t+\tau_{ps})+M_A(t+\tau_{1A})$<br>$GeneA_1(t) \xrightarrow{k_1} GeneA_1(t+\tau_{ps})+M_A(t+\tau_1)$<br>$M_A(t) \xrightarrow{k_2} M_A(t+\tau_{rbs})+P_A(t+\tau_2 A)$ | $GeneA_0+P_A \xrightarrow{k_3} GeneA_1$<br>$GeneA_1 \xrightarrow{k_4} GeneA_0+P_A$ | $P_A \xrightarrow{k_5} \emptyset$<br>$M_A \xrightarrow{k_6} \emptyset$ |
| $GeneB_0(t) \xrightarrow{\varepsilon\cdot k_1} GeneB_0(t+\tau_{ps})+M_B(t+\tau_{1B})$<br>$GeneB_1(t) \xrightarrow{k_1} GeneB_1(t+\tau_{ps})+M_B(t+\tau_1)$<br>$M_B(t) \xrightarrow{k_2} M_B(t+\tau_{rbs})+P_B(t+\tau_2 B)$ | $GeneB_0+P_A \xrightarrow{k_3} GeneB_1$<br>$GeneB_1 \xrightarrow{k_4} GeneB_0+P_A$ | $P_B \xrightarrow{k_5} \emptyset$<br>$M_B \xrightarrow{k_6} \emptyset$ |
| | | $P_A+P_B \xrightarrow{k_7} P_C$<br>$P_C \xrightarrow{k_8} P_B$ |

**Table 2.3** The complete set of reactions behind the delayed Barkai-Leibler relaxation oscillator model. The delays $\tau_{1A}$ and $\tau_{1B}$ are explicitly different, to emphasize the subsequent independent investigation. The subscript notation for the genes refers to how many proteins are bound to the gene as transcription factors. $GeneX_0$ is an actively transcribing gene, while $GeneX_1$ is repressed. The translation delay $\tau_{2X}$ assumes generic 1000 base pair genes. For reactions rate parameters see [Vilar *et al.*, 2002].

switch repressilator model, the relaxation oscillator uses activation promoter sites, and combinatorial operators were not used when modeling this network.

The relaxation oscillator does not possess the simple symmetry of the repressilator, and thus the delay-dependance is much more interesting. In fact, when delaying the relaxation oscillator, the stability of the limit cycle attractor is either increased or decreased, depending on which pathway is delayed. This fact begins to hint at the complexity that emerges when investigating large regulatory networks, and further underscores the need for a computationally efficient method of analysis, whereby individual pathways can be easily studied in isolation.

Figure 2.13 shows a histogram of the relaxation oscillator state-space projected

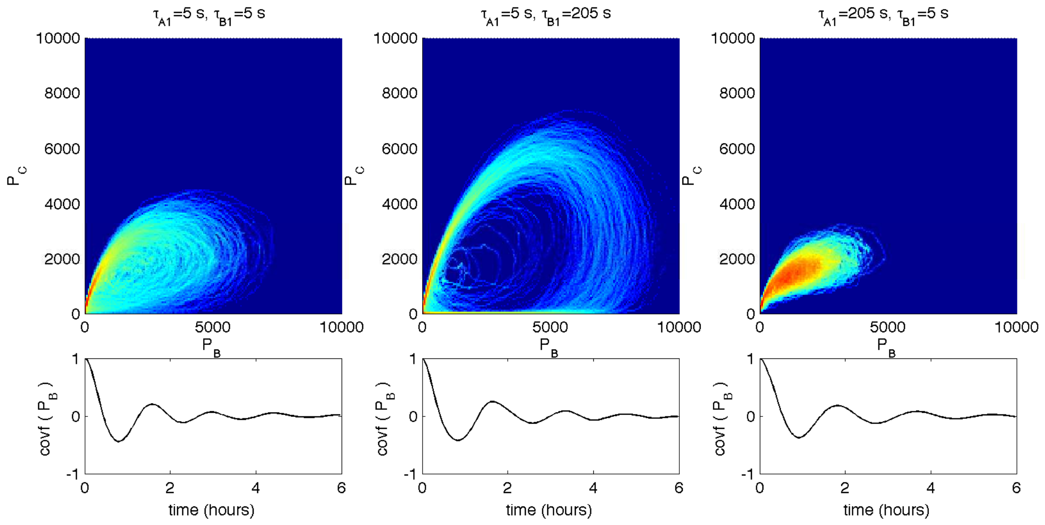

**Figure 2.13** Histograms across the projected phase space at (left) the nominal delay configuration, (center) when gene B is delayed, and (right) when gene A is delayed, along with the autocorrelation function of $P_B(t)$ at that parameterization. The histogram colormap is logarithmic in scale. While the phase space histogram shows the delay-dependancy clearly, the autocorrelation functions provide no insight at all.

into the $(P_B \times P_C)$ plane. Introducing a delay onto the gene A mRNA transcript decreases the stability of the oscillator, while a delay on gene B increases the stability. This is consistent with the separation of timescales dynamics at work in this hysteresis-based oscillator, whereby delaying gene B retards the oscillators slow repression pathway, keeping the fast auto-activation pathway untouched.

Unfortunately, the autocorrelation peaks that summarized the stability improvement of the repressilator so well say little or nothing here. The reason for this is not clear, but if we permit speculation, it seems to relate to the direction of the noise with respect to the limit cycle. For the repressilator, the noise reduction brought by delays seems to primarily suppress disturbances *along* the limit cycle, leading to a reduction of phase noise. Meanwhile, the delay-induced noise reduction for the relaxation oscillator seems to be primarily perpendicular to the trajectory of the limit cycle, and delays do little to effect the phase noise of the stochastic oscillations. We emphasize that this explanation is merely speculation, and no clear answer is immediately apparent.

Analyzing the stability of the relaxation oscillator limit cycle reveals that even the simplest networks can have complex delay-dependancies. Our qualitative analysis nonetheless revealed the possibility of engineering stability by delaying the secondary helper gene of the network (Gene B).

As we have iterated many times by this point, performing anything more than a crude local analysis of a discrete, stochastic model of genetic regulatory network is prohibitively slow. For this reason, we now (finally) turn to DDE analysis of simpler, deterministic models, where we develop more precise measures of delay-dependent stability using a wide range of results from numerical analysis and the theory of dynamical systems.

# 3. DDE Stability Analysis

In this chapter, we outline methods from the stability analysis of delay-differential equations (DDEs), and demonstrate their applicability to our bioengineering inquiry. The analysis presented here is well grounded in the theory of dynamical systems, and if offers a method for quantifying the delay-dependance of the attractor stability qualitatively ‘observed’ during the previous stochastic analysis.

First, we study the stability of fixed points of systems of DDEs, applying linear stability analysis and principles from numerical analysis and control theory. Next, we study the more complicated stability of limit cycles of systems of DDEs, where our approach is based on Floquet theory and methods from numerical analysis for boundary value problems (BVPs). In each section we first review the theoretical foundations of the analysis before presenting its application to bioengineering. This stability analysis of DDE models provides a succinct summary of how delays can be used to engineer stable genetic oscillators.

## 3.1 Delay-differential equations

Often genetic regulatory networks are modeled as a system of differential equations, where species assume positive, continuous concentrations, and interact instantaneously and deterministically in continuous time, in a time invariant fashion. Mathematically speaking, such models simply describe the reaction network as an initial value problem (IVP):

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t)) \tag{3.1}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n, \tag{3.2}$$

where $\mathbf{x}$ is the vector of $n$ species concentrations, and $\mathbf{x}_0$ is the initial concentrations. In reality the concentrations are positive, and $\mathbf{x} \in \mathbb{R}^n_+$, but we will consider $\mathbf{x} \in \mathbb{R}^n$, and consistent models will naturally confine the system to the positive subspace.

As should be clear to the reader at this point, in reality, these reaction networks very often contain delays. Such systems are more accurately modeled through a dependance upon a series of historical values, forming the following DDE:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{x}(t-\tau_1), \ldots, \mathbf{x}(t-\tau_m)), \tag{3.3}$$

$$\mathbf{x}_t(s) = \phi(s) \in C^0([-\tau, 0], \mathbb{R}^n), \quad , \tag{3.4}$$

where $\mathbf{x}_t(s) := \mathbf{x}(t+s)$, $s \in [-\tau, 0]$ will be explained below. This is a special case of what is more generally known as a functional differential equation (FDE), where the right hand side can potentially involve functionals (e.g. integrals) operating on historic values over some distributed interval of time (as opposed to discrete sample times). This more general case need not be considered here as it does not occur in our regulatory network systems. See [Hale and Lunel, 1993] for details.

There are many fundamental differences between ODEs and DDEs, and in spite of their superficial similarity, the introduction of even a single delay introduces several complications. The most significant difference, with widespread consequences, is that the state of the system at time $t$ is no longer completely defined by the state of the system at that time ($\mathbf{x}(t) \in \mathbb{R}^n$), but rather by the historical values of the state on the entire interval $[t-\tau, t]$, here denoted $\mathbf{x}_t(s)$. This continuous vector-valued *history function* defined over the entire interval is in fact the infinite-dimensional state for the system. This in turn complicates the initial conditions, which then must be defined as a history function. For an excellent overview directed towards pointing out computational challenges, see [Bellen and Zennaro, 2003].

To speak in less general terms, in physical situations the function $f$ is widely separable, owing to the fact that direct interactions between different historical states constitute a physical impossibility. This effectively restricts our study to formulations of the following type:

$$\frac{d\mathbf{x}(t)}{dt} = f_0(\mathbf{x}(t)) + \sum_{i=1}^{m} f_i(\mathbf{x}(t-\tau_i)), \tag{3.5}$$

$$\mathbf{x}(t) = \phi(t), \quad t \in [-\tau, 0], \tag{3.6}$$

where $f_0$ might model species decay, and $f_i$ might model different transcriptional regulation pathways, while all map from $\mathbb{R}^n \to \mathbb{R}^n$, as with $f$. This separated formulation does not in itself admit any novel methods of analysis, but can be useful for developing intuition.

## 3.2 Stability of fixed points

Here we present three methods for analyzing the stability of fixed points in delay-differential equations. First, we present a restricted explicit approach with limited (but nonetheless some) applicability. Second, we present a general numerical method based on well understood approximations. These two methods are largely borrowed from [Jarlebring, 2008], a recent PhD thesis that surveys the subject of numerical methods for linear DDE stability well. The methods are then applied to the Gardner toggle switch model to study the delay-dependent stability of the DDE model. Model reduction techniques for studying fixed points of DDE models of genetic regulatory networks as presented in [Chen and Aihara, 2002a] are omitted here due to their severely restricted applicability.

### Linearization

Fixed points in delayed systems can be derived in the same manner as for ordinary differential equations. Taking $\frac{d\mathbf{x}(t)}{dt} = 0$ and $\mathbf{x}(t) = \mathbf{x}(t-\tau_i) = \mathbf{x}^*$, we solve for the zeros of the following equation:

$$0 = f(\mathbf{x}^*, \mathbf{x}^*, \ldots, \mathbf{x}^*). \tag{3.7}$$

Because we are generally studying multi-stable systems, there will typically be several fixed points. It can be noted that the Gardner toggle switch exhibits symmetries that result in two stable fixed points (and one unstable fixed point) with identical stability and delay-sensitivity. But in general, the procedure that follows must be repeated at each fixed point that one wishes to study.

In order to study stability, we then linearize (3.5) over the current and delayed states, at the chosen fixed point, to obtain:

$$\frac{d\mathbf{x}(t)}{dt} = A_0\mathbf{x}(t) + \sum_{i=1}^{m} A_i\mathbf{x}(t-\tau_i), \tag{3.8}$$

$$A_i = \left.\frac{\partial f_i(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*}. \tag{3.9}$$

Here $f_i$ and $\mathbf{x}$ are all vector valued, and so the matrices $A_i$ constitute the typical Jacobian matrices.

### Explicit approach

In this section, we offer an explicit approach for systems of linear DDEs as presented in (3.8), but the applicability is limited to systems with one delay. These types of systems are in fact rather common in control and other disciples, and for a limited subset of these problems, explicit expressions for the eigenvalues have been derived. The central tool for these results is the Lambert W function, the inverse function of $f(W) = We^W$, which is in fact of curiously old origins [Lambert, 1758; Euler, 1779]. To the best of the our knowledge, the relation to delay differential equations was first made in [Wright, 1948], and it is also discussed in [Hale and Lunel, 1993]. The review by [Corless *et al.*, 1996] can be seen as ushering in its modern era of application.

The Lambert W function arises when considering solutions to the characteristic equation of linear DDE problems as in (3.8), with one delayed dependancy:

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= A_0\mathbf{x}(t) + A_1\mathbf{x}(t-\tau), && (3.10) \\ \mathbf{x}(t) &= \phi(t), \quad t \in [-\tau, 0]. && (3.11) \end{aligned}$$

The characteristic equation can be derived through an ansatz of the solution type $\mathbf{x}(t) = \mathbf{x}_0 e^{At}$, from which we obtain the transcendental characteristic equation:

$$0 = \det(-sI + A_0 + A_1 e^{-s\tau}) \qquad (3.12)$$

The roots of this function are infinite, owing to its transcendental nature, and in agreement the infinite dimensional structure of DDEs. The roots can be expressed using the Lambert W function, according to the following theorem [Jarlebring and Damm, 2007]:

THEOREM 3.2.1
If $A_0$ and $A_1$ are simultaneously triangularizable (with ($A_0$, $A_1$) commuting being a sufficient special case), then

$$\sigma(\mathscr{A}) = \bigcup_k \sigma\left(\frac{1}{\tau}W_k(A_1\tau e^{-A_0\tau}) + A_0\right) \qquad (3.13)$$

□

Here $W_k$ denotes the $k$th branch of the $W$ function. In practice, the branches of the $W_k$ function are evaluated efficiently to machine accuracy using Halley's method [Hildebrand, 1987], as discussed in [Corless *et al.*, 1996]. The requirement that $A_0$ and $A_1$ be simultaneously triangularizable is rather constraining, but fortunately some systems do exhibit this property, by virtue of commuting. In our biological context, if the instantaneous dependance $A_0$ only contains decay terms with a symmetric decay rate across chemical species ($A_0 = \beta I_n$), then ($A_0$, $A_1$) certainly commute.

It is worth noting that in the case where $A_0$ and $A_1$ are scalar, we obtain a scalar delay-differential equation, and the result presented in Theorem 3.2.1 is well known. The correct generalization to systems of equations came only recently, with the work of Jarlebring.

Some properties of the $W$ function operating on matrices will now be briefly discussed. We note that the Lambert $W$ function operates block-wise on matrices in Jordan normal form, where different branches can be accessed by different blocks:

$$W_k(J) = \operatorname{diag}(W_{k_1}(J_{n_1}(\lambda_1)), \ldots, W_{k_m}(J_{n_m}(\lambda_m))). \qquad (3.14)$$

By also noting the permissible transformation $W_k(A) = SW_k(J)S^{-1}$, we know enough to analyze the spectrum of basic systems where all eigenvalues are simple. For systems possessing Jordan blocks of dimension greater than one, additional techniques presented in [Jarlebring and Damm, 2007] are needed.

### Numerical approach

The numerical approach is not limited to only one delay term, and offers a general approach that can be validated against the explicit approach for single-delay, simultaneously triangularizable problems. For pedagogical simplicity, we will however limit our presentation to systems with a single delay. The approach consists of several steps, where the linearized system (3.10) is first translated to a partial differential equation (PDE) formulation, exactly describing the DDE, and then the spectrum of the PDE operator are analyzed through Chebyshev spectral semi-discretization.

Linear delay differential equations can be though of as abstract Cauchy transport problems with non-local boundary conditions [Hale and Lunel, 1993; Jarlebring, 2008]. By introducing a dummy variable $\theta$ to describe a *memory dimension* in a clever way, the system in (3.10) can be described by $u(t,\theta) = \mathbf{x}(t+\theta)$ solving:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial u}{\partial \theta}, && (3.15)\\ \frac{\partial u(t,0)}{\partial \theta} &= A_0 u(t,0) + A_1 u(t,-\tau), && (3.16)\\ u(0,\theta) &= \phi(\theta), \quad \theta \in [-\tau, 0], && (3.17) \end{aligned}$$

where (3.16) is the boundary value mimicking (3.10), and (3.17) is the initial condition. Note that this formulation is consistent even for vector valued problems ($n > 1$). This linear PDE formulation is also infinite-dimensional, and it is rigorously identical to the linearized problem studied in (3.10). This means that we can study the stability of (3.15), and it will correspond to the stability properties of the DDE, linearized at the fixed point of interest.

By introducing $\mathscr{A}$ as the differentiation operator in $\theta$, (3.15) becomes

$$\frac{\partial}{\partial t} u(t,\theta) = \mathscr{A} u(t,\theta), \tag{3.18}$$

and we can now formulate a tractable approximation of the time dynamics by discretizing the memory dimension $\theta$, with a grid $\Theta = \{\theta_0, \ldots, \theta_N\}$, and then approximate the differentiation operator $\mathscr{A}$ with the differentiation matrix $A_N$, which will be especially constructed for the task. The history function can be sampled, and the solution elements $u_N^i(t) \approx u(t,\theta_i) = \mathbf{x}(t+\theta_i)$ (still vector valued) represents the solution at the $(N+1)$ gridpoints in $\Theta$:

$$\begin{aligned} \frac{d}{dt} u_N(t) &= A_N u_N(t), && (3.19)\\ u_N(0) &= [\phi(\theta_0)^T, \ldots, \phi(\theta_N)^T]^T && (3.20) \end{aligned}$$

This is a finite-dimensional linear system of ordinary differential equations, approximating the infinite-dimensional PDE, brought about by discretization in $\theta$, and we can now readily access the approximate spectral properties of the operator $\mathscr{A}$ by analyzing $A_N$, and thereby grasp the stability of the system evolving in time, which is precisely our interest.

The discretization of the operator is a somewhat intricate matter. A crude approach would involve selecting a uniform grid $\Theta$, and approximate the differentiation using a (forward Euler) finite difference [Bellen and Maset, 2000]. This results in the classic differentiation matrix, with the boundary condition $\frac{d}{dt}u_N^N(t) = A_0 u_N^0(t) + A_1 u_N^N(t)$ absorbed into the matrix:

$$
\begin{aligned}
A_N &= \begin{bmatrix} D_N \otimes I_n \\ A_1\ 0 \cdots 0\ A_0 \end{bmatrix} \in \mathbb{R}^{n(N+1)\times n(N+1)} && (3.21)\\
D_N &= \frac{1}{h}\begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{N\times(N+1)} && (3.22)
\end{aligned}
$$

where $h = \tau/N$ is the uniform grid spacing, and the Kronecker product operates on an identity matrix of the same dimension as the original system ($n$), keeping our derivation relevant for higher dimensional systems.

This crude finite difference approach is in practice very crude, and a much better approach involves discretizing the memory dimension $\theta$ with non-uniform Chebyshev nodes, and using a Chebyshev differentiation matrix in place of Euler forward differentiation [Breda *et al.*, 2005]. The theory of spectral differentiation will not be reviewed here, see [Trefethen, 2000] for details. In practice, we select a non-uniform discretization grid $\Theta = \{\cos(j\pi/N)\}_{j=0}^{N} \in [-1,1]$ (which is rescaled below), and then constructing the nominal Chebyshev differentiation matrix $D_N \in \mathbb{R}^{(N+1)\times(N+1)}$ as:

$$
\begin{aligned}
(D_N)_{00} &= \tfrac{2N^2+1}{6}, \\
(D_N)_{NN} &= -\tfrac{2N^2+1}{6}, \\
(D_N)_{ii} &= \tfrac{-\theta_i}{2(1-\theta_i^2)}, \qquad j = 1,\ldots,N-1, \\
(D_N)_{ij} &= \tfrac{c_i}{c_j}\tfrac{(-1)^{i+j}}{(\theta_i-\theta_j)}, \quad i,j = 0,\ldots,N-1, \quad i \neq j
\end{aligned}
\qquad (3.23)
$$

where

$$
c_i = \begin{cases} 2, & i = 1, N \\ 1, & \text{otherwise.} \end{cases} \qquad (3.24)
$$

This nominal Chebyshev differentiation matrix is then transformed to the relevant interval as $\hat{D}_N = -\frac{2}{\tau}D_N$, which places the interval on $[-\tau/2, \tau/2]$, but since we are ultimately interested only in the stability properties of the operator here, and not in simulation, we do not need to shift the interval. The multiplication by $-1$ assure agreement with the sequential order of the solution vector, where $u_N^0(t) = u(t,\theta_0) = u(t,-\tau) = \mathbf{x}(t-\tau)$ and $u_N^N(t) = \mathbf{x}(t)$. The boundary condition is again imposed by replacing the last row of the matrix by the boundary condition. We thus achieve our operator approximation $A_N$ (in matlab syntax, enumerating from 1), as:

$$
A_N = \begin{bmatrix} -\frac{2}{\tau}\hat{D}_N(1:N, 1:N+1) \otimes I_n \\ A_1\ 0 \cdots 0\ A_0 \end{bmatrix} \in \mathbb{R}^{n(N+1)\times n(N+1)} \qquad (3.25)
$$

We can now study the eigenvalues of (3.25) as an approximation of the differential operator $\mathscr{A}$ with so-called “spectral accuracy”. Loosely speaking, spectral accuracy implies that if the solution $u$ has $p-1$ continuous derivatives in $L^2(\mathbb{R})$ for some $p \geq 1$, then the convergence is $O(N^{-(p+1)})$, and if $u$ is analytic then the convergence is $O(c^N)$, where $c \in (0,1)$. Again, see [Trefethen, 2000] and the references therein for details. This rapid convergence allows us to analyze the stability of even large networks in a computationally tractable manner. Thus, the previously derived explicit spectrum of a limited class of linear systems of DDEs is useful as a benchmark for the numerical method by comparison, but the numerical method is broadly preferable.

## 3.3 Analysis: Toggle switch

We will now apply both the explicit and numerical approaches to study the stability of the fixed points of the Gardner toggle switch [Gardner *et al.*, 2000] as the delay parameters are varied. The behavior of the toggle switch can be understood to follow the following reduced (in the sense that we do not model mRNA and protein separately) model, with decay, transcriptional regulation, and promoter leakiness:

$$\begin{aligned} \frac{dx_1(t)}{dt} &= -\beta x_1(t) + \frac{\alpha_1}{1+x_2(t-\tau_A)^2} + \alpha_0 && (3.26) \\ \frac{dx_2(t)}{dt} &= -\beta x_2(t) + \frac{\alpha_1}{1+x_1(t-\tau_B)^2} + \alpha_0, && (3.27) \end{aligned}$$

where $\beta = \log(2)/600$, $\alpha_1 = 10$, $\alpha_0 = 0.01$. Here we aggregate the many transcriptional and translational delays from the stochastic models into single delay terms. The system has two stable fixed points (and one unstable) in the positive quadrant, and we will focus out attention on the stable fixed points. Due to the symmetry of the model, we need only study the stability of one of the two, and the behavior of the other equilibrium point will be transposed but identical. We solve for the fixed points $x^*$ using a symbolic solver and obtain the linearized system:

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \underbrace{\begin{bmatrix} -\beta & 0 \\ 0 & -\beta \end{bmatrix}}_{A_0} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & \alpha_h \\ \alpha_l & 0 \end{bmatrix}}_{A_1} \begin{bmatrix} x_1(t-\tau) \\ x_2(t-\tau) \end{bmatrix}. \qquad (3.28)$$

Here the values $\alpha_h$ and $\alpha_l$ come from linearization at the fixed point, and are exchanged to study the other equilibrium. For the first equilibrium, where $x_1$ is “high” and $x_2$ is “low”, $\alpha_h = -0.1225$ and $\alpha_l = -5.6783 \cdot 10^{-8}$.

For this system, the matrices $(A_0, A_1)$ clearly commute, and so the explicit result presented in Theorem 3.2.1 becomes applicable. Below in Figure 3.1 we present a root locus plot of the first six eigenvalues of the system, found using the branches $W_0$, $W_{-1}$, and $W_1$ of the $W$-function, studied as we vary the delay parameter $\tau$. The spectrum from Theorem 3.2.1 reduces quite nicely for this system using the properties of the $W$-function presented earlier, and with $(\lambda_1, \lambda_2)$ as the eigenvalues of $A_1$, we obtain the following union of sets of two eigenvalues per branch of the $W$-function:

$$\sigma(\mathscr{A}) = \bigcup_k \sigma\left(\frac{1}{\tau}W_k(A_1\tau e^{-A_0\tau}) + A_0\right) \tag{3.29}$$

$$= \bigcup_k \sigma\left(\frac{1}{\tau}W_k(A_1\tau e^{\beta\tau I}) - \beta I\right) \tag{3.30}$$

$$= \bigcup_k \left\{\frac{1}{\tau}W_k(\lambda_1\tau e^{\beta\tau}) - \beta, \frac{1}{\tau}W_k(\lambda_2\tau e^{\beta\tau}) - \beta\right\}. \tag{3.31}$$

It can be noted from the plot that the eigenvalues continuously approach the eigenvalues of the non-delayed system as $\tau \longrightarrow 0$. This can also be deduced by evaluating the limit of (3.31).

We also present the same six eigenvalues calculated using the numerical method based on Chebyshev semidiscretization of the reformulated PDE, where we computer the spectrum of $A_N$ from (3.25). Our discretization uses a mere $N = 10$ nodes, and the resulting eigenvalues are for our purposes essentially identical to the explicit eigenvalues.

From this root locus plot we can deduce that the principle eigenvalue, understood to be the eigenvalue with the greatest real portion, is initially robust to delay, though it eventually moves slowly towards the imaginary axis (the stability boundary) as the delay $\tau$ is increased beyond $\tau = 500$ seconds. In fact, all of the eigenvalues

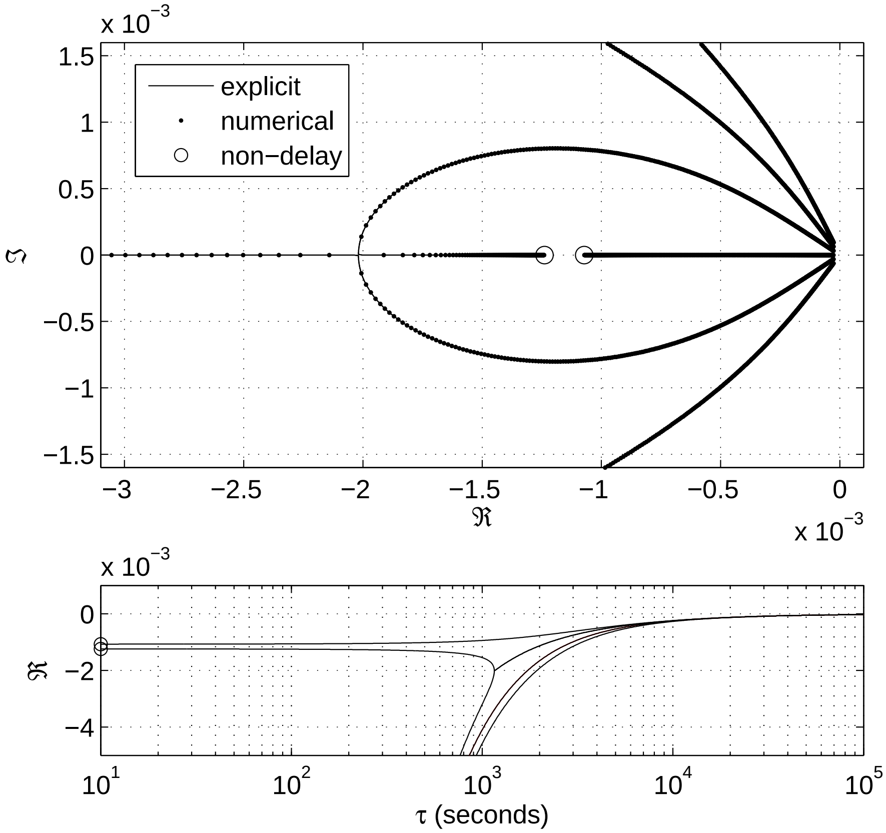

**Figure 3.1** A root locus plot showing the movement of the six principle eigenvalues as the delay parameter $\tau$ is varied (above), and the real portion of the same eigenvalues plotted versus the delay parameter, in seconds (below).

(even the infinitely many eigenvalues not shown) converge upon the origin as $\tau \to \infty$. This agrees with the delay-indifference observed when studying the stochastic toggle switch model.

In summary, this section provides a method for analyzing the delay-dependence of the stability of fixed points, and we find that for our toggle switch network, the stability of the operating points are generally indifferent to all but the largest delays, explaining the behavior of the stochastic model seen earlier. This destabilizing effect of delay can not be considered surprising nor particularly exciting, but as we shift our focus to limit cycles and oscillatory genetic networks, once again more complicated consequences can be suspected.

## 3.4 Stability of limit cycles

In this section we present the primary result of this thesis, a method for analyzing the delay-dependent stability of oscillators in genetic regulatory networks using tools from the theory of dynamical systems. These oscillatory networks can be modeled as nonlinear systems of delay-differential equations exhibiting limit cycles. The method of analysis first finds periodic solutions to nonlinear DDEs using a cleverly modified Newton's method-type solver, and then constructs a linear time integration operator, linearized at the periodic solution. The stability properties of the operator reveals the stability properties of the genetic regulatory network underlying the nonlinear system of DDEs being studied, and a measure of the oscillators rigidity can then be presented. We apply our analysis to the Elowitz ring oscillator (repressilator) and the Barkai-Leibler relaxation oscillator, showing how changes in delay topology affect the stability of the oscillations.

In previous work regarding the stability analysis of limit cycles to delay differential models of genetic regulatory networks, mathematical assumptions necessary to obtain analytic results have greatly restricted the class of relevant models. In [Chen and Aihara, 2002b], less quantitative conclusions are drawn regarding the stability of limit cycles based on the measure of the parameter region that leads to oscillatory behavior. This result uses singular perturbation theory and so it furthermore applies only to relaxation/hysteresis oscillators, exploiting the separation of timescales found there. In our work, we develop a more general quantitative result using well-understood numerical methods to analyze general networks.

### Linearizing at a periodic solution

The *monodromy operator* is a linear time integration operator mapping solutions to the variational equation from one period to the next, linearized around the periodic solution. The theory which underlies this periodic linearization is generally referred to as Floquet theory, see [Stokes, 1977; Hale and Lunel, 1993] for details. For a system with $m$ delays, the variational equation is given by,

$$\frac{d}{dt}\mathbf{y} = TA_0^*(t)\mathbf{y}(t) + T\sum_{i=1}^{m} A_i^*(t)\mathbf{y}(t-\frac{\tau_i}{T}) \tag{3.32}$$

where $A_i^*(t) = \frac{\partial f}{\partial x_i}\Big|_{u^*}$ are periodic linearizations of $f$ at the periodic solution $\mathbf{u}^*(t)$. Notice that for constant solutions (fixed points), this is the same as the linearization in (3.8). Floquet theory tells us that the solution to this system of equations has the separable form

$$\mathbf{y}(t) \quad = \quad P(t)e^{Bt}, \tag{3.33}$$

where $P(t)$ is $T$-periodic, $P(t) = P(t+T)$, $\forall t$. Then the monodromy operator $\mathscr{M}$ : $C_0\big([-\frac{\tau}{T},0],\mathbb{R}^n\big) \mapsto C_0\big([-\frac{\tau}{T},0],\mathbb{R}^n\big)$ corresponds to forward integration over the period $T$, and the spectrum is given by

$$\sigma(\mathscr{M}) \quad = \quad \sigma(e^{BT}), \tag{3.34}$$

see [Hale and Lunel, 1993] for details.

The monodromy operator always possesses a *trivial* eigenvalue (in this context also called the *characteristic (Floquet) multiplier*) of $\mu_1 = 1$, corresponding to a perturbation along the periodic solution. By studying the spectrum of characteristic multipliers, we can determine and quantify the stability of the periodic solution. For stable limit cycles,

$$|\mu_i| < 1, \qquad \forall i > 1,$$

and the smaller the characteristic multipliers $\mu_i$, the faster any perturbations away from the periodic solution along the corresponding eigenfunction $\psi_i$ die out. In this format, we are studying a simple problem of linear stability theory for a discrete map, and using this we can deduce the delay-dependent stability by studying the system under a range of delay parameterizations.

We seek periodic solutions $\mathbf{x}(t)$ to our DDE, where the history functions at time $t = 0$ and $t = T$ are identical. At this point it becomes appropriate to rescale the time dimension, such that $t := t/T$. With this, we seek a periodic solution where $\mathbf{x}_0 = \mathbf{x}_1$, by notation analogous to that which was introduced earlier, $\mathbf{x}_t \in C^0([t-\frac{\tau}{T},t],\mathbb{R}^n)$, where $\mathbf{x}_t(s) := \mathbf{x}(t+\frac{s}{T})$, $s \in [-\tau,0]$. We will only derive our results for systems with a single delay, but the generalization to systems with multiple (discrete) delays-dependencies is straight-forward. For notational simplicity, we also assume that $\tau < T$, a reasonable constraint that is however not necessary. In rescaled time we obtain a 'functional' boundary value problem on $t \in [0,1]$, as:

$$\frac{d\mathbf{x}(t)}{dt} \quad = \quad T f(\mathbf{x}(t),\mathbf{x}(t-\frac{\tau}{T})), \tag{3.35}$$

$$\mathbf{x}_0 \quad = \quad \mathbf{x}_1, \tag{3.36}$$

$$p(\mathbf{x}(t),T) \quad = \quad 0. \tag{3.37}$$

Here $T$ is treated as an unknown parameter (with nonlinear dependancies) that must be solved for alongside our solution. In (3.37) above we've been forced to add a *phase constraint*, introduced to arbitrarily fix a starting point in the period. The general aspects of this approach applied to finding limit cycles in ODEs with unknown periods is well outlined in [Parker and Chua, 1989], but as we will now see, DDEs present some complications, resolved primarily in [Engelborghs *et al.*, 2000].

Because the dependance from one point in the solution to another is non-trivial (when the period of the oscillation is not an even multiple of the delay time(s)), no simple discretization exists, and constructing a complete set of points with a closed dependancy is prohibitively difficult or impossible. Thus, instead of solving for the point values of a frequently sampled solution, instead the solution is parameterized as an interpolated solution across a mesh of points. In the work of [Engelborghs *et al.*, 2000], which is the basis of the implementation in the DDE-BIFTOOL toolbox [Engelborghs *et al.*, 2002], the set of piecewise polynomials (splines) are used to parameterize the solution. in order to fully document the methodology we use in this work, we will now outline the approach in sufficient detail.

Let $\Pi$ be a collection of meshpoints $0 = t_0 < t_1 < \ldots < t_L = 1$ that partition the interval $[0,1]$, typically a uniform grid. Set $h_i := t_{i+1} - t_i$ for $i = 0,\ldots,L-1$. Let

$\pi_m$ denote the set of all (vector-valued) polynomials of degree not exceeding $m$. We will approximate the solution to (3.35) by an element from the following space of piecewise polynomials:

$$S_m(\Pi) := \{\mathbf{u} \in C([0,T],\mathbb{R}^n) : \mathbf{u}|_{[t_i,t_{i+1}]} \in \pi_m, i = 0,\dots,L-1\}. \tag{3.38}$$

In each interval $[t_i, t_{i+1}]$ we then have a vector-valued polynomial of degree $m$, represented at the *representation points* within the interval:

$$t_{i+\frac{j}{m}} := t_i + \frac{j}{m}h_i,\ l = 1,\dots,m-1, \tag{3.39}$$

These points are a further refined uniform grid, and it is on this grid of points that we will obtain our solution. We can then formulate the interpolated solution $\mathbf{u}(t)$ with unknowns $\mathbf{u}(t_{i+\frac{j}{m}})$ as:

$$\mathbf{u}(t) := \sum_{j=0}^{m} \mathbf{u}(t_{i+\frac{j}{m}})P_{ij}(t), \qquad t \in [t_i, t_{i+1}] \tag{3.40}$$

where $P_{ij}(t)$ are the *Lagrange polynomials*,

$$P_{ij}(t) := \prod_{r=0,r\neq j}^{m} \frac{t - t_{i+\frac{r}{m}}}{t_{i+\frac{j}{m}} - t_{i+\frac{r}{m}}}, \quad j = 0,\dots,m. \tag{3.41}$$

Thus we have a degree-$m$ polynomial on each interval, across $L$ intervals, in $n$ dimensions, and so $\dim(S_m(\Pi)) = n \times (m \times L + 1)$ (note that interval endpoints of the piecewise polynomials are shared), and $u \in S_m(\Pi)$ is continuous but not necessarily continuously differentiable (at the mesh points). When not continuous, either a left-hand or right-hand derivative can be arbitrarily chosen, and the effect upon the solution is insignificant.

The solution parameterized in (3.40) can then be inserted into (3.35), and the Lagrange polynomials differentiated and inserted on the left hand side to obtain the *collocation equations* for each interval, which can be solved to evaluate the solution to (3.35) correctly at all *collocation points* $c$ per interval:

$$\begin{aligned}
\frac{d\mathbf{u}(c)}{dt} &= Tf(\mathbf{u}(c), \mathbf{u}(c - \frac{\tau}{T})), \quad \text{when } c - \frac{\tau}{T} \geq 0 && (3.42)\\
\frac{d\mathbf{u}(c)}{dt} &= Tf(\mathbf{u}(c), \mathbf{u}(c - \frac{\tau}{T} + 1)), \quad \text{when } c - \frac{\tau}{T} < 0, &&\\
\mathbf{u}_0 &= \mathbf{u}_L, && (3.43)\\
p(u,T) &= 0, && (3.44)
\end{aligned}$$

The collocation points $c$ in each interval $[t_i, t_{i+1}]$ can be chosen either uniformly, thus replicating the *representation points* in (3.39), or as the roots of Gauss-Legendre polynomials. Where a uniform (or any other general mesh) would result in $\mathcal{O}(h^m)$ convergence, Gauss-Legendre collocation points result in $\mathcal{O}(h^{m+1})$ convergence, and are implemented in the DDE BIFTOOL software package [Engelborghs *et al.*, 2000]. The collocation points $c_{i,l}$ are then:

$$\begin{aligned}
c_{i,l} &:= t_i + c_l h_i,\ l = 1,\dots,m, && (3.45)\\
0 &\leq c_1 < c_2 < \cdots < c_m \leq 1. && (3.46)
\end{aligned}$$

where $c_l$ are the zeros to the Gauss-Legendre polynomials, on the unit interval. We thus obtain one instance of (3.42) for each $i = 0, \dots, L-1$, $l = 1, \dots, m$. If we write

$$\tilde{c}_{i,l} = \begin{cases} c_{i,l} - \frac{\tau}{T} & , \quad \text{when } c - \frac{\tau}{T} \geq 0, \\ c_{i,l} - \frac{\tau}{T} + 1 & , \quad \text{when } c - \frac{\tau}{T} < 0, \end{cases} \tag{3.47}$$

and let $k$ be the integer such that $t_k \leq \tilde{c}_{i,l} < t_{k+1}$, then the collocation equations (3.42) take the form:

$$\sum_{j=0}^{m} \mathbf{u}_{i+\frac{j}{m}} P'_{ij}(c_{i,l}) = T f\left( \sum_{j=0}^{m} \mathbf{u}_{i+\frac{j}{m}} P_{ij}(c_{i,l}), \sum_{j=0}^{m} \mathbf{u}_{k+\frac{j}{m}} P_{kj}(\tilde{c}_{i,l}) \right), \quad \forall i, l. \tag{3.48}$$

This, for all intents and purposes, is merely a system of nonlinear algebraic equations, with unknowns $\mathbf{u}_{i+\frac{j}{m}} := \mathbf{u}(t_{i+\frac{j}{m}})$ and $T$, and we can now apply Newton's method to solve for these unknowns. We note that this problem provides $n \times (m \times L + 1)$ equations for $n \times (m \times L + 1) + 1$ unknowns, and the phase constraint provides the last equation.

### The phase constraint

A phase condition is required in order to remove translational invariancy and fix a starting point in the period, for otherwise an unconstrained set of phase shifted periodic solutions would all satisfy our problem. Typically the classical integral phase condition is applied,

$$\int_0^1 \dot{\mathbf{u}}^{(0)}(s) \Big( \mathbf{u}^{(0)}(s) - \mathbf{u}^{(v)}(s) \Big) ds = \mathbf{0}, \tag{3.49}$$

where $\mathbf{u}^{(0)}(s)$ is the initial solution at the start of the Newton iteration, and $\mathbf{u}^{(v)}(s)$ is the current estimate. Notice that this phase constraint respects the phase of the initial estimate.

### Constructing the Newton solver

To solve (3.48) for the periodic solution, we reformulate the equation for each collocation point $c_{i,l}$ as:

$$\mathbf{H}_{i,l}(\mathbf{u}, T) := \sum_{j=0}^{m} \mathbf{u}_{i+\frac{j}{m}} P'_{ij}(c_{i,l}) - T f\left( \sum_{j=0}^{m} \mathbf{u}_{i+\frac{j}{m}} P_{ij}(c_{i,l}), \sum_{j=0}^{m} \mathbf{u}_{k+\frac{j}{m}} P_{kj}(\tilde{c}_{i,l}) \right) = \mathbf{0}. \tag{3.50}$$

Note that $\dim(\mathbf{H}_{i,l}) = \dim(\mathbf{u}) = n$, the size of the system of DDEs. Performing a Taylor expansion for each $(i,l)$ around an estimate $\mathbf{u}^{(v)}$, we construct our Newton solver by identifying the linear error estimates $(\Delta\mathbf{u}^{(v)}, \Delta T^{(v)})$, which can then be solved for in a linear setting, and an update scheme applied:

$$
\begin{aligned}
\mathbf{0} &= \mathbf{H}_{i,l}(\mathbf{u},T) && (3.51)\\
\mathbf{0} &\approx \mathbf{H}_{i,l}(\mathbf{u}^{(\nu)},T^{(\nu)})+ && (3.52)\\
&\quad \Big[D_{\mathbf{u}^{(\nu)}}\mathbf{H}_{i,l}(\mathbf{u}^{(\nu)},T^{(\nu)})\Delta\mathbf{u}^{(\nu)}\Big]+\Big[D_{T^{(\nu)}}\mathbf{H}_{i,l}(\mathbf{u}^{(\nu)},T^{(\nu)})\Delta T^{(\nu)}\Big]\\
-\mathbf{H}_{i,l}(\mathbf{u}^{(\nu)},T^{(\nu)}) &= \Bigg[\sum_{j=0}^{m}P'_{ij}(c_{i,l})\Delta\mathbf{u}^{(\nu)}_{i+\frac{j}{m}}+ && (3.53)\\
&\quad T^{(\nu)}A_0^{(\nu)}\sum_{j=0}^{m}P_{ij}(c_{i,l})\Delta\mathbf{u}^{(\nu)}_{i+\frac{j}{m}}+T^{(\nu)}A_1^{(\nu)}\sum_{j=0}^{m}P'_{kj}(\tilde{c}_{i,l})\Delta\mathbf{u}^{(\nu)}_{k+\frac{j}{m}}\Bigg]+\\
&\quad \Bigg[f\Big(\sum_{j=0}^{m}\mathbf{u}^{(\nu)}_{i+\frac{j}{m}}P_{i,j}(c_{i,l}),\sum_{j=0}^{m}\mathbf{u}^{(\nu)}_{k+\frac{j}{m}}P_{k,j}(\tilde{c}_{i,l})\Big)\Delta T^{(\nu)}-\\
&\quad T^{(\nu)}A_1^{(\nu)}\Big(\sum_{j=0}^{m}\mathbf{u}^{(\nu)}_{k+\frac{j}{m}}P'_{k,j}(\tilde{c}_{i,l})\Big)\frac{\tau}{(T^{(\nu)})^2}\Delta T^{(\nu)}\Bigg]
\end{aligned}
$$

This is then a system of linear equations over the error estimates $(\Delta\mathbf{u},\Delta T)$, with $A_0^{(\nu)}$ and $A_1^{(\nu)}$ being the $\mu$-iteration Jacobians to $f$, evaluated at the collocation point $c_{i,l}$:

$$A_0^{(\nu)} := \frac{\partial f}{\partial \xi}(\xi,\eta) \qquad A_1^{(\nu)} := \frac{\partial f}{\partial \eta}(\xi,\eta) \tag{3.54}$$

Now the typical Newton method for systems of equations can be applied. To illustrate, we present the structure of a fictitious system of equations with $m=2$ below:

$$
\left[\begin{array}{c|c} A^{(\nu)} & \mathbf{b}^{(\nu)} \\ \hline (\mathbf{c}^{(\nu)})^T & d^{(\nu)} \end{array}\right]
\underbrace{\left[\begin{array}{c} \Delta\mathbf{u}_0 \\ \Delta\mathbf{u}_{0+\frac{1}{2}} \\ \Delta\mathbf{u}_1 \\ \Delta\mathbf{u}_{1+\frac{1}{2}} \\ \Delta\mathbf{u}_2 \\ \vdots \\ \hline \Delta T \end{array}\right]^{(\nu)}}_{\delta^{(\nu)}}
= -\left[\begin{array}{c} \mathbf{H}_{0,0}(u^{(\nu)},T^{(\nu)}) \\ \mathbf{H}_{0,1}(u^{(\nu)},T^{(\nu)}) \\ \mathbf{H}_{1,0}(u^{(\nu)},T^{(\nu)}) \\ \mathbf{H}_{1,1}(u^{(\nu)},T^{(\nu)}) \\ \mathbf{H}_{2,0}(u^{(\nu)},T^{(\nu)}) \\ \vdots \\ \hline 0 \end{array}\right], \tag{3.55}
$$

where $A$ is a sparse matrix with band-like structures (see [Engelborghs *et al.*, 2000]), $\mathbf{b}$ is a vector incorporating the $T$-dependance, and $\mathbf{c}$ and $d$ are derived by linearizing the phase constraint (3.49), a process that is not shown here.

This system can then be solved for $\delta^{(\nu)}$ (via e.g. LU decomposition to avoid matrix inversion), and the solution can be updated until convergence as:

$$
\begin{aligned}
\mathbf{u}^{(\nu+1)} &= \mathbf{u}^{(\nu)}+\Delta\mathbf{u}^{(\nu)} && (3.56)\\
T^{(\nu+1)} &= T^{(\nu)}+\Delta T^{(\nu)}. && (3.57)
\end{aligned}
$$

**Obtaining an initial estimate**

We now wish to apply this method at many different delay parameter values, to find the limit cycle of the system at those parameter values, with a stability analysis as our goal. But in order to run our Newton iterator, a descent initial estimate is required for convergence. This is an infamous property of the Newton method, and it is overcome by obtaining an initial estimate of the limit cycle from a numerical simulation. By applying a numerical solver (such as `dde23` or `ddesd` in matlab) roughly until convergence ("for a long while"), and extracting one period of the solution translated to a proper mesh grid $\Pi$ and set of representation points $t_{i+\frac{j}{m}}$ (see (3.38) and (3.39)), we are able to supply a sufficiently close initial estimate, that in practice allows our Newton solver to converge. If convergence is not achieved, stricter requirements can be made for the DDE numerical solver.

Once we have obtained a periodic solution at a nominal set of delay parameters, we have successfully entered the 'solution space', and this nominal solution can be used as an initial estimate to re-run the Newton solver at a nearby set of parameters, and here convergence is typically excellent. In this fashion, one can traverse the parameter space to obtain periodic solutions to nearby parameter values rapidly, without having to re-run a DDE numerical solver.

### The time integration operator

All of the work so far has been for the purpose of *finding* the limit cycle. With the periodic solution in hand, $\mathbf{u}^*(t)$ with period $T^*$, our attention turns to studying the stability properties of the time integration operator applied to the system over a period $T$, also called the *monodromy operator*.

The *monodromy matrix* $M$ approximating this operator is the matrix corresponding to a numerical integration of (3.32) spanning $[-\frac{\tau}{T}, 1]$, and extracting the mapping of the elements in $[-\frac{\tau}{T}, 0]$ to $[1-\frac{\tau}{T}, 1]$. This matrix can be derived largely from the linear system in (3.55), where the $A$-matrix is essentially this discretized, linearized operator. An exact mathematical formulation of the monodromy matrix is weighed down in notation, and therefore a detailed formulation is omitted. For an implementation of the methodology, we rely on DDE BIFTOOL.

The eigenvalues of the monodromy matrix approximate the eigenvalues of the operator, providing a measure of the stability of the periodic solution. Thus, we can derive the dominant non-trivial multiplier $\mu_2$ (recall that $\mu_1 = 1$), and study its dependance on the delay parameters of the delay-differential equation.

### Sensitivity analysis

In addition to studying the delay-dependance of the dominant non-trivial multiplier, to succinctly study the delay-dependance, we will also examine partial derivative sensitivities. By calculating the sensitivity of the dominant non-trivial multiplier with respect to delay $\tau_i$ as a central difference after a logarithmic transformation,

$$S_i := -\frac{d}{d\tau_i}\Big(\ln|\mu_2(\tau)|\Big) \tag{3.58}$$

$$S_i \approx -\left(\frac{\ln|\mu_2(\tau+\varepsilon_i)| - \ln|\mu_2(\tau-\varepsilon_i)|}{2\varepsilon_i}\right) \tag{3.59}$$

$$S_i \approx -\frac{1}{2\varepsilon_i}\ln\left(\frac{|\mu_2(\tau+\varepsilon_i)|}{|\mu_2(\tau-\varepsilon_i)|}\right), \tag{3.60}$$

we obtain a mapping where a positive sensitivity corresponds to an increase in stability, and a negative sensitivity corresponds to a decrease. Here $\tau$ is the vector of delays, and $\varepsilon_i$ is a perturbation to delay $i$.

## 3.5 Analysis: Repressilator

Having now outlined the mathematical methodology, we can turn our attention to the bioengineering inquiry that motivates our efforts. How might delays improve the stability of genetic regulatory oscillators? In terms of our newly derived methodology, how do the characteristic multipliers of the limit cycle depend upon delay?

We once again begin our oscillator analysis with the Elowitz repressilator, which was introduced in the previous chapter. The repressilator is a ring oscillator consisting of three proteins repressing each other, and is commonly formulated in a six-dimensional ODE, with proteins and mRNA:

$$\begin{aligned}
\frac{dm_A}{dt} &= \alpha_0 + \frac{\alpha}{1+p_C(t)^2} - m_A \\
\frac{dm_B}{dt} &= \alpha_0 + \frac{\alpha}{1+p_A(t)^2} - m_B \\
\frac{dm_C}{dt} &= \alpha_0 + \frac{\alpha}{1+p_B(t)^2} - m_C \\
\frac{dp_A}{dt} &= -\beta(p_A(t) - m_A(t)) \\
\frac{dp_B}{dt} &= -\beta(p_B(t) - m_B(t)) \\
\frac{dp_C}{dt} &= -\beta(p_C(t) - m_C(t)).
\end{aligned} \tag{3.61}$$

The reduced ODE formulation of the repressilator is a case-in-point of why non-delayed models do not accurately describe the system dynamics. The reduced ODE model (where $dm_i/dt \approx 0$, and parameters $\alpha, \alpha_0, \beta$ are appropriately converted),

$$\begin{aligned}
\frac{dp_A}{dt} &= -\beta p_A(t) + \alpha_0 + \frac{\alpha}{1+p_C(t)^2} \\
\frac{dp_B}{dt} &= -\beta p_B(t) + \alpha_0 + \frac{\alpha}{1+p_A(t)^2} \\
\frac{dp_C}{dt} &= -\beta p_C(t) + \alpha_0 + \frac{\alpha}{1+p_B(t)^2},
\end{aligned} \tag{3.62}$$

in fact leads to a stable equilibrium fixed point, and no oscillations at all. For parameter values, see the ODE model in [Elowitz and Leibler, 2000]. The more detailed six-dimensional model is used by the authors to maintain a delay in the interactions between proteins, via the intermediate mRNAs. This further motivates the delay differential formulation, which is diagramed below in Figure 3.2:

$$\begin{aligned}
\frac{dp_A}{dt} &= -\beta p_A(t) + \alpha_0 + \frac{\alpha}{1+p_C(t-\tau_A)^2} \\
\frac{dp_B}{dt} &= -\beta p_B(t) + \alpha_0 + \frac{\alpha}{1+p_A(t-\tau_B)^2} \\
\frac{dp_C}{dt} &= -\beta p_C(t) + \alpha_0 + \frac{\alpha}{1+p_B(t-\tau_C)^2}.
\end{aligned} \tag{3.63}$$

While the non-delayed system is dominated by a stable fixed point, when $\tau_A, \tau_B, \tau_C \approx$ 21 seconds, a Hopf bifurcation occurs, and for delays longer than 21 seconds we obtain a stable limit cycle surrounding an unstable fixed point. This alone is a noteworthy realization, seeing as the classic repressilator operates only marginally beyond the bifurcation point, at about 27 seconds of delay.

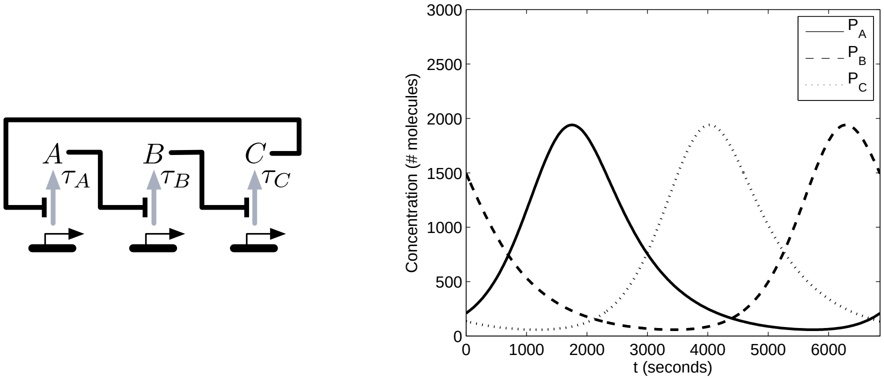

**Figure 3.2** A schematic of the Elowitz ring oscillator delay model, and the nominal limit cycle. Three promoter sites produce three different proteins, and are negatively regulated (repressed) in a 'ring'. Transcription and translation are modeled as one delay.

The extended delays we choose to work with are $\tau_1 = \tau_2 = \tau_3 = 127$ seconds. In contrast to the more detailed stochastic formulation, here the delay from promoter activity to protein is summarized in one term, and so we are modeling a 100 second transcriptional delay, ahead of a 5 second period to transcribe the ribosome binding site (RBS), and 22 seconds to translate a generic 1000 bp mRNA to protein, at 45 bp/s [Phillips *et al.*, 2008].

How does the stability of the limit cycle depend upon delay? The two principle characteristic Floquet multipliers are shown below, as a function of varying one delay parameter. Due to symmetry, all three delay parameters clearly have the same effect.

We can here confidently conclude that increasing delays on any (and more intelligently, all) of the transcriptional delays in the repressilator increases the stability of the oscillator, making it less susceptible to perturbations from the oscillator limit cycle. This is a significant theoretical result, seeing as the nominal characterization of the repressilator only offers a barely functioning oscillator when studied *in vivo*. This result also agrees with the delay-dependancy of the more detailed stochastic model of the repressilator regulatory network.

To offer a more concise investigation, we take the sensitivity measure (3.60) at the nominal parameterization, and see that our sensitivities are symmetrical across the pathways, $S = (0.00377, 0.00377, 0.00377)$. For large systems, these sensitivities evaluated at the nominal case can give a clear summary of where delay can improve the stability of the system, at a fraction of the computational cost required to search

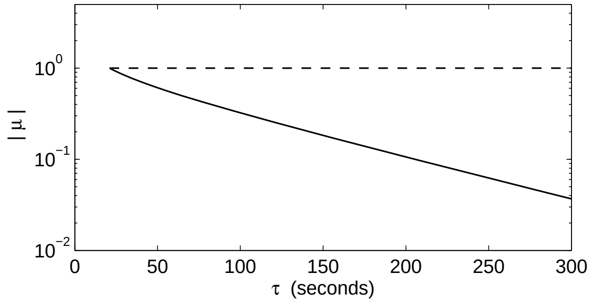

**Figure 3.3** The two principle characteristic Floquet multipliers as a function of varying one delay parameter. The first multiplier is constant at $\mu_1 = 1$ (dashed), while the second multiplier $\mu_2$ (solid) rapidly becomes smaller as we increase the delay (note the logarithmic scale).

the delay parameter space. There is no indication that the delay-dependence is necessarily monotonically increasing or decreasing (though no counter example has been found). While a sensitivity analysis by no means paints the complete picture, it can offer a concise overview for complex networks, evaluating the delay-dependance of each pathway at the nominal parameterization.

## 3.6 Analysis: Relaxation oscillator

The DDE model of the Barkai-Leibler relaxation oscillator will now be examined, where transcriptional regulation is used for activation, and a hypothetical protein-protein interaction between proteins A and B is used to form a decaying protein C, see [Barkai and Leibler, 2000] for details. The protein-protein interaction is an instantaneous reaction, and we choose to investigate delays only in the context of transcription. A diagram of the circuit as well as the nominal limit cycle are shown in Figure 3.4, and the delay-differential formulation is given as:

$$\begin{aligned} \frac{dp_A}{dt} &= -\beta p_A(t) + \alpha_0 + \alpha \frac{p_A(t-\tau_A)}{1+p_A(t-\tau_A)^2} - \gamma p_A(t) p_B(t) \qquad (3.64) \\ \frac{dp_B}{dt} &= -\beta p_B(t) + \alpha_0 + \alpha \frac{p_A(t-\tau_B)}{1+p_A(t-\tau_B)^2} - \gamma p_A(t) p_B(t) + \beta p_C(t) \\ \frac{dp_C}{dt} &= -\beta p_C(t) + \gamma p_A(t) p_B(t). \end{aligned}$$

For parameter values, see [Vilar *et al.*, 2002].

Recalling the delay-dependent stability of the stochastic relaxation oscillator in the previous chapter, again we find that increasing the delay for the primary self-activation pathway would decrease the stability, while increasing the delay on the other pathway would in fact increase the stability, just as with the stochastic model, but now we can compute the characteristic multipliers as a function of delay, plotted below in Figure 3.5, and obtain a much more quantitative understanding. We can again summarize the stability dependancy at the nominal parameters with the sensitivity measure $(S_A, S_B) = (-0.0694, 0.0525)$. The sensitivity measure, which is

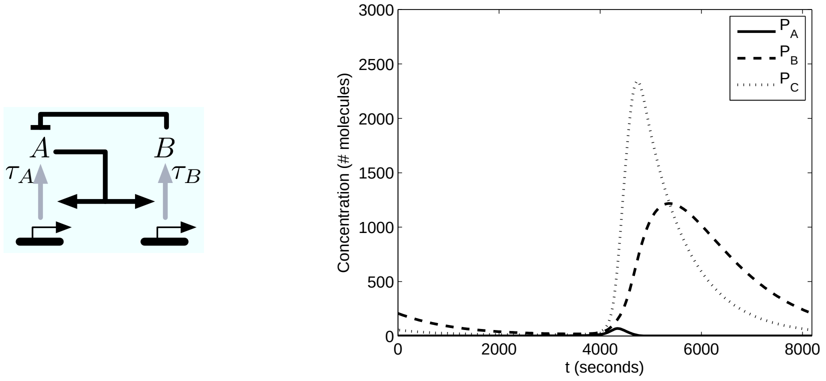

**Figure 3.4** A schematic of the Barkai-Leibler relaxation oscillator delay model, and the nominal limit cycle. Note that the protein-protein interaction is direct, and plays no part in transcription or translation (which are again modeled as one delay). The protein C is now shown in the diagram, but serves as an intermediate species along the decay pathway when protein B represses A.

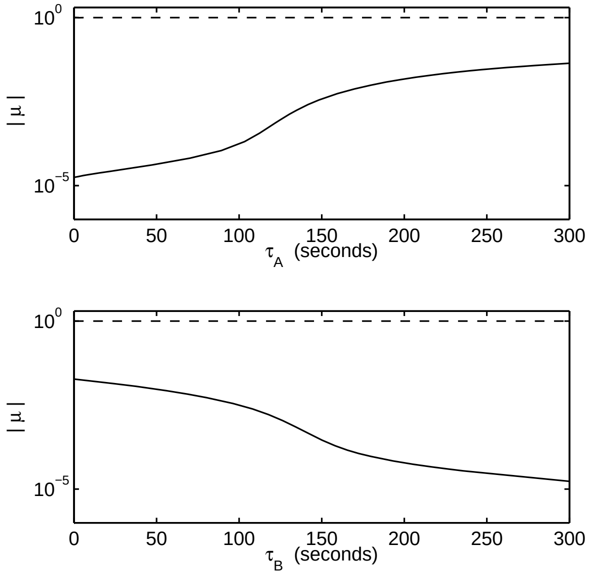

**Figure 3.5** The two principle characteristic Floquet multipliers as a function of varying the delay parameters for the relaxation oscillator with $\tau_A$ above, and $\tau_B$ below. The first multiplier is in both cases constant at $\mu_1 = 1$ (dashed), while the second multiplier $\mu_2$ (solid), which quantifies the stability of the limit cycle, is delay-dependent, and varies as we move either $\tau_A$ or $\tau_B$ from the nominal configuration (where $\tau_A = \tau_B = 127$ seconds).

significantly faster to calculate, summarizes the delay-dependance well, given that the dependancies are still monotonic (though opposite) even in this case. The usefulness of such local sensitivity measures, along with the question of monotonicity, are topics open for future study.

# 4. Conclusions

By examining the role of delay in both stochastic and deterministic models of genetic regulatory networks, we establish a theory for delay sensitivity and tuning that proposes a new paradigm for bioengineering. Our analysis shows that delays play a particularly important role in the dynamics of oscillatory networks, and delay engineering can be used to considerably improve the stability of limit cycles in oscillator models. By applying the methods of sensitivity analysis for DDE models developed here, large regulatory networks (natural or synthetic) from a wide range of biological inquires can potentially be examined for delay tuning. This thesis also outlines methods for stochastic model analysis that can be performed to validate the dependancy at a greater level of detail.

The strong agreement between the stochastic modeling approach, derived from the Chemical Master Equation, and the deterministic approach, formulated using delay-differential equations (DDEs), suggests that the delay effects observed in this thesis are valid to the extent that the CME and DDE modeling approaches are valid. But whether or not biology agrees is admittedly another matter.

It is important to emphasize that the results presented in this thesis are theoretical in nature, and that at this time these predictions have not been verified in biology. A ‘wet’ investigation is the obvious next step, and towards this goal, an initial investigation is currently being undertaken within the Murray Group at Caltech.

# 5. Acknowledgements

This thesis is the result of research completed at the Department of Control and Dynamical Systems (CDS) at the California Institute of Technology during the spring of 2008, under the supervision of Professor Richard Murray. I would like to thank Professor Murray for his guidance and for hosting me during this productive visit, as well as for inviting me on my initial visit to the CDS department as a Summer Undergraduate Research Fellow (SURF) during the summer of 2006. I would also like to thank Professor Anders Rantzer from Lund for introducing me to Professor Murray, and for guidance during my studies in Lund. Visiting Caltech has been a truly wonderful experience, both academically and personally.

Furthermore, I would like to extend a special thanks to Mary Dunlop, my SURF advisor during the summer of 2006 and ACC co-author, for all her assistance then and now, and the Caltech Bio-Control reading group, for stimulating discussions and helping to fill the holes in my biology education. I'd also like to thank Francisco Zabala and Dionysios Barmpoutis, my officemates at Caltech, for their company, assistance, and support during a turbulent spring in Pasadena. Their friendship played no small role in making this thesis possible. But most of all, I'd like to thank my family: my brothers, Martin and Olof, for being incredible brothers, and my parents Mikael and Margareta, for everything they've done for me, and for always being there.

# 6. Bibliography

Alberts, B., D. Bray, A. Johnson, J. Lewis, M. Raff, and K. Roberts (2003): *Essential cell biology*. Garland Science, New York.

Alon, U. (2007): *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC.

Barkai, N. and S. Leibler (2000): “Circadian clocks limited by noise.” *Nature*, **403:6767**, pp. 267–8.

Barrio, M., K. Burrage, A. Leier, T. Tian, and P. Hunter (2006): “Oscillatory Regulation of Hes1: Discrete Stochastic Delay Modelling and Simulation.” *PLoS Comput Biol*, **2:9**, p. e117.

Becskei, A. and L. Serrano (2000): “Engineering stability in gene networks by autoregulation.” *Nature*, **405:6786**, pp. 590–3.

Bellen, A. and S. Maset (2000): “Numerical solution of constant coefficient linear delay differential equations as abstract Cauchy problems.” *Numerische Mathematik*, **84:3**, pp. 351–374.

Bellen, A. and M. Zennaro (2003): *Numerical Methods for Delay Differential Equations*. Oxford University Press, USA.

Bratsun, D., D. Volfson, L. Tsimring, and J. Hasty (2005): “Delay-induced stochastic oscillations in gene regulation.” *Proceedings of the National Academy of Sciences*, **102:41**, pp. 14593–14598.

Breda, D., S. Maset, and R. Vermiglio (2005): “Pseudospectral differencing methods for characteristic roots of delay differential equations.” *SIAM Journal on Scientific Computing*, **27:2**, pp. 482–495.

Cai, X. (2007): “Exact stochastic simulation of coupled chemical reactions with delays.” *The Journal of Chemical Physics*, **126**, p. 124108.

Cao, Y., H. Li, and L. Petzold (2004): “Efficient formulation of the stochastic simulation algorithm for chemically reacting systems.” *The Journal of Chemical Physics*, **121**, p. 4059.

Cao, Y. and L. Petzold (2006): “Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems.” *Journal of Computational Physics*, **212:1**, pp. 6–24.

Carlson, J. and J. Doyle (1999): “Highly optimized tolerance: A mechanism for power laws in designed systems.” *Physical Review E*, **60:2**, pp. 1412–1427.

Carlson, J. and J. Doyle (2002): “Complexity and robustness.” *Proceedings of the National Academy of Sciences*, **99:90001**, p. 2538.

Chen, L. and K. Aihara (2002a): “Stability of genetic regulatory networks with time delay.” *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on]*, **49:5**, pp. 602–608.

Chen, L. and K. Aihara (2002b): “Stability of genetic regulatory networks with time delay.” *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on]*, **49:5**, pp. 602–608.

Corless, R., G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth (1996): "On the Lambert W function." *Advances in Computational Mathematics*, **5:1**, pp. 329–359.

Elowitz, M. and S. Leibler (2000): "A synthetic oscillatory network of transcriptional regulators." *Nature*, **403:6767**, pp. 335–8.

Engelborghs, K., T. Luzyanina, K. In't Hout, and D. Roose (2000): "Collocation Methods for the Computation of Periodic Solutions of Delay Differential Equations." *SIAM Journal on Scientific Computing*, **22:5**, pp. 1593–1609.

Engelborghs, K., T. Luzyanina, and D. Roose (2002): "Numerical Bifurcation Analysis of Delay Differential Equations Using DDE-BIFTOOL." *ACM Transactions on Mathematical Software*, **28:1**, pp. 1–21.

Epshtein, V. and E. Nudler (2003): "Cooperation Between RNA Polymerase Molecules in Transcription Elongation."

Euler, L. (1779): "De serie Lambertina plurimisque eius insignibus proprietatibus." *Acta Acad. Sci. Petro*, pp. 29–51.

Gardner, T., C. Cantor, and J. Collins (2000): "Construction of a genetic toggle switch in Escherichia coli." *Nature*, **403:6767**, pp. 339–42.

Gibson, M. and J. Bruck (2000): "Efficient exact stochastic simulation of chemical systems with many species and many channels." *J. Phys. Chem. A*, **104:9**, pp. 1876–1889.

Gillespie, D. (1976): "A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions." *Journal of Computational Physics*, **22**, p. 403.

Gillespie, D. (1977): "Exact stochastic simulation of coupled chemical reactions." *The Journal of Physical Chemistry*, **81:25**, pp. 2340–2361.

Glass, L. (1975): "Classification of biological networks by their qualitative dynamics." *J Theor Biol*, **54:1**, pp. 85–107.

Greive, S. and P. von Hippel (2005): "Thinking quantitatively about transcriptional regulation." *Nature Reviews Molecular Cell Biology*, **6:3**, pp. 221–232.

Gunawan, R., Y. Cao, L. Petzold, and F. Doyle (2005): "Sensitivity Analysis of Discrete Stochastic Systems." *Biophysical Journal*, **88:4**, pp. 2530–2540.

Hale, J. and S. Lunel (1993): *Introduction to Functional Differential Equations*. Springer.

Haseltine, E. and J. Rawlings (2002): "Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics." *The Journal of Chemical Physics*, **117**, p. 6959.

Hasty, J., D. McMillen, and J. Collins (2002): "Engineered gene circuits." *Nature*, **420:6912**, pp. 224–230.

Herbert, K., A. La Porta, B. Wong, R. Mooney, K. Neuman, R. Landick, and S. Block (2006): "Sequence-Resolved Detection of Pausing by Single RNA Polymerase Molecules." *Cell*, **125:6**, pp. 1083–1094.

Hildebrand, F. (1987): *Introduction to Numerical Analysis*. Courier Dover Publications.

Jacob, F. and J. Monod (1961): “Genetic regulatory mechanisms in the synthesis of proteins.” *J Mol Biol*, **3**, pp. 318–56.

Jarlebring, E. (2008): *The spectrum of delay-differential equations: numerical methods, stability and perturbation.* PhD thesis, Inst. Comp. Math., TU Braunschweig.

Jarlebring, E. and T. Damm (2007): “The Lambert W function and the spectrum of some multidimensional time-delay systems.” *Automatica*, **43:12**, pp. 2124–2128.

Kaern, M., T. Elston, W. Blake, and J. Collins (2005): “Stochasticity in gene expression: from theories to phenotypes.” *Nat Rev Genet*, **6:6**, pp. 451–464.

Kashtan, N. and U. Alon (2005): “Spontaneous evolution of modularity and network motifs.” *Proceedings of the National Academy of Sciences*, **102:39**, pp. 13773–13778.

Kosuri, S. (2007): *Simulation, Models, and Refactoring of Bacteriophage T7 Gene Expression.* PhD thesis, Massachusetts Institute of Technology.

Kosuri, S., J. Kelly, and D. Endy (2007): “TABASCO: A single molecule, base-pair resolved gene expression simulator.” *BMC Bioinformatics*, **8**, p. 480.

Lambert, J. (1758): “Observationes variae in mathesin puram.” *Acta Helvetica*, **3**, pp. 128–168.

Li, H. and L. Petzold (2006): “Logarithmic Direct Method for Discrete Stochastic Simulation of Chemically Reacting Systems.” Technical Report. Department of Computer Science, University of California, Santa Barbara.

Mangan, S. and U. Alon (2003): “Structure and function of the feed-forward loop network motif.” *Proceedings of the National Academy of Sciences*, **100:21**, p. 11980.

McAdams, H. and A. Arkin (1997): “Stochastic mechanisms in gene expression.” *Proceedings of the National Academy of Sciences*, **94:3**, p. 814.

Milo, R., S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon (2002): “Network Motifs: Simple Building Blocks of Complex Networks.”

Munsky, B. and M. Khammash (2006): “The finite state projection algorithm for the solution of the chemical master equation.” *The Journal of Chemical Physics*, **124**, p. 044104.

Parker, T. and L. Chua (1989): *Practical numerical algorithms for chaotic systems.* Springer-Verlag New York, Inc. New York, NY, USA.

Phillips, R., J. Kondev, and J. Theriot (2008): *Physical Biology of the Cell.* Garland Science.

Plyasunov, S. and A. Arkin (2007): “Efficient stochastic sensitivity analysis of discrete event systems.” *Journal of Computational Physics*, **221:2**, pp. 724–738.

Raser, J. and E. O’Shea (2005): “Noise in Gene Expression: Origins, Consequences, and Control.” *Science*, **309:5743**, pp. 2010–2013.

Sjöberg, P., P. Lötstedt, and J. Elf (2008): “Fokker–Planck approximation of the master equation in molecular biology.” (to appear).

Stokes, A. (1977): “Local coordinates around a limit cycle of a functional differential equation with applications.” *J. Differential Equations*, **24**, pp. 153–172.

Tian, T. and K. Burrage (2004): “Binomial leap methods for simulating stochastic chemical kinetics.” *The Journal of Chemical Physics*, **121**, p. 10356.

Tolic-Norrelykke, S., A. Engh, R. Landick, and J. Gelles (2004): “Diversity in the Rates of Transcript Elongation by Single RNA Polymerase Molecules.” *Journal of Biological Chemistry*, **279:5**, p. 3292.

Trefethen, L. (2000): *Spectral Methods in MATLAB*. Society for Industrial Mathematics (SIAM).

Ugander, J., M. Dunlop, and R. Murray (2007): “Analysis of a Digital Clock for Molecular Computing.” *Proceedings of the 26th American Control Conference (ACC)*, pp. 1595–1599.

Van Kampen, N. (2007): *Stochastic Processes in Physics and Chemistry*. North Holland.

Vilar, J., H. Kueh, N. Barkai, and S. Leibler (2002): “Mechanisms of noise-resistance in genetic oscillators.” *Proceedings of the National Academy of Sciences*, **99:9**, p. 5988.

Wright, E. (1948): “The linear difference-differential equation with constant coefficients.” *Proc. R. Soc. Edinb., Sect. A*, **62**, pp. 387–393.