

Linear Control Theory for Gene Network Modeling

Yong-Jun Shin^{1*}, Leonidas Bleris^{1,2*}

1 Department of Electrical Engineering, University of Texas at Dallas, Richardson, Texas, United States of America, **2** Department of Bioengineering, University of Texas at Dallas, Richardson, Texas, United States of America

Abstract

Systems biology is an interdisciplinary field that aims at understanding complex interactions in cells. Here we demonstrate that linear control theory can provide valuable insight and practical tools for the characterization of complex biological networks. We provide the foundation for such analyses through the study of several case studies including cascade and parallel forms, feedback and feedforward loops. We reproduce experimental results and provide rational analysis of the observed behavior. We demonstrate that methods such as the transfer function (frequency domain) and linear state-space (time domain) can be used to predict reliably the properties and transient behavior of complex network topologies and point to specific design strategies for synthetic networks.

Citation: Shin Y-J, Bleris L (2010) Linear Control Theory for Gene Network Modeling. PLoS ONE 5(9): e12785. doi:10.1371/journal.pone.0012785

Editor: Diego Di Bernardo, Fondazione Telethon, Italy

Received: March 18, 2010; **Accepted:** July 21, 2010; **Published:** September 16, 2010

Copyright: © 2010 Shin, Bleris. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: These authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: yshin@cornell.edu (Y-JS); bleris@utdallas.edu (LB)

‡ Current address: School of Electrical and Computer Engineering, Cornell University, Ithaca, New York, United States of America

Introduction

Cells comprise of multiple, heterogeneous subunits that operate in a well-orchestrated manner [1,2]. Although extremely complex, phenotypes such as cell division and environmental adaptation are the outcome of discrete changes that lead to a deterministic sequence of information transfer and processing within the cell. This information is encoded and transferred via multiple pathways, in different time-scales, and is typically processed in parallel by multi-component networks. Such networks comprise of genes, gene products and small molecules that mutually affect each other or interconvert through biochemical reactions. The number of possible network topologies for a given set of elements is large and it grows exponentially with the number of elements.

Systems biology is an interdisciplinary field that aims at understanding such complex interactions in cells, via the use of a wide spectrum of theoretical and experimental techniques [3]. One of the main thrusts of systems biology is the study of gene networks [4–7], via top-down and bottom-up approaches [8]. A top-down approach aims at unraveling the complexity of network dynamics without or with little prior knowledge of the network components and the relationships between them [9–12]. On the other hand, a bottom-up approach (closely related to synthetic biology) aims at constructing and studying small-scale biological networks from modular components [13–17]. Gene networks are inherently stochastic [18,19], which renders any modeling effort nontrivial. Furthermore, as the size of a network increases, it becomes increasingly difficult to predict its dynamic behavior. Another characteristic feature is the existence of nonlinearities in biological networks, which further complicates any modeling effort.

Towards analyzing the dynamic behavior of gene networks, a range of mathematical and computational modeling methods have been developed, including Boolean networks, Petri nets, state-charts, ordinary differential equations, and stochastic simulation

algorithms [4,20–30]. These approaches can be further organized into two larger categories: logical and continuous models. Logical models deal with the logical sequence of events while continuous models describe the dynamics that depend on molecular concentrations and time.

In this paper, we develop and use linear models [31–34], which are shown to capture the dynamics of gene networks in an intuitive and efficient way. We argue that tools of linear control theory, including transfer function (frequency domain) and linear state-space (time domain) methods can be exceptionally practical for systems and synthetic biologists towards unraveling the properties of gene networks and engineering novel systems. We provide several examples of the application of the transfer function method for the analysis of gene networks, starting with network motifs [5], the basic building blocks of gene networks. The transfer function method is sometimes regarded as “classical” in control theory. The state-space or “modern” approach describes a system as a set of input, output, and state variables related by first-order differential or difference equations. One of the advantages of using the state-space method is that it can be used to model multiple-input multiple-output (MIMO) gene networks in a compact manner, utilizing vectors and matrices. Furthermore, we show that the linear state-space approach enables us to utilize a spectrum of tools available for optimal/robust estimation/control for gene network modeling [35–37]. As an example, we illustrate that the Kalman filter, one of the well-established optimal estimation tools, can be applied for stochastic modeling of a simple two-gene network. Finally, using a six-node gene network, we demonstrate that our linear approaches can reduce the modeling complexity and provide rapid insight about its dynamic behavior, as compared to conventional non-linear modeling approaches.

The manuscript is organized as follows. We commence our analysis with modeling a simple gene regulation case and subsequently provide the details behind the proposed linearization scheme. Using these results we present the transfer function

method for gene network modeling and exemplify the methodology for collapsing cascade and parallel forms to transfer functions. We then provide six case studies that cover a wide range of systems and synthetic biology problems including: cascaded simple regulations, synthetic gene oscillators, effect of the basal production rate, four cascaded simple regulation loops, and finally interconnected feedforward loops. Subsequently, we present the linear state-space method for gene network modeling and we outline its use for the analysis of cascaded simple regulations. Using the linear state-space formulation, we conclude this manuscript with two additional case studies: the use of optimal estimation in gene network measurements and the analysis of the six-node gene network.

A. Modeling simple gene regulation

A two-gene network (or simple regulation) can be regarded as a fundamental unit that serves as a basic building block for constructing elaborate networks. In simple regulation, one gene (Y_{gene}) can be activated by another gene (X_{gene}), as indicated by the notation $X \rightarrow Y$ in **Figure 1**. This notation, however, involves in reality multiple steps. First, X_{gene} is transcribed into a messenger RNA, X_{mRNA} , which is then translated into a protein ($X_{protein}$). In the presence of a signal S_x , $X_{protein}$ shifts to its active form $X^*_{protein}$ (also called a transcription factor of Y_{gene}) and binds the promoter of Y_{gene} , transcribing Y_{gene} into Y_{mRNA} . Finally, as Y_{mRNA} is translated, $Y_{protein}$ is produced. The signal S_x acts like a switch, by determining the amount of active form $X_{protein}$ or $X^*_{protein}$. The production rate of $Y_{protein}$ can be expressed as a function of time $F(t)$ (units of concentration per unit time). The production is balanced by processes that decrease $Y_{protein}$, namely degradation (protein destruction) and dilution (concentration reduction due to the increase of cell volume) [13]. Degradation and dilution can be collectively denoted as a time-dependent function $d(t)$. The change in the concentration of $Y_{protein}$ depends on both $F(t)$ and $d(t)$, and its dynamics can be described as an ordinary differential equation:

$$\frac{dy(t)}{dt} = F(t) - d(t)y(t) \tag{1}$$

where $y(t)$ stands for the concentration of $Y_{protein}$. If we assume that $d(t)$ depends only on dilution (for example, in growing bacteria, many proteins are stable and they are not actively degraded), $d(t)$ can be expressed as a constant D , based on following equation [13]:

$$d(t) = D = \frac{\ln 2}{T_{1/2}} \tag{2}$$

where $T_{1/2}$ is the response time, the time to reach one half of the steady-state $Y_{protein}$ concentration. The response times of bacteria are approximately 30 min to a few hours, while those of eukaryotic cells can be longer [13]. Equation 1 can now be restated by substituting $d(t)$ by the constant D :

$$\frac{dy(t)}{dt} = F(t) - Dy(t) \tag{3}$$

For the analysis of the function $F(t)$, we need to consider a number of additional factors. As stated earlier, $X_{protein}$ must be converted to $X^*_{protein}$ by the signal S_x in order to initiate the $Y_{protein}$ production. The concentration of $X^*_{protein}$ can be expressed as a function of S_x , which is an activating switch. This switch-like relationship can be described using the non-linear Hill function [13], and the relationship between $X^*_{protein}$ (we use $x^*(t)$) and S_x (we use $s_x(t)$) can be expressed as:

$$x^*(t) = \frac{x(t) \cdot s_x^l(t)}{K_{1A}^l + s_x^l(t)} \tag{4}$$

where $x(t)$ stands for the concentration of total $X_{protein}$ that includes both inactive and active forms. It is the maximal level of $X^*_{protein}$ or $x^*(t)$ (in units of concentration) that is reached when $s_x(t) \gg K_{1A}$. K_{1A} is the concentration of $s_x(t)$, at which half-maximal concentration of $x^*(t)$ is reached. The Hill coefficient l determines the steepness of the function. Note that K_{1A} and l are determined by a number of factors, such as enzymatic activity, pH, and temperature, and can be estimated from experimental data. When $s_x(t)$ acts as a repressor, the corresponding Hill function is :

$$x^*(t) = \frac{x(t)}{1 + \left[\frac{s_x(t)}{K_{1B}} \right]^m} \tag{5}$$

where $x(t)$ is the maximal level of the $x^*(t)$ production that is reached when $s_x(t) = 0$. In other words, $x(t)$ is reached ($x^*(t) = x(t)$) when there is no signal causing repression. K_{1B} is the concentration

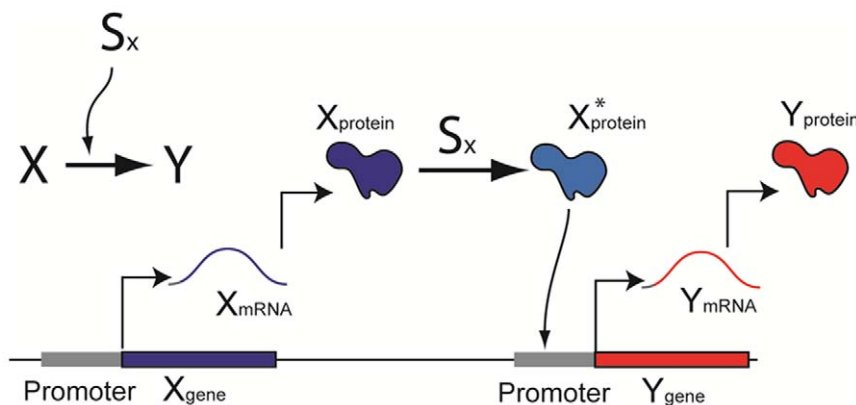


Figure 1. A schematic illustration of simple gene regulation.
doi:10.1371/journal.pone.0012785.g001

of $s_x(t)$, at which half-maximal repression of the $x^*(t)$ production is reached. Again, m is the Hill coefficient and determines the steepness of the function.

As a generalization, we can assume that $F(t)$ is the sum of a basal promoter production rate F_0 and the Hill function:

$$F(t) = F_0 + \frac{F_{max} \cdot [x^*(t)]^n}{K_{2A}^n + [x^*(t)]^n} \quad (6)$$

F_{max} is the maximal level of the $Y_{protein}$ production (in units of concentration per unit time) that is reached when $x^*(t) \gg K_{2A}$. K_{2A} is the concentration of $x^*(t)$ at which half-maximal production of $Y_{protein}$ is reached. n is the Hill coefficient, and K_{2A} and n are determined by a number of factors such as the translation rate, pH, and temperature. Similarly, when $x^*(t)$ acts as a repressor, the corresponding Hill function is:

$$F(t) = F_0 + \frac{F_{max}}{1 + \left[\frac{x^*(t)}{K_{2B}} \right]^o} \quad (7)$$

where F_{max} is the maximal level of the $Y_{protein}$ production that is reached when $x^*(t) = 0$. F_{max} is reached when there is no transcription factor causing repression. K_{2B} is the concentration of $x^*(t)$ at which half-maximal repression of the $Y_{protein}$ production is reached. o is the Hill coefficient.

B. Linearization

A non-linear model can be linearized around an equilibrium point using a Jacobian matrix, providing insight about the network behavior near that point [38]. For systems that involve Hill functions, there have been various linearization approaches that can be used to study the dynamics outside of the equilibrium [13,39–41]. These methods enable us to convert a non-linear model into a linear model and use linear tools for studying network dynamics over a wide range of parameter and variable values.

Two such approaches widely discussed in the literature [13,39–41] are the two-section piecewise linearization (step function or logic approximation) and the three-section piecewise linearization (Figure 2). In the figure, $F_x(t)$ represents the production rate of $Y_{protein}$ due to $X_{protein}$. In this section we introduce an alternative linearization approach and in later sections we will illustrate how the particular approach enables us to capture the dynamics of gene networks in an intuitive and efficient way.

Our method is a variation of a two-section piecewise linearization. In the step function approach (yellow in Figure 2), $F_x(t)$ is zero for all the values of $x^*(t)$ between 0 and a threshold point (TP) and takes its maximum rate when $x^*(t)$ is above TP. Therefore, the step function is inherently less accurate than the Hill Function and results to loss of information. When using the three-section piecewise linearization approach (green in Figure 2) one has to identify the two threshold values. As shown in Figure 2, using our approach we are required to identify one threshold. Furthermore, we may set our threshold higher than $2K_{2A}$, the $x^*(t)$ value required for reaching F_{max} , thus eliminating the need for explicitly incorporating the threshold into our model.

The first step towards applying linear control theory tools is to reduce (3) to a linear equation, in other words express $dy(t)/dt$ as a linear combination of $x(t)$ and $y(t)$. The concentration $x^*(t)$ of (4) can be expressed as the product of $x(t)$ and a function $F_I(t)$ that depends only on $s_x(t)$ as shown in (8). Note that l and K_{1A} are constant values determined experimentally, and $F_I(t)$ is a scaling factor that always lies between 0 and 1, depending on the value of $s_x(t)$. When $s_x(t)$ is zero, $F_I(t)$ is also zero, and if $s_x(t)$ is very large, then $F_I(t)$ approaches one.

$$x^*(t) = \frac{x(t) \cdot s_x^l(t)}{K_{1A}^l + s_x^l(t)} = x(t) \cdot F_I(t) \quad \left(F_I(t) = \frac{s_x^l(t)}{K_{1A}^l + s_x^l(t)} \right) \quad (8)$$

The Hill function relationship between $F_x(t)$ and $x^*(t)$ in (6) can be plotted as shown in Figure 2 (red line). As described earlier,

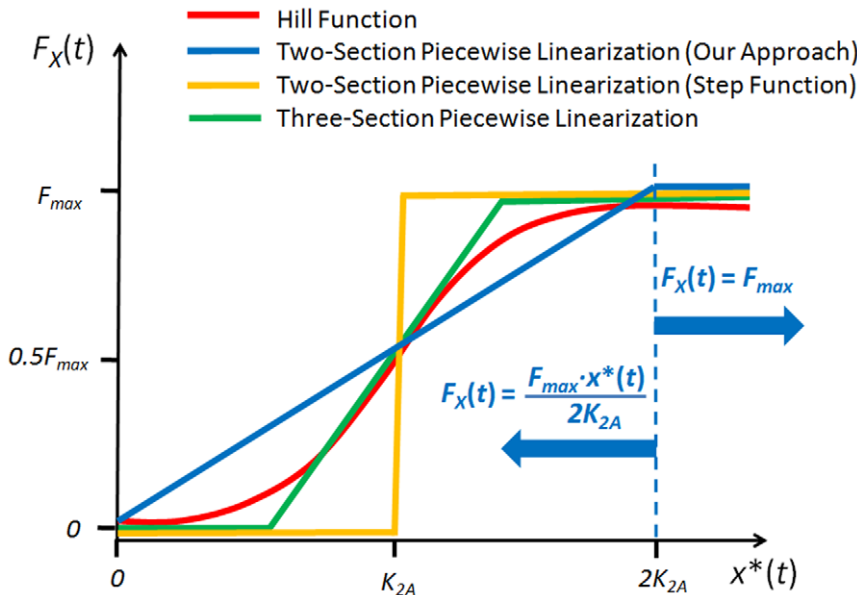


Figure 2. Linearization of the relationship between $F(t)$ and $x^*(t)$. The red line represents the Hill function. The linear approximation is represented by a blue line in two distinct regions separated by a dotted line. The logic approximation (step function) is shown in yellow and the three-section piecewise linearization is shown in green. doi:10.1371/journal.pone.0012785.g002

F_{max} is the maximal level of the $Y_{protein}$ production that is reached when $x^*(t) \gg K_{2A}$, and K_{2A} is the concentration of $x^*(t)$ at which half-maximal production of $Y_{protein}$ is reached.

For linearization, we will consider two cases: a case when $x^*(t)$ is smaller than $2K_{2A}$ and another when it is greater than $2K_{2A}$ (Figure 2). For the first case, we will assume that a line that passes through the origin (0,0) and $(2K_{2A}, F_{max})$ represents the linear approximation of the Hill function, and the slope of the line is:

$$slope = \frac{F_{max} - 0}{2K_{2A} - 0} = \frac{F_{max}}{2K_{2A}} \tag{9}$$

Now, denoting the slope as F_2 , (6) can be expressed as:

$$\begin{aligned} F(t) &= F_0 + \frac{F_{max} \cdot [x^*(t)]^n}{K_{2A}^n + [x^*(t)]^n} = F_0 + F_x(t) \\ &= F_0 + \frac{F_{max}}{2K_{2A}} \cdot x^*(t) = F_0 + F_2 \cdot x^*(t) \end{aligned} \tag{10}$$

Substituting $x^*(t)$ in (10) by $x(t) \cdot F_1(t)$ from (8), (10) becomes:

$$F(t) = F_0 + F_2 x^*(t) = F_0 + F_2 x(t) F_1(t) \tag{11}$$

Denoting $F_2 \cdot F_1(t)$ as $f(t)$, (11) becomes:

$$F(t) = F_0 + f(t)x(t) \tag{12}$$

Substituting $F(t)$ by $f(t)x(t)$, (1) can now be expressed as:

$$\frac{dy(t)}{dt} = F_0 + f(t)x(t) - d(t)y(t) \tag{13}$$

where $dy(t)/dt$ is shown as a linear combination of $x(t)$ and $y(t)$.

For the $x^*(t)$ values greater than $2K_{2A}$, $F(t)$ is equal to the constant F_{max} (Figure 2). Therefore, (1) can be expressed as:

$$\frac{dy(t)}{dt} = F_0 + F_{max} - d(t)y(t) \tag{14}$$

which again shows that $dy(t)/dt$ is a linear combination of F_{max} and $y(t)$. (13) and (14) can be shown together as:

$$\frac{dy(t)}{dt} = \begin{cases} F_0 + f(t)x(t) - d(t)y(t) & (0 \leq x^*(t) \leq 2K_{2A}) \\ F_0 + F_{max} - d(t)y(t) & (x^*(t) > 2K_{2A}) \end{cases} \tag{15}$$

Similarly, for the case when $s_x(t)$ acts as a repressor as shown in (5), $x^*(t)$ can be expressed as :

$$x^*(t) = \frac{x(t)}{1 + \left[\frac{s_x(t)}{K_{1B}}\right]^m} = x(t) \cdot F_1(t) \quad \left(F_1(t) = \frac{1}{1 + \left[\frac{s_x(t)}{K_{1B}}\right]^m} \right) \tag{16}$$

When $x^*(t)$ acts as a repressor, based on the linearization scheme shown in (9–12), the corresponding Hill function (7) can be expressed as:

$$F(t) = F_0 + \frac{F_{max}}{1 + \left[\frac{x^*(t)}{K_{2B}}\right]^o} = \begin{cases} F_0 - f(t)x(t) & (0 \leq x^*(t) \leq 2K_{2B}) \\ F_0 & (x^*(t) > 2K_{2B}) \end{cases} \tag{17}$$

From now on, we will consider only the non-trivial cases when $x^*(t)$ is between zero and $2K_{2A}$.

Results and Discussion

A. Transfer function method for gene network modeling

A transfer function can be derived from a linear, time-invariant differential equation using Laplace transform [42]. It is a mathematical representation of the relationship between the input and output in the frequency domain. Towards analyzing a gene network, the transfer function method can provide useful information about the behavior and stability of a system. Furthermore, using block diagrams, the transfer function method can represent large complex structures in a simple and intuitive way. Notably, the stochastic nature of gene networks has also been analyzed in the frequency domain [43–45].

1. Simple regulation. Starting from equation (8), if we assume that the concentration of the signal $s_x(t)$ is constant or time-invariant (for example, being at saturation), then we can assume $F_1(t)$ is also a constant value, F_1 . Now (3) can be rewritten as:

$$\frac{dy(t)}{dt} = F(t) - Dy(t) = F_0 + Fx(t) - Dy(t) \tag{18}$$

This time-invariance is required for the application of the transfer function method [42]. Using the Laplace transform, (18) can be expressed as:

$$sY(s) - y(0) = F(s) - Dy(s) = \frac{F_0}{s} + FX(s) - DY(s) \tag{19}$$

where $F(s)$, $X(s)$, and $Y(s)$ represent the Laplace transform of $F(t)$, $x(t)$, and $y(t)$. Once the Laplace transform is evaluated, computing in Laplace domain is algebraic and the complexity of solving differential equation is eliminated [42][41][41][42][43][48]. Assuming the initial concentration of $Y_{protein}$, $y(0)$, is zero, the transfer function $G(s)$, which relates the input $F(s)$ and output $Y(s)$ in the frequency domain, can be expressed as:

$$G(s) = \frac{Y(s)}{F(s)} = \frac{Y(s)}{\frac{F_0}{s} + FX(s)} = \frac{1}{s + D} \tag{20}$$

Note that the constant D plays a critical role in $G(s)$. It implies that the critical factor that characterizes this system is the dilution/ degradation constant. Also in this case the transfer function $G(s)$ is not affected by the input F_0 . However, as the network becomes more complicated (e.g., with positive and negative feedbacks), F_0/s is incorporated into the transfer function. The block diagram representation of the input, output, and transfer function of simple regulation is shown in Figure 3. Using the inverse Laplace transform, the output in the time domain, $y(t)$, can be calculated as following :

$$y(t) = \frac{F_0 + Fx(t)}{D} (1 - e^{-Dt}) \tag{21}$$

In case $x(t)$ is not changing (has reached its steady state) and set as a constant value X , $F \cdot x(t)$ becomes FX and the resulting step response is:

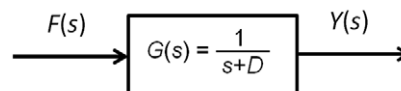


Figure 3. Block diagram representation of simple regulation.
doi:10.1371/journal.pone.0012785.g003

$$y(t) = \frac{F_0 + FX}{D} (1 - e^{-Dt}) \tag{22}$$

where $y(t)$ will approach the steady state value as time t goes to infinity, and the constant D in e^{-Dt} determines how fast the steady state is reached. The step response in Laplace domain can be expressed as:

$$Y(s) = \frac{F_0 + FX}{s(s + D)} \tag{23}$$

Solving a differential equation or evaluating the inverse Laplace transform enables us to evaluate the output response of a system. However, these techniques can be laborious and time-consuming. The use of poles and zeros, fundamental to the analysis and design of control systems, is a technique that can simplify the evaluation. Given the first-order transfer function $G(s)$ in (20), a pole, the value for s which makes the denominator of the transfer function equal to zero, exists at $s = -D$. Since D (the degradation/dilution factor) is greater than zero in a real biological system, $-D$ can be assumed to be a negative number. This indicates that the system is stable because stable systems do not have any positive poles [42]. For a system to be stable, its natural response must decay to zero, as time approaches infinity, or oscillate. There are various specifications (time constant, rise time, and settling time, etc) that may reveal useful information about the network behavior (the reader is referred to [42]). $1/D$, called the time constant of the response, is the time for the step response to rise 63% of its final or steady state value. Rise time, defined as the time for the system response to go from 0.1 to 0.9 of its steady state value, is equal to $2.2/D$. Settling time is the time for the response to reach and stay within 2% of its final value and it is equivalent to $4/D$.

2. Cascade and parallel forms. Based on the block diagram representation of simple regulation shown in **Figure 3**, we can now decompose any complex gene network using four interconnection topologies: cascade forms, parallel forms, feedback loops, and feedforward loops. These topologies are often intermingled with one another as shown in later examples. In this section, we describe the cascade and parallel forms. **Figure 4(a)** shows an example of cascaded simple regulation blocks. The first output is the product of $F(s)$ and $G_Y(s)$. It is also the input for the second simple regulation. The second or final output is the sum of $F(s)G_Y(s)$ and the basal production rate F_0/s multiplied by $F(s)$ and $G_Z(s)$. This cascade form can be simplified

using an equivalent transfer function $G(s)$ as shown at the bottom of the **Figure 4(a)**. **Figure 4(b)** shows an example of parallel simple regulations. In this case, the equivalent transfer function $G(s)$ is the algebraic sum of $G_Y(s)$ and $G_Z(s)$. Using these simplification methods, any large complex block diagram can be reduced into a single transfer function.

3. Feedback loops. **Figure 5(a)** shows the block diagram of a feedback loop. Compared to **Figure 3**, there is an additional feedback element $H(s)$. The block diagram can be expressed in the Laplace domain as:

$$Y(s) = F(s) \frac{G(s)}{1 \pm G(s)H(s)} = F(s)G(s)' \left(G(s)' = \frac{G(s)}{1 \pm G(s)H(s)} \right) \tag{24}$$

where $G(s)'$ is the simplified equivalent transfer function.

4. Case Study I: Autoregulations. Autoregulation is the simplest form of feedback loops and consists of one simple regulation and one feedback loop (**Figure 5(b)**). In simple regulation, $x^*(t)$ is the only transcription factor that determines the production of $y(t)$. On the other hand, in autoregulation, both $x^*(t)$ and $y^*(t)$, the active form of $y(t)$, can affect the $y(t)$ production. The $y(t)$ production function due to $y^*(t)$ or $f_y(t)$ requires two Hill functions as in the case of $y(t)$ production function due to $x^*(t)$. Assuming that the signal s_y that controls the $y^*(t)$ formation is an activator:

$$y^*(t) = \frac{y(t) \cdot s_y^l(t)}{K_{1A}^l + s_y^l(t)} \tag{25}$$

The production function $F_y(t)$ with respect to $y^*(t)$ can be expressed as:

$$F_y(t) = \frac{F_{\max} \cdot [y^*(t)]^n}{K_{2A}^n + [y^*(t)]^n} \tag{26}$$

Applying the linearization scheme used in (9–12), we can rewrite (26) as:

$$F_y(t) = H_2 y^*(t) = H_2 y(t) H_1(t) = h(t) y(t) \quad (h(t) = H_2 H_1(t)) \tag{27}$$

$h(t)$ in (27) is equivalent to $f(t)$ in (12). As in (18), if we assume that the signal $s_y(t)$ is constant then we can express $H_1(t)$ as a constant value H_1 and, consequently, $h(t)$ as H ($H = H_1 H_2$).

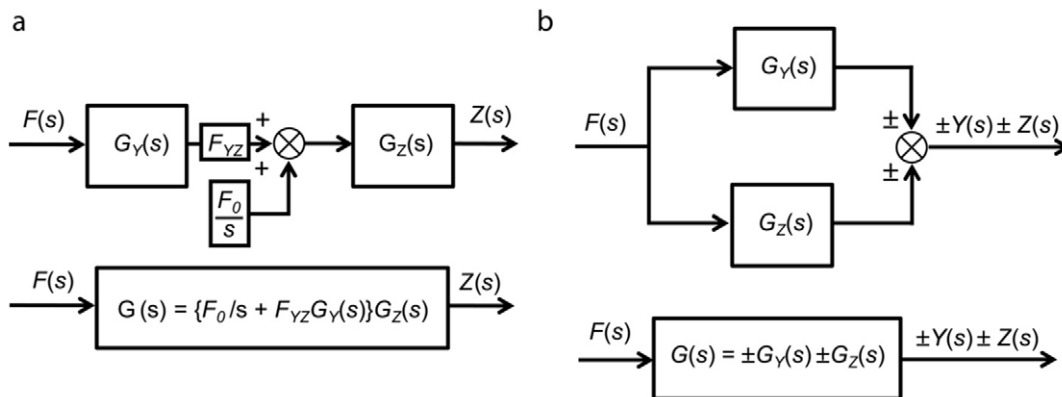


Figure 4. Cascade and Parallel forms. (a) Two cascaded simple regulations (above) and their equivalent simplified form (below). (b) Parallel simple regulations (above) and their equivalent simplified form (below). doi:10.1371/journal.pone.0012785.g004

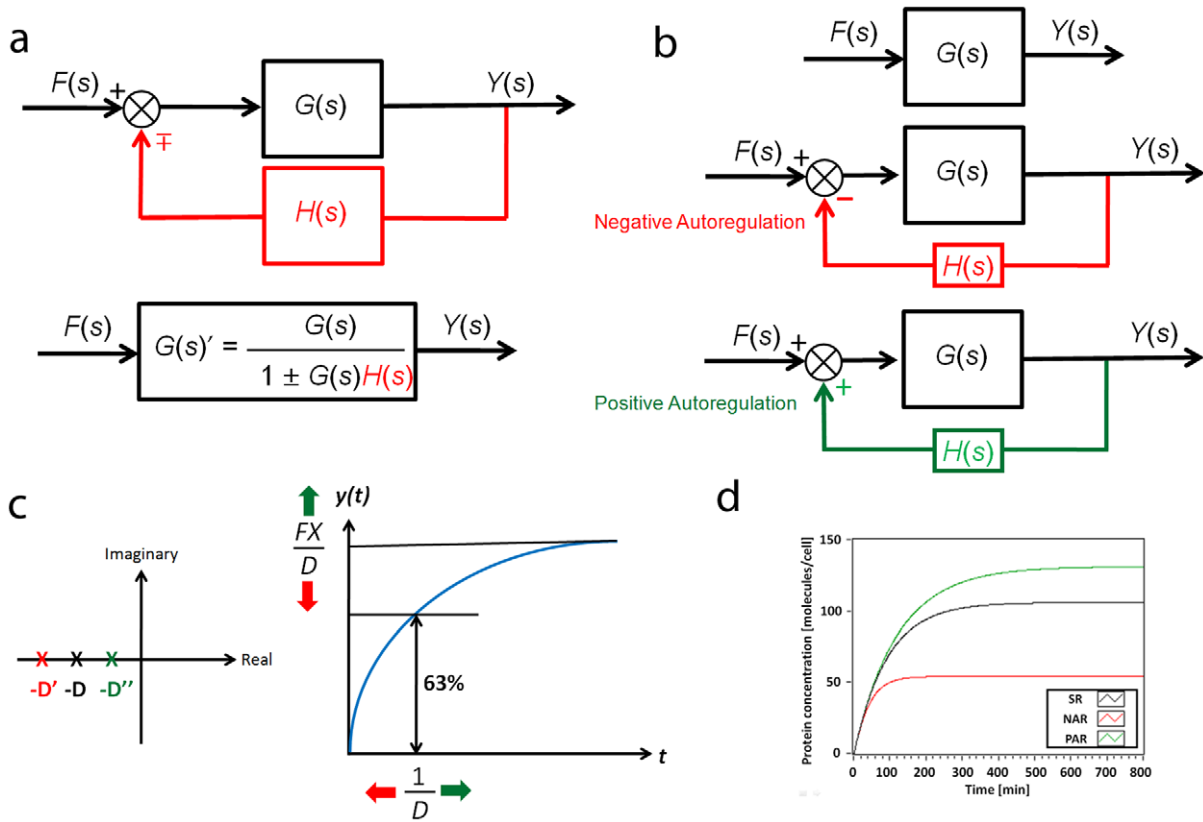


Figure 5. Feedback loops. (a) Block diagram of a feedback loop and its simplified equivalent form. (b) Block diagram of negative autoregulation (NAR) and positive autoregulation (PAR). (c) The effect of H on the stability and behavior. NAR speeds up the response time of the gene expression and decreases the steady state value. In contrast, PAR increases both the response time and steady state value. (d) Simulation results. SR stands for simple regulation. The effects of NAR and PAR on the response time and steady state value are shown. doi:10.1371/journal.pone.0012785.g005

Negative autoregulation (NAR) occurs when a transcription factor represses the transcription of its own gene (negative feedback). It has been demonstrated that NAR speeds up the response time of the gene expression, decreases the steady state value, and reduces cell-cell variation in protein levels [13,46]. Using (20) and (24), the transfer function $G'(s)$ can be derived as:

$$G(s)' = \frac{G(s)}{1 + G(s)H(s)} = \frac{G(s)}{1 + G(s)H} = \frac{\frac{1}{s+D}}{1 + \frac{1}{s+D}H} \quad (28)$$

$$= \frac{1}{s+D+H} = \frac{1}{s+D'} \quad (D' = D+H)$$

Note that we have a new degradation/dilution constant D' . Since both D and H have positive values in biological systems, D' is also positive, indicating that the system is stable based on the pole-zero analysis described previously (Figure 5(c)). Furthermore, the time constant of the response ($1/D'$) is greater than $1/D$ since H is positive. This explains the experimental observation of the decrease of the response time using NAR (Figure 5(c) and (d)).

Positive autoregulation (PAR) occurs when a transcription factor controls its own protein production rate. In contrast to NAR, the response time is extended and the steady state value and cell-cell variation are increased [13]. Using (20) and (24), the

transfer function of PAR is:

$$G(s)'' = \frac{G(s)}{1 - G(s)H(s)} = \frac{G(s)}{1 - G(s)H} = \frac{\frac{1}{s+D}}{1 - \frac{1}{s+D}H} \quad (29)$$

$$= \frac{1}{s+D-H} = \frac{1}{s+D''} \quad (D'' = D-H)$$

Equation 29 illustrates that, compared to simple regulation, the time constant ($1/D''$) is increased because D'' is equal to D subtracted by H , a positive number. This explains the increase of the response time (Figure 5(c) and (d)). Additionally, for the system to be stable, $-D''$ must be negative (or H must be smaller than D). In other words, if the positive feedback is “too strong” (H is greater than D) then the system may become unstable.

5. Case Study II: Negative feedback loop involving two cascaded simple regulations. Figure 6 shows the block diagram of a negative feedback loop that consists of two cascaded simple regulations. X_{gene} activates Y_{gene} in the presence of signal $S_{X\gamma}$. Y_{gene} then activates Z_{gene} in the presence of $S_{Y\gamma}$, and Z_{gene} at the same time represses Y_{gene} in the absence of $S_{Z\gamma}$. If we assume that the basal protein production rate is negligible (zero) the transfer function becomes second-order system and there are well-established linear control theory tools for analyzing second-order systems. However, as we will show in subsequent section, the results can be extended to cases where the basal rates are not zero.

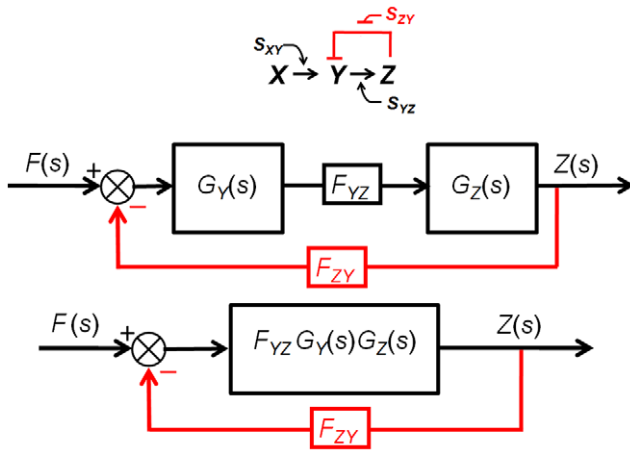


Figure 6. Negative feedback loop involving two cascaded simple regulations. The top block diagram is reduced (simplified) into an equivalent block diagram shown at the bottom. Note that the transfer function in the reduced form is the product of the transfer functions $G_Y(s)$ and $G_Z(s)$ multiplied by F_{YZ} . doi:10.1371/journal.pone.0012785.g006

Using (24), the transfer function can be expressed as:

$$G(s) = \frac{F_{YZ} G_Y(s) G_Z(s)}{1 + F_{ZY} G_Y(s) G_Z(s)} \quad (30)$$

Substituting $G_Y(s)$ and $G_Z(s)$ by $1/(s+D_Y)$ and $1/(s+D_Z)$, (30) becomes:

$$G(s) = \frac{F_{YZ}}{s^2 + (D_Y + D_Z)s + D_Y D_Z + F_{YZ} F_{ZY}} \quad (31)$$

Now we have a second-order transfer function $G(s)$. Whereas varying the first-order transfer function parameter D changes only the response time ($1/D$), changing the parameters (F_{YZ} , F_{ZY} , D_Y and D_Z) of a second-order transfer function can influence both the speed and form of the system response [42]. $G(s)$ can be restated in a general second-order form:

$$G(s) = \frac{F_{YZ}}{s^2 + as + b} \quad (a = D_Y + D_Z, b = D_Y D_Z + F_{YZ} F_{ZY}) \quad (32)$$

The natural frequency ω_n of a second-order system is defined as the frequency of an undamped oscillation [42].

$$\omega_n = \sqrt{b} = \sqrt{D_Y D_Z + F_{YZ} F_{ZY}} \quad (33)$$

Without damping, the poles would be on the imaginary axis as shown in **Figure 7(b)**. Equation (33) shows that the natural frequency can be tuned by varying the strength of both F_{YZ} and F_{ZY} .

For a damped system, the damping ratio ζ is defined as [42]:

$$\zeta = \frac{a}{2\omega_n} = \frac{D_Y + D_Z}{2\sqrt{D_Y D_Z + F_{YZ} F_{ZY}}} \quad (34)$$

Various step responses of a second-order system with respect to the damping ratio ζ are shown in **Figure 7**. The damped natural frequency ω_d is defined as [42]:

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \quad (35)$$

The poles of an undamped oscillating system lie on the imaginary axis, and thus the system is said to be marginally stable [42]. Also, note that all the poles in **Figure 7** lie on the left half of the plane except for the ones that belong to an undamped oscillation. For an underdamped oscillation, additional specifications can be defined and used to predict the response in detail, including the peak time, percent overshoot, settling time, and rise time. Using a numerical approach, rise time can be computed even though there is no precise analytical solution [42]. All these specifications can be expressed in terms of F_{YZ} , F_{ZY} , D_Y and D_Z .

6. Case Study III: Tunable synthetic gene oscillators.

Building tunable synthetic gene oscillators has been a central area of focus for systems and synthetic biologists [47–49]. An oscillating system is basically an undamped second-order transfer function (**Figure 7(b)**). For a system to have no damping, a (being equal to $D_Y + D_Z$) in (34) must be zero. Therefore, an additional negative term is required to decrease the value of a to zero (or close to zero from a practical point of view) since the degradation/dilution constants D_Y and D_Z are positive numbers. Using the transfer function method, we will show that the use of positive autoregulation can lead to undamped oscillations and that by varying the strength of positive autoregulation we can tune the oscillation frequency.

A tunable synthetic gene oscillator is shown in **Figure 8** [47]. The genes *araC* (denoted as 1) and *lacI* (denoted as 2) have identical hybrid promoters that can be activated by *AraC* in the presence of arabinose and repressed by *LacI* in the absence of IPTG. As shown in the figure, *araC* has a positive autoregulation while *LacI* a negative autoregulation. F_1 and F_3 are determined by the signals, arabinose and IPTG, respectively. The uppermost block diagram in **Figure 8** can be simplified by removing autoregulations (28 and 29). It can be further simplified since it has two cascaded elements. The final equivalent block diagram with a single transfer function is shown at the bottom of the figure. Using (24), the overall transfer function can be expressed as:

$$G(s) = \frac{F_2}{s^2 + (D_1 + D_2 - F_1 + F_2)s + D_1 D_2 + D_1 F_3 - D_2 F_1 - F_1 F_3 + F_2 F_4} \quad (36)$$

According to (36), $a (= D_1 + D_2 - F_1 + F_2)$ must be zero (or close to zero) for an undamped oscillation. It is clear that F_1 , the only negative term created by the positive autoregulation of *araC*, can contribute to decreasing a to zero. This explains why positive autoregulation is needed for an undamped oscillation in biological systems. The natural frequency of the response is:

$$\omega_n = \sqrt{b} = \sqrt{D_1 D_2 + D_1 F_3 - D_2 F_1 - F_1 F_3 + F_2 F_4} \quad (37)$$

Equation 37 shows that both the parameters influenced by arabinose (F_1 and F_2) and parameters influenced by IPTG (F_3 and F_4) can change or tune the oscillating frequency. Since F_1 and F_2 in (37) contribute to both positive and negative terms, it is difficult to predict their effect on the frequency. Likewise, F_3 and F_4 are also found in both positive and negative terms in (37), again making the prediction difficult. Experimentally, the IPTG concentration and oscillating frequency show a non-monotonic relationship [47].

Using the transfer function method, we can identify a way of changing such a non-monotonic relationship into a monotonic one. For example, if the negative autoregulation for *lacI* is removed from the network (by removing the repression operator site from the promoter), $D_1 F_3$ and $-F_1 F_3$ are eliminated from (37). The natural frequency then becomes:

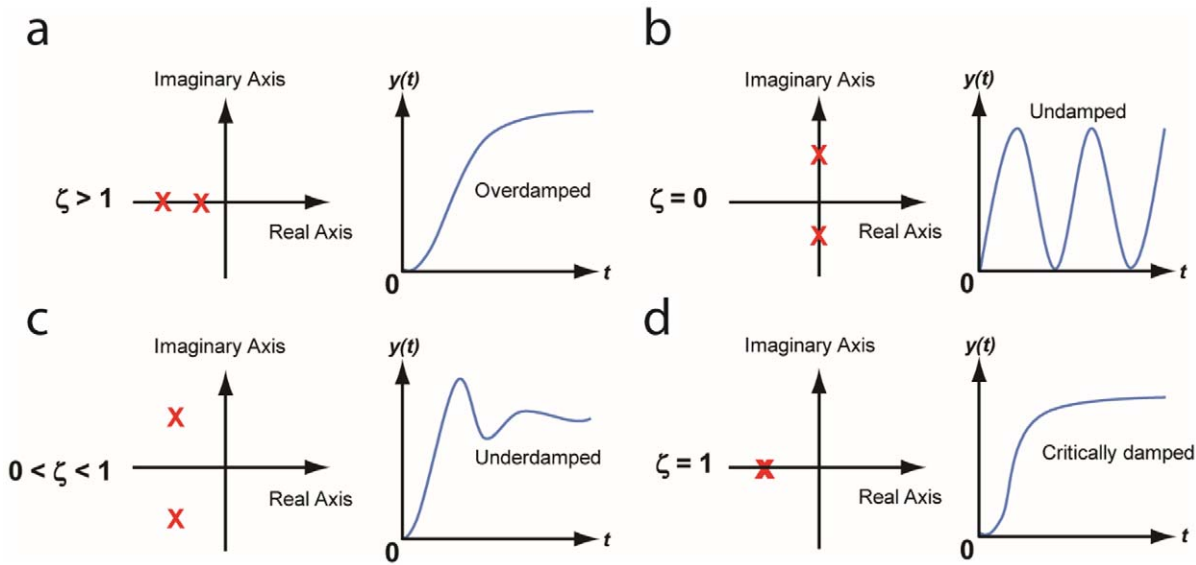


Figure 7. Step response of a second-order system with respect to the damping ratio ζ (the poles are shown as X). (a) Overdamped oscillation. The damping ratio is greater than 1 and the poles are both negative real numbers. The system reaches its steady state without oscillation. As the damping ratio increases, it reaches the steady state slower. (b) Undamped oscillation. Note that all the poles are on the imaginary axis. The damping ratio is zero and there is an oscillation without damping. (c) Underdamped oscillation. The damping ratio is between 0 and 1, and the poles are complex numbers with the negative real part. The oscillation gradually decreases to zero as the system reaches its steady state. (d) Critically damped oscillation. The steady state is reached in the fastest way without oscillation. The two poles have the same negative value [42].
doi:10.1371/journal.pone.0012785.g007

$$\omega_n = \sqrt{D_1 D_2 - D_2 F_1 + F_2 F_4} \quad (38)$$

Now, F_4 belongs to a positive term, meaning that as the IPTG concentration is raised the oscillating frequency also increases, and

vice versa, indicating that IPTG can now cause a monotonic behavior.

In order for an undamped system ($a=0$) to be marginally stable, b in (37) must be greater than zero so that the two poles lie separately on the imaginary axis [42]:

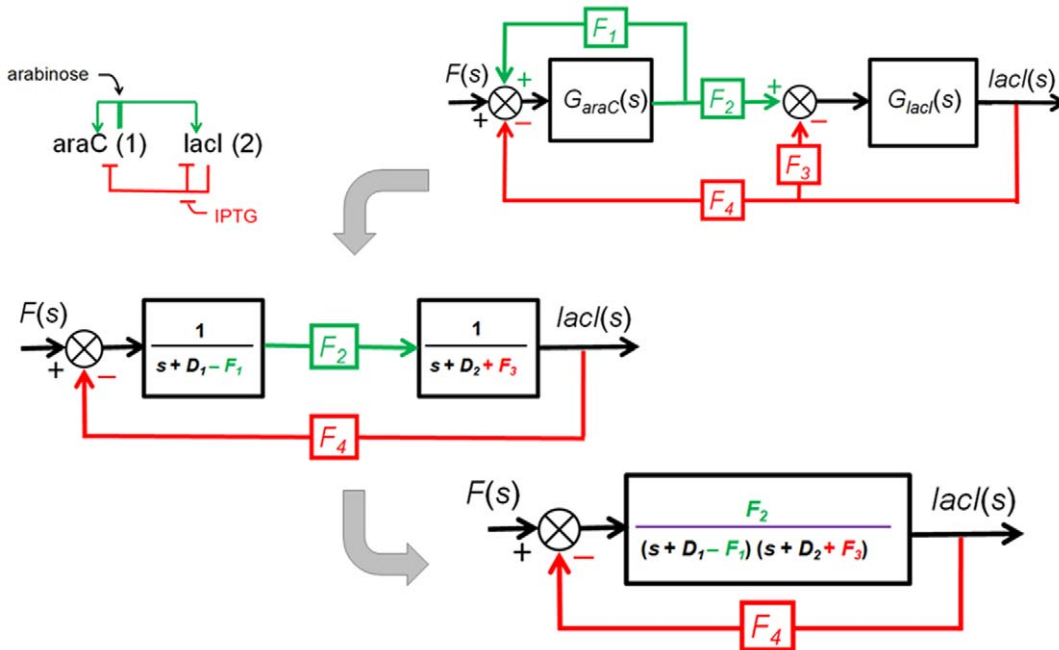


Figure 8. Tunable synthetic gene oscillator. The top block diagram is reduced into the block diagram in the middle by removing positive and negative autoregulation. Then, the two cascaded transfer functions are reduced into a single equivalent transfer function, as shown in the bottom block diagram.
doi:10.1371/journal.pone.0012785.g008

$$D_1D_2 + D_1F_3 - D_2F_1 - F_1F_3 + F_2F_4 > 0 \quad (39)$$

There are two negative terms ($-D_2F_1$ and $-F_1F_3$) that can make the system unstable and they are related to both arabinose (F_1) and IPTG (F_3). However, since F_1 is present in both terms and not in any of the positive terms, decreasing the arabinose concentration (and not the IPTG concentration) seems to be an effective solution when there is an issue of instability.

7. Case Study IV: Eliminating the effect of the basal production rate. In this section, we will demonstrate how the effect of the basal production rate $f_b(t)$ can be removed using a second-order negative feedback system described earlier. If we add the Laplace transform of the basal Z protein production rate $F_{0,Z}(s)$ to the model shown in the Figure 6, as illustrated in Figure 9, the transform of the output $Z(s)$ is given by:

$$Z(s) = E(s)G_Y(s)F_{YZ}G_Z(s) + F_{0,Z}G_Z(s) \quad (40)$$

Note that $E(s)$ is the error (difference) between $F(s)$ and $F_{ZY}Z(s)$. It is analogous to the concept of steady-state error in the presence of a disturbance ($F_{0,Z}(s)$ in our case) [42]. Substituting $Z(s) = \frac{[F(s) - E(s)]}{F_{ZY}}$ into (40) and solving for $E(s)$, we get:

$$E(s) = \frac{1}{1 + F_{YZ}F_{ZY}G_Y(s)G_Z(s)} F(s) - \frac{F_{YZ}G_Z(s)}{1 + F_{YZ}F_{ZY}G_Y(s)G_Z(s)} F_{0,Z}(s) \quad (41)$$

where the first term can be regarded as a transfer function relating $E(s)$ to $F(s)$ and the second term relating $E(s)$ to $F_{0,Z}(s)$.

Applying the final value theorem to Eq. (41), we obtain:

$$e(\infty) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \frac{s}{1 + F_{YZ}F_{ZY}G_Y(s)G_Z(s)} F(s) - \lim_{s \rightarrow 0} \frac{sF_{YZ}G_Z(s)}{1 + F_{YZ}F_{ZY}G_Y(s)G_Z(s)} F_{0,Z}(s) \quad (42)$$

As we assume the basal rate is constant $F_{0,Z}$, $F_{0,Z}(s)$ becomes $F_{0,Z}/s$. Substituting this value into the second term of (42), the steady-state error component due to the basal rate can be

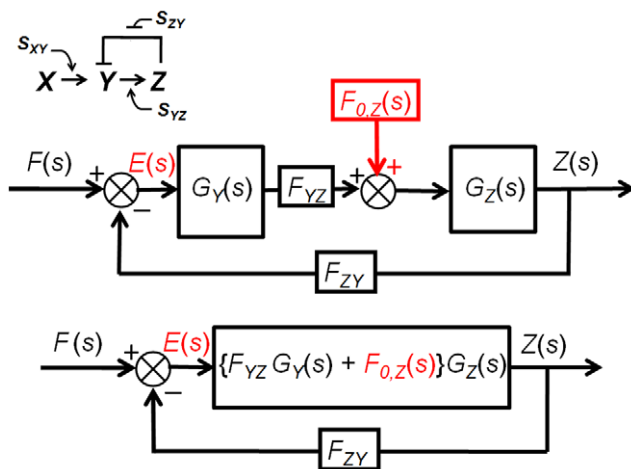


Figure 9. Negative feedback loop involving two cascaded simple regulations with the Laplace transform of the basal production rate $F_{0,Z}(s)$ added. The top block diagram is reduced (simplified) into an equivalent block diagram shown at the bottom. doi:10.1371/journal.pone.0012785.g009

found as:

$$\begin{aligned}
 & - \lim_{s \rightarrow 0} \frac{sF_{YZ}G_Z(s)}{1 + F_{YZ}F_{ZY}G_Y(s)G_Z(s)} F_{0,Z}(s) \\
 & = - \lim_{s \rightarrow 0} \frac{sF_{0,Z}F_{YZ}G_Z(s)}{s[1 + F_{YZ}F_{ZY}G_Y(s)G_Z(s)]} \\
 & = - \frac{1}{\lim_{s \rightarrow 0} \frac{1}{F_{0,Z}F_{YZ}G_Z(s)} + \lim_{s \rightarrow 0} \frac{F_{YZ}G_Y(s)}{F_{0,Z}}}
 \end{aligned} \quad (43)$$

Equation (43) can provide important insight about reducing the steady-state error due to basal activity by tuning the strength of F_{YZ} or F_{ZY} .

8. Case Study V: A synthetic oscillator involving four cascaded simple regulation loops. A second order system with a negative feedback loop described earlier could be efficiently analyzed using a pole/zero plot and various formulas. Even though we do not have such formulas for higher order systems, pole/zero plots can still be useful for predicting their behavior.

Figure 10 shows the block diagram of a fourth order system with various feedbacks. It is a synthetic network called IRMA (*In vivo* Reverse-engineering and Modeling techniques Assessment) that consists of four genes (*CBF1*, *GAL4*, *SWI5*, and *ASH1*) in *Saccharomyces Cerevisiae* [50]. It has been demonstrated both computationally (using a non-linear model) and experimentally that the network could be turned into an oscillator by changing various parameters (Michaelis-Menten coefficient, Hill coefficient, etc) that control the strength of the interactions among genes.

An equivalent model was built using the transfer function method, as shown in **Figure 10(a)**, where the strength of the interactions was determined by changing the values of the constants (F_{12} , F_{23} , F_{31} , F_{33} , F_{34} , and F_{41}) shown in the figure. The numbers 1, 2, 3, and 4 in subscript represent the four genes mentioned earlier, respectively. As described in the paper [50], we were able to demonstrate that increasing the values of F_{12} , F_{33} , F_{34} , and F_{41} contributes to generating an oscillatory behavior. **Figure 10(b)** shows that as we increase the values of those constants, the dominant poles (two poles that have the least real values) cross the imaginary axis. When the dominant poles are exactly on the axis, the system exhibits an undamped oscillation, similar to the behavior of a second order system shown in **Figure 7** (see Movie S1 and File S1).

9. Case Study VI: Interconnected feedforward loops. Feedforward loops (FFL) are network motifs [13] that combine into more complex and larger networks (e.g. in *Bacillus Subtilis* [51]). The coherent type-1 feedforward loop (C1-FFL) and incoherent type-1 feedforward loop (Ic1-FFL) are the most abundant FFL types [52]. In this section, we will describe how transfer function method can be used to model a network that consists of interconnected feedforward loops. **Figure 11(a)** shows a simplified schematic diagram of the network. Two sets of C1-FFL and Ic1-FFL in parallel are connected in cascade. Ic1-FFLs generate pulses of C1 and C2, and C1-FFLs create delays in C2 and C3 expression. **Figure 11(b)** shows a schematic diagram that illustrates the sequential expression of C1, C2, and C3. **Figure 11(c)** shows the block diagram of the network. Using the transfer function method, the sequential expression of the three genes can be simulated as shown in **Figure 11(d)**. Note that if the values of the protein concentrations are normalized, the expression pattern will be equivalent to the one shown in **Figure 11(b)**.

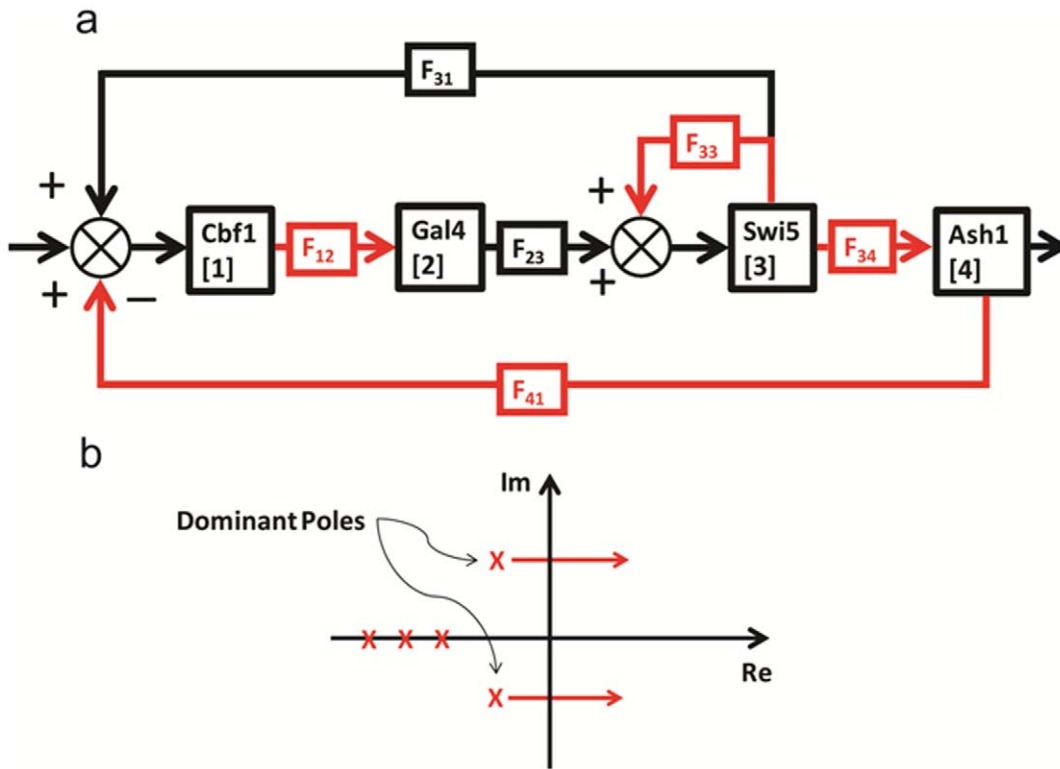


Figure 10. Negative feedback loop involving four-cascaded simple. (a) Increasing the values of F_{12} , F_{33} , F_{34} , and F_{41} contributes to generating an oscillatory behavior (shown in red). (b) Increasing the values of F_{12} , F_{33} , F_{34} , and F_{41} causes the dominant poles to cross the imaginary axis. When the dominant poles are exactly on the axis, the system exhibits an undamped oscillation. doi:10.1371/journal.pone.0012785.g010

B. Linear State-Space Method for Gene Network Modeling

A state is a complete summary of the status of a system at a particular point in time, and it is described by the values of a set of state variables [53]. Based on the linearization scheme presented previously, a linear state-space method can be applied to model gene networks. There are many benefits in using the linear state-space approach. First, it allows time-varying systems, meaning that the models can include time-varying signals. This was not possible in the case of transfer function-based models described earlier. Secondly, large complex networks that are multiple-input multiple-output (MIMO) systems can be represented in a compact way, using vectors and matrices. Thirdly, the dynamic behavior of a system can be understood using the eigenvalues and eigenvectors. The eigenvalues are equivalent to the poles discussed in the transfer function method. Finally, we can build a stochastic linear state-space model that enables us to utilize a spectrum of tools available for optimal/robust estimation/control for gene network modeling [35–37]. As an example, we illustrate that the Kalman filter, one of the well-established optimal estimation algorithms in science and engineering, can be applied towards modeling a simple two-gene network.

In the linear model of simple regulation (linear ODE form shown in (13)), the system involves two proteins, $x(t)$ and $y(t)$, with units of concentration per cell. If we consider gene y as our system of interest, then and the state vector \mathbf{s} can be expressed as:

$$\mathbf{s} = [y(t)] \tag{44}$$

The state-space model can be represented as [53]:

$$\dot{\mathbf{s}} = \mathbf{A}\mathbf{s} + \mathbf{B}\mathbf{u} \rightarrow \frac{dy(t)}{dt} = -d(t)y(t) + F_0 + f(t)x(t) = -d(t)y(t) + [1 \ f(t)] \begin{bmatrix} F_0 \\ x(t) \end{bmatrix}$$

$$\left(\mathbf{A} = -d(t), \mathbf{B} = [1 \ f(t)], \mathbf{u} = \begin{bmatrix} F_0 \\ x(t) \end{bmatrix} \right) \tag{45}$$

where \mathbf{u} is the input vector. Note that we do not have to assume that $f(t)$ and $d(t)$ are time-invariant as in the transfer function case (18).

The state-space method is analogous to the transfer function method in many ways. Therefore, in this section we present only the case of a negative feedback loop involving two cascaded simple regulations. The network shown in **Figure 6** can be written as a state-space model:

$$\begin{bmatrix} \frac{dy(t)}{dt} \\ \frac{dz(t)}{dt} \end{bmatrix} = \begin{bmatrix} -d(t) & -f_{zy}(t) \\ f_{zy}(t) & -d_z(t) \end{bmatrix} \begin{bmatrix} y(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} 1 & f_{xy}(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{0,y} \\ x(t) \\ F_{0,z} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} -d(t) & -f_{zy}(t) \\ f_{zy}(t) & -d_z(t) \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 & f_{xy}(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{u} = \begin{bmatrix} F_{0,y} \\ x(t) \\ F_{0,z} \end{bmatrix} \tag{46}$$

where $f_{ab}(t)$ denotes a function (shown in (12)) related to the production of protein b by protein a . $F_{0,a}$ represents the basal production rate of protein a . We can predict the behavior and

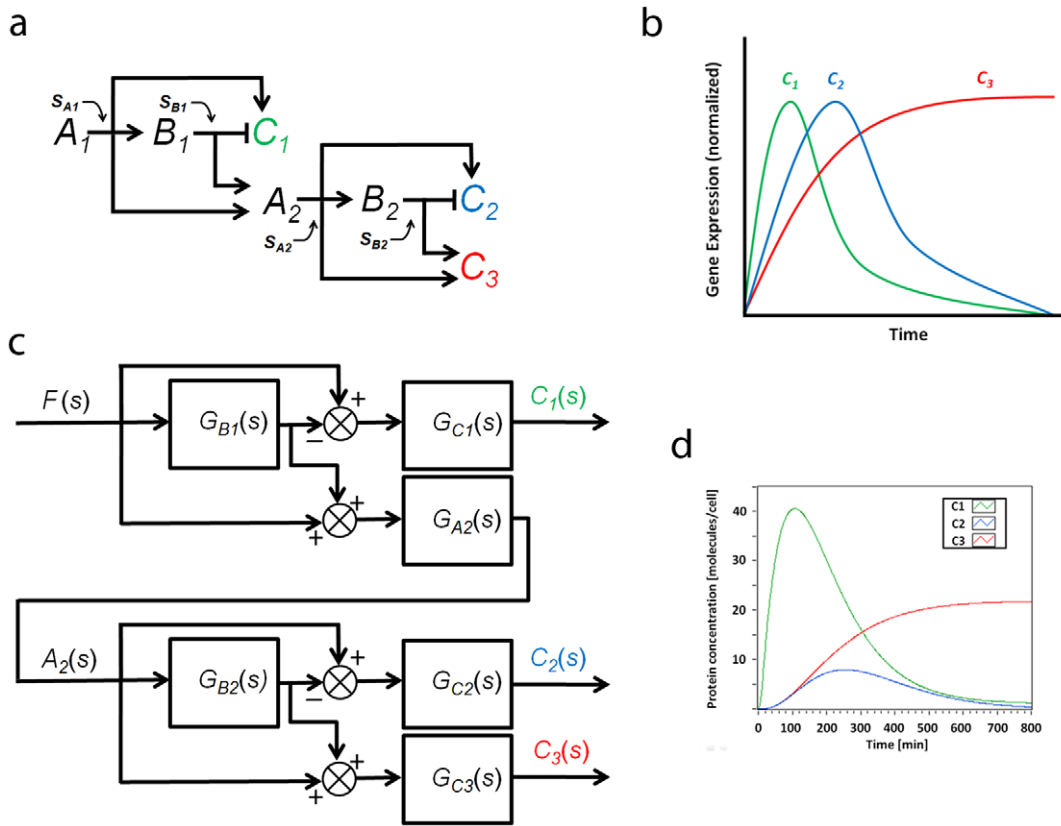


Figure 11. Interconnected feed forward loops. (a) **A simplified schematic diagram.** Two sets of C1-FFL and I1-FFL in parallel are connected in cascade. (b) The sequential expression of C1, C2, and C3. I1-FFLs generate pulses of C1 and C2, and C1-FFLs create delays in C2 and C3 expression. (c) Block diagram. The protein production constants “F”s are not shown in the diagram. (d) Simulation result. The sequential expression of the three genes is shown. Note that if the values of the protein concentrations are normalized, the expression pattern will be equivalent to the one shown in (b). doi:10.1371/journal.pone.0012785.g011

stability of the system using the eigenvalues and eigenvectors of the matrix \mathbf{A} [53,54]. For example, assuming $d_y(t) = d_z(t) = 0.01$ (constant), $f_{zy}(t) = 0.01$ (constant), $f_{yz}(t) = 0.04$ (constant), $f_{xy}(t) \cdot x(t) = 10$ (constant), the eigenvalues (λ_1 and λ_2) and eigenvectors (ϕ_1 and ϕ_2) of \mathbf{A} can be calculated as:

$$\lambda_1 = -0.01 + 0.02i, \lambda_2 = -0.01 - 0.02i,$$

$$\phi_1 = \begin{bmatrix} -0.4472i \\ -0.8944 \end{bmatrix}, \phi_2 = \begin{bmatrix} 0.4472i \\ -0.8944 \end{bmatrix} \quad (47)$$

Since the eigenvalues are equivalent to the poles, we can immediately predict that y and z will exhibit underdamped oscillations and the system is stable. The solution can be written as:

$$\begin{bmatrix} y(t) \\ z(t) \end{bmatrix} = C_1 e^{\lambda_1 t} \phi_1 + C_2 e^{\lambda_2 t} \phi_2 = C_1 e^{(-0.01+0.02i)t} \begin{bmatrix} -0.4472i \\ -0.8944 \end{bmatrix} + C_2 e^{(-0.01-0.02i)t} \begin{bmatrix} 0.4472i \\ -0.8944 \end{bmatrix} \quad (48)$$

where C_1 and C_2 are determined by the initial values of y and z . Depending on the initial values, the time-dependent values of y and z can follow an infinite number of paths as shown in the vector field (Figure 12). However, regardless of the initial values, the figure shows that all the trajectories will eventually converge into an equilibrium point. This equilibrium point can also be computed by

solving $dy(t)/dt = 0$ and $dz(t)/dt = 0$ in (46). Note that the damped oscillatory behaviors that were predicted using the eigenvalues are observed in both Y and Z expressions (Figure 12).

1. Case Study VII: Optimal estimation in gene network measurements.

Stochastic optimal estimation [18,19,43,45] and control methods aim to determine the best strategy for estimating or controlling a dynamic system in the presence of uncertainty [35]. This objective is common to many fields, including engineering, science, and economics [35]. Assume that, in the case of simple regulation $x \rightarrow y$, we are given noisy time-series fluorescence measurement data of y protein [55–57]. Would it be possible to “optimally” estimate protein x which is also fluctuating due to noise? In this section, we will illustrate that the Kalman filter, a well-established optimal estimation algorithm, can be applied for stochastic modeling of a simple regulation (two-gene network). Before we discuss the stochastic model of simple regulation, it is important to note that there are two different noise contributions in gene networks [18,58,59]. “Intrinsic noise” is generated by the inherent stochasticity of biochemical processes, such as transcription and translation, which are directly related to the expression of a specific gene. On the other hand, the environment including other cellular components (mitochondria, microtubules, etc) that indirectly influence the expression of y gene, contribute to the “extrinsic noise”.

First, the least squares estimation is used to estimate the constant concentration of protein x (without noise), given noisy measurement data of protein y . Using the Euler’s method, equation (13) can be expressed as:

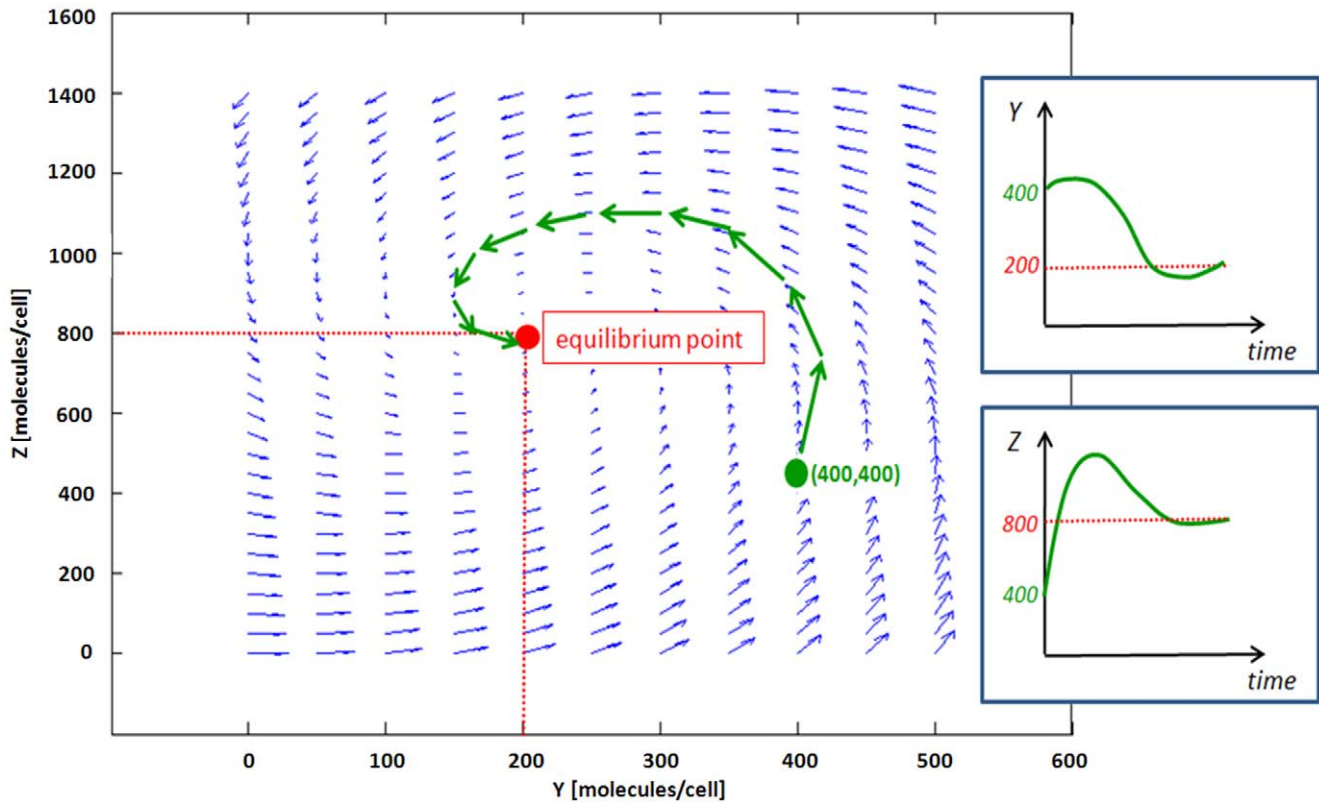


Figure 12. Vector field of the negative feedback loop. The direction and magnitude of each vector is determined by (46). When the initial value is (400,400), the trajectory eventually converges to the equilibrium point (200,800) (green arrows). Damped oscillations of both Y and Z expressions are also shown in small boxes. doi:10.1371/journal.pone.0012785.g012

$$y_{n+1} = y_n + \frac{dy_n}{dn} \cdot h = y_n + [F_{0,n} + f(t)x_n - d(t)y_n] \cdot h \tag{49}$$

$$= F_{0,n}h + f(t)hx_n + (1 - d(t)h)y_n \quad (n = 1, 2, \dots, N-1)$$

where h is the step size. For simplicity, we will assume $f(t)$ and $d(t)$ are constant (F and D) and the basal protein production rate and initial value of y are zero ($F_{0,n} = y_1 = 0$):

$$y_2 = Fhx_1 + (1 - Dh)y_1 = Fhx_1$$

$$y_3 = Fhx_2 + (1 - Dh)y_2 = Fhx_2 + (1 - Dh)Fhx_1$$

$$y_4 = Fhx_3 + (1 - Dh)y_3 = Fhx_3 + (1 - Dh)Fhx_2 + (1 - Dh)^2 Fhx_1$$

$$\vdots$$

$$y_{N+1} = Fhx_N + (1 - Dh)y_N = Fhx_N + (1 - Dh)Fhx_{N-1} + \dots + (1 - Dh)^{N-2} Fhx_1 \tag{50}$$

$$\rightarrow \begin{bmatrix} y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{N+1} \end{bmatrix} = \begin{bmatrix} Fh & 0 & \dots & 0 \\ (1 - Dh)Fh & Fh & \dots & 0 \\ (1 - Dh)^2 Fh & (1 - Dh)Fh & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ (1 - Dh)^{N-2} Fh & (1 - Dh)^{N-1} Fh & \dots & Fh \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix}$$

$$\rightarrow y = Hx$$

which can be compactly shown as $y = Hx$. Since the measurement data vector y has noise:

$$y = Hx + e^y \tag{51}$$

where e^y is assumed to be a Gaussian zero-mean white noise vector.

Σ , the covariance matrix of e^y , can be shown as:

$$\Sigma = E[e^y(e^y)^T] = \begin{bmatrix} \sigma_2 & \dots & \sigma_{2,N+1} \\ \vdots & \ddots & \vdots \\ \sigma_{N+1,2} & \dots & \sigma_{N+1} \end{bmatrix} \tag{52}$$

Note that in this formulation protein x is noise-free and thus has no stochastic influence on y . Using the least squares method, the estimated x can be calculated as:

$$\hat{x} = (H^T H)^{-1} H^T y \tag{53}$$

Now, we will use the Kalman filter for estimating noisy x from noisy y (Methods: Derivation of the discrete-time Kalman filter for simple regulation). For estimating the concentration of x with a noise vector e^x (intrinsic and extrinsic noise combined), following equations can be used:

$$\hat{x}_{N+1|N} = \hat{x}_{N|N}$$

$$\hat{x}_{N+1|N+1} = \hat{x}_{N+1|N} + K_N (y_{N+1} - h_{N+1} \hat{x}_{N+1|N})$$

$$P_{N+1|N} = P_{N|N} + q_N \tag{54}$$

$$K_{N+1} = P_{N+1|N} h_{N+1} [P_{N+1|N} (h_{N+1})^2 + \sigma_{N+2}]^{-1}$$

$$P_{N+1|N+1} = P_{N+1|N} (1 - K_{N+1} h_{N+1})^2 + K_{N+1} \sigma_{N+2} (K_{N+1})^T$$

Figure 13(a) shows x , which is the sum of a constant value (200 molecules/cell) vector and Gaussian zero-mean white noise

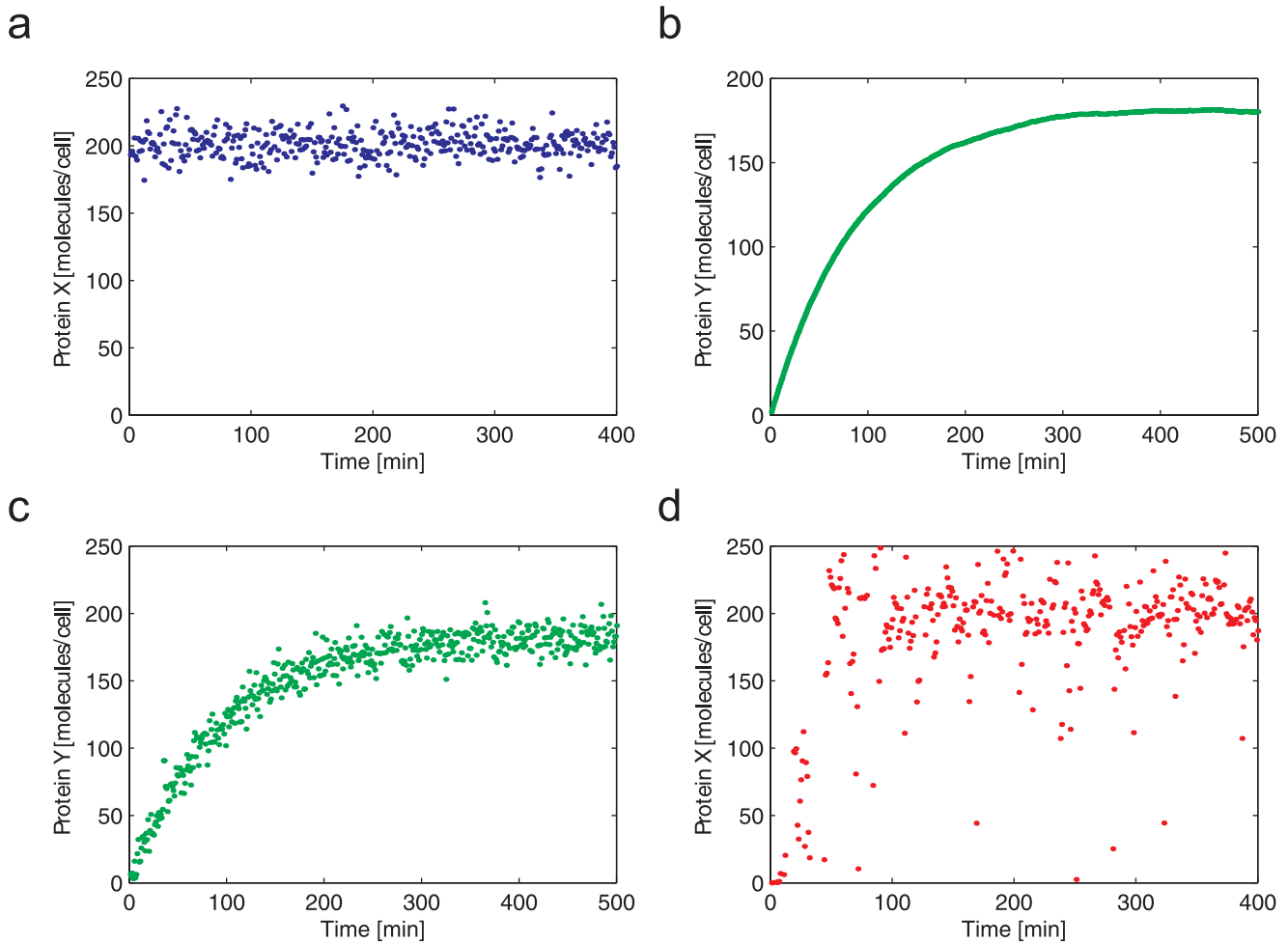


Figure 13. Estimation of x using the Kalman filter. (a) x is the sum of a constant value (200 molecules/cell) vector and Gaussian zero-mean white noise vector with a standard deviation of 10. It is assumed to have both intrinsic and extrinsic noise components. (b) y produced by the action of x illustrated in panel a. (c) Extrinsic noise e^y is added to y shown in panel b. (d) The estimated x using the Kalman filter is shown. doi:10.1371/journal.pone.0012785.g013

vector with a standard deviation of 10. The noise is assumed to have both intrinsic and extrinsic components. y produced by the action of x is shown in **Figure 13(b)**. Note that the noise has been reduced significantly. The noise in **Figure 13(b)** can be regarded to be as intrinsic only, indicating that a simple regulation can behave as a low-pass filter (removes high-frequency signals which are noise components in this example) [32]. In **Figure 13(c)**, extrinsic noise e^y described in (50) is added to y , and the total noise is composed of both intrinsic and extrinsic components. For the simulation, we assume that the extrinsic noise e^y is a Gaussian zero-mean white noise vector with the standard deviation 9. **Figure 13(d)** shows the estimated x using the Kalman filter, given y in **Figure 13(c)**. Even though it does not exactly duplicate x shown in (a), it is the “optimal” estimation of x .

2. Case Study VIII: Analysis of a six-node gene network. Here, we will illustrate the utility of the state-space method for the analysis of a 6-node network (**Figure 14(a)**). There are a few dynamic features that we may intuitively extract from the figure based on our previous discussions. For example, there is a coherent type-1 feedforward loop (C1-FFL) embedded in the network (that consists of A, B, and C), which suggests that gene C is expressed after the expression of gene B (there is a delay in the C expression).

The 6-node gene network can be modeled using the following state-space formulation:

$$\begin{aligned}
 \begin{bmatrix} \dot{A}(t) \\ \dot{B}(t) \\ \dot{C}(t) \\ \dot{D}(t) \\ \dot{E}(t) \\ \dot{F}(t) \end{bmatrix} &= \begin{bmatrix} -d_A(t) & 0 & 0 & -f_{DA}(t) & 0 & 0 \\ f_{AB}(t) & -d_B(t) & -f_{CB}(t) & 0 & 0 & 0 \\ f_{AC}(t) & f_{BC}(t) & -d_C(t) & 0 & -f_{EC}(t) & 0 \\ 0 & 0 & f_{CD}(t) & -d_D(t) & 0 & 0 \\ f_{AE}(t) & 0 & 0 & 0 & -d_E(t) & f_{FE}(t) \\ 0 & 0 & 0 & f_{DF}(t) & 0 & -d_F(t) \end{bmatrix} \begin{bmatrix} A(t) \\ B(t) \\ C(t) \\ D(t) \\ E(t) \\ F(t) \end{bmatrix} \\
 &+ \begin{bmatrix} 1 & f_{XA}(t) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{0,A} \\ X(t) \\ F_{0,B} \\ F_{0,C} \\ F_{0,D} \\ F_{0,E} \\ F_{0,F} \end{bmatrix} \tag{55}
 \end{aligned}$$

where $f_{AB}(t)$ is analogous to $f(t)$ in (13) and accounts for the relationship between $A(t)$ (transcription factor) to $B(t)$ (expressed protein). It is

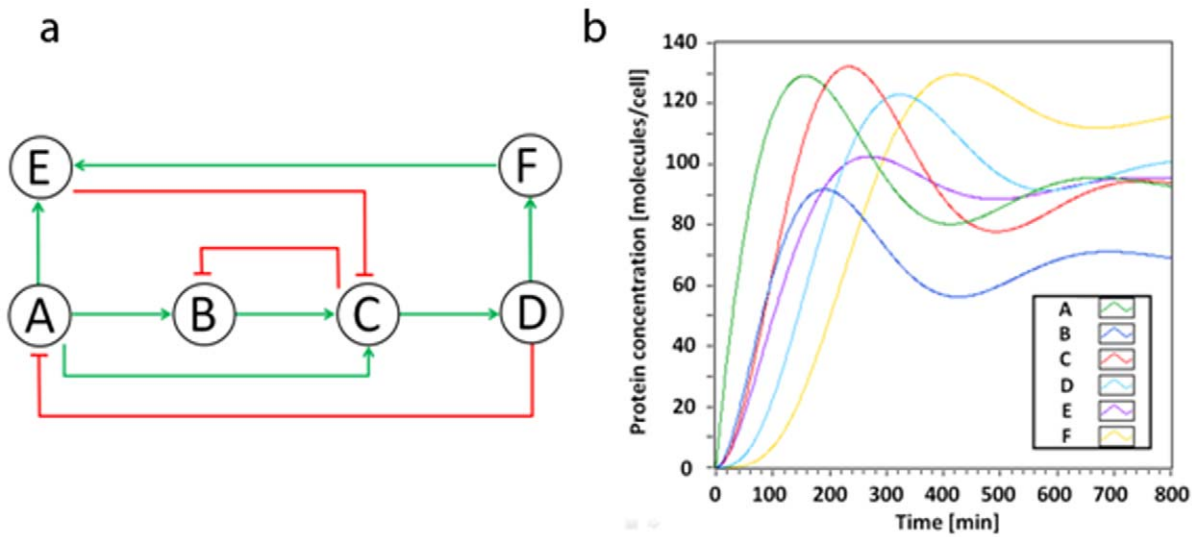


Figure 14. A six-node gene network. (a) A schematic illustration. (b) Simulation result based on the linear state-space method. doi:10.1371/journal.pone.0012785.g014

assumed that gene A is expressed by an arbitrary gene X , which is shown as an input in the model. **Figure 14(b)** illustrates that intuitive and qualitative information about the dynamic behavior can be acquired using the linear state-space method. The delay in gene C expression compared to the expression of gene B is also shown.

C. Further Discussion and Conclusion

We demonstrated that well-established tools of linear control theory can be used to model gene networks and explain experimental observations in a highly intuitive way. Methods such as the transfer function (frequency domain) and linear state-space (time domain) were applied to reveal inherent characteristics and predict the dynamic behavior of various gene network topologies, including cascade/parallel forms, feedback loops, and feedforward loops. Additionally, we showed that well-established optimal estimation tools, such as the Kalman filter, can be used in the context of gene network modeling in the presence of noise.

While we assumed that multiple transcription factors act on a single gene in an additive/subtractive way, in biological systems positive or negative cooperativity, and mutual exclusion can also be observed. Combinational logic is one approach that can be used for determining the net effect of multiple inputs at such junctions [13,30]. We expect that hybrid models that combine both continuous and logical approaches may provide additional insight.

Methods

A. Simulations

For simulations, the value of degradation/dilution constant (D) was 0.01/min and $f(t)$ was set as a constant (F) that ranged from 0.01 to 0.1/min. The input $x(t)$ had a fixed value between 100 and 1000 molecules/cell and the typical basal protein production rate was set as 0.1 molecules/min.

B. Derivation of the discrete-time Kalman filter for simple regulation

1. Recursive least squares estimation. For the estimation of \mathbf{x} , a new piece of information y_{new} can be used without repeating calculations already done on \mathbf{y}_{old} through the recursive

least squares estimation [34,36]. With a new piece of updated information, (51) can be expressed as:

$$\begin{bmatrix} y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{N+2} \end{bmatrix} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,N+1} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,N+1} \\ h_{3,1} & h_{3,2} & \cdots & h_{3,N+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N+1,1} & h_{N+1,2} & \cdots & h_{N+1,N+1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N+1} \end{bmatrix} + \begin{bmatrix} e^y_2 \\ e^y_3 \\ e^y_4 \\ \vdots \\ e^y_{N+2} \end{bmatrix} \quad (56)$$

From (56), it can be stated that:

$$y_{N+2} = [h_{N+1,1} \quad h_{N+1,2} \quad \cdots \quad h_{N+1,N+1}] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N+1} \end{bmatrix} + e^y_{N+2} \quad (57)$$

$$\rightarrow y_{N+2} = \mathbf{h}_{N+1} \mathbf{x}_{N+1} + e^y_{N+2}$$

A linear recursive estimator can be written as [36]:

$$\hat{\mathbf{x}}_{N+1} = \hat{\mathbf{x}}_N + \mathbf{K}_N (y_{N+2} - \mathbf{h}_{N+1} \hat{\mathbf{x}}_N) \quad (58)$$

where \mathbf{K}_N is a gain matrix.

The mean of the estimation error $\mathbf{e}^{\hat{\mathbf{x}}}_{N+1} (= \mathbf{x}_{N+1} - \hat{\mathbf{x}}_{N+1})$ can be computed as [36]:

$$E[\mathbf{e}^{\hat{\mathbf{x}}}_{N+1}] = (\mathbf{I} - \mathbf{K}_{N+1} \mathbf{h}_{N+1}) E[\mathbf{e}^{\hat{\mathbf{x}}}_N] - \mathbf{K}_{N+1} E[e^y_{N+2}] \quad (59)$$

The covariance matrix \mathbf{P} of the estimation error $\mathbf{e}^{\hat{\mathbf{x}}}$ is:

$$\mathbf{P} = E[\mathbf{e}^{\hat{\mathbf{x}}} (\mathbf{e}^{\hat{\mathbf{x}}})^T] = \begin{bmatrix} p_1 & \cdots & p_{1,N} \\ \vdots & \ddots & \vdots \\ p_{N,1} & \cdots & p_N \end{bmatrix} \quad (60)$$

A recursive formula for the calculation of the estimation-error covariance also can be shown as [36]:

$$\mathbf{P}_{N+1} = (1 - \mathbf{K}_{N+1} \mathbf{h}_{N+1}) \mathbf{P}_N (1 - \mathbf{K}_{N+1} \mathbf{h}_{N+1})^T + \mathbf{K}_{N+1} \sigma_{N+2} (\mathbf{K}_{N+1})^T \quad (61)$$

The gain matrix can be expressed as [36]:

$$\mathbf{K}_{N+1} = \mathbf{P}_N (\mathbf{h}_{N+1})^T \left[\mathbf{h}_{N+1} \mathbf{P}_N (\mathbf{h}_{N+1})^T + \sigma_{N+2} \right]^{-1} \quad (62)$$

2. Propagation of the mean and covariance. Suppose we have the following linear discrete-time system [36]:

$$\mathbf{x}_{N+1} = \mathbf{A}_N \mathbf{x}_N + \mathbf{B}_N \mathbf{u}_N + \mathbf{e}_N^x \quad (63)$$

where \mathbf{u}_N is a known input and \mathbf{e}_N^x is Gaussian zero-mean white noise. The covariance matrix of \mathbf{e}^x can be shown as:

$$\mathbf{Q} = E \left[\mathbf{e}^x (\mathbf{e}^x)^T \right] = \begin{bmatrix} q_1 & \cdots & q_{1,N} \\ \vdots & \ddots & \vdots \\ q_{N,1} & \cdots & q_N \end{bmatrix} \quad (64)$$

If we take the expected value of both sides of (63):

$$E[\mathbf{x}_{N+1}] = E[\mathbf{A}_N \mathbf{x}_N + \mathbf{B}_N \mathbf{u}_N + \mathbf{e}_N^x] \rightarrow \bar{\mathbf{x}}_{N+1} = \mathbf{A}_N \bar{\mathbf{x}}_N + \mathbf{B}_N \mathbf{u}_N \quad (65)$$

which shows the propagation of the mean of \mathbf{x}_N [36]. The propagation of the covariance can be shown as [36]:

$$P_{\bar{\mathbf{x}}|N+1} = E \left[(\mathbf{x}_{N+1} - \bar{\mathbf{x}}_{N+1}) (\mathbf{x}_{N+1} - \bar{\mathbf{x}}_{N+1})^T \right] = \mathbf{A}_N P_{\bar{\mathbf{x}}|N} \mathbf{A}_N^T + q_N \quad (66)$$

3. Derivation of the discrete-time Kalman filter. If we have all of the measurements up to and including time $N+1$ available for our estimate of \mathbf{x}_{N+1} , then we can form a *posteriori* estimate denoted as $\hat{\mathbf{x}}_{N+1|N+1}$. If we have all of the measurements before (not including) time $N+1$ available for our estimate of \mathbf{x}_{N+1} , then we can form a *priori* estimate denoted as $\hat{\mathbf{x}}_{N+1|N}$. Similarly, $P_{N+1|N+1}$ denotes the covariance of $\hat{\mathbf{x}}_{N+1|N+1}$ and $P_{N+1|N}$ denotes the covariance of $\hat{\mathbf{x}}_{N+1|N}$. Assuming that the initial state $E[x_1]$ is given:

$$\hat{\mathbf{x}}_{1|1} = E[x_1] = \bar{\mathbf{x}}_{1|1} \quad (67)$$

Using (65) that describes the propagation of the mean of \mathbf{x}_N , we

References

- Nelson DL, Cox MM (2005) Lehninger Principles of Biochemistry. New York: W H Freeman and Company.
- Alberts B, Johnson A, Lewis J, Raff M, Roberts K, et al. (2002) Molecular Biology of the Cell. New York: Garland Science.
- Kitano H (2002) Systems biology: a brief overview. Science 295: 1662–1664.
- Hasty J, McMillen D, Isaacs F, Collins JJ (2001) Computational studies of gene regulatory networks: in numero molecular biology. Nat Rev Genet 2: 268–279.
- Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, et al. (2002) Network motifs: simple building blocks of complex networks. Science 298: 824–827.
- Iglesias PA, Ingalls BP (2009) Control Theory and Systems Biology. Cambridge: The MIT Press.
- Alon U (2007) Network motifs: theory and experimental approaches. Nat Rev Genet 8: 450–461.
- Cuccato G, Della Gatta G, di Bernardo D (2009) Systems and Synthetic biology: tackling genetic networks and complex diseases. Heredity 102: 527–532.

obtain:

$$\begin{aligned} \bar{\mathbf{x}}_{2|1} &= \mathbf{A}_1 \bar{\mathbf{x}}_{1|1} + \mathbf{B}_1 \mathbf{u}_1 \rightarrow \hat{\mathbf{x}}_{2|1} = \mathbf{A}_1 \hat{\mathbf{x}}_{1|1} + \mathbf{B}_1 \mathbf{u}_1 = \hat{\mathbf{x}}_{1|1} \\ (\mathbf{A}_1 = \mathbf{B}_1 = 1, \mathbf{u}_1 = 0) \end{aligned} \quad (68)$$

The reasoning can be extended as following:

$$\hat{\mathbf{x}}_{N+1|N} = \hat{\mathbf{x}}_{N|N} \quad (69)$$

Assuming $P_{1|1}$, the covariance of the initial estimate $\hat{\mathbf{x}}_{1|1}$, is given and using (66):

$$\begin{aligned} P_{1|1} &= P_{\bar{\mathbf{x}},1|1} \\ P_{\bar{\mathbf{x}},2|1} &= \mathbf{A}_1 P_{\bar{\mathbf{x}},1|1} \mathbf{A}_1 + q_1 \rightarrow P_{2|1} = \mathbf{A}_1 P_{1|1} \mathbf{A}_1 + q_1 = P_{1|1} + q_1 (\mathbf{A}_1 = 1) \end{aligned} \quad (70)$$

The reasoning can be extended as following:

$$P_{N+1|N} = \mathbf{A}_N P_{N|N} \mathbf{A}_N + q_N = P_{N|N} + q_N \quad (\mathbf{A}_N = 1) \quad (71)$$

Now we have the time-update equations for $\hat{\mathbf{x}}$ and P and we need the measurement-update equations for $\hat{\mathbf{x}}$ and P . Utilizing the recursive least squares development (56–62), the discrete-time Kalman filter for simple regulation can be summarized as:

$$\begin{aligned} \hat{\mathbf{x}}_{N+1|N+1} &= \hat{\mathbf{x}}_{N+1|N} + \mathbf{K}_N (y_{N+2} - h_{N+1} \hat{\mathbf{x}}_{N+1|N}) \\ \mathbf{K}_{N+1} &= P_{N+1|N} h_{N+1} \left[P_{N+1|N} (h_{N+1})^2 + \sigma_{N+2} \right]^{-1} \end{aligned} \quad (72)$$

$$P_{N+1|N+1} = P_{N+1|N} (1 - \mathbf{K}_{N+1} h_{N+1})^2 + \mathbf{K}_{N+1} \sigma_{N+2} (\mathbf{K}_{N+1})^T$$

Supporting Information

File S1 LabVIEW VI file for Case Study V.

Found at: doi:10.1371/journal.pone.0012785.s001 (0.06 MB TXT)

Movie S1 Movie file for Case Study V.

Found at: doi:10.1371/journal.pone.0012785.s002 (1.16 MB MOV)

Author Contributions

Wrote the paper: YJS LB. Conceived the modeling approach and its theoretical foundation: Y-JS. Designed the simulations: Y-JS, LB. Performed the simulations: Y-JS. Analyzed the simulation results: Y-JS, LB.

16. Bleris LG, Xie Z, Adadey A, Sontag E, Benenson Y (2009) Transcriptional feedforward motif as stable expression unit. *Systems Biology of Human Disease Conference Proceeding*. Boston.
17. Lu TK, Khalil AS, Collins JJ (2009) Next-generation synthetic gene networks. *Nat Biotechnol* 27: 1139–1150.
18. Dunlop MJ, Cox RS, 3rd, Levine JH, Murray RM, Elowitz MB (2008) Regulatory activity revealed by dynamic correlations in gene expression noise. *Nat Genet* 40: 1493–1498.
19. Raj A, van Oudenaarden A (2008) Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell* 135: 216–226.
20. Thomas R (1973) Boolean formalization of genetic control circuits. *J Theor Biol* 42: 563–585.
21. Smolen P, Baxter DA, Byrne JH (2000) Modeling transcriptional control in gene networks—methods, recent results, and future directions. *Bull Math Biol* 62: 247–292.
22. Bower JM, Bolouri H (2001) *Computational Modeling of Genetic and Biochemical Networks*. Cambridge: MIT Press.
23. Shmulevich I, Dougherty ER, Zhang W (2002) Gene perturbation and intervention in probabilistic Boolean networks. *Bioinformatics* 18: 1319–1331.
24. Goutsias J, Lee NH (2007) Computational and experimental approaches for modeling gene regulatory networks. *Curr Pharm Des* 13: 1415–1436.
25. Steggle IJ, Banks R, Shaw O, Wipat A (2007) Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach. *Bioinformatics* 23: 336–343.
26. Schilstra (2008) Bio-logic: Gene expression and the laws of combinatorial logic. *Artif Life* 14: 121.
27. Karlebach G, Shamir R (2008) Modelling and analysis of gene regulatory networks. *Nat Rev Mol Cell Biol* 9: 770–780.
28. Garg A, Di Cara A, Xenarios I, Mendoza L, De Micheli G (2008) Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics* 24: 1917–1925.
29. Bolouri H (2008) *Computational Modeling of Gene Regulatory Networks*. London: Imperial College Press.
30. Shin YJ, Nourani M (2010) Statecharts for gene network modeling. *PLoS One* 5: e9376.
31. Anderson BDO, Moore JB (1990) *Optimal Control: Linear Quadratic Methods*. Mineola: Dover.
32. Lathi BP (2002) *Linear Systems and Signals*. Oxford: Oxford University Press.
33. Schweickhardt T, Allgöwer F (2004) *Quantitative nonlinearity assessment – An introduction to nonlinearity measurement*. Amsterdam: Elsevier Science.
34. Strang G (2007) *Computational Science and Engineering*. Wellesley: Wellesley-Cambridge Press.
35. Stengel RF (1994) *Optimal Control and Estimation*. Mineola: Dover.
36. Simon D (2006) *Optimal State Estimation*. New York: John Wiley & Sons.
37. Skogestad S, Postlethwaite I (2005) *Multivariable Feedback Control*. New York: John Wiley & Sons.
38. Iglesias PA, Ingalls BP (2010) *Control Theory and Systems Biology*. Cambridge: MIT Press.
39. De Jong H, Gouze JL, Hernandez C, Page M, Sari T, et al. (2004) Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull Math Biol* 66: 301–340.
40. Casey R, de Jong H, Gouze JL (2006) Piecewise-linear models of genetic regulatory networks: equilibria and their stability. *J Math Biol* 52: 27–56.
41. Polynikis A, Hogan SJ, di Bernardo M (2009) Comparing different ODE modelling approaches for gene regulatory networks. *J Theor Biol* 261: 511–530.
42. Nise NS (2000) *Control Systems Engineering*. New York: John Wiley & Sons.
43. Simpson ML, Cox CD, Sayler GS (2003) Frequency domain analysis of noise in autoregulated gene circuits. *Proc Natl Acad Sci U S A* 100: 4551–4556.
44. Cox CD, McCollum JM, Austin DW, Allen MS, Dar RD, et al. (2006) Frequency domain analysis of noise in simple gene circuits. *Chaos* 16: 026102.
45. Austin DW, Allen MS, McCollum JM, Dar RD, Wilgus JR, et al. (2006) Gene network shaping of inherent noise spectra. *Nature* 439: 608–611.
46. Becskei A, Serrano L (2000) Engineering stability in gene networks by autoregulation. *Nature* 405: 590–593.
47. Stricker J, Cookson S, Bennett MR, Mather WH, Tsimring LS, et al. (2008) A fast, robust and tunable synthetic gene oscillator. *Nature* 456: 516–519.
48. Tsai TY, Choi YS, Ma W, Pomeroy JR, Tang C, et al. (2008) Robust, tunable biological oscillations from interlinked positive and negative feedback loops. *Science* 321: 126–129.
49. Tigges M, Marquez-Lago TT, Stelling J, Fussenegger M (2009) A tunable synthetic mammalian oscillator. *Nature* 457: 309–312.
50. Marucci L, Barton DA, Cantone I, Ricci MA, Cosma MP, et al. (2009) How to turn a genetic circuit into a synthetic tunable oscillator, or a bistable switch. *PLoS One* 4: e8083.
51. Eichenberger P, Fujita M, Jensen ST, Conlon EM, Rudner DZ, et al. (2004) The program of gene transcription for a single differentiating cell type during sporulation in *Bacillus subtilis*. *PLoS Biol* 2: e328.
52. Mangan S, Alon U (2003) Structure and function of the feed-forward loop network motif. *Proc Natl Acad Sci U S A* 100: 11980–11985.
53. Brogan WL (1991) *Modern Control Theory*. Englewood Cliffs New Jersey: Prentice-Hall.
54. Luenberger DG (1979) *Introduction to Dynamic Systems*. New York: John Wiley & Sons.
55. Suel GM, Garcia-Ojalvo J, Liberman LM, Elowitz MB (2006) An excitable gene regulatory circuit induces transient cellular differentiation. *Nature* 440: 545–550.
56. Dunlop MJ, Cox RS, 3rd, Levine JH, Murray RM, Elowitz MB (2008) Regulatory activity revealed by dynamic correlations in gene expression noise. *Nat Genet* 40: 1493–1498.
57. Locke JC, Elowitz MB (2009) Using movies to analyse gene circuit dynamics in single cells. *Nat Rev Microbiol* 7: 383–392.
58. Raser JM, O’Shea EK (2005) Noise in gene expression: origins, consequences, and control. *Science* 309: 2010–2013.
59. Swain PS, Elowitz MB, Siggia ED (2002) Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc Natl Acad Sci U S A* 99: 12795–12800.