PARAMETER ESTIMATION OF OSCILLATORY SYSTEMS

(WITH APPLICATION TO CIRCADIAN RHYTHMS)

ANG KOK SIONG

NATIONAL UNIVERSITY OF SINGAPORE

2009

PARAMETER ESTIMATION OF OSCILLATORY SYSTEMS

(WITH APPLICATION TO CIRCADIAN RHYTHMS)

ANG KOK SIONG
(*B.Eng.*(*Hons.*), *NUS*)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF CHEMICAL AND
BIOMOLECULAR ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2009

Acknowledgements

The author wishes to thank Dr. Rudiyanto Gunawan for his guidance and as a source of inspiration and role model for research. The development of the author's knowledge and skills in research would not have been possible without Dr. Gunawan. For this, his input and advice during the project is gratefully acknowledged.

The author would also like to thank the members of the Gunawan group for the camaraderie and fruitful discussions on various topics. Many thanks as well to the rest of the process systems engineering community in NUS for the friendship and intellectually stimulating atmosphere. A few of the senior research students in the department have particularly helpful to the author with their highly appreciated suggestions.

Finally, special thanks to an old friend and fellow graduate student who understands pains of research and has the ability to laugh at it.

Contents

Summary

Many important biological systems are known to exhibit oscillatory behavior, with examples such as the cell cycle and circadian rhythms. Consequently, mathematical models are built to study system properties like stability, robustness and stability. A review of the literature shows that parameter estimation techniques are rarely employed when building most models of oscillatory systems. Instead, model parameters are often arbitrarily chosen to yield desired qualitative behavior. Unfortunately, this may lead to misleading conclusions from the analysis of the model. Therefore, the purpose of this work is to study the problem of parameter estimation for oscillatory systems.

The output of oscillatory systems exhibits two characteristics, shape (state trajectory) and periodicity, while typical non-oscillatory systems only possess shape. The periodicity property also results in the unbounded increase of sensitivity coefficients with time. As a result, application of traditional gradient-based methods is not feasible.

In this work, the effect of shape and periodicity was decoupled and a suitable objective function using maximum likelihood estimation was derived. Due to the nature of the solution space, a stochastic global optimizer was selected as the search algorithm. An alternate approach using maximum a posteriori estimation by combining Phase Response Curve data with time series data was also investigated. The developed methodology was tested on three circadian rhythm models and its effectiveness was clearly shown in the results obtained.

Keywords: Parameter estimation, oscillator, identifiability analysis

List of Tables

List of Figures

List of Symbols

c	Equality constraint function
Δx	Small finite change in x
η	Isochron
f	Function
g	Output function
Γ	Green's function
γ	Scaling factor between 0 and 1
$\Gamma^{\dagger}$	Adjoint Green's function
h	Inequality constraint function
$\mathbf{I}$	Identity matrix
$\mathbf{J}$	Jacobian matrix
l	Lower bound of parameter
λ	Damping factor in Levenberg-Marquardt
$\mathcal{L}$	Likelihood function
m	Number of system parameters
μ	Distribution mean
N	Total number of measurements
n	Number of system states
n_c	Number of equality constraints
N_{CPU}	Number of processors
n_h	Number of inequality constraints
n_p	Number of bounded parameters
p	System parameter
P_{code}	Proportion of parallelizable code

Φ Objective function

ϕ Phase

Q Phase sensitivity

s Local sensitivity coefficient

σ Standard deviation

τ Period

θ Duration of finite pulse perturbation

t Time

u Upper bound of parameter

v_τ Variance of period

$\mathbf{V}$ Covariance matrix

x System State

y Observable system state

Differential Evolution

$\mathbf{b}_\mathrm{u}$ Upper bounds of parameters

$\mathbf{b}_\mathrm{l}$ Lower bounds of parameters

Cr Crossover factor

D Length of solution vectors

F Mutation factor

λ Tuning parameter for /target-to-best/ variant of DE

N_p Size of solution population

$\mathbf{P_u}$ Population of trial solution vectors $\mathbf{u}$

$\mathbf{P_v}$ Population of mutant solution vectors $\mathbf{v}$

$\mathbf{P_x}$ Population of current solution vectors $\mathbf{x}$

$\mathbf{u}$ Trial solution vector

$\mathbf{v}$ Mutant solution vector

$\mathbf{x}$ Current solution vector

2-state Tyson Model

J_p Michaelis constant for protein kinase (DBT)

K_eq Equilibrium constant for dimerization

k_{m} First-order rate constant for mRNA degradation

k_{p1} Maximum rate for monomer phosphorylation

k_{p2} Maximum rate for dimer phosphorylation

k_{p3} First-order rate constant for proteolysis

M mRNA concentration

P_{crit} Dimer concentration at the half-maximum transcription rate

P_{t} Total protein concentration

q Algebraic expression

v_{m} Maximum rate of synthesis of mRNA

v_{p} Rate constant for translation of mRNA

5-state Goldbeter Model

K_1 Michaelis constant of forward reaction in first phosphorylation reaction

k_1 Rate constant for transportation of bi-phosphorylated PER protein from cytosol into the nucleus

K_2 Michaelis constant for backward reaction in first phosphorylation reaction

k_2 Rate constant for transportation of bi-phosphorylated PER protein from nucleus into the cytosol

K_3 Michaelis constant for forward reaction in second phosphorylation reaction

K_4 Michaelis constant for backward reaction in second phosphorylation reaction

K_{D} Michaelis constant for degradation of bi-phosphorylated PER protein

K_{I} Threshold constant for repression

K_{m} Michaelis constant for degradation of PER mRNA

k_{s} Synthesis rate for PER protein

M PER mRNA concentration

n Hill parameter for degree of cooperativity

P_0 Non-phosphorylated PER protein concentration

P_1 Mono-phosphorylated PER protein concentration

P_2 Bi-phosphorylated PER protein concentration

P_{N} Bi-phosphorylated PER protein concentration in nucleus

V_1 Maximum rate for forward reaction first in phosphorylation reaction

V_2 Maximum rate for backward reaction first in phosphorylation reaction

V_3 Maximum rate for forward reaction second in phosphorylation reaction

V_4 Maximum rate for backward reaction second in phosphorylation reaction

v_{d} Maximum rate for degradation of bi-phosphorylated PER protein

v_{m} Maximum rate for degradation of PER mRNA

v_{s} Maximum accumulation rate of PER mRNA in the cytosol

10-state Goldbeter Model

C PER-TIM dimer concentration

C_{N} PER-TIM dimer concentration in nucleus

k_1 Transportation rate of PER-TIM dimer from cytosol to nucleus

$K_{1\mathrm{P}}$ Michaelis constant for forward reaction in first phosphorylation reaction of PER protein

$K_{1\mathrm{T}}$ Michaelis constant for forward reaction in first phosphorylation reaction of TIM protein

k_2 Transportation rate of PER-TIM dimer from nucleus to cytosol

$K_{2\mathrm{P}}$ Michaelis constant for backward reaction in first phosphorylation reaction of PER protein

$K_{2\mathrm{T}}$ Michaelis constant for backward reaction in first phosphorylation reaction of TIM protein

k_3 Rate of PER and TIM proteins association to form dimer

$K_{3\mathrm{P}}$ Michaelis constant for forward reaction in second phosphorylation reaction of PER protein

$K_{3\mathrm{T}}$ Michaelis constant for forward reaction in second phosphorylation reaction of TIM protein

k_4 Rate of PER-TIM dimer disassociation

$K_{4\mathrm{P}}$ Michaelis constant for backward reaction in second phosphorylation reaction of PER protein

$K_{4\mathrm{T}}$ Michaelis constant for backward reaction in second phosphorylation reaction of TIM protein

k_{d} Non-specific degradation rate of mRNAs and proteins

k_{dC} Degradation rate for PER-TIM dimer in cytosol

k_{dN} Degradation rate for PER-TIM dimer in nucleus

K_{dP} Michaelis constant of biphosphorylated PER protein degradation in cytosol

K_{dT} Michaelis constant of biphosphorylated PER protein degradation in cytosol

K_{IP} Threshold constant for repression on PER mRNA synthesis by dimer

K_{IT} Threshold constant for repression on TIM mRNA synthesis by dimer

K_{mP} Michaelis constant of PER mRNA degradation in cytosol

K_{mT} Michaelis constant of TIM mRNA degradation in cytosol

k_{sP} Rate of PER protein synthesis

k_{sT} Rate of TIM protein synthesis

M_{p} PER mRNA concentration

M_{T} TIM mRNA concentration

n Hill parameter for degree of cooperativity

P_1 Non-phosphorylated PER protein concentration

P_1 Monophosphorylated PER protein concentration

P_2 Biphosphorylated PER protein concentration

T_0 Non-phosphorylated TIM protein concentration

T_1 Monophosphorylated TIM protein concentration

T_2 Biphosphorylated TIM protein concentration

$V_{1\mathrm{P}}$ Maximum rate constant for forward reaction in first phosphorylation reaction of PER protein

$V_{1\mathrm{T}}$ Maximum rate for forward reaction in first phosphorylation reaction of TIM protein

$V_{2\mathrm{P}}$ Maximum rate constant for backward reaction in first phosphorylation reaction of PER protein

$V_{2\mathrm{T}}$ Maximum rate for backward reaction in first phosphorylation reaction of TIM protein

$V_{3\mathrm{P}}$ Maximum rate for forward reaction in second phosphorylation reaction of PER protein

$V_{3\mathrm{T}}$ Maximum rate for forward reaction in second phosphorylation reaction of TIM protein

$V_{4\mathrm{P}}$ Maximum rate for backward reaction in second phosphorylation reaction of PER protein

$V_{4\mathrm{T}}$ Maximum rate for backward reaction in second phosphorylation reaction of TIM protein

v_{dP} Maximum rate of biphosphorylated PER protein degradation in cytosol

v_{dT} Maximum rate of biphosphorylated TIM protein degradation in cytosol

v_{mP} Maximum rate of PER mRNA degradation in cytosol

v_{mT} Maximum rate of TIM mRNA degradation in cytosol

v_{sP} Accumulation rate of PER mRNA in cytosol

v_{sT} Accumulation rate of TIM mRNA in cytosol

Chapter 1

Introduction

In the study of natural phenomena, mathematical models are often created for analysis purposes to gain insights on system properties such as stability, robustness and parametric sensitivity, and their predictive powers used for systems design and to guide further experiments. The model building process for dynamical systems is composed of iterative steps that include the specification of model structure and equations, identifiability analysis, experimental design, execution of experiments, parameter estimation, and model invalidation [1]. This thesis concerns the parameter estimation step [2]. A top-down approach of prescribing model parameters $\mathbf{p}$ is to fit the model output $\mathbf{y}$ to available experimental data $\hat{\mathbf{y}}$ in a process called parameter estimation. An objective function Φ such as the sum of errors squared is often selected to measure the goodness of fit:

$$\Phi(\mathbf{p}) = \sum_{i=1}^{N} \Big(\hat{y}_i - y_i(\mathbf{p})\Big)^2, \tag{1.1}$$

and a search algorithm is used to obtain the set of parameters $\hat{\mathbf{p}}$ that minimize the objective function $\Phi(\mathbf{p})$, *i.e.*

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \Phi(\mathbf{p}). \tag{1.2}$$

Presently, parameter estimation theory and techniques have matured and are regularly used in many areas of science and engineering [3, 4].

In biology, model construction and parameter estimation are also commonly employed. As in the use of models in physics and engineering, analysis of biological models enables greater understanding of cellular and organism behavior, and more recently, the use of models to guide drug development [5, 6]. The systems approach to biology, called Systems Biology [7, 8], has been taken up in the recent years to deal with the complexities inherent to biological systems, made possible by the explosion of biological data resulting from technological advances in the past decade and the continued growth in computing power. Instead of the reductionist approach of viewing genes, proteins and other metabolites, these components are now studied as an integrated system of interacting parts of a network, in parallel to the systems approach used in engineering. Tools routinely used in other scientific disciplines and engineering have found new applications, sometimes appropriately modified, to study biological systems. The usage of such analysis tools can produce non-intuitive insights that are not possible with a simple inspection of reaction networks.

Unfortunately, the main obstacle to building models in such a quantitative manner is the quality of data available. Experimental data from biological experiments suffers a variety of problems, including significant measurement noise, inherent stochastic nature of the process, missing or incomplete data and unknown components, all of which complicate parameter estimation.

Within biology, a number of important systems exhibit rhythmic behavior, including the cell cycle [9], circadian rhythms [10], glycolysis [11] and cyclic AMP production [12]. Although mathematical models of these systems have been constructed, parameter estimation techniques were not routinely applied. While some kinetic parameters are available from independent or direct measurements, the vast majority are not. Instead, the parameters are often tweaked ad-hoc such that the model outputs match qualitative features of experimental data.

In this work, the models used in parameter estimation are drawn from the study of circadian rhythms. The following section will introduce the biology of circadian rhythms to serve as background information.

1.1 Circadian Rhythms

Circadian rhythms are approximately 24 hour cycles which regulate physiology, biochemistry and behavior of most living organisms. In humans, the rhythm is most obvious in the sleep-wake cycle. The rhythms are controlled by a circadian oscillator that is endogenous but also responsive to external cues such as light, and can entrain the rhythms to the local environment.

1.1.1 Structure and Characteristics

An important milestone in the molecular biology of circadian rhythms is the discovery of the *Period* gene in the fruit fly *Drosophila melanogaster* by Konopka and Benzer using mutant screens, and thus establishing the role of genes in the circadian clock [13]. Subsequent studies identified similar clock genes and proteins (homologues) in other living organisms. Experimental evidence to date show that circadian clocks such as those found in *Drosophila*, *Neurospora* and mammals are based on transcriptional-translational feedback loops, involving coupled positive and negative feedback [14, 15].

The three main characteristics displayed by circadian oscillators are: an approximately 24 hour period, entrainment to the environment, and temperature compensation [16, 17]. In particular, entrainment is of relevance to the generation of Phase Response Curve (PRC), a commonly used analysis to study the phase behavior of circadian rhythms. Since the Free Running Period (FRP) of the circadian clocks is not exactly 24 hours, the rhythms need to be reset daily to maintain synchrony with the environment. Some of the known resetting cues of circadian oscillators include light, ambient temperature, feeding and physical activities [16]. However, circadian response to these cues is not uniform over the cycle [16]. Depending on the timing, the cue may produce a phase advance, a phase delay or virtually no phase shift. Plotting the resulting phase shift over the phase of the circadian rhythm at which the cue was given produces a PRC. Figure 1.1 shows the PRC obtained from the *Drosophila* in response to light pulses. Examples of PRCs for different organisms can also be found in the PRC

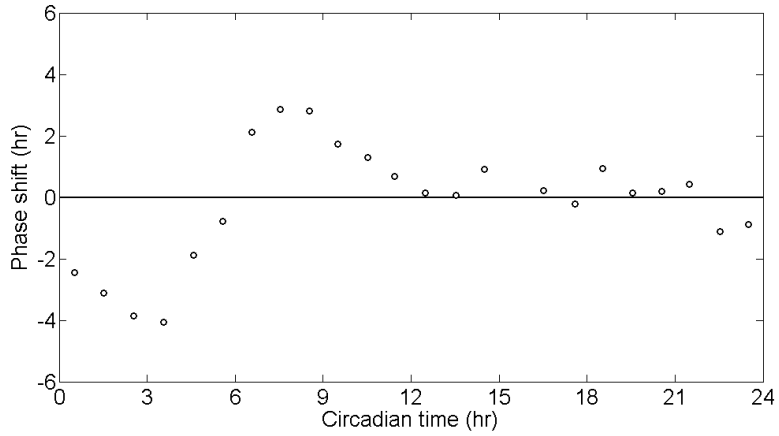


Figure 1.1: PRC obtained for the *Drosophila melanogaster* using 1 min light pulses (Adapted from Hall and Rosbash [19]). A positive phase shift is phase advance and a negative phase shift is phase delay.

Atlas compiled by Johnson [18].

1.1.2 *Drosophila melanogaster*

The fruit fly *Drosophila melanogaster* is one of the model organisms commonly used in biological studies. Its popularity primarily stems from its small size, short lifespan, ease in maintaining a large population, and the knowledge accumulated from the long history of use. In 2000, sequencing of the *Drosophila melanogaster* genome was completed [20].

Figure 1.2 shows a simplified *Drosophila* diagram of the circadian clock mechanism. The core of the clock consists of 2 interlocking feedback loops, the first consisting of PER (period) and TIM (timeless) and the second composed of CLK (clock), VRI (vrille) and PDP1 (PAR-domain protein 1) [14, 21, 22]. Both loops are connected due to interaction via CLK.

In the PER-TIM loop, CLK and CYC (cycle) form a complex that activates *per* and *tim* transcriptions. By the start of evening, both *per* and *tim* mRNA levels reach their maximum while their protein levels only peak $4 \sim 6$ hours later [23]. This delay is attributed to the phosphorylation-induced destabilization of PER when bound to DBT (double-time) [24]. Stabilization of PER by binding with TIM allows it to translocate into the nucleus, but PER and TIM have also

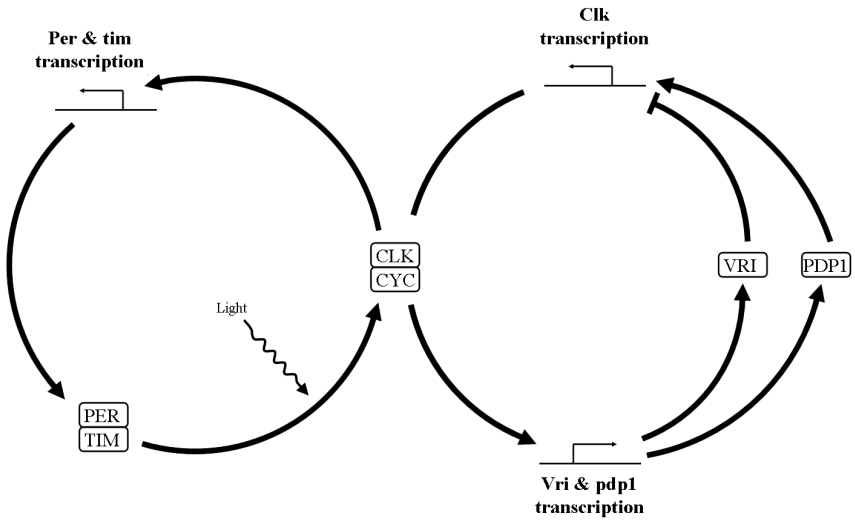


Figure 1.2: Simple schematic of the *Drosophila melanogaster* circadian clock.

been observed to translocate separately and re-associate in the nucleus [25]. The PER-TIM complex level builds up during the night in the nucleus. The complex binds to CLK and inhibit transcription of *per* and *tim* [26]. Coupled with the inhibition, PER and TIM levels are lowered by phosphorylation induced degradation of PER, and degradation of TIM by CRY (cryptochrome) [27]. This CRY dependent degradation is particularly important as it enables entrainment with the external environment through light. DBT also binds to CLK and induces phosphorylation for degradation. However, this does not mean that the overall CLK protein levels cycle in phase with PER and TIM as hypophosphorylated CLK accumulates from new synthesis or dephosphorylation. By noon the next day, both proteins are at their lowest levels and CLK can again activate PER and TIM transcriptions, starting the cycle anew.

In the other loop, the transcriptions of VRI and PDP1 are promoted by CLK while PDP1 promotes the transcription of CLK. At noon, CLK induces VRI and PDP1 transcriptions. VRI protein level increases more rapidly than PDP1 and represses the transcription of CLK by competitively binding to PDP1 in the evening. By night, PDP1 levels exceed VRI and reactivates *clk* transcription. This leads to *clk* mRNA cycling in an opposite phase with the other mRNA levels (*per, tim, vrille, pdp*) [28]. However, this mRNA cycling does not affect the total

CLK protein level, though hyperphosphorylated and hypophosphorylated CLK are known to accumulate in anti-phase with one another [29].

The function of the second (CLK) feedback loop is presently not yet well understood. A single negative feedback loop with delay is sufficient for generating oscillations and mathematical models of the circadian clock such as the *Drosophila* had been modeled with only the PER-TIM feedback loop [30–33]. The time delay may take the form of an explicit delay in the equations or a series of intermediate species. Due to this time delay, the system repeatedly undershoots or overshoots the steady state, thus generating oscillations. Alternatively, some oscillatory models incorporate positive feedback to introduce hysteresis into the system, preventing it from reaching a steady state. Removing the positive feedback loop will naturally abolish the oscillations. For models that do not rely on positive feedback to generate sustained oscillations, it was hypothesized that the additional loop increases the system's robustness to parameter perturbations although this was not supported by simulation studies of models by Smolen *et al.* [34, 35]. More experimental evidence will be needed to shed further light on the second feedback loop.

1.2 Thesis Aim

The purpose of this work is to investigate parameter estimation in oscillatory systems. A methodology was developed to estimate the model parameters from time-series oscillatory data. Although the circadian rhythm models of the *Drosophila melanogaster* were used as case studies, the methodology is generic and applicable to general oscillatory systems. The confidence intervals of the parameter estimates were subsequently computed using the Fisher Information Matrix (FIM) to determine practical identifiability of the parameters.

The effect of noise and sampling time on parametric identifiability was also studied. This is useful in guiding lab experiments on the decision of noise reduction with repeated samplings or reducing sampling time with more samples.

Finally, the possibility of using Phase Response Curve (PRC) of circadian

rhythms for parameter estimation was investigated. This was motivated by the abundance of PRC data from numerous circadian rhythm studies over the years and greater accessibility than time series mRNA and protein data.

1.3 Thesis Organization

The thesis is organized as follows: Chapter 2 explains the basic concepts behind parameter estimation and briefly surveys the current parameter search methods available. Chapter 3 discusses sensitivity analysis of oscillatory systems. The problem formulation and development of the parameter estimation methodology are explained in Chapter 4. Chapter 5 presents the results from the case studies. The work is then concluded in Chapter 6.

Chapter 2

Parameter Estimation

This chapter gives a short introduction to parameter estimation, reviews search algorithms available, and summarizes past works on parameter estimation of circadian systems. The problem statement is restated and relevant concepts of parameter estimation are first discussed in Section 2.1. A brief survey on the popular parameter search methods available is covered in Section 2.2. In Section 2.3, recent works on parameter estimation of oscillatory systems are reviewed.

2.1 Problem statement

In parameter estimation, problem formulation requires selection of a suitable objective function as a measure of the goodness of fit. The ordinary least squares estimator is commonly employed, as well as other approaches such as maximum likelihood and Bayesian maximum a posteriori [3]. The objective function for ordinary least squares is

$$\Phi(\mathbf{p}) = \sum_{i=1}^{N} \Big(\hat{y}_i - y_i(\mathbf{p})\Big)^2 \tag{2.1}$$

where $\mathbf{p}$ is the vector of model parameters, $\hat{y}$ are the measurements, and $y(\mathbf{p})$ is the model output. There are alternatives such as an observer-based approach [36], or the Belief Propagation [37] method that produces probability distributions for the parameters as opposed to point estimates.

Nonlinear parameter estimation problems can be considered a subset of general Non-Linear Programming (NLP) problems and may be stated in the following manner:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \Phi(\mathbf{p}) \tag{2.2}$$

subject to

$$h_i(\mathbf{p}) < 0 \qquad i = 1, ..., n_h \tag{2.3}$$

$$c_j(\mathbf{p}) = 0 \qquad j = 1, ..., n_c \tag{2.4}$$

$$l_k \leq \mathbf{p} \leq u_k \qquad k = 1, ..., n_p. \tag{2.5}$$

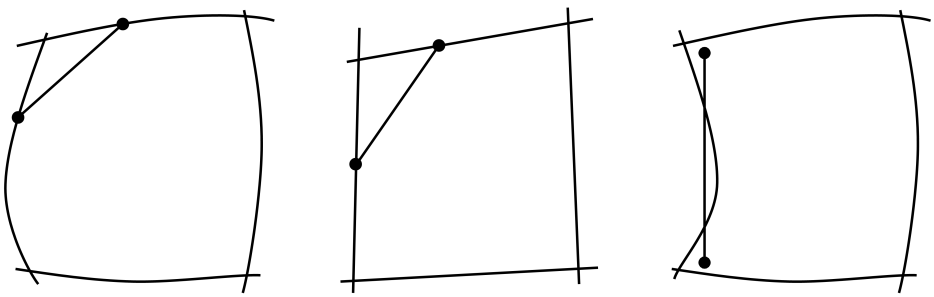
Here, $\hat{\mathbf{p}}$ is the vector of n_p parameter estimates that minimize the objective function $\Phi(\mathbf{p})$ subject to n_h inequality and n_c equality constraint functions. The variables l_k and u_k are lower and upper bounds specified on the parameter estimates, respectively. The space defined by the constraints and bounds is called the feasible region. Simple parameter estimation problems often have no constraints, but bounds on the parameter estimates are often applied to ensure that realistic estimates are obtained. This is particularly true for kinetic parameters of irreversible (bio)chemical reaction which cannot be negative by definition.

2.1.1 Convexity and Multiple Optima

Convexity is an important concept in optimization and is useful in understanding local and global optima. The presence of multiple local optima in the feasible region has an impact on the choice of optimization algorithms used.

Convex Set and Function

A convex set is defined as a set of points in n-dimensional space where all pairs of points can be joined by a straight line that is also entirely within the set. The concept is illustrated in Figure 2.1 for two dimensions. Similarly, Figure



(a) Convex set.

(b) Convex set.

(c) Nonconvex set. Note: not all of the line segment joining the two points is within the set

Figure 2.1: 2D convex and nonconvex sets. Adapted from Edgar *et al.* [38].

2.2 illustrates the concept of convexity for a single variable function. A function $f(\mathbf{x})$ defined on a convex set is said to be a convex function if the following holds:

$$f[\gamma \mathbf{x}_1 + (1-\gamma)\mathbf{x}_2] \leq \gamma f(\mathbf{x}_1) + (1-\gamma) f(\mathbf{x}_2) \tag{2.6}$$

for any two points $\mathbf{x}_1$ and $\mathbf{x}_2$, and γ is a scalar in the range $0 \leq \gamma \leq 1$ [38]. The function is strictly convex only if the strict inequality holds [38].

If the objective function is convex within the convex feasible region, the resulting problem is a convex programming problem where only a single optimum exists. This optimal point for the entire feasible region is called the *global* optimum. In contrast, for a nonconvex optimization, a point can be the optimum only for a neighborhood around the point and is referred to as a *local* optimum. For many problems, the feasible region is non-convex and contains multiple local optima. Within this set of local optima, the member with the smallest function score is the global optimum for the entire feasible region. Figure 2.3 shows a single variable function with three local optima (A, B, C) of which B is the global optimum.

2.2 Optimization Methods

For general nonlinear parameter estimation problems, closed form solutions do not exist and optimization algorithms are needed to solve for the parameter es-

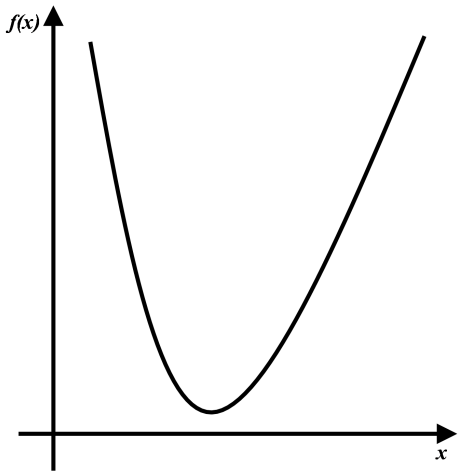


Figure 2.2: A convex single variable function

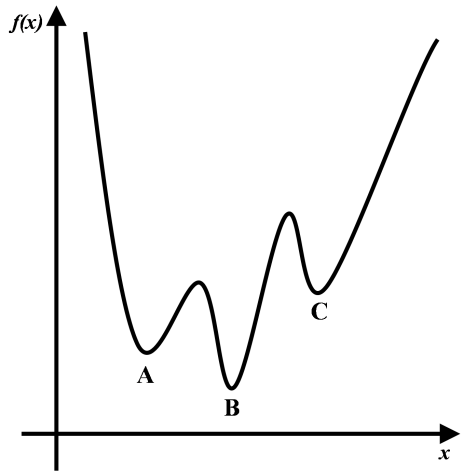


Figure 2.3: Multiple optima in a single variable function

timates. A variety of optimization methods are available to compute the best parameter estimates and the choice of the most appropriate algorithm is dependent on the nature of the problem.

All optimization algorithms start with an initial starting guess (*e.g.* derivative based methods) or multiple guesses (*e.g.* stochastic search methods), and iteratively improve the solution(s) until a termination criterion is satisfied. To improve a solution, local search methods utilize only local information from its neighborhood (*e.g.* gradient for derivative-based methods) to search for a better solution. On the other hand, global methods utilize information from the entire solution space to improve current solution(s). Within local and global classes, the methods can be further subdivided.

2.2.1 Local Search

Derivative-based methods are extremely popular in solving NLP problems due to their computation efficiency and mathematical proofs of convergence. These methods rely on the first order derivative (gradient) or even second order derivative (Hessian) information to determine the direction taken for the search step. For nonlinear parameter estimation problems, derivative-based methods such as Gauss-Newton and Levenberg-Marquardt [4] for least squares problems are

usually very efficient in terms of number of iterative steps. The Gauss-Newton method approximates the Hessian matrix used in the Newton method with $\mathbf{J}^{\mathrm{T}}\mathbf{J}$ where $\mathbf{J}$ is the Jacobian of the model. The Levenberg-Marquardt method further modifies the Hessian matrix approximation with an additional $\lambda\mathbf{I}$ term, where λ is a non-negative damping factor and $\mathbf{I}$ is the identity matrix. The damping factor may be modified during each iteration to adjust the speed of convergence.

Since the parameter estimation problem also falls into the general class of NLP problems in optimization, various NLP algorithms can be used as well. NLP algorithms are divided into derivative-based and direct search methods. For derivative-based methods, gradient descent, Newton's method and Quasi-Newton methods fall into this category. The first utilizes gradient information while the second incorporates the Hessian matrix as well. Due to the difficulty of computing the Hessian, Quasi-Newton methods use various techniques to approximate the Hessian matrix. When there are constraints to be satisfied, methods available include Successive Linear Programming (SLP), Successive Quadratic Programming (SQP) and Generalized Reduced Gradient (GRG) [38]. While these derivative-based methods are usually efficient in the number of iterations, the overall speed is dependent on the computation cost of accurate gradient values.

The most popular direct search method is the classic Nelder-Mead simplex method [39]. The basic Nelder-Mead algorithm searches for the optimum by first creating a simplex of $n+1$ vertices in the n-dimensional solution space and replacing the worst vertex with a better point reflected through the centroid of the other n vertices. More sophisticated enhancements allow the simplex to adaptively expand or shrink during the search. Implementations of the algorithm can be found in a large number of software platforms and libraries such as MATLAB [40], Mathematica [41], COPASI (successor to Gepasi) [42] and Systems Biology Workbench [43].

Another well known direct search method is the Hooke and Jeeves pattern search [44]. With an initial starting solution or base solution, an exploratory

search is executed by perturbing the base solution along search directions that span the solution space. The base solution is replaced if a superior solution is found and a subsequent pattern move is made in the direction of the earlier successful exploratory search. If the exploratory search fails, the magnitude of the search perturbations is reduced and another exploratory search is executed. The algorithm is implemented in software packages such as LANCELOT [45] and COPASI [42].

As mentioned earlier, local search methods rely only on information from the neighborhood of the current solution. Using gradient information, derivative-based methods will search "downhill" for a minimum solution. If successful convergence is achieved, the converged solution is the optimum of the subregion containing the initial guess. For nonconvex problems, depending on the initial guess, the final solution is not guaranteed to be the global optimum. In general, local direct search methods are "greedy" by making locally optimal choices and thus suffer from the same drawback as derivative-based methods. However, direct search methods may be modified to have the ability to escape from local optima and this gives them the ability to better explore the solution space. Nevertheless, there is still no guarantee that the global optimum will be found.

2.2.2 Global Search

Though many challenges remain, research in the field of global optimization has seen much progress in the recent decades, with many examples of successful applications [46–48]. This is coupled with advances in computing power that allow the methods developed to be applied to practical problems of realistic size.

The main advantage of global methods is their ability to handle nonconvex problems better than local methods. As mentioned earlier, global methods use information from the entire solution space to improve on current solution(s), and thus search the entire solution space more effectively. However, the ability to effectively search for the global optimum necessitates a much heavier computational load. Most stochastic methods employ a population of solutions to explore

the entire solution space, while deterministic methods divide the solution space into subregions for investigation. In contrast, local methods only explore a single convex region in a "downhill" manner for the local optimum. Within a convex region, local methods are far more efficient than global stochastic methods in reaching the optimum.

Another drawback of global methods is the difficulty in implementation as compared to local methods. While a number of stochastic methods such as the Evolutionary Algorithms [48] (see below) are often touted as easy to apply [49], much effort can be expended in "tuning" the algorithm for a particular problem in order to obtain satisfactory results. For deterministic methods, specification of convex envelopes has a huge impact on the chosen algorithm's performance (this is further discussed below).

The global optimization methods currently in use can be divided into deterministic and stochastic methods. Deterministic methods are more rigorous and convergence proofs exist for certain problem classes, while this is not the case for stochastic methods. However, stochastic methods are comparatively easier to implement and remain popular.

Deterministic Methods

A number of deterministic methods are available [50], but the most efficient ones are based on spatial branch and bound (BB) methods. The BB method was originally developed by Land and Doig in 1960 [51] for Linear Programming but it can also be applied to nonconvex Nonlinear Programming (NLP) through a reformulation of the problem. For NLP problems, convex envelopes or underestimators are first used to approximate the solution space, thus creating a convex Mixed Integer Nonlinear Programming (MINLP) problem. This is then solved using derivative-based NLP methods for the subproblems and BB methods for the global Mixed Integer problem. While convergence proofs of BB methods for certain problem classes exist, the search tree is not guaranteed to be finite. If the underestimating functions are not suitable, the search will become an ex-

haustive enumeration of the solution space and the resulting computation cost is prohibitive. In the past two decades, software packages offering implementations of BB method, such as BARON [52] and αBB [53], have been developed and the number of successful applications is growing [54].

Stochastic Methods

In the earlier discussion of local search methods, it was noted that derivative-based methods are computationally efficient for convex problems. One straightforward method to avoid local optima is to employ multiple starting points with local NLP solvers [38]. However, using naive and randomly chosen starting points tend to result in multiple convergence to identical local optima and consequently result in an inefficient search. To improve efficiency, the Multilevel Single Linkage method was proposed by Rinnooy Kan and Timmer [55]. The algorithm iteratively generates randomly sampled points and selects a fraction of these points based on objective function score and proximity to one another, as well as previous solutions for improvement with local NLP algorithms.

Metaheuristic methods belong to a popular class of stochastic methods used for optimization. These methods are stochastic in nature, incorporating probabilistic elements in the generation of new solutions. Interestingly, many of these algorithms are based on various real-life phenomena (evolution, physical phenomena, behavior of organisms, *etc.*) or a combination of heuristic rules for geometric exploration of the solution space and are designed to avoid local optima. The objective function is usually treated as a black-box function, thus allowing the methods to be applied to different problem classes with minimal modifications. Combining such flexibility with relatively simpler implementation effort compared to deterministic methods, metaheuristics are often a practical choice. With the ability to avoid local optima, they usually produce better solutions compared to local methods which are reliant on a good initial guess for success [56]. Metaheuristics can also obtain the global optimum, although there is no guarantee. In some applications, a time consuming search for the global op-

timum solution is not necessary but good, suboptimal solutions obtained within a much shorter time frame are preferred.

Metaheuristic search methods include the well-known Genetic Algorithm (GA) [57] and Evolutionary Strategy (ES) [58], both classified under Evolutionary Algorithms (EA) [48]. These algorithms can be characterized as population based stochastic optimizers that rely on evolutionary-inspired processes such as crossover and mutation to generate fresh solutions during each iteration to update the current population.

Another class of metaheuristics is the Swarm Intelligence class of algorithms with examples like Particle Swarm [59] and Ant Colony [60]. These algorithms are based on the collective behavior of a large group of individual organisms (or agents in Artificial Intelligence research). The movement of individuals across the solution space during the search is guided by individual records of good solutions encountered previously and group knowledge that is facilitated by communications between individuals.

Outside of these two major classes of algorithms, there are other population based stochastic optimizers such as Differential Evolution [47] and Scatter Search [61]. Differential Evolution (DE) is a population based stochastic optimizer that bears many similarities to other EA algorithms such as GA and ES, although it is not always classified as an EA. The algorithm was originally developed by Price and Storn in 1995 [62] to solve the Chebyshev polynomial fitting problem but has since evolved into its current form of a versatile and popular optimization algorithm [47]. Unlike Genetic Algorithm which operates on bit strings, DE operates on real numbers, making it particularly suited for nonlinear optimization.

The defining characteristic of DE is its unique method of generating new solution vectors by perturbing each existing solution with a scaled difference of two other randomly selected solutions. Another differing characteristic is the application of selection pressure. EAs usually place selection pressure by only selecting superior parents to generate new solutions while in DE, the generation

of new solutions is unbiased and the selection pressure is instead applied through the replacement of current solutions only with new solutions that are superior.

Scatter Search (SS) uses a much smaller population size and relies on structured combinations of existing solutions to produce new solutions and (optionally) improve them with other (local) methods. Although one implementation of SS [61] used the Nelder-Mead simplex algorithm to improve promising solutions (intensification phase), other local NLP solvers can also be applied [63]. By strict definition, the original Scatter Search is a hybrid method. However, the algorithm can be used without the local search, thus making it pure metaheuristic method.

Other metaheuristics also include the popular Simulated Annealing (SA) [64, 65] and related methods like Stochastic tunneling [66] and Tabu Search [67], which only maintain a single solution during the search iterations. The list of metaheuristics methods discussed above is not meant to be exhaustive. The research activity in the field does not show any sign of slowing down as new algorithms and modifications of existing methods have been proposed within the past decade and more can be expected within the foreseeable future.

Due to the problem formulation and solution screening method (Section 4.2), the solution space contains discontinuities between oscillating and non-oscillating solutions, and multiple local optima may exist. These preclude the use of local methods, especially derivative-based methods. The flexibility and ease of implementation makes metaheuristic methods very attractive for application to the present parameter estimation problem.

2.2.3 Hybrids

Although stochastic global search methods have no guarantees for locating the global optima, they are generally good at avoiding local optima in which local search methods tend to get trapped. Unfortunately, stochastic search methods are computationally expensive. Even when the search has located the convex region of an optimum, convergence to the optimum is far slower than a derivative-

based search method. Thus, it had been suggested to combine the strengths of both classes in a synergistic way, *i.e.* using the stochastic search to avoid poor local optima and the rapid convergence of local search methods when a good optimum region is found.

In one hybrid structure, the global method is used sequentially with the local method. The first global step (*e.g.* GA) searches the solution space to avoid poor local optima and then the search switches to a local method (*e.g.* LM) for rapid convergence with the best known solution as the starting point. Alternatively, the local search can be integrated into the global search. This usually entails the use of local search to improve interim solutions obtained within the global search. An example is the Scatter Search algorithm discussed previously.

2.3 Parameter Estimation of Oscillatory Systems: Circadian Rhythms

As discussed in Chapter 1, parameter estimation methodology is not routinely employed by modelers of biological oscillators. In the review of literature, a small number of recent works were found to apply parameter estimation techniques to build models of circadian rhythms.

Forger and Peskin [68] performed parameter estimation of their 74 states, 36 parameters mammalian circadian rhythm model with experimentally measured protein and mRNA levels under entrained conditions. The data available is sparse, containing only 3 mRNA time profiles each with 6 measurements and 4 protein time profiles each with 13 measurements. The model was fitted to the data over a single oscillation with a simple coordinate search algorithm which cycles over each parameter to modify and compute the resulting objective function score. An initial guess with a suitable period was obtained by trial and error and then used in the parameter search. The objective function does not include error in the free running period, though the best solution obtained shows a physiologically acceptable free running period.

In the modeling of the *Arabidopsis* circadian oscillator, Locke *et al.* [69] used an alternate approach to construct an objective function that scores based on qualitative features of the model output. These features include free running period, phase difference, strength of oscillations and entrainment ability. The model used in this work is composed of 6 states with 23 parameters, which is relatively small compared to the other works discussed in this section. The search procedure consists of an initial phase that enumerates a large number (1 million) of quasi-random points in the parameter space and selecting the best 50 for optimization with SA. This methodology was again used in the construction of an extended model in a subsequent work [70].

In another modeling effort of the *Arabidopsis* circadian oscillator, Zeilinger *et al.* [71] constructed an objective function with terms that measure the phase relationships between identified genes to the light-dark cycles, free running period under constant light and dark conditions, as well as the period of one mutant type. This last term (period of mutant type) is particularly interesting as it is not used in the other parameter estimation efforts. The model used in this study consists of 19 states and 87 parameters, which is also the largest number of parameters estimated among the works discussed in this section. The search algorithm used is ES with the initial population composed of oscillating solutions obtained from a random search of 10,000 solutions. The final solution obtained is further refined using a local hill climbing optimizer.

A recent work by Bagheri *et al.* [72] on the *Drosophila* circadian rhythms shares some similarity to Locke *et al.* and Zeilinger *et al.* in the spirit behind the objective function constructed. The model is composed of 29 states and 84 parameters but the problem size was reduced to 36 parameters by using assumptions of similar reaction rate constants for different species to lump parameters together. The parameter space was further reduced by discretization. By using relative sensitivity distributions obtained from studies of similar models, groups of parameters, in the descending order of sensitivities, were allowed accuracy to the hundredths, the tens and the ones. The parameter estimation is composed

of 3 successive stages solved using GA. Successful solutions from each stage are fed into the next as the initial population. In the first stage, parameter sets are screened for autonomous oscillations and the objective function measures how close the free running period is to the circadian 24 hours. For the second stage, an objective function is constructed to measure qualitative characteristics of the system such as phase relationships and amplitude of certain proteins. The objective function for the second stage is further modified with additional terms that measure entrainment characteristics, creating the objective function for the final stage.

With exception to Forger and Peskin, the works discussed above use objective functions that measure the match in features such as phase relationships as opposed to matching time based profiles of mRNA and proteins. In this work, the case studies used follow Forger and Peskin in using time series data of mRNA and proteins. However, the approach taken in problem formulation and the resulting objective function is different, as well as the use of a global search algorithm. Further, the methodology also considers the confidence intervals of the parameter estimates for use in identifiability analysis.

Chapter 3

Sensitivity Analysis of Oscillatory Systems

In the parameter estimation of dynamical systems, the search modifies parameter values to find the trajectory for the best fit. Sensitivity analysis described in Section 3.2 enables the understanding of system behavior with respect to parametric perturbations and is also useful in computing the Fisher Information Matrix (FIM) to estimate the variance of parameter estimates (Section 4.6). However, sensitivity analysis of oscillatory systems requires a different approach due to their periodic nature and the properties of interest (period and phase) are not addressed by conventional analysis. The appropriate sensitivity measures and associated computation methods for oscillatory systems are reviewed in Section 3.3. Instead of infinitesimal perturbations used in sensitivity analysis, finite perturbations can be utilized as well. In Section 3.4, a commonly used tool in the study of circadian rhythms, the Phase Response Curve, is introduced. Through sensitivity analysis, problems encountered in parameter estimation of oscillatory systems can be better understood, namely due to the property of periodicity which is absent in typical non-oscillatory systems.

3.1 Oscillatory Systems

In the modeling of dynamical physical and biological systems, coupled ordinary differential equations (ODE) are commonly used. In vector notation, the system can be written as:

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

with initial conditions

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{3.2}$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes the states, $\mathbf{p} \in \mathbb{R}^m$ the parameters, $\mathbf{f}$ the vector of nonlinear equations and t is time. Such a system can be used to model an oscillator with stable, attractive limit cycle behavior. A limit cycle is defined as an enclosed periodic orbit in phase space. Biological oscillators are commonly modeled to exhibit such behavior, since the oscillations are maintained while being subjected to external perturbations, as well as the inherent stochastic nature of biological processes (as opposed to orbits) [31, 73–75]. In this work, such models of biological oscillators are considered.

3.2 Sensitivity Analysis

Sensitivity analysis is the study of system output changes due to the perturbations in parameters and initial conditions. Sensitivity analysis is widely applicable, including chemical systems [76]. In this work, local sensitivities are used in the computation of FIM from which the variance of parameter estimates can be bounded (Section 4.6).

The first order local sensitivity coefficient is defined as:

$$s_{i,j} = \frac{\partial y_i}{\partial p_j} \tag{3.3}$$

where $s_{i,j}$ is the sensitivity coefficient of dependent variable y_i with respect to parameter p_j. Higher order sensitivity coefficients are available but only first order sensitivities are considered here. Generally, output sensitivity coefficients can be computed from the state sensitivities since the outputs are functions of the system states by:

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) \tag{3.4}$$

where $\mathbf{g}$ is the output function and $\mathbf{x}$ are system states. For an ODE system (Equation 3.1), there are 3 methods of computing the state sensitivities: direct, finite-difference and Green's function [76].

Direct Method

The direct method is the conceptually most straightforward method of computing the state sensitivities by solving, together with the original ODE system (Equation 3.1):

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathbf{x}}{\partial \mathbf{p}}(t) = \frac{\mathrm{d}\mathbf{S}}{\mathrm{d}t}(\mathbf{p}, \mathbf{x}, t) = \mathbf{J}(\mathbf{x}, t)\frac{\partial \mathbf{x}}{\partial \mathbf{p}}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{p}}(\mathbf{x}, t) = \mathbf{J}(\mathbf{x}, t)\mathbf{S}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{p}}(\mathbf{x}, t) \tag{3.5}$$

where $\mathbf{J}$ is the Jacobian matrix of $\mathbf{f}$ with respect to $\mathbf{x}$. The initial conditions are given by:

$$s_{i,j}\big|_{t=0} = \begin{cases} 0, & p_j \neq x_i \\ 1, & p_j = x_i. \end{cases} \tag{3.6}$$

Finite Difference Method

Finite difference avoids the necessity of solving the model and sensitivity differential equations, and approximates the local sensitivity coefficients by:

$$s_{i,j} = \frac{\partial x_i}{\partial p_j} \approx \frac{\Delta x_i}{\Delta p_j} = \frac{x_i(t, p_j + \Delta p_j) - x_i(t, p_j)}{\Delta p_j} \tag{3.7}$$

using a finite perturbation Δp_j. To determine the state sensitivities with respect to a parameter p_j, the model equations are solved twice for p_j and $p_j + \Delta p_j$.

When using finite difference, there are two sources of error, simulation error and truncation error. Simulation error is due to the use of numerical integration methods and truncation error is caused by the omission of higher order terms in the approximation (Equation 3.7). The simulation error is present in other methods (direct method and Green's function method); it cannot be completely eliminated and can only be controlled by the choice of integration step size. The second error type, truncation error, can be adjusted by reducing Δp_j, but the error magnitude cannot be reduced beyond the magnitude of error due to numerical integration. To determine a suitable Δp_j at each time step, multiple function evaluations may be computed and this translates to considerable computation cost as one has to solve the model equations more than twice to obtain sufficiently accurate sensitivity of one state with respect to one parameter.

The finite difference method can be useful if only a single local sensitivity value at one time point is needed, or if the model is given in a complex functional or non-mathematical form that does not allow derivation of the Jacobian (*i.e.* black box model).

Green's function method

The Green's function method solves for the sensitivities from the equation system (3.5) in a different manner. The homogenous part is solved first and the particular solution for each parameter is solved next. The homogenous part of the sensitivity equation system corresponding to the Green's function $\mathbf{\Gamma}(t, t')$ problem is:

$$\frac{d}{dt}\mathbf{\Gamma}(t, t') = \mathbf{J}(t)\mathbf{\Gamma}(t, t'),\ t \geq t';\ \mathbf{\Gamma}(t', t') = \mathbf{I}. \tag{3.8}$$

The parametric sensitivities can be computed with:

$$\frac{\partial \mathbf{x}}{\partial p_j}(t) = \mathbf{\Gamma}(t, 0)\frac{\partial \mathbf{x}}{\partial p_j}(0) + \int_0^t \mathbf{\Gamma}(t, t')\frac{\partial \mathbf{f}}{p_j}(t')dt'. \tag{3.9}$$

In the computation of sensitivities over $0 \leq t \leq t_{\text{end}}$, $\mathbf{\Gamma}(t, t')$ value for $0 \leq t$ are required for each time interval. A more efficient method is to compute the adjoint Green's function $\mathbf{\Gamma}^{\dagger}(t', t)$ using

$$\frac{d}{dt}\mathbf{\Gamma}^{\dagger}(t', t) = -\mathbf{\Gamma}^{\dagger}(t', t)\mathbf{J}(t),\ 0 \leq t' \leq t;\ \mathbf{\Gamma}(t', t') = \mathbf{I} \tag{3.10}$$

which is integrated backwards. Since $\mathbf{\Gamma}(t, t') = \mathbf{\Gamma}^{\dagger}(t', t)$, the adjoint can be used in Equation 3.9 instead.

In the direct method, $(m+1) \times n$ differential equations are solved while $n \times n$ differential equations and n integrals are solved in Green's function method. In the case of $m \gg n$, the Green's function method is more computationally efficient.

3.3 Sensitivity Analysis of Oscillatory Systems

Parametric state sensitivity computed for oscillatory systems show a divergence as time increases towards infinity, as illustrated in Figure 3.1. Sensitivity analysis of the properties of interest for oscillatory systems, amplitude, period and phase, cannot be directly obtained using the methods described in Section 3.2. Instead, a different treatment is required [77]. This section will focus on sensitivity analysis of phase and period due to their importance to the problem formulation.

Before proceeding further, phase needs to be defined first. Here, phase (ϕ) in an oscillation is the relative position on the limit cycle with respect to a reference point. It is measured by the time difference between the point and the reference modulo the period. When comparing two oscillation trajectories from the same limit cycle, the phase difference is the difference in time (modulo period) for both trajectories to attain the same phase on the limit cycle.

An important concept used in the analysis of oscillator limit cycle is isochrons (η). Isochrons are defined as the set of initial conditions that give oscillations with the same phase as $t \to \infty$. Figure 3.2 shows hypothetical isochrons of a

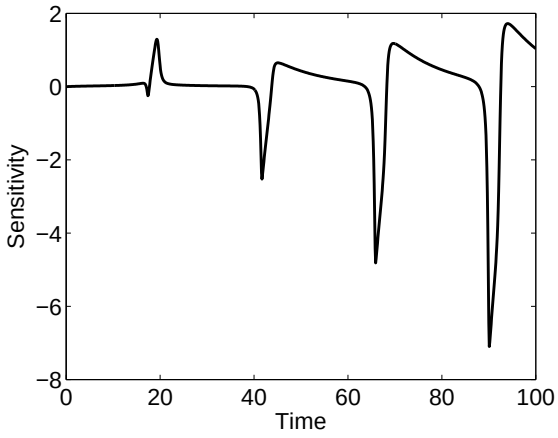


Figure 3.1: Sensitivity of state M to parameter v_m in the Tyson *et al.* model

2-state limit cycle. When comparing a perturbed limit cycle to the nominal, the phase difference can be measured with respect to the isochrons of the nominal limit cycle. Use of the isochron concept with sensitivity analysis of oscillatory behavior was explored by Gunawan and Doyle [78].

3.3.1 Sensitivity of Phase to Initial Condition

It is assumed here that the system exhibits stable limit cycle behavior and that the perturbed initial conditions lie within the basin of attraction of the limit cycle. Perturbations in the initial conditions do not alter the steady state oscillations but only the phase after the initial transient effects (See Figure 3.3). To compute the phase sensitivity to initial conditions, the Green's function matrix may be used:

$$\begin{aligned} Q_j(t=0) = \frac{\partial \phi}{\partial x_j(0)} &= -\left(\frac{\partial x_i(t')}{\partial x_j(0)}\right) \Big/ \left(\frac{dx_i(t')}{dt}\right) \\ &= -\lim_{t' \to \infty} \mathbf{\Gamma}(t', 0) \Big/ \left(\frac{dx_i(t')}{dt}\right) \end{aligned} \tag{3.11}$$

where only a single row of the adjoint Green's function matrix is needed. The trajectory is required to reach the limit cycle, but given that it approaches the limit cycle asymptotically, this only occurs at $t' = \infty$. For practical applications however, sufficient accuracy can usually be obtained after a small number of cycles, though the exact number for a given accuracy varies for different systems.

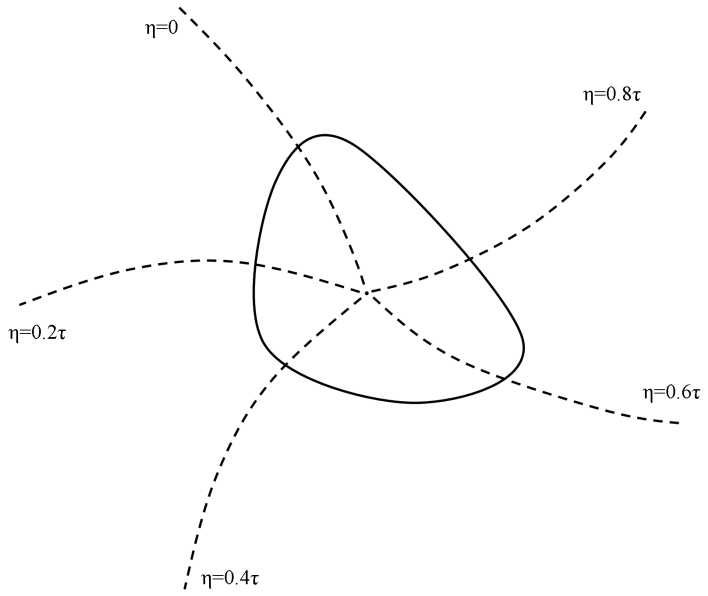


Figure 3.2: Isochrons of a 2-state limit cycle.

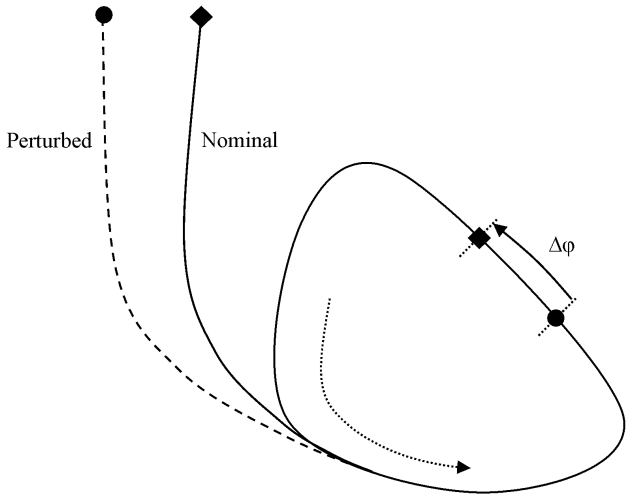


Figure 3.3: Evolution of trajectory from different initial conditions and resulting phase difference. Adapted from Gunawan and Doyle (2006).

The choice of state x_i is immaterial as long as the trajectory belongs to the limit cycle since phase is a system property rather than a property of individual states. The above computation can also be used to compute phase sensitivities with respect to perturbations of the states at any time t.

3.3.2 Parametric Phase Sensitivity

Perturbations to the parameters not only alter the phase of the system, but also the limit cycle itself and consequently the isochrons. Thus, this requires the comparison of phase between two different limit cycles. To resolve this, phase is compared using the isochrons of the nominal limit cycle as shown in the earlier section. Parametric phase sensitivity can be then defined as [77]:

$$\left(\frac{\partial\phi(t)}{\partial p_j}\right)_{\eta} = \sum_{i=1}^{n} Q_i(t)\frac{\partial x_i(t)}{\partial p_j} \tag{3.12}$$

where η denotes the phase difference measured with respect to a given isochron $\eta(t)$. The phase sensitivity computed is the sum total of the phase shifts due to state changes caused by parametric perturbations. Figure 3.4 illustrates the phase difference between the perturbed and nominal trajectories measured using isochrons of the nominal cycle.

3.3.3 Period Sensitivity

When the period τ changes due to parameter perturbations, the phase difference accumulates with every cycle. The accumulated phase difference over a single cycle is in fact equal to the period change, and the period sensitivity can be computed from the phase sensitivity according to:

$$\frac{\partial\tau}{\partial p_j} = \left(\frac{\partial\phi(t+\tau)}{\partial p_j}\right)_{\eta} - \left(\frac{\partial\phi(t)}{\partial p_j}\right)_{\eta} \tag{3.13}$$

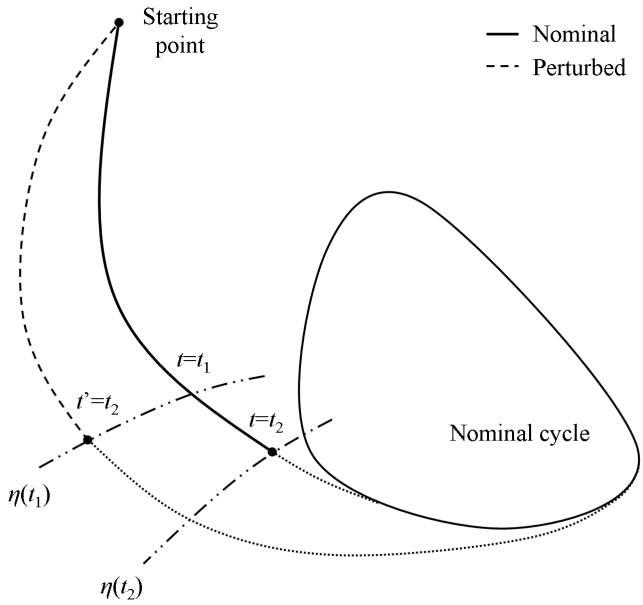


Figure 3.4: Phase difference measured with the isochrons ($\eta(t)$) of the nominal cycle. The isochrons are in illustrated with dash-dot lines. The dotted lines denote trajectory with nominal parameters. Adapted from Gunawan and Doyle (2006).

3.3.4 Parametric Sensitivity

At the beginning of Section 3.3, the parametric sensitivities of oscillatory systems was shown to diverge as time increases towards infinity. The parametric state sensitivities can be decomposed in the following manner [79]:

$$\frac{\partial x_i}{\partial p_j} = \left(\frac{\partial x_i}{\partial p_j}\right)_\tau - \frac{t}{\tau}\frac{\partial \tau}{\partial p_j}\frac{dx_i}{dt} \tag{3.14}$$

to give the respective shape and period contributions. On the right hand side, the first term contains the parametric state sensitivity with respect to same period and is periodic. The second term depends linearly on t as well as the period sensitivity. If the period sensitivity is not zero (*i.e.* the system period is affected by the parametric perturbation), then the second term and consequently the state sensitivity grows unbounded as $t \to \infty$. Figure 3.5 shows the divergence in two oscillatory behaviors with different periods.

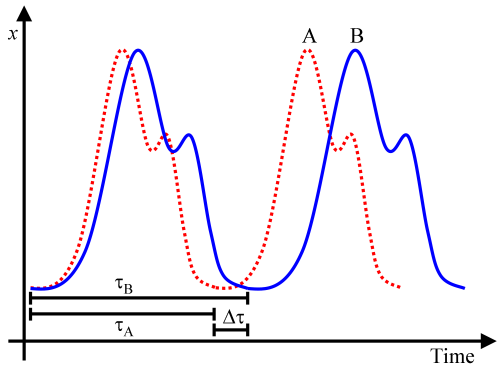


Figure 3.5: Comparing two oscillating signals with different periods.

3.4 Phase Response Curve

In circadian oscillators, an important property is the ability to entrain to the environmental cues. Depending on the time at which it is administered, these entrainment cues induce different magnitudes of phase shift in the oscillator. When plotting the phase shifts or phase response over one period, a Phase Response Curve (PRC) is obtained. The period is normalized to 24 hr in this work.

The entrainment cues, usually light, are typically modeled as a finite parametric perturbation to the system. To compute the phase response, the parameters affected by light are perturbed by $\mathbf{\Delta p}$ for a time period of θ. The system is then allowed to return to the limit cycle. The phase shift is measured by taking the difference between points of the same phase from the perturbed and unperturbed system. Figure 3.6 illustrates the phase response of the system.

PRCs can be classified in two ways. The first divides PRCs into type 1 and type 0 by winding number [80]. Type 1 PRCs show small phase shifts and are continuous over the entire cycle. Type 0 PRCs show large shifts with a discontinuity between phase delay and phase advance shifts (12 hr delay = 12 hr advance). An example of each type is given in Figure 3.8. The second classification divides PRCs into type I and type II by bifurcation structure [81]. Type I PRCs only exhibit advance phase shifts while type II can exhibit both advance and delay phase shifts. Figure 3.7 illustrates the two types.

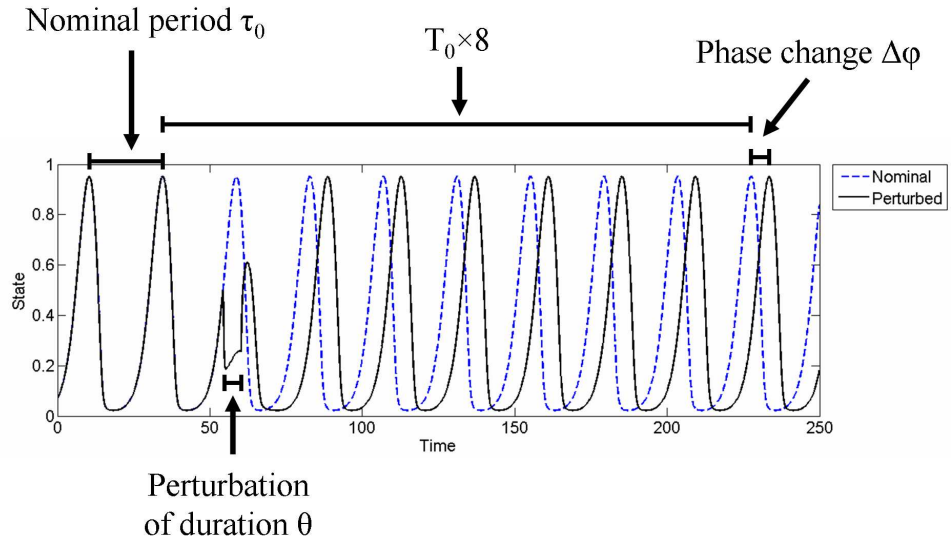
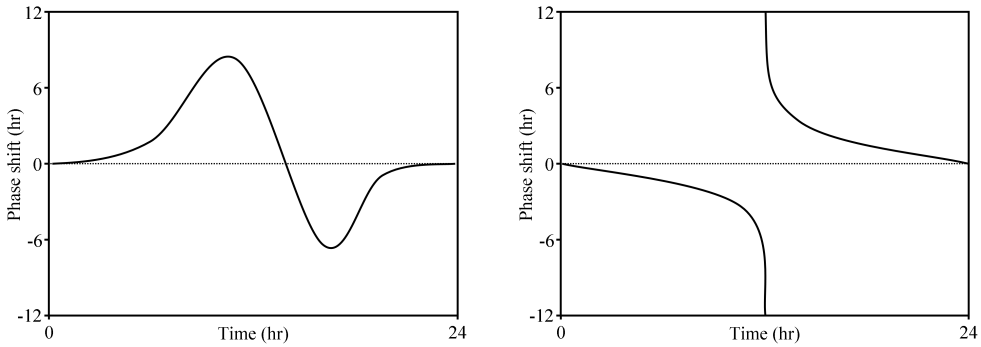
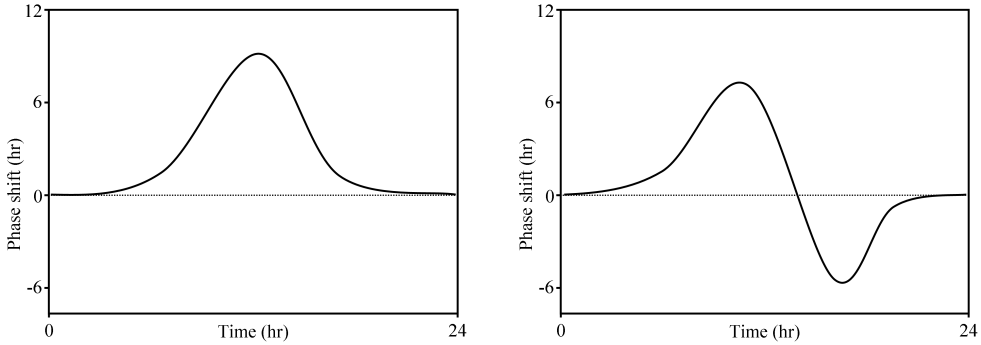


Figure 3.6: Phase response to perturbation.



(a) Type 1 PRC (b) Type 0 PRC

Figure 3.7: PRCs classified by winding number



(a) Type I PRC (b) Type II PRC

Figure 3.8: PRCs classified by bifurcation structure

Chapter 4

Methodology

The parameter estimation problem of oscillatory systems is first formulated in Section 4.1, based on the analysis in the previous chapter. Next, the presentation of the parameter estimation methodology is divided into two parts: the objective function formulation and the optimization algorithm. The objective function computation is further broken down into three consecutive steps, each described in Sections 4.2, 4.3 and 4.4. Thereafter, Section 4.5 discusses the optimization algorithm selected and modifications made for application to the problem at hand. With the estimated parameters, covariance matrix of the parameters can be estimated using the Fisher Information Matrix to determine the parameter identifiability, which is described in Section 4.6.

4.1 Problem Formulation

Oscillatory systems possess two key characteristics of periodicity and shape, where shape describes the state trajectory over one cycle. In contrast, non-oscillatory systems only possess the shape characteristic and have infinite period. The periodicity characteristic prevents the direct application of standard parameter estimation methods (sum of errors squared) to oscillatory systems, since parametric sensitivities and consequently the gradient of the objective function grows unbounded over time from the accumulation of phase differences due to periodicity mismatch. In Section 3.3.4, the corresponding unbounded increase in

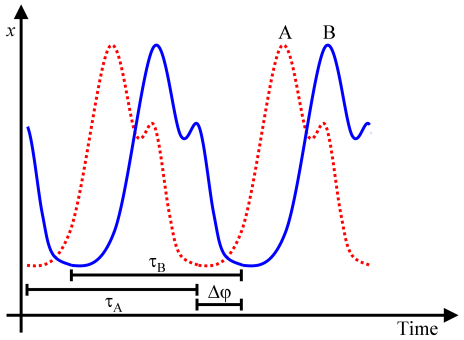


Figure 4.1: Comparing two oscillating signals at different phases

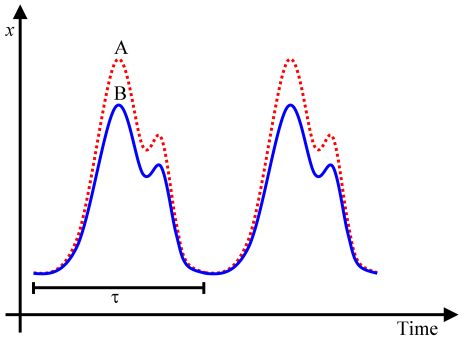


Figure 4.2: Comparing two oscillating signals with different shapes

parametric sensitivity was discussed. Again we turn to Figure 3.5 in the previous chapter with two oscillations with different periodicity compared on time basis. Due to the period mismatch, an error will be computed between the two trajectories. However, if we scale time with period, both trajectories are identical in shape and thus no shape error. In other words, the only difference between the two oscillations is the period mismatch.

Another distinguishing characteristic between two oscillations is the phase. Due to noise, the initial phase of the data cannot be determined accurately and results in a phase difference between the data and model simulation. This is illustrated in Figure 4.1, where two oscillations with different starting phases are compared. Both oscillations are identical in shape and period but due to the difference in phase, an incorrect shape error is computed. To resolve this, the initial conditions are cast as parameters to be estimated with the system parameters. Figure 4.2 shows two oscillations compared at the same starting phase and on phase basis, thus clearly illustrating the shape error. In this way, shape error and period error are decoupled and ensured that a correct shape error is computed.

Based on the formulation above, the computation strategy for the objective function is divided into the following three steps. In the first step, the feasible oscillatory solutions are screened out. In the second step, period of the feasible solutions are estimated. Finally, the error between phase series data and the model is computed using the objective function in the third step. These steps

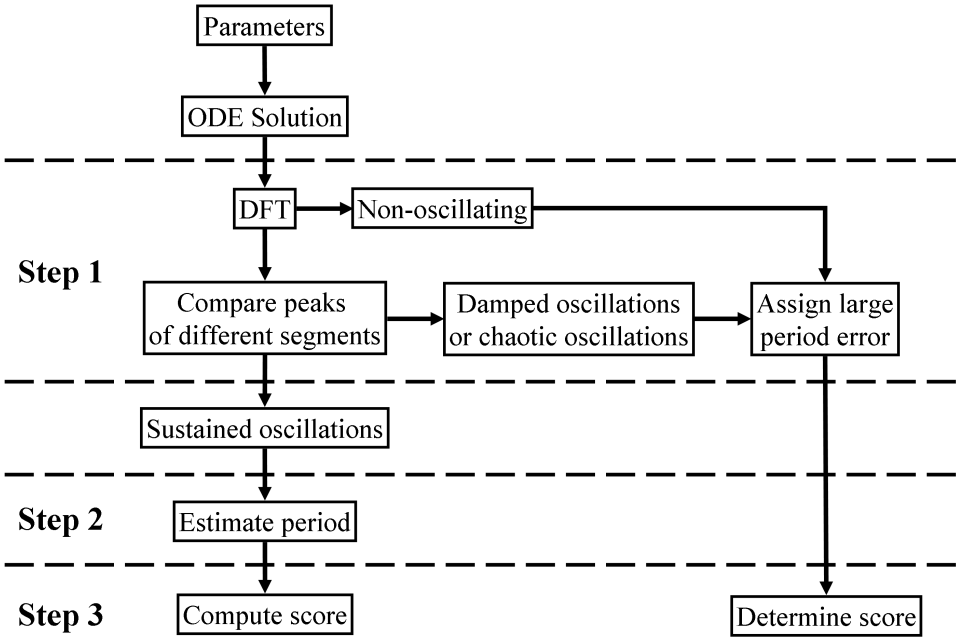


Figure 4.3: Parameter screening and scoring in the objective function.

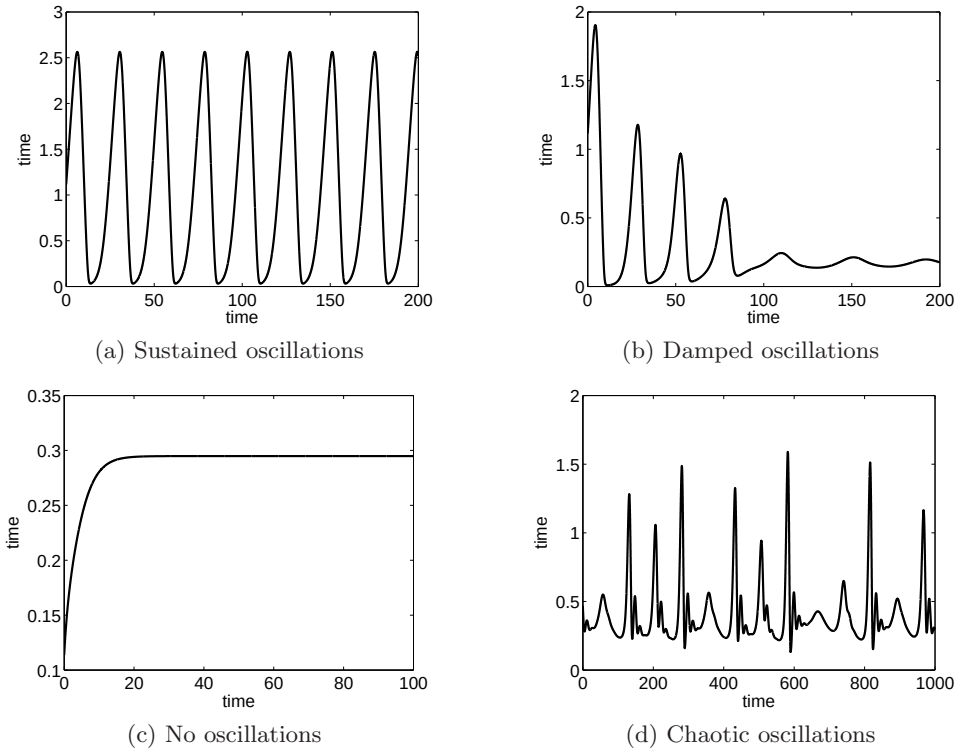
are summarized in Figure 4.3.

4.2 Feasible Oscillatory Behavior

In the parameter estimation problem of oscillatory systems, the parameter space contains parameter vectors that produce model dynamics of different natures. Among the numerically stable solutions, the following types of dynamics are possible:

1. Sustained oscillations
2. Damped oscillations
3. Non-oscillating
4. Chaotic oscillations

Figure 4.4 illustrates the dynamic behaviors listed above. Since solutions exhibiting sustained oscillations (*i.e.* constant period and constant amplitude) are desired, parameter sets that produce Type 1 solutions need to be screened out. To accomplish this, two screening processes are carried out. The first employs a Discrete Fourier Transform (DFT) of the model output to detect oscillations with a constant period and the second checks for variations in crest heights.



(a) Sustained oscillations

(b) Damped oscillations

(c) No oscillations

(d) Chaotic oscillations

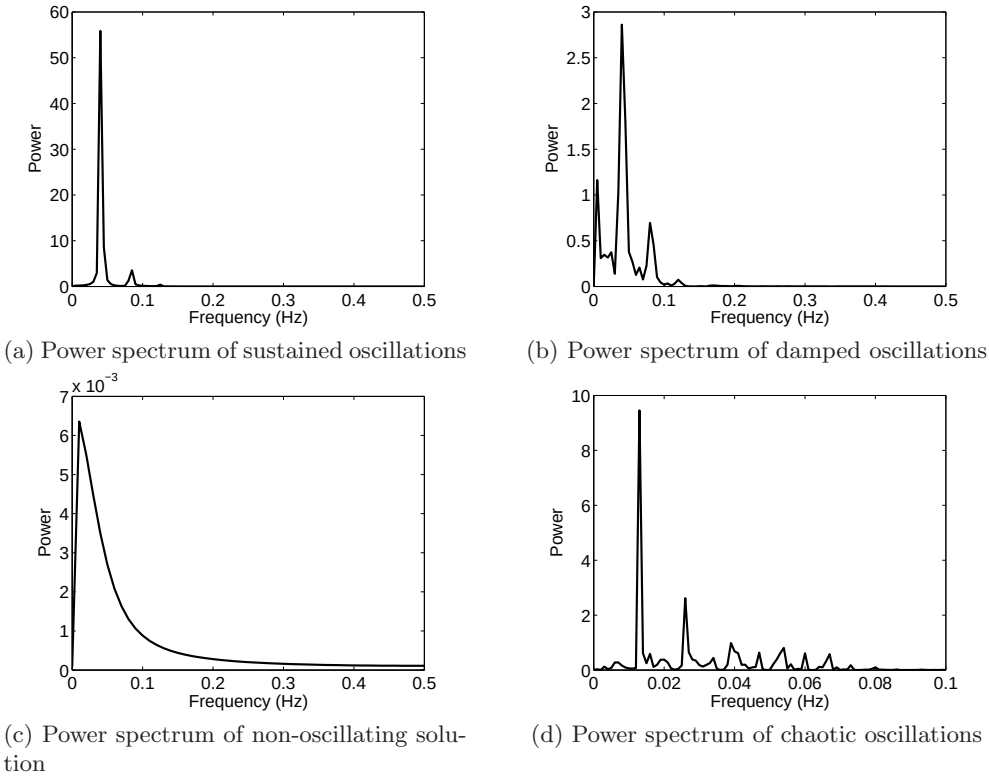
Figure 4.4: Solutions types.

4.2.1 Discrete Fourier Transform

To screen for oscillating solutions with a constant period, a DFT [82] of model output is used. Figure 4.5 shows the power spectrum of the solution types discussed above. For presentation clarity, the power at 0 Hz is removed, since it is much larger in magnitude than the power at all other frequencies, and represents the power at steady state which is irrelevant to the analysis. An oscillating solution produces a prominent peak in the power spectrum at the oscillating frequency. On the other hand, a non-oscillating dynamic response produces only a single peak at 0 Hz (not shown). In this manner, non-oscillating solutions can be screened out.

4.2.2 Peak Comparison

From the model output, the crest heights of the oscillation cycles are recorded. If the comparison of heights shows minimal deviation (*e.g.* $< 2\%$) between each consecutive crest, the solution is assumed to exhibit sustained oscillations.



(a) Power spectrum of sustained oscillations

(b) Power spectrum of damped oscillations

(c) Power spectrum of non-oscillating solution

(d) Power spectrum of chaotic oscillations

Figure 4.5: Power spectrum of solutions types.

After classifying each solution (and its associated parameter vector) accordingly, the period of Type 1 solutions are estimated (See Section 4.3 below). The remaining solutions either have infinite period (non-oscillating and damped oscillations) or multiple frequency components (chaotic oscillations). For these solutions, a large finite value is assigned as the period, *i.e.* the time of the final reading in the dataset (which is also the maximum period possible for the dataset). This places a large selection pressure in the search method against these solutions. Unfortunately, it also introduces discontinuities between oscillating and non-oscillating regions in the solution space, but derivative-free optimization algorithms can handle such discontinuities.

The method described above for screening out stable limit cycle solutions relies on properties of their state trajectories. The advantage of this approach is its applicability to nonlinear systems in general and ease of use. The drawback is its lack of rigor and possible incorrect classification of solutions due to the criteria used (*e.g.* oscillations damped at less than 1% per cycle), though such

a problem did not arise in the case studies used.

There are a number of analytical methods that can be used to determine the existence (or non-existence) and stability of periodic orbits (limit cycles inclusive) [83]. Unfortunately, these methods are difficult to apply to general nonlinear systems. These methods instead may be useful in analysis of the model prior to parameter estimation, where the results can help determine suitable parameter bounds for parameter estimation.

4.3 Period Estimation

Although the DFT of the model output can be used to estimate the system period, the period estimates obtained were found to be inaccurate, thus requiring an alternative technique to produce better period estimates.

One method to estimate the period is to find the time differential between two points of the same phase on adjacent cycles of an output y_i. These two points are marked with crosses in Figure 4.6. The best period estimate is obtained by using two points at the phase with the largest gradient ($\mathrm{d}y/\mathrm{d}t$) as this gives the smallest error Δy_i for a given sampling time Δt_i. To obtain the time of the chosen points, linear interpolation is used. A cubic interpolating spline can be used to produce more accurate estimates but the gain in accuracy is minor considering the choice of points (with the largest gradient) and the additional computational effort of generating a spline at every objective function evaluation.

Alternatively, a simpler method is to use mean-crossings for period estimation. In Figure 4.6, the mean value is indicated by the horizontal line and the two circular markings are the zero-crossings used. The use of zero-crossings eliminates the need to search for the point with the highest gradient but with small sacrifices in accuracy for most cases. Figure 4.7 illustrates a possible case in which the zero-crossing method produces poor accuracy in the period estimated. Such cases are however rare and do not appear the case studies used in this work.

For the period estimation of experimental data, the data is first treated using a moving average filter to reduce the measurement noise. The presence of noise

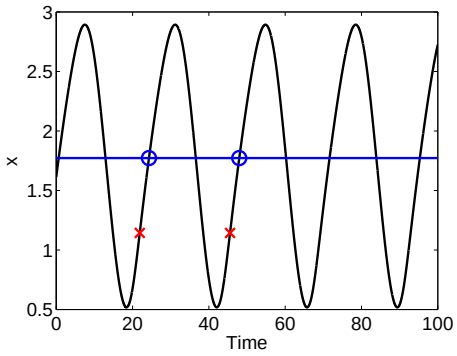


Figure 4.6: Period estimation

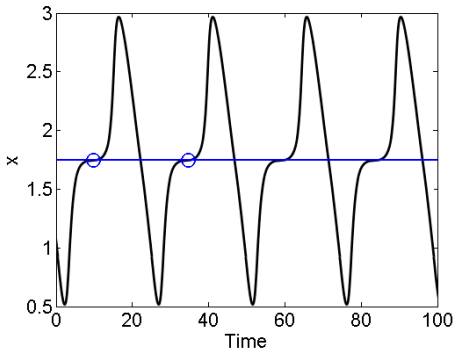


Figure 4.7: Poor period estimation.

also means that gradient estimation is difficult and thus the period is estimated by the simpler method of mean-crossings. It is important that an accurate period estimate is used to scale time, since an inaccurate period used will naturally lead to the comparison of data points at the wrong phase and incorrect shape errors computed.

For a given set of noisy experimental data, multiple period estimates can be obtained (one per oscillation cycle, per state). The population variance of N period estimates τ can be estimated by:

$$\sigma_\tau^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\tau_i - \bar{\tau})^2, \tag{4.1}$$

since it is difficult to compute the period variance using the variance of data points.

4.4 Error Computation

A number of objective functions are viable for the parameter estimation problem. These include [3]:

- Ordinary Least Squares (OLS)
- Maximum Likelihood Estimation (MLE)
- Maximum A Posteriori (MAP)

From the three methods listed, MAP estimation requires the most amount of information: the *a priori* distribution of the parameters and the measurement

data. MLE requires only an assumed distribution of the measurement noise while OLS requires only the data.

Ordinary Least Squares is the simplest method in common use. The OLS objective is commonly modified with the use of weights as Weighted Least Squares (WLS). For example, in the case where multiple states are measured, the state measurements may be weighted by the peak value of the state. When measurement error variance are used as the weights, the WLS is identical to the MLE with the assumption of independent and identically distributed (iid), and Gaussian distribution of errors.

In MLE and MAP, a statistical approach is taken. The measurements are considered as samples of random variables. These random variables have a joint probability density function (PDF) and the estimator in MLE maximizes the likelihood of observing the dataset. The derivation with the assumption of Gaussian distribution is given in Section 4.4.1. For MAP estimation, the parameters are also considered as random variables with a prior distribution. Using the prior distribution and the Bayes' rule, the MAP estimator is derived in Section 4.4.2.

4.4.1 Maximum Likelihood Estimation

A likelihood function is defined as the joint probability function of the data sample, reflecting the likelihood of the parameters producing a given dataset. The likelihood $\mathcal{L}(\mathbf{p}; \hat{\mathbf{y}})$ of obtaining a vector of parameters $\mathbf{p}$ given the dataset $\hat{\mathbf{y}}$ of N observations is:

$$\mathcal{L}(\mathbf{p}; \hat{\mathbf{y}}) = f_y(\hat{\mathbf{y}}; \mathbf{p}) \tag{4.2}$$

where f_y is the probability density function (PDF) of $\hat{\mathbf{y}}$ given $\mathbf{p}$. For MLE under Gaussian assumption, $\mathbf{p}$ can be estimated by maximizing the log of the likelihood function:

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} \Big\{ \log \mathcal{L}(\mathbf{p}; \hat{\mathbf{y}}) \Big\} = \arg\max_{\mathbf{p}} \Big\{ \log f_y(\hat{\mathbf{y}}; \mathbf{p}) \Big\} \tag{4.3}$$

where $\hat{\mathbf{p}}$ is the Maximum Likelihood Estimator of $\mathbf{p}$. The logarithm of a function is a monotone transformation and thus does not alter the maxima. The logarithm operation is used to convert multiplication terms to addition terms, which are easier to manipulate.

In the absence of additional information, the assumption of Gaussian distribution for f_y is commonly made:

$$f_y(\hat{\mathbf{y}};\, \mathbf{p}) = \left(\frac{1}{(2\pi)^{N/2}|\mathbf{V}|^{1/2}}\right) \exp\left(-\frac{1}{2}(\hat{\mathbf{y}}-\mathbf{y})^{\mathrm{T}}\mathbf{V}^{-1}(\hat{\mathbf{y}}-\mathbf{y})\right) \tag{4.4}$$

where $\mathbf{y}$ is the output of model $\mathbf{f}$ and $\mathbf{V}$ is the covariance matrix of the observations. Substituting back into Equation 4.3 gives:

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} \left\{ \log\left(\frac{1}{(2\pi)^{N/2}|\mathbf{V}|^{1/2}}\right) - \left(\frac{1}{2}(\hat{\mathbf{y}}-\mathbf{y})^{\mathrm{T}}\mathbf{V}^{-1}(\hat{\mathbf{y}}-\mathbf{y})\right)\right\}. \tag{4.5}$$

Removing the first constant term and converting the maximization of a negative function to the minimization of a positive function yields

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \left\{ (\hat{\mathbf{y}}-\mathbf{y})^{\mathrm{T}}\mathbf{V}_{\mathbf{y}}^{-1}(\hat{\mathbf{y}}-\mathbf{y})\right\}. \tag{4.6}$$

If the measurements are independent, then the covariance matrix $\mathbf{V}$ is diagonal and the parameter estimation problem then reduces to:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \left\{ \sum_{i=1}^{N} \frac{(\hat{y}_i - y_i)^2}{\sigma_i^2} \right\} \tag{4.7}$$

for N data points and σ_i is the i-th diagonal element of $\mathbf{V}$. This is identical to weighted least squares using measurement variance as the weights.

4.4.2 Maximum a Posteriori

In the Bayesian approach of MAP estimation, the parameters are treated as random variables. These parameters, having a prior distribution $f(\mathbf{p})$, are correlated with the measurements $\hat{\mathbf{y}}$ in the dataset. With these, we seek the maximum of the posterior joint probability distribution function of $\mathbf{p}$ and $\hat{\mathbf{y}}$. Using Bayes'

rule, the conditional probability distribution function for $\mathbf{p}$ is given by [3]:

$$f(\mathbf{p}|\hat{\mathbf{y}}) = \frac{f_y(\hat{\mathbf{y}}|\mathbf{p}) \cdot f(\mathbf{p})}{f(\hat{\mathbf{y}})}. \tag{4.8}$$

The MAP estimate can then stated as:

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} \left\{ \frac{f_y(\hat{\mathbf{y}}|\,\mathbf{p}) \cdot f(\mathbf{p})}{f(\hat{\mathbf{y}})} \right\} \tag{4.9}$$

where the denominator is not dependent on $\mathbf{p}$ and the numerator is then used to derive the objective function. As in the case for MLE, maximizing the logarithm of a function is the same as maximizing the function itself:

$$\begin{aligned} \hat{\mathbf{p}} &= \arg\max_{\mathbf{p}} \left\{ \log\left[f_y(\hat{\mathbf{y}}|\,\mathbf{p}) \cdot f(\mathbf{p}) \right] \right\} \\ &= \arg\max_{\mathbf{p}} \left\{ \log\left[f_y(\hat{\mathbf{y}}|\,\mathbf{p}) \right] + \log\left[f(\mathbf{p}) \right] \right\}. \end{aligned} \tag{4.10}$$

With the assumption of Gaussian distribution for both the measurements as well as the prior distribution of parameters $\tilde{\mathbf{p}}$, manipulation of the above equations gives

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \left\{ (\hat{\mathbf{y}} - \mathbf{y})^{\mathrm{T}} \mathbf{V}_{\mathbf{y}}^{-1} (\hat{\mathbf{y}} - \mathbf{y}) + (\tilde{\mathbf{p}} - \mathbf{p})^{\mathrm{T}} \mathbf{V}_{\mathbf{p}}^{-1} (\tilde{\mathbf{p}} - \mathbf{p}) \right\} \tag{4.11}$$

and

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \left\{ \sum_{i=1}^{N_y} \frac{(\hat{y}_i - y_i)^2}{\sigma_{y,i}^2} + \sum_{i=1}^{N_p} \frac{(\tilde{p}_i - p_i)^2}{\sigma_{p,i}^2} \right\} \tag{4.12}$$

for diagonal covariance matrices $\mathbf{V_y}$ and $\mathbf{V_p}$.

4.4.3 Objective Function for Oscillatory Systems

As described in Section 4.1, the time series data are converted into phase (ϕ) series data by scaling time with the period. By comparing data points at the same phase, the shape error is computed, while the period error is accounted for in an additional term. Assuming no knowledge of a prior distribution for

parameters $\mathbf{p}$, for a system with N_y states, the objective function $\Phi(\mathbf{p})$ to be minimized can be constructed using MLE as follows:

$$\Phi(\mathbf{p}) = \left(\sum_{j=1}^{N_y} \sum_{i=1}^{N} \frac{1}{\sigma_{j,i}^2} \big(\hat{y}_j(\phi_i) - y_j(\phi_i)\big)^2 \right) + \left(\sum_{i=1}^{N_\tau} \frac{1}{\sigma_\tau^2} (\hat{\tau}_i - \tau)^2 \right) \tag{4.13}$$

where N_τ is the number of period estimates obtained from the data. The period variance σ_τ^2 uses the estimated population variance from Equation 4.1.

However, the variation in period estimates from the experimental data used is due to the additive measurement noise in the time series data as opposed to a stochastic system with varying period. One can consider each period estimates as a different reading of the same data point (period), and since the simulated model has only one period estimate, a simplified formulation using the mean period $\hat{\tau}_{ave}$ is:

$$\Phi(\mathbf{p}) = \left(\sum_{j=1}^{N_y} \sum_{i=1}^{N} \frac{1}{\sigma_{j,i}^2} \big(\hat{y}_j(\phi_i) - y_j(\phi_i)\big)^2 \right) + \left(\frac{1}{\sigma_{\tau,\text{ave}}^2} (\hat{\tau}_{ave} - \tau)^2 \right) \tag{4.14}$$

with $\sigma_{\tau,\,\text{ave}}^2$ being the estimated variance of the period mean, *i.e.* $\sigma_{\tau,\,\text{ave}}^2$ can be obtained by dividing σ_τ^2 with N_τ.

4.4.4 Stochasticity in Gene Expression

Due to low copy numbers within cells, many cellular processes, such as gene expression, are affected by stochastic noise. With development of the Green Fluorescence Protein (GFP) and derivatives, intracellular noise can be directly studied [84, 85]. Through wet-lab experiments and simulations, it was shown that proteins at low copy numbers exhibited a long-tailed distribution that was fitted to a log-normal distribution [86,87]. As the amount of protein increased, a crossover to the Gaussian distribution was observed. This crossover was modeled by the following exponential function:

$$\mathbf{p}(N) = \left(\frac{N + N_0}{\sqrt{2\pi} N N_0} \right) \exp \left[\frac{-\big(\ln\big(\frac{N}{N_0}\big) + \frac{N}{N_0} - \mu \big)^2}{2\sigma^2} \right] \tag{4.15}$$

where N is the number of proteins copies, N_0 is an independent scaling parameter, μ is the distribution mean, and σ is standard deviation (Supporting Text of [87]).

For simplicity, only the Gaussian distribution of noise is considered in this work. However, the objective function developed in the earlier section can easily be modified for the case of log-normal distribution or the distribution for lognormal-to-Gaussian crossover in Equation 4.15.

4.5 Differential Evolution

Due to the problem formulation in the previous section, the solution space contains discontinuities and multiple local optima may exist. These exclude the use of local methods, especially derivative-based methods. On the other hand, most metaheuristic methods such as Differential Evolution (DE) do not require gradient information and are much better at handling multiple optima. Combined with flexibility and ease of implementation, the DE algorithm is an attractive choice.

DE utilizes two vector populations each containing N_p D-dimensional vectors. For the parameter estimation problem at hand, each vector contains the model parameters to be optimized. The current population $\mathbf{P}_{\mathbf{x}}$ contains vectors $\mathbf{x}_{i,g}$ that are either initial solution points or selected surviving solutions from the last iterate:

$$\begin{aligned} &\mathbf{P}_{\mathbf{x},g} = \{\mathbf{x}_{i,g}\}, && i = 0, 1, ..., N_p - 1,\ g = 0, 1, ..., g_{\max}, \\ &\mathbf{x}_{i,g} = \{x_{j,i,g}\}, && j = 0, 1, ..., D - 1. \end{aligned} \tag{4.16}$$

The subscript index g (0 to $g_{\max}$) indicates the generation, i indicates population index (0 to $N_p - 1$) and finally j (0 to D) indicates parameter index.

During each iteration, randomly chosen vectors from $\mathbf{P}_{\mathbf{x},g}$ are first mutated

to produce the mutant population $\mathbf{P}_{\mathbf{v},g}$ that consists of N_p mutant vectors $\mathbf{v}_{i,g}$:

$$\begin{aligned} &\mathbf{P}_{\mathbf{v},g} = \{\mathbf{v}_{i,g}\}, && i = 0, 1, ..., N_p - 1,\ g = 0, 1, ..., g_{\max}, \\ &\mathbf{v}_{i,g} = \{v_{j,i,g}\}, && j = 0, 1, ..., D - 1. \end{aligned} \tag{4.17}$$

Each mutation is next recombined with a randomly chosen target vector from the current population to produce the trial population $\mathbf{P}_{\mathbf{u}}$ of N_p trial vectors $\mathbf{u}_{i,g}$:

$$\begin{aligned} &\mathbf{P}_{\mathbf{u},g} = \{\mathbf{u}_{i,g}\}, && i = 0, 1, ..., N_p - 1,\ g = 0, 1, ..., g_{\max}, \\ &\mathbf{u}_{i,g} = \{x_{j,i,g}\}, && j = 0, 1, ..., D - 1. \end{aligned} \tag{4.18}$$

Since the mutant vectors are overwritten by trials vectors, only a single array is needed to hold both $\mathbf{P}_{\mathbf{v},g}$ and $\mathbf{P}_{\mathbf{u},g}$ populations and thus only two arrays are necessary for storage in the program.

4.5.1 Initialization

In most standard implementations of DE, the population is initialized in an uniform random manner. Using the upper ($\mathbf{b}_{\mathrm{U}}$) and lower ($\mathbf{b}_{\mathrm{L}}$) bounds for each parameter:

$$x_{j,i,0} = b_{j,\mathrm{L}} + \mathrm{rand}_j(0,1) \cdot (b_{j,\mathrm{u}} - b_{j,\mathrm{L}}) \tag{4.19}$$

where $\mathrm{rand}_j(0,1)$ is a random number generator that returns uniformly distributed random numbers within the range $[0, 1)$. Alternative initialization methods are possible [47] or the parameter space may be initialized with a logarithmic distribution for parameters with bounds spanning multiple orders of magnitude, a scheme applied to another metaheuristic optimizer [63].

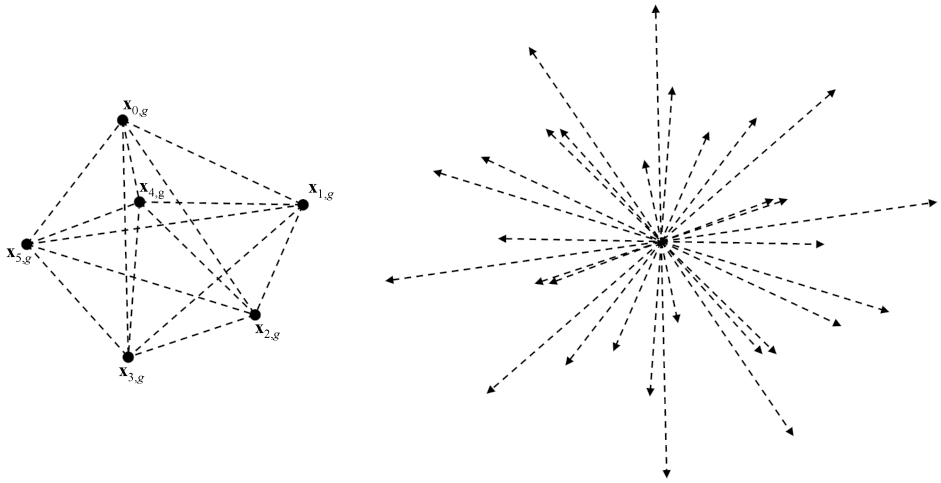


Figure 4.8: Vector differences and resulting difference vector distribution. Adapted from Price *et al.* (2005).

4.5.2 Differential Mutation

The main mechanism of producing new solutions is differential mutation, where a scaled, randomly sampled vector difference is added to a third vector. Here, a mutant vector $\mathbf{v}_{i,g}$ is generated using three randomly selected vectors from the current population $\mathbf{P}_{\mathbf{x},g}$:

$$\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}). \tag{4.20}$$

Index $r0$ denotes the *base vector* while $r1$ and $r2$ denote the *difference vectors*. Index i specifies the *target vector* which the resulting mutant undergoes recombination and competes against during selection. The scaling factor F, a positive real number, is the primary tuning variable used to control the evolution rate of the population. Typically, F falls between 0.0 and 1.0, although it may assume values > 1.0.

Figure 4.8 shows how a population of six solution vectors on the left can generate a set of unscaled vector differences on the right. Note that the distribution is symmetric about zero because each pair of vectors gives two vector differences of the same magnitude but opposite direction.

The three indices ($r0$, $r1$, $r2$) can be selected in a few ways. The most

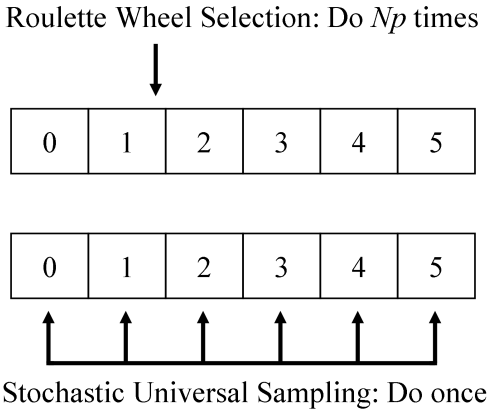


Figure 4.9: Stochastic universal sampling and roulette wheel selection. Adapted from Price *et al.* (2005).

straightforward is a random picking of all three indices but this may lead to some vectors being chosen repeatedly while others are omitted completely. If indices ($r0$, $r1$, $r2$) are not mutually exclusive, this can result in degenerate vector combinations. The differential mutation operation can thus be reduced into simple arithmetic recombination ($r1 = r0$ or $r2 = r0$), no mutation ($r1 = r2$) or even base vector duplication if no crossover occurs in the following step. In addition, degenerate combinations can also appear due to the choice of trial vector index i. If $i = r0$, the crossover step becomes mutation only. These degenerate combinations discussed are of first-order; higher order degenerate combinations are possible, but with much lower probability. An in-depth analysis of various degenerate vector combinations can be found in Price *et al.* [47].

To prevent repeated selections for base vectors, *stochastic universal sampling* can be used [88]. The N_p vectors are chosen in a single trial with the same probability for all vectors. This is contrasted by the commonly used *roulette wheel selection* where N_p single selection trials are executed [88]. Figure 4.9 contrasts the two methods.

There are two methods that adhere to stochastic universal sampling: permutation selection and random offset. In permutation sampling, base vector of indices consecutively drawn from an array of randomly permuted number sequence are paired with the sequentially numbered target vectors. For random

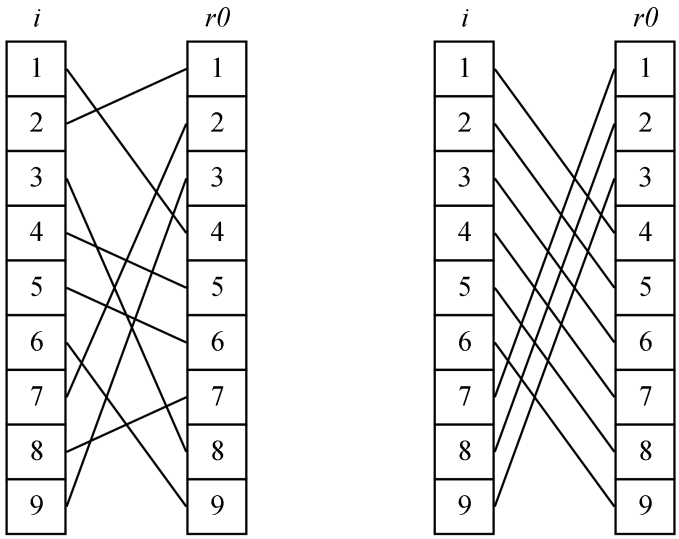


Figure 4.10: Pairing base vectors to target vectors. Adapted from Price *et al.* (2005).

offset, the base vector index is computed as the sum, modulo N_p, of the target index and a randomly generated offset. Figure 4.10 illustrates both selection techniques.

To prevent degenerate vector combinations between indices, a combination of permutation and random offset selection can be used. When pairing base vector indices to the target vectors, a single array of permuted numbers is first generated for $r0$. The $r1$ indices array can next be generated by adding a small random offset ($< N_p/2$) to the $r1$ indices array, modulo N_p. Likewise, the $r2$ indices array can be generated from $r1$'s array in a similar fashion. This ensures that no degenerate vector combinations appear.

4.5.3 Crossover

Crossover is an operator that is complementary to differential mutation in generating new solutions. In the basic algorithm, uniform (or binary) crossover of one target vector and a mutant vector is performed:

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } \left(\text{rand}_j(0,1) \leq Cr \text{ or j} = j_{\text{rand}}\right) \\ x_{j,i,g} & \text{otherwise.} \end{cases} \tag{4.21}$$

The proportion of trial parameters taken from the mutant vector is controlled by the control variable Cr. This user defined value is kept between 0 and 1.

For each j-th parameter, a uniform random number is generated and compared to the crossover probability Cr. If the random value is less than or equal to Cr, the mutant parameter $v_{j,i,g}$ is passed on to trial vector $\mathbf{u}_{i,j}$, else the target parameter $x_{j,i,g}$ passed on instead. To prevent producing a trial vector $\mathbf{u}_{i,g}$ that is identical to a target vector $\mathbf{x}_{i,g}$, a mutant parameter of a randomly chosen index j_{rand} is automatically assigned to the trial parameter.

The rule-of-thumb for the initial settings of DE are $10 \cdot D$ as the population size N_p, 0.8 for F and 0.9 for Cr [89]. Prior experiences of the DE community [47] showed that settings for most successful solutions fall within the range of [0.5,1].

In the selection of vectors for the new generation, each trial vector $\mathbf{u}_{i,g}$ is paired with its corresponding target vector $\mathbf{x}_{i,g}$ and a simple comparison of objective function scores (Φ) is made. If the trial vector is as good or better, it replaces the target vector:

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & \text{if } \Phi(\mathbf{u}_{i,g}) \leq \Phi(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise.} \end{cases} \tag{4.22}$$

Age does not play a role in selection here and the best-so-far solution is always retained regardless of age (*i.e.* elitist).

The algorithm iterates over mutation, crossover and selection until the termination condition is satisfied. Like EAs, a variety of termination conditions are possible, such as maximum allowed CPU time, maximum number of generations, the best objective function score dropping below an acceptable limit, or population statistics [47, 48]. Figure 4.11 is a flowchart that illustrates the algorithm during each iteration. The flowchart however does not explicitly indicate that $r0$, $r1$, $r2$ and i are distinct.

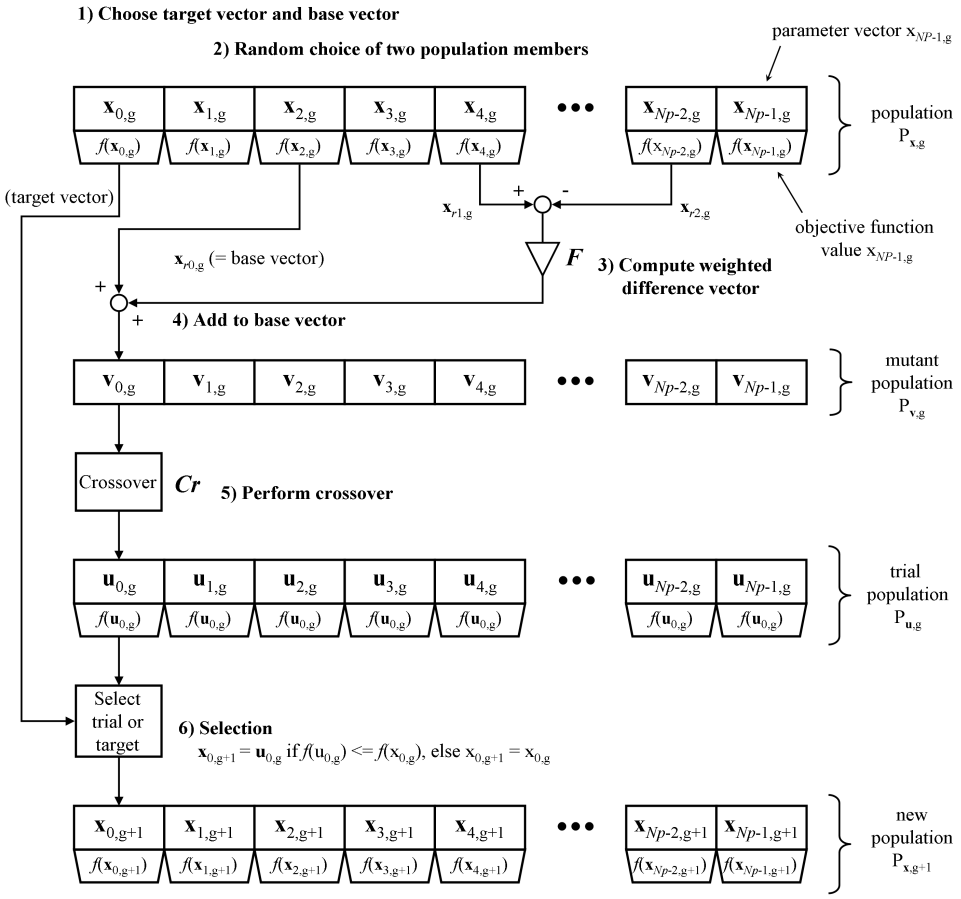


Figure 4.11: Flow chart of DE's generate-and-test loop. Adapted from Price *et al.* (2005).

4.5.4 Variants

There exist many variants of DE, two of which are discussed further below [47]. The basic DE algorithm is referred to as DE/rand/1/bin in shorthand. The first term after "DE" indicates the choice of base vector in the mutation step, while the next term "1" denotes how many vector differences added to the base vector. The last term refers to the crossover method, which in this case is binomial (bin).

DE/best/1/bin is a variant of DE where the base vector in differential mutation is replaced by the best vector currently known. This DE variant is tailored for small populations and trades reliability for fast convergence. Jitter is commonly used in this variant to improve the algorithm's reliability by adding small random perturbations to F for every parameter.

In DE/target-to-best/1/bin, each base vector lies on the line between the target vector and the best vector. The position of the base vector on the line is controlled by λ, another control variable that can be manipulated separately or set to match F. The modified differential mutation operator can be written as:

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + \lambda(\mathbf{x}_{best,g} - \mathbf{x}_{i,g}) + F(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}). \tag{4.23}$$

A minor variant uses the /rand-to-best/ option, where the target vector $\mathbf{x}_{i,g}$ in Equation 4.23 is randomly selected instead (*i.e.* $\mathbf{x}_{j,g}$ is used and $i \neq j$).

4.5.5 Application to Parameter Estimation

In the parameter estimation of oscillatory systems, the solution space poses problems to the type of search method that can be used. As described in Section 4.2, the solution space contains dynamics of different natures, of which only the limit cycle is desired. Using the formulation in the earlier sections, the solution space becomes discontinuous between oscillating and non-oscillating solutions (See Section 4.2.2). Hence, the use of DE is appropriate since it does not require gradient information in the search. Depending on the definition of parametric bounds, the solution space may be dominated by non-oscillatory solutions.

Using naive uniform seeding, a small fraction of the initial solution population containing oscillatory solutions will drastically slow down the convergence speed of the search, if not prevent convergence. There are three possible methods to improve the initial population. The first is to randomly generate solutions continuously and draw out oscillating solutions until a quota is met. The initial population is then formed using these oscillating solutions and the remaining filled by randomly generated solutions. The second method is to generate an initial population which contains at least one oscillating solution (if more than one, the best in terms of period or score) that serves as the center and generating new solutions that lie on the line between the remaining solutions and the center. The solutions are successively drawn towards the center until the quota of oscillating solutions in the population is met. The third is adapted from Evolutionary Strategy [58], where a single oscillating solution is mutated with a normally distributed random variable to produce the initial population.

4.5.6 Alternative Search Algorithms

Two other search methods, Evolutionary Strategy [58] and Scatter Search [90] were also tested for our parameter estimation. Unfortunately, both methods have drawbacks that rendered them inefficient in solving the parameter estimation problem. For Evolutionary Strategy, the main problem is the choice of step size. The geometry and size of the oscillating region is generally unknown and thus it is difficult to tune the optimizer to suit the problem. For Scatter Search, the algorithm utilizes a small primary population of "elite" vectors and is heavily reliant on random reseeding of the parameter space in generating diversity. If only oscillating solutions are admitted into the elite population vector, excessive computational effort is required if the proportion of oscillating solutions in the parameter space is relatively small. On the other hand, if non-oscillating solutions are admitted, the discontinuity between oscillating and non-oscillating solutions will prevent the leveraging of local search methods to improve non-oscillating solutions. This defeats Scatter Search's main advantage

of integrating of local search methods into the global search.

It should be stressed here that the parameter estimation problem is not restricted to any particular search algorithm. Three global search algorithms were investigated and Differential Evolution was found to be the most efficient. As surveyed in Chapter 2, a variety of methods exist and it is possible that a number of them may also be suitable. However, the purpose of this work is not to evaluate efficiency of optimization algorithms in solving the current class of problems but to show that the effectiveness of the developed framework (*i.e.* objective function).

4.6 Confidence Intervals and Identifiability

After the best parameter estimates are obtained, we can evaluate the lower bounds on the confidence intervals of these estimates using the Fisher Information Matrix (FIM) and determine the practical identifiability of the parameters. The definition of the FIM given by:

$$\text{FIM} = \text{E}\left\{ \left[\frac{\partial}{\partial \mathbf{p}} \log f(\hat{\mathbf{y}}|\mathbf{p}) \right] \left[\frac{\partial}{\partial \mathbf{p}} \log f(\hat{\mathbf{y}}|\mathbf{p}) \right]^{\text{T}} \right\}. \tag{4.24}$$

Under Gaussian assumption, the FIM is computed by:

$$\text{FIM} = \sum_{i=1}^{N} \mathbf{S}^{\text{T}}(t_i)\mathbf{V}^{-1}(t_i)\mathbf{S}(t_i) \tag{4.25}$$

where $\mathbf{S}$ is the $n \times m$ sensitivity matrix of $\mathbf{y}$ with respect to $\mathbf{p}$ and $\mathbf{V}$ is the measurement covariance matrix. The derivation of the formula is given in Appendix A. In this work, the objective function depends on period and shape. Thus, the FIM is constructed as follows:

$$\text{FIM} = \mathbf{s}_\tau^{\text{T}} \text{v}_\tau^{-1} \mathbf{s}_\tau + \sum_{i=1}^{N} \mathbf{S}_{\text{c}}^{\text{T}}(\theta_i)\mathbf{V}^{-1}(\theta_i)\mathbf{S}_{\text{c}}(\theta_i) \tag{4.26}$$

where $\mathbf{S}_{\text{c}}$ is the parametric sensitivity matrix with respect to constant period in Equation 3.14, $\mathbf{s}_\tau$ is the period sensitivity vector and v_τ is the variance of

the period average. In this work, it is assumed that the measurements are independent and identically distributed (iid), and thus $\mathbf{V}$ is diagonal.

By the Cramèr-Rao Inequality [3]:

$$\mathbf{V_p} \geq \text{FIM}^{-1} \tag{4.27}$$

gives the lower bound of the parameter covariance matrix $\mathbf{V_p}$ as the inverse of the FIM. Using the diagonal elements of $\mathbf{V_p}$, the 95% confidence intervals can be constructed as $[p^0 - 1.96\sigma_p,\ p^0 + 1.96\sigma_p]$, where p^0 are the parameter estimates and σ_p are the estimated standard deviations from the FIM.

Practically identifiable parameters are often defined as not to encompass zero within their confidence intervals [91]. If the parametric confidence intervals encompass zero, this implies that we are unable to determine with 95% confidence that the parameters are non-zero. Thus we consider these parameters to be practically unidentifiable. For p^0 to be considered practically identifiable, the following equation should hold:

$$\frac{\sigma}{p^0} < \frac{1}{1.96} \tag{4.28}$$

where σ/p^0 is also known as the Coefficient of Variation (CV). For a parameter to be practically identifiable at a 95% confidence interval, the CV needs to be less than 51%. For the ease of analysis, we use 50%.

4.7 Violation of Assumptions

In the use of MLE in this work, a number of assumptions were made. We now state them as follows:

1. Additive errors
2. Zero mean errors
3. Uncorrelated errors
4. Errors have a normal distribution with known statistical parameters.

Clearly, violation of these assumptions will have a deleterious effect on the parameter estimates and associated confidence intervals. For example, if the errors do not possess a zero mean, the resulting parameter estimates will be biased.

A common violation of assumption is the presence of correlated errors. This can be detected by analyzing the residuals with a lag plot where a random scatter shot pattern indicates an absence of correlation in the errors. If correlation is detected, the error covariance matrix equation is no longer diagonal and Equation 4.6 should be used instead. In the case of errors that are autocorrelated, one approach is assume a model for the autocorrelated errors.

Although the MLE was derived above for the normal distribution, it can also be derived for other error distributions(this also applies to MAP). This is illustrated in [4] for the family of exponential distributions including Poisson. However, the situation may arise where the error distribution is unknown. Using knowledge of the system under study, an error distribution may be guessed but caution must be applied as the estimates obtained are likely to be biased and the confidence intervals computed are no longer accurate.

Chapter 5

Parameter Estimation

In this chapter, the parameter estimation methodology was applied to three circadian rhythm models. For each model, an *in-silico* dataset of 200 hourly samples corrupted with Gaussian noise of 10% standard deviation was generated. Parameters were then fitted to this dataset and identifiability analysis was performed on the estimates obtained by computing the parameter variance using the FIM. Since the datasets are generated *in-silico* using known parameters and without plant-model mismatch, accuracy of the estimates could be checked, particularly for the identifiable parameters. The purpose of this approach is to judge efficacy of the methodology developed in recovering model parameters.

To study the effects of noise and sampling time on the identifiability of parameters, datasets with different noise levels (5%−25%) and sampling times (0.25 hr−2 hr) were generated. Parameters were re-estimated for the respective datasets and the corresponding variance computed. By observing the parameter variance changes, the effects of reducing sampling time and noise reduction on parameter identifiability were compared.

For the parameter estimation, the Differential Evolution algorithm was implemented in C language [47] with the initial population seeding modified as described in Section 4.5.5. The DE/rand-to-best/1/bin variant was used and the additional control variable λ was manipulated separately. Parameter bounds used for the initial population seeding were enforced during the subsequent search

due to the large number of numerically unstable solutions generated if the bounds were removed. Although these solutions were naturally discounted due to their large objective function scores, the effectiveness of the search algorithm was severely affected.

The CVODE solver package [92] was used to numerically solve the model ODEs and the FFTW3 library [93] was used for DFT computation. Further, the Intel® MPI (Message Passing Interface) library was used to enable parallel computation. All parameter estimation computations were performed on an Intel Dual Core 1.6GHz computational cluster with 112 compute nodes. For the estimation of parameter variance based on the FIM, MATLAB® [94] was used as the computation platform and the ode15s solver was used to solve the relevant ODE systems.

5.1 2-state Tyson Model

In the first and simplest example, a two state, nine parameter *Drosophila* circadian model by Tyson *et al.* [32] was used. The present model was simplified from a larger model of six differential equations with the mechanism illustrated in Figure 5.1. To reduce the 6 state model, PER and TIM were lumped into a single specie as experimental data showed similar time profiles for both proteins. The model was further reduced by assuming that dimerization reactions are fast enough for the monomer and dimer concentrations to be at equilibrium. The final model comprises of two differential equations and one algebraic expression given by:

$$\begin{aligned} \frac{\mathrm{d}M}{\mathrm{d}t} &= \frac{2}{1+(P_\mathrm{t}(1-q)/2P_\mathrm{crit})^2} - k_\mathrm{m}M \\ \frac{\mathrm{d}P_\mathrm{t}}{\mathrm{d}t} &= v_\mathrm{p}M - \frac{k_\mathrm{p1}P_\mathrm{t}q + k_\mathrm{p2}P_\mathrm{t}}{J_\mathrm{p}+P_\mathrm{t}} - k_\mathrm{p3}P_\mathrm{t} \end{aligned} \tag{5.1}$$

with

$$q = \frac{2}{1+\sqrt{1+8K_\mathrm{eq}P_\mathrm{t}}}$$

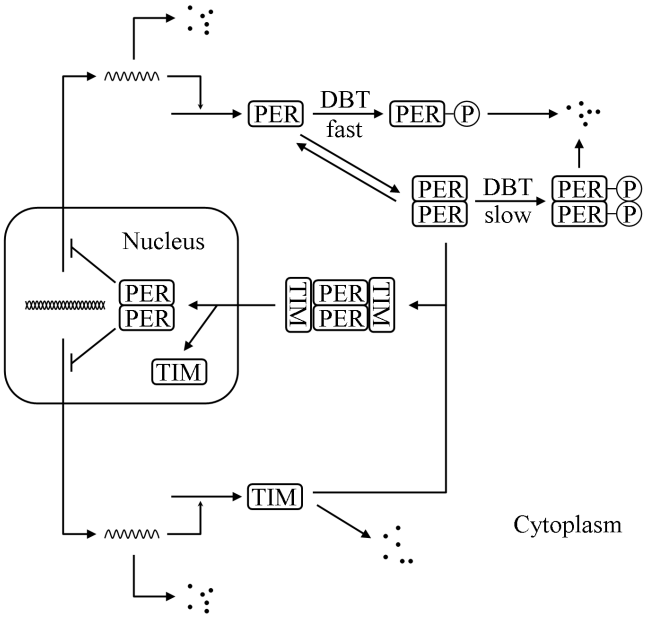


Figure 5.1: Molecular mechanism of the circadian clock, adapted from Tyson et al. (1999).

where M and P_{t} represents the *per* mRNA and PER protein concentrations respectively.

5.1.1 Parameter Estimation

As explained in Section 4.1, the initial conditions (or concentrations) were included in the parameter search and therefore the number of search variables totaled to 11. The nine kinetic parameters were constrained to 0.1 to 10 times of their true values and $\pm 2\sigma$ for both initial concentrations. The DE strategy used was DE/rand-to-best/1/bin with the following settings: $F = 0.8$, $\lambda = 0.8$, $Cr = 0.9$ and a population size of 100.

Figure 5.2 compares the simulation using best fit parameter estimates with *in-silico* data, showing a good agreement. The best fit parameter estimates are listed in Table 5.1 with the corresponding true values, as well as the percent deviations and Coefficient of Variations (CV). The CVs were computed based on the FIM discussed in Section 4.6. Among the nine parameters, six are within 10% of their true values while the remaining (k_{p1}, k_{p2}, K_{eq}) show deviations

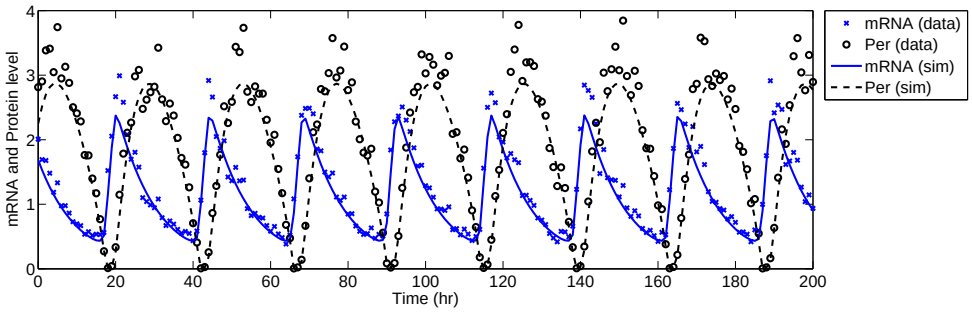


Figure 5.2: Comparison of best fit simulation with data (10% noise, 200 samples) for the 2-state Tyson model.

Parameters	True	Estimates	% Deviation	Std. Dev.	% CV
v_{m}	1	0.99766	0.23449	0.02393	2.39810
k_{m}	0.1	0.10224	2.23587	0.00171	1.66785
v_{p}	0.5	0.48697	2.60533	0.00966	1.98346
k_{p1}	10	7.87814	**21.21864**	3.37335	42.81916
k_{p2}	0.03	0.01820	**39.31867**	0.04880	**268.05608**
k_{p3}	0.1	0.10156	1.56184	0.01204	11.85233
K_{eq}	200	137.89132	**31.05434**	107.51092	**77.96787**
P_{crit}	0.1	0.09742	2.57611	0.00646	6.63155
J_{p}	0.05	0.04557	8.86806	0.00782	17.17000
M	1.86056	1.81217	2.60050	0.02758	1.52200
P_{t}	2.60609	2.64742	1.58564	0.13786	5.20748

Table 5.1: Best fit parameter estimates of the 2-state Tyson model.

up to 40% (highlighted in bold). k_{p1}, k_{p2} are the V_{max} constants for monomer and dimer phosphorylation respectively, and K_{eq} is the equilibrium constant for dimerization. These three parameters also have large CVs and two of them (k_{p2}, K_{eq}) are practically unidentifiable based on the 95% confidence interval, that is the CVs are greater than 50%. The results match the analysis of the model which found that the oscillation period was approximately 24 hr and was insensitive to K_{eq} or k_{p1} when $K_{\mathrm{eq}} > 100$ [32].

The estimated parameters generally show good agreement with the true parameter values and the simulation result fits well with the data, indicating a successful parameter search. Although the model is simple with a small number of parameters, this example demonstrates the feasibility of the proposed methodology.

5.1.2 Effect of Noise

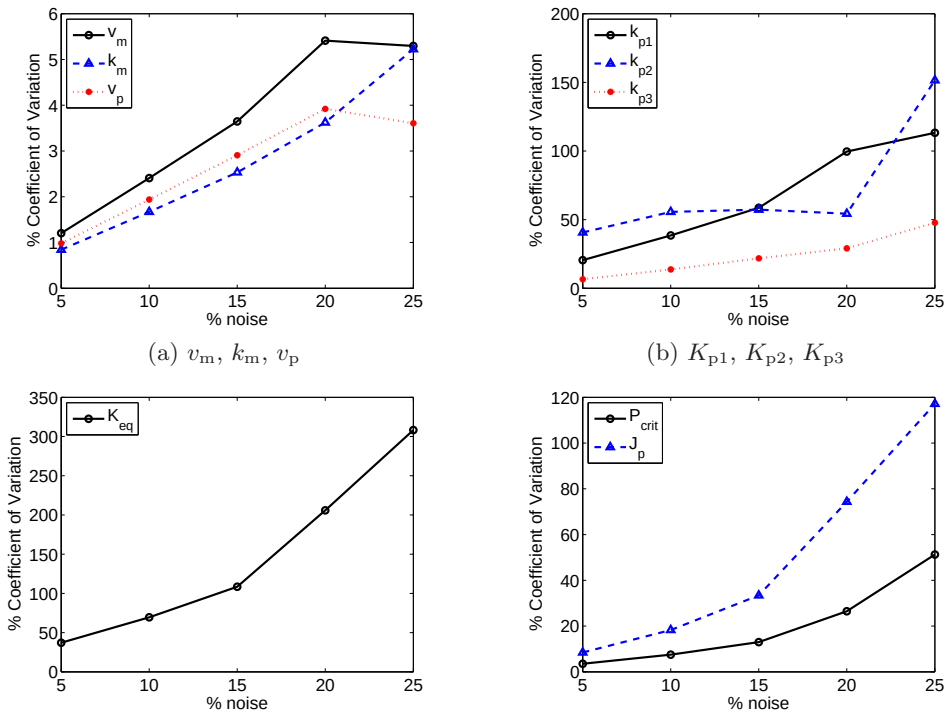
To study the effect of noise, the parameter search was repeated for different noise levels from 5% to 25% in 5% increments while the sampling time (1 hr) and number of samples (200) were kept the same as in the initial parameter estimation. Figure 5.3 shows how the CVs of the parameters vary with different noise levels. As expected, decreasing noise levels lead to corresponding decreases in CVs, though exceptions exist. If the parameter estimates obtained from one dataset (*e.g.* with 10% noise) are reused to compute the CVs for all noise levels, smooth monotonic plots for all parameters will be obtained, and this is expected from the FIM computation in Equation 4.26. However, parameters re-estimated for each dataset will differ due to different noise realizations. For v_{m} and v_{p}, the parameter estimates for 25% noise show larger deviations compared to estimates at lower noise levels and the effect of the deviations on the CV is larger than the effect of higher noise level. For k_{p2}, the parameter is practically unidentifiable and the estimates for each noise level show large deviations from the true value ($> 100\%$).

5.1.3 Effect of Sampling Time

To study the effect of sampling time on parameter identifiability, sampling time was also varied between 0.25 hr and 2 hr, while noise was fixed at 10% and the simulation time remained at 200 hr. Figure 5.4 shows how CVs of the parameters vary with different sampling times. As expected, decreasing sampling times also lead to decreasing CVs.

5.1.4 Noise vs. Sampling Time

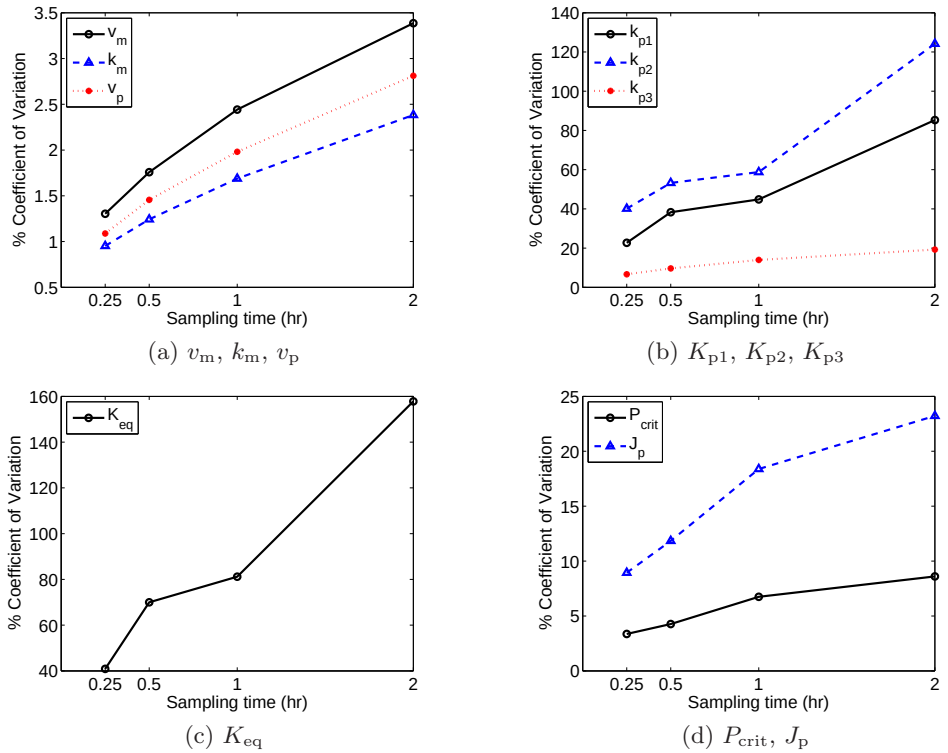
Increasing the number of measurements will generally increase the information content of the dataset and consequently improve parametric identifiability. More measurements can be taken at a faster rate or in more replicates. The purpose of this comparison is to determine which of these gives a greater improvement in terms of reducing CV. In this case, the number of readings N was quadrupled,



(a) v_{m}, k_{m}, v_{p} (b) K_{p1}, K_{p2}, K_{p3}

(c) K_{eq} (d) P_{crit}, J_{p}

Figure 5.3: % CVs for different noise levels in the 2-state Tyson model.



(a) v_{m}, k_{m}, v_{p} (b) K_{p1}, K_{p2}, K_{p3}

(c) K_{eq} (d) P_{crit}, J_{p}

Figure 5.4: % CVs for different sampling times in the 2-state Tyson model.

Parameter	Sampling	Noise
v_{m}	-1.62873	**-3.00346**
k_{m}	-1.14077	**-1.95367**
v_{p}	-1.35604	**-1.98040**
k_{p1}	-47.02695	**-61.10426**
k_{p2}	-70.94138	1.32942
k_{p3}	-9.65761	**-15.34328**
K_{eq}	-87.80901	**-136.37960**
P_{crit}	-4.34053	**-18.94790**
J_{p}	-11.37933	**-56.06638**

Table 5.2: Comparison of % CV changes due to sampling time decrease and noise reduction in the 2-state Tyson model.

by either taking four replicates and thereby reducing noise by half (from 20% to 10%), or decreasing the sampling time from 2hr to 0.5hr. The changes in CV are shown in Table 5.2. With the exception of k_{p2}, a reduction in measurement noise (*i.e.* more replicates) results in a greater improvement in CV (highlighted in bold) over a decrease in the sampling time. For k_{p2}, it counter-intuitively shows a slight increase in CV when the percent noise was reduced. As mentioned earlier in Section 5.1.2, the parameter is unidentifiable and exhibit large deviations in the parametric estimates. This renders the CVs computed unreliable and thus inappropriate for the comparison. Excluding k_{p2}, all other parameters unequivocally show that a noise reduction gives a larger improvement in terms of CV. These results suggest that in wet-lab experiments, having more replicates is better than a higher sampling rate in improving parameter identifiability for the same increase in total readings.

5.2 5-state Goldbeter Model

In the second study, the *Drosophila* circadian model by Goldbeter [30] was used. The circadian oscillations of PER is modeled with five species: *per* mRNA (M), PER protein (P_0), and the mono- and bi- phosphorylated forms (P_1, P_2), and nuclear PER (P_N). The source of oscillations is the negative feedback of nuclear PER suppressing the transcription of *per* mRNA and delay due to the phosphorylation of PER protein before transportation into the nucleus. This is different

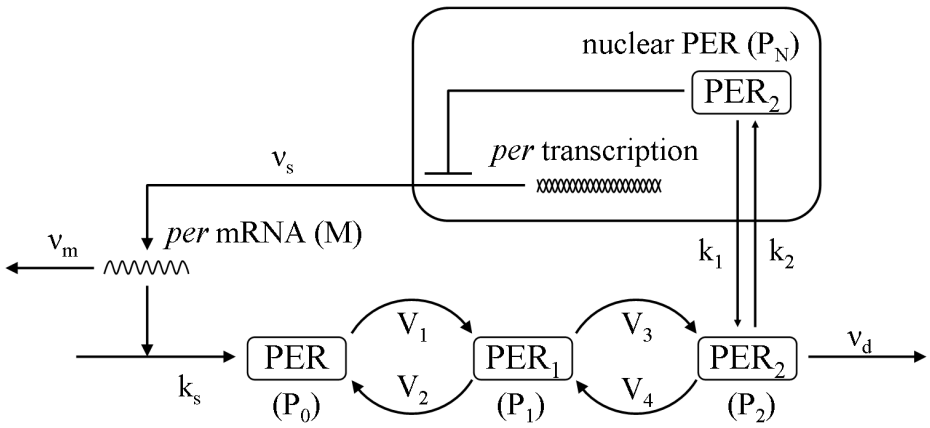


Figure 5.5: Molecular mechanism of the circadian clock, adapted from Goldbeter (1995).

from the coupled action of positive and negative feedback, as well as the role of phosphorylation generating the positive feedback in the previous Tyson *et al.* model. In the Goldbeter model, a delay in the feedback loop favors sustained oscillations and also gives a phase relationship between *per* mRNA and total PER protein levels that is consistent with experimental observations, in which *per* mRNA level peaks about 4 hours prior to the maximum of total PER protein. Figure 5.5 shows the regulatory network model.

The 5-state ODE model:

$$
\begin{aligned}
\frac{\mathrm{d}M}{\mathrm{d}t} &= v_\mathrm{s}\frac{K_\mathrm{I}^n}{K_\mathrm{I}^n + P_\mathrm{N}^n} - v_\mathrm{m}\frac{M}{K_\mathrm{m} + M} \\
\frac{\mathrm{d}P_0}{\mathrm{d}t} &= k_\mathrm{s}M - V_1\frac{P_0}{K_1 + P_0} + V_2\frac{P_1}{K_2 + P_1} \\
\frac{\mathrm{d}P_1}{\mathrm{d}t} &= V_1\frac{P_0}{K_1 + P_0} - V_2\frac{P_1}{K_2 + P_1} - V_3\frac{P_1}{K_3 + P_1} - V_4\frac{P_2}{K_4 + P_2} \\
\frac{\mathrm{d}P_2}{\mathrm{d}t} &= V_3\frac{P_1}{K_3 + P_1} - V_4\frac{P_2}{K_4 + P_2} - k_1P_2 + k_2P_\mathrm{N} - v_\mathrm{d}\frac{P_2}{K_\mathrm{d} + P_2} \\
\frac{\mathrm{d}P_\mathrm{N}}{\mathrm{d}t} &= k_1P_2 - k_2P_\mathrm{N}
\end{aligned}
\tag{5.2}
$$

where M is the *per* mRNA and the PER protein levels are represented by P with the subscripts $0-2$ denoting non-phosphorylated, mono-phosphorylated and bi-phosphorylated PER respectively, and N indicating bi-phosphorylated PER levels in the nucleus. There are 18 kinetic parameters and 5 initial concentrations

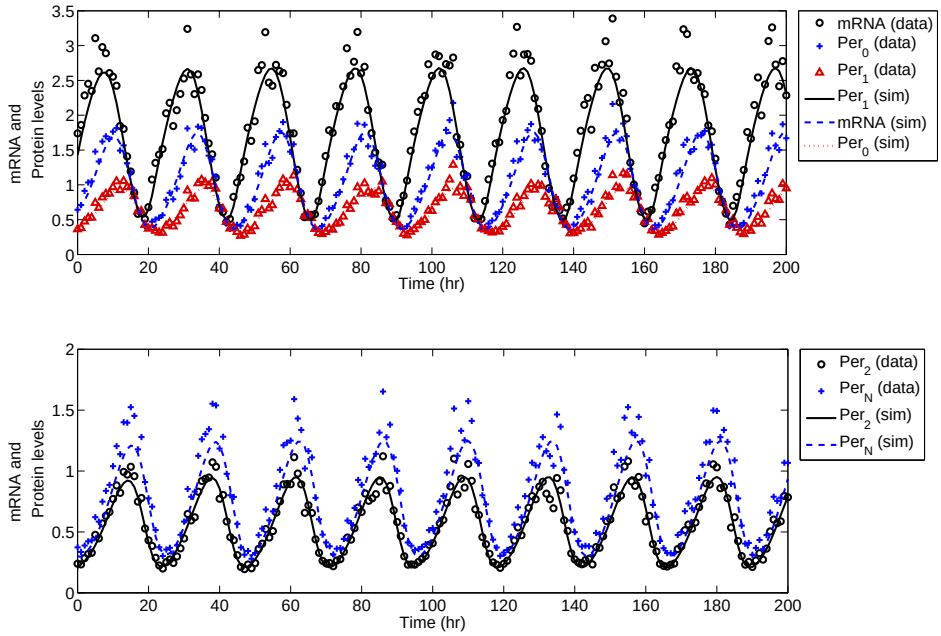


Figure 5.6: Comparison of best fit simulation with data (10% noise, 200 samples) for the 5-state Goldbeter model.

in this model, a total of 23 parameters to be estimated. The size of the present problem is twice that of the previous example and thus the DE population size was doubled to 200, while the strategy and control settings were kept the same (DE/rand-to-best/1/bin, $F = 0.8$, $\lambda = 0.8$, $Cr = 0.9$).

5.2.1 Parameter Estimation

The true parameters used in this study were taken from Goldbeter [30] and given in Table 5.3. Again, an hourly sampled dataset for 200 hours with 10% Gaussian noise was generated. The parameter search space was constrained to between 0.1 to 10 times of the true values and $\pm 2\sigma$ for the initial concentrations. The resulting best fit simulation is compared with the data in Figure 5.6. Table 5.3 gives the parameter estimates of the best fit simulation, as well as percent deviations and CVs.

In summary, 16 of the 18 parameter deviate less than 20% from their true values while the remaining two (K_2 and V_2) show deviations up to 50% (high-

Parameters	True	Estimates	% Deviation	Std. Dev.	% CV
v_{s}	0.76	0.78292	3.01593	0.09480	12.10793
v_{m}	0.65	0.69508	6.93508	0.06727	9.67859
K_{m}	0.5	0.52143	4.28494	0.15038	28.84065
k_{s}	0.38	0.37126	2.30029	0.00676	1.82142
v_{d}	0.95	0.87293	8.11313	0.01793	2.05358
k_1	1.9	1.90992	0.52233	0.17180	8.99521
k_2	1.3	1.30667	0.51291	0.11597	8.87530
K_{I}	1	0.99541	0.45948	0.05704	5.73062
K_{d}	0.2	0.17143	14.28481	0.01446	8.43180
n	4	3.84130	3.96758	0.61767	16.07971
K_1	2	2.15445	7.72256	0.53161	24.67488
K_2	2	2.97533	**48.76670**	3.41867	**114.90020**
K_3	2	1.89402	5.29892	0.59143	31.22588
K_4	2	1.81320	9.34015	1.31103	**72.30470**
V_1	3.2	3.22470	0.77189	0.35931	11.14242
V_2	1.58	2.03879	**29.03728**	1.62021	**79.46930**
V_3	5	4.75132	4.97363	0.83768	17.63045
V_4	2.5	2.34909	6.03628	0.97479	41.49627
M	1.61043	1.55651	3.34791	0.02708	1.73973
P_0	0.59515	0.60087	0.96124	0.04156	6.91702
P_1	0.33635	0.35086	4.31383	0.03344	9.53003
P_2	0.22122	0.23398	5.76935	0.02311	9.87596
P_{N}	0.34592	0.35838	3.60310	0.03122	8.71086

Table 5.3: Best fit parameter estimates of the 5-state Goldbeter model.

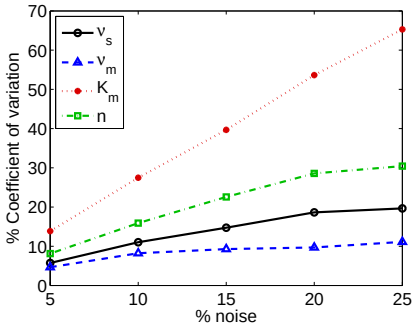
lighted in bold). The standard deviations computed from the FIM show that K_2, K_4 and V_2 are practically unidentifiable based on the 95% confidence interval (highlighted in bold), thus explaining the large deviations of K_2 and V_2. These unidentifiable parameters make up three of the four Michaelis-Menten kinetic parameters of the backward phosphorylation reactions. The fourth (V_4), although practically identifiable, still exhibits a large CV of 41%. This implies that the oscillations are not sensitive to these parameters.

5.2.2 Effect of Noise

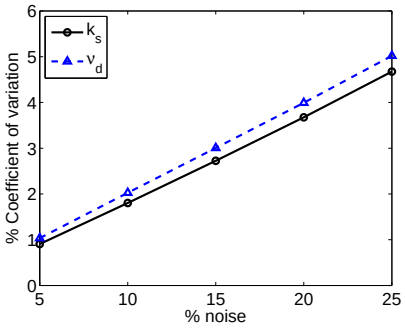
To investigate the effects of noise, a study similar to the previous example was performed. The sampling time (1 hr) and number of samples (200) were maintained while the noise level was varied from 5% to 25% at 5% increments. Figure 5.7 shows the CVs computed for different noise levels. Again, decreasing noise levels lead to decreasing CVs, though with a major exception in V_2. The CVs computed for V_2 show that the parameter is identifiable for all noise levels and this in conflict with the analysis above (Section 5.2.1), where the computed CV indicate that the parameter is unidentifiable. In the initial parameter estimation, the value of 2.04 was obtained for V_2 while the estimate obtained in this section is 1.54 and closer to the true value of 1.58. This shows that identifiability analysis for parameters can be affected by deviations in the parameter estimates. In the estimates obtained, most parameters exhibit larger deviations for higher noise levels. For most parameters, the effect on the trends of computed CVs is fairly small except for V_2, where an opposite trend in CV is observed between 10% and 20% noise levels.

5.2.3 Effect of Sampling Time

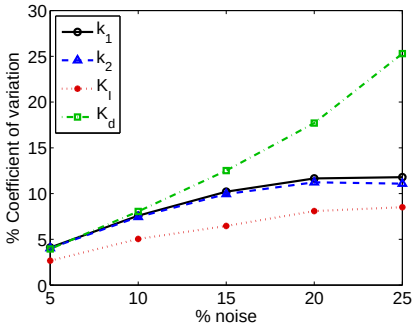
Figure 5.8 shows how the CV of the parameters vary with different sampling times. Again, the noise level was fixed at 10% and the total time remained at 200 hr. As expected, decreasing sampling times lead to decreases in CV for most parameters. However, K_2 and V_2 in Figure 5.8e show abnormally large values of



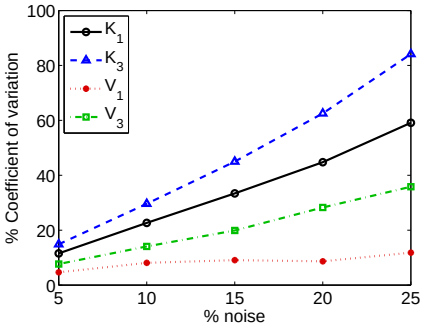
(a) v_{s}, v_{m}, K_{m}



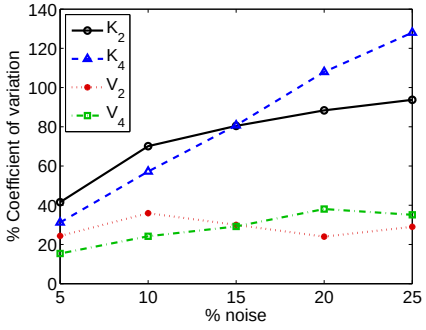
(b) k_{s}, v_{d}



(c) k_1, k_2, K_{I}, K_{d}

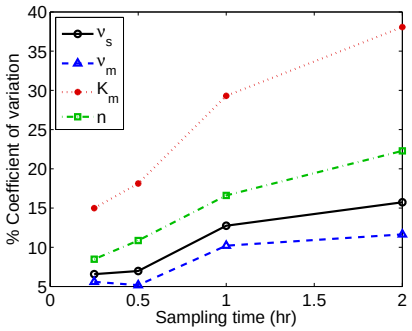


(d) K_1, K_3, V_1, V_3

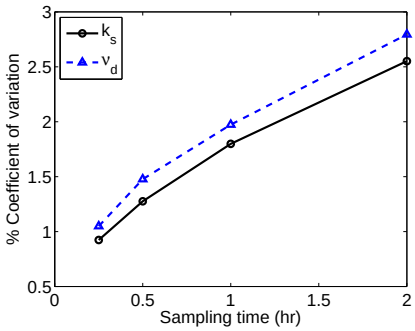


(e) K_2, K_4, V_2, V_4

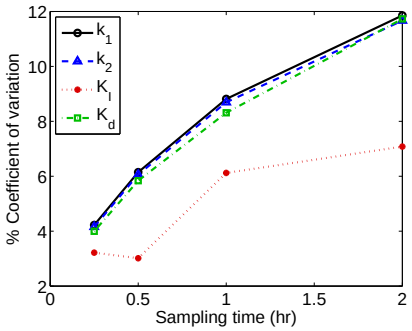
Figure 5.7: % CVs for different noise levels in the 5-state Goldbeter model.



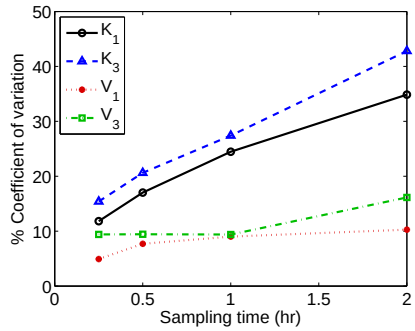
(a) v_{s}, v_{m}, K_{m}



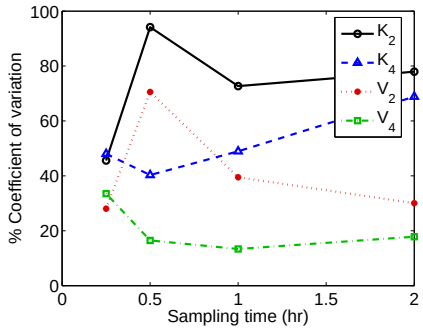
(b) k_{s}, v_{d}



(c) k_1, k_2, K_{I}, K_{d}



(d) K_1, K_3, V_1, V_3



(e) K_2, K_4, V_2, V_4

Figure 5.8: % CVs for different sampling times in the 5-state Goldbeter model.

CVs at the sampling time of 0.5hr and this can be attributed to the large positive deviations in the parameter estimates ($> 74\%$). In addition, K_2 is practically unidentifiable for all but the smallest sampling time (0.25hr). For K_4 and V_4, the large increase in the CV for the smallest sampling rate (0.25hr) can also be attributed to the positive deviations in parameter estimates, as opposed to the negative deviations for estimates at other sampling rates.

5.2.4 Noise vs. Sampling Time

As in the 2-state Tyson model, changes in the CV from noise reduction (20% to 10%) and decrease in sampling time (2 hr to 0.5 hr) are compared in Table 5.4 with the larger changes in the CV highlighted in bold. For K_2 and V_2, the comparisons are ignored due to positive changes in the CV. As mentioned in the above sections, the parameter estimates obtained show large deviations and the effect of these deviations on the CV is larger than those of reducing noise or sampling time. This results in unexpected positive changes for reduced noise or sampling time. For the remaining 16 parameters, sampling time decrease produce greater improvement for 6 parameters while the remaining 10 show larger decreases in the CV due to noise reduction. This suggests that noise reduction is still preferable in improving parameter identifiability.

In Figure 5.7, four parameters (K_{m}, K_2, K_3 and K_4) are unidentifiable at higher noise levels ($> 15\%$). Inspection of Table 5.4 shows that, with the exception of K_2, noise reduction results in greater improvement on identifiability for these parameters, further reinforcing the earlier suggestion on the effectiveness of noise reduction.

The comparisons of replicate versus sampling time for the 2-state Tyson (previous example) and 5-state Goldbeter models show that poor parameter estimates due to different noise realizations and parametric unidentifiability affect the reliability of the CVs computed. An alternative approach is to employ the Monte Carlo method to estimate parameter variance for different noise levels and sampling times.

5.2.5 Limited Dataset

Presently, modeling efforts on biological systems are often hampered by incomplete datasets. Measurements are available for only certain mRNA or protein concentrations and this poses a problem during data reconciliation with models. Here, the effect of missing measurements on certain states is investigated. For this study, only measurements of the mRNA (M) and non-phosphorylated

Parameter	Sampling	Noise
v_s	**-8.75210**	-7.60981
v_m	**-6.47071**	-1.46025
K_m	-19.94299	**-26.16486**
k_s	-1.27653	**-1.87398**
v_d	-1.31863	**-1.96850**
k_1	**-5.70837**	-4.06932
k_2	**-5.60854**	-3.79022
K_I	**-4.06717**	-3.04671
K_d	-5.89206	**-9.65584**
n	-11.42775	**-12.65854**
K_1	-17.83286	**-22.12054**
K_2	16.20834	-18.26299
K_3	-22.19166	**-32.93368**
K_4	-28.56870	**-50.76704**
V_1	**-2.57046**	-0.54661
V_2	40.52213	11.96134
V_3	-6.70453	**-14.17805**
V_4	-1.34974	**-13.88948**

Table 5.4: Comparison of % CV changes due to sampling time decrease and noise reduction in the 5-state Goldbeter model.

PER (P_0) were assumed to be available. The noise level (10%) and number of samples (200) were unchanged. The true parameters and bounds were also kept unchanged except for the initial concentrations of the unmeasured states, P_1, P_2 and P_N. Since these states were not measured and thus no *a priori* knowledge, they were allowed to range between 0 and 2.

Figure 5.9 shows the best fit simulation compared to the data. In the upper panel, the simulated system fits well to the measured states. However, the lower panel shows considerable discrepancy between the model and the unmeasured states. This is not unexpected since these states were not measured. One interesting observation is that despite the discrepancy in the magnitude of the peaks and troughs between model simulation and data, the phase behavior is captured by the model. Since phase is arguably the more important aspect of circadian rhythms, the resulting fit may be considered acceptable under such circumstances with missing data.

Table 5.5 compares the true and estimated values of the parameters. From the percent deviations, only 5 out of 18 estimates show deviations of less than

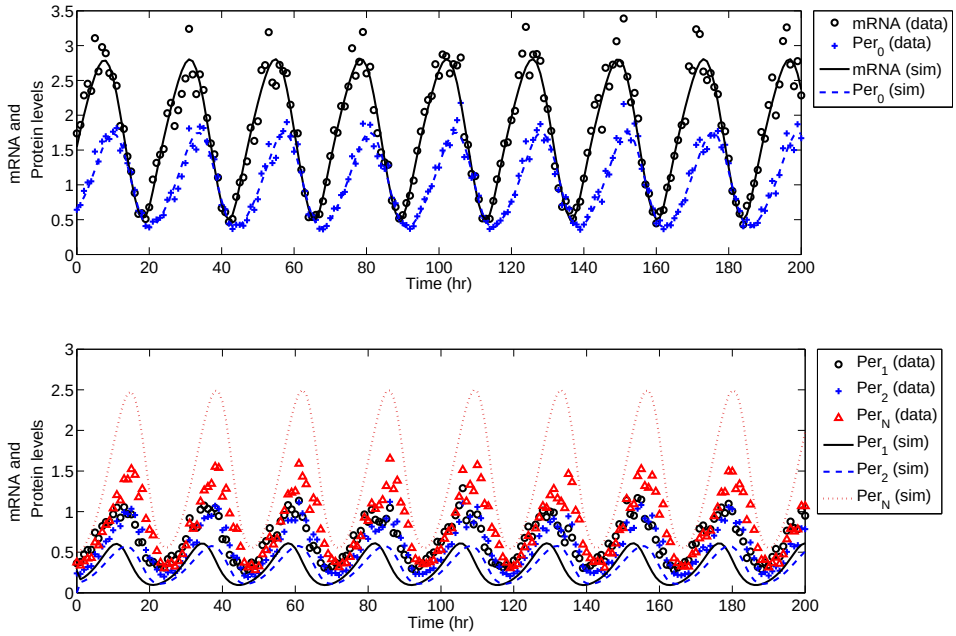


Figure 5.9: Comparison of best fit simulation with limited data (2 measured states) for the 5-state Goldbeter model.

10%, while computation of standard deviation with the FIM shows that all 18 parameters are practically unidentifiable (highlighted in bold). Some of the estimates for parameters K_{I}, K_4 and V_4 hit their respective upper or lower bounds, implying that a better fit can be found outside these bounds. For population based search algorithms such as DE to be efficient, the global optimum should be within the initial parameter bounds defined [47], or it may result in a failure to find the global optimum. However, no suitable bounds could be found for these parameters as they grew unbounded towards infinity or 0 when the bounds were removed. To obtain reasonable estimates, the bounds for these parameters were maintained. Wider bounds can be also used but these give negligible improvement in the objective function value and thus the fit (results not shown).

Among the estimates of the initial concentrations, the measured states expectedly show small deviations and are thus identifiable, while the remaining three show large deviations and are unidentifiable. Since these states are unmea-

Parameters	True	Estimates	% Deviation	Std. Dev.	% CV
v_s	0.76	0.70416	7.34697	0.37529	**53.29635**
v_m	0.65	0.60042	7.62779	0.37248	**62.03733**
K_m	0.5	0.60069	**20.13817**	0.90999	**151.48975**
k_s	0.38	0.37814	0.49021	0.32500	**85.94695**
v_d	0.95	0.96724	1.814321	1.35379	**139.96488**
k_1	1.9	0.58607	**69.15404**	25.52322	**4354.9538**
k_2	1.3	0.60981	**53.09169**	1.59578	**261.68575**
K_I	1	0.10000	**90.00000**	2.31706	**2317.0595**
K_d	0.2	0.03102	**84.48976**	0.72009	**2321.3506**
n	4	2.46687	**38.32831**	8.66061	**351.07723**
K_1	2	1.56966	**21.51679**	2.34826	**149.60269**
K_2	2	8.91748	**345.87378**	91.48826	**1025.9435**
K_3	2	17.04129	**752.06458**	635.25133	**727.71824**
K_4	2	19.99971	**899.98549**	1059016.4	**5295158.9**
V_1	3.2	2.89102	9.65556	5.26519	**182.12210**
V_2	1.58	2.42469	**53.46108**	20.75342	**855.92225**
V_3	5	6.26821	**25.36420**	218.04875	**3478.6448**
V_4	2.5	0.25001	**89.99977**	12917.435	**5166855.3**
M	1.61043	1.78891	11.08290	0.17367	9.70807
P_0	0.59515	0.64496	8.36923	0.06384	9.89807
P_1	0.33635	0.84551	**151.37990**	1.64446	**194.49391**
P_2	0.22122	0.43009	**94.41387**	1.94377	**451.94983**
P_N	0.34592	5.4×10^{-09}	**100.00000**	1.28592	$\mathbf{2.4 \times 10^{10}}$

Table 5.5: Best fit parameter estimates of the 5-state Goldbeter model with incomplete measurements.

sured, the deviations and lack of identifiability for their initial concentrations can be expected.

5.3 10-state Goldbeter Model

In the third and final example, a 10 state, 38 parameter model of *Drosophila* circadian oscillations [95] was used. Both PER and TIM proteins are modeled in the system as two coupled negative feedback loops. The mechanism that produces oscillating protein levels is the negative feedback due to the repression of PER and TIM mRNA transcription by the nuclear PER-TIM protein complex. Similar to Goldbeter model in the second example, protein phosphorylation of PER and TIM in this model serves as time delay for the feedback loops. Figure

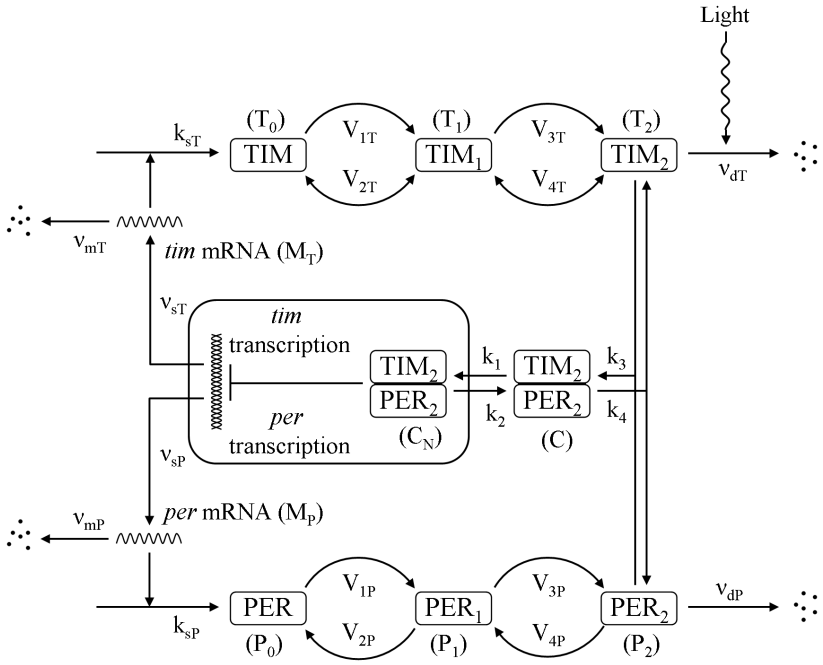


Figure 5.10: Molecular mechanism of the circadian clock, adapted from Goldbeter (1998).

5.10 shows the model scheme. The system was initially modeled with symmetric kinetic parameters for PER and TIM but alternate asymmetric parameters [96] were chosen for this study instead.

5.3.1 Parameter Estimation

With 38 unknown parameters and 10 initial concentrations, this nonlinear parameter estimation problem was challenging. As in the previous examples, an *in-silico* data set of 200 hourly samples and 10% noise was generated and used for parameter estimation. The kinetic parameters were also constrained to 0.1 to 10 times of the true values except for k_{d}, k_{dC}, k_{dN} and n. Parameters k_{d}, k_{dC}, k_{dN} are reaction constants for non-specific degradation that are included in the model to ensure existence of steady solutions during the inhibition of other specific degradation processes [95]. They were constrained between 0 and 10 times of their true values to allow the processes to be switched off with $k = 0$. n is the Hill coefficient and was constrained between 1 and 10, and the initial concentrations were constrained within $\pm 2\sigma$. The model equations are provided

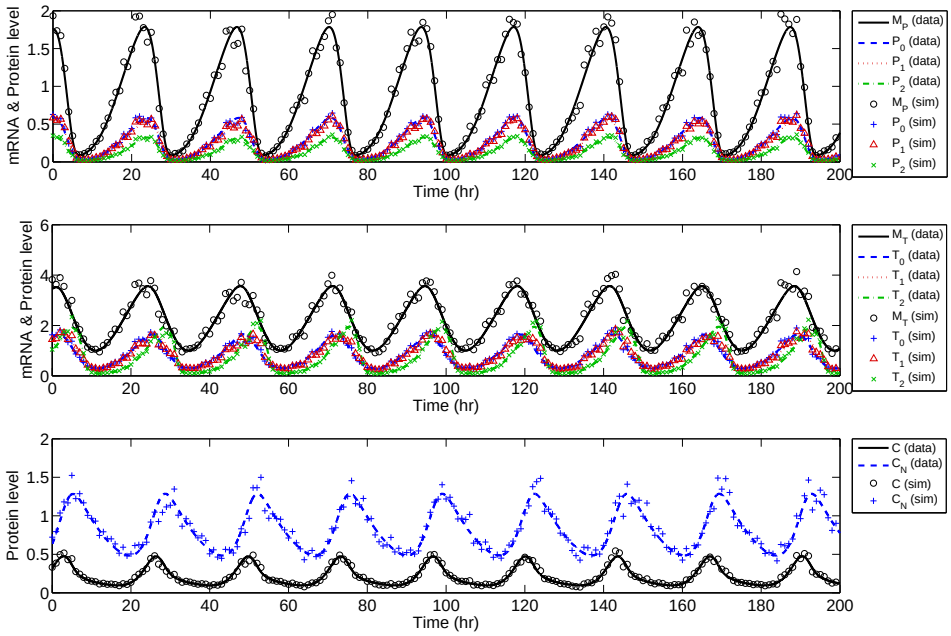


Figure 5.11: Comparison of best fit simulation with data (10% noise, 200 samples) for the 10-state Goldbeter model.

in Appendix B for brevity.

For the DE search, the strategy remained as DE/rand-to-best/1/bin and, the settings found to solve the problem were $F = 0.5$, $\lambda = 0.3$ and $Cr = 0.9$. The recommended settings for F and λ did not work for this example and the above settings were determined by trial and error to successfully solve the problem. In the initial attempts to estimate the parameters, it was found that estimates of a few parameters ($K_{2\mathrm{P}}$, $K_{4\mathrm{P}}$ and $K_{2\mathrm{T}}$) hit their respective upper bounds, implying that a better fit can be found outside these bounds. This is similar to the situation encountered with limited measurements for the 5 state Goldbeter model in Section 5.2.5. No suitable bounds could again be found for these parameters as they increased unbounded towards infinity when the bounds were removed. Thus, the bounds for these parameters were again maintained in order to recover reasonable estimates. Wider bounds were tested, but they give negligible improvement in the the objective function value. In the identifiable analysis below, it was found that these parameters are not identifiable and it is

suggested that they can be removed from future parameter estimation efforts.

The best parameter estimates and corresponding CVs are listed in Table 5.6 and 5.7. The corresponding model output is compared with the data in Figure 5.11. The fit between data and model is again excellent, even with the use of bounds on the parameters.

Among the 16 Michaelis-Menten kinetic parameters for phosphorylation of PER and TIM (K_{1P-4P}, K_{1T-4T}, V_{1P-4P}, V_{1T-4T}), 12 are not practically identifiable (highlighted in bold). In particular, kinetic parameters of the backward reactions are unidentifiable and the estimates also show large deviations ($> 20\%$) from their true values (highlighted in bold). With respect to the forward reactions, phosphorylation kinetic parameters for PER protein are also not identifiable, although the estimates show less than 13% deviation. In contrast, the forward reaction kinetic parameters of TIM protein are identifiable with correspondingly low deviations. We can deduce that the system is insensitive to the backward reaction parameters from these results and this is similar to the second example. In addition, due to the choice of parameters for the asymmetric model, the system is also not sensitive to the PER protein phosphorylation forward reaction parameters.

The degradation parameters (k_{d}, k_{dC}, k_{dN}) are unidentifiable and the parameter estimates also show large deviations, with the estimate for k_{dN} approaching 0 while k_{dC} is about 8 times its true value. As mentioned earlier, these parameters are non-specific degradation terms that are not critical for oscillatory behavior and this helps to explain their large standard deviations.

5.3.2 Parameter Estimation with Phase Response Curve

The Phase Response Curve (PRC) is a tool regularly used to characterize circadian rhythms [19, 97, 98] since phase behavior is a very important aspect of circadian rhythms, and PRCs are useful and easily available measures of phase behavior. Experimental PRC measurements are easy to elicit and widely available, in contrast to protein and mRNA datasets which are lacking and are sparse

Parameters	True	Estimates	% Deviation	Std. Dev.	% CV
v_{sP}	1.1	1.05998	3.63825	0.03758	3.54540
v_{sT}	1.0	0.96380	3.61994	0.05177	5.37104
v_{mP}	1.0	0.95440	4.56041	0.02870	3.00755
v_{mT}	0.7	0.68269	2.47248	0.05018	7.34999
v_{dP}	2.2	2.14085	2.68845	0.15183	7.09206
v_{dT}	3.0	2.90303	3.23241	0.06579	2.26609
k_{sP}	0.9	0.88257	1.93725	0.03359	3.80643
k_{sT}	0.9	0.88444	0.01729	0.02548	0.02880
k_1	0.8	0.80030	1.72932	0.02548	2.88037
k_2	0.2	0.21032	5.16084	0.01466	6.97240
k_3	1.2	1.26984	5.81955	0.06749	5.31498
k_4	0.6	0.58417	2.63841	0.04767	8.15967
K_{mP}	0.2	0.20544	2.71880	0.02944	14.32909
K_{mT}	0.2	0.29027	**45.13398**	0.19963	**68.77576**
K_{IP}	1.0	0.94769	5.23134	0.01883	1.98728
K_{IT}	1.0	0.92226	7.77434	0.05063	5.49006
K_{dP}	0.2	0.20133	0.66351	0.01652	8.20367
K_{dT}	0.2	0.19648	1.76144	0.00892	4.53784
$K_{1\mathrm{P}}$	2.0	2.25786	12.89278	1.63931	**72.60449**
$K_{2\mathrm{P}}$	2.0	19.99365	**899.6827**	1144.6725	**5725.179**
$K_{3\mathrm{P}}$	2.0	2.12346	6.17317	1.26231	**59.44557**
$K_{4\mathrm{P}}$	2.0	19.99991	**899.9952**	2748.46522	**13742.39**
$K_{1\mathrm{T}}$	2.0	2.22182	11.09093	0.45575	20.51227
$K_{2\mathrm{T}}$	2.0	20.00000	**899.9999**	228.95293	**1144.765**
$K_{3\mathrm{T}}$	2.0	1.97684	1.157968	0.165439	8.368837
$K_{4\mathrm{T}}$	2.0	5.88634	**194.3168**	10.67921	**181.4238**
$V_{1\mathrm{P}}$	8.0	8.49962	6.245185	5.63568	**66.30509**
$V_{2\mathrm{P}}$	1.0	6.73086	**573.0859**	380.312705	**5650.285**
$V_{3\mathrm{P}}$	8.0	8.10973	1.37165	4.26614	**52.60521**
$V_{4\mathrm{P}}$	1.0	6.27738	**527.7378**	854.34227	**13609.86**
$V_{1\mathrm{T}}$	8.0	8.35513	4.43911	1.00311	12.00596
$V_{2\mathrm{T}}$	1.0	6.43818	**543.8175**	67.27627	**1044.959**
$V_{3\mathrm{T}}$	8.0	7.64987	4.37668	0.47861	6644
$V_{4\mathrm{T}}$	1.0	1.80439	**80.43869**	2.35390	**130.4541**
k_{d}	0.01	0.00161	**83.86298**	0.01665	**1031.896**
k_{dC}	0.01	0.09069	**806.8997**	0.06735	**74.26728**
k_{dN}	0.01	2.8×10^{-08}	**99.99972**	0.01380	**4903365**
n	4	3.84846	3.78852	0.16016	4.16155

Table 5.6: Best fit parameter estimates of the 10-state Goldbeter model.

Parameters	True	Estimates	% Deviation	Std. Dev.	% CV
M_{P}	1.79120	1.75321	2.12081	0.04525	2.58122
P_0	0.58218	0.60647	4.17288	0.05666	9.34281
P_1	0.54007	0.55968	3.63112	0.05431	9.70296
P_2	0.32128	0.33881	5.45373	0.03416	10.08106
M_{T}	3.54281	3.44471	2.76905	0.05885	1.70853
T_0	1.50822	1.56290	3.62520	0.13020	8.33049
T_1	1.33985	1.33979	0.00442	0.12142	9.06268
T_2	0.97060	0.97471	0.42296	0.09407	9.65086
C	0.30794	0.31174	1.23461	0.03152	10.11138
C_{N}	0.66946	0.63435	5.24453	0.02962	4.66999

Table 5.7: Best fit initial concentrations estimates of the 10-state Goldbeter model

even when available. Due to the importance of phase behavior, experimental phase response data is also used [32,35,95] for model verification in model building. Thus, the purpose of this section is to investigate parameter estimation of circadian oscillators with PRC data.

For this study, the 10 state circadian rhythm model was used. As described in Chapter 1, the light induces the TIM protein degradation. The effect of light on the system was thus modeled with an increase in v_{dT}, the kinetic parameter for TIM protein degradation. The PRC was generated by introducing a square pulse variation of v_{dT} using a multiplicative factor m into the system and then allowing it to return to its original limit cycle over a number of oscillation cycles. Figure 3.6 illustrates the computation of phase change at eight cycles after the perturbation to eliminate transient effects. The resulting phase change was measured by using reference points of the same phase. The oscillation peaks are used in Figure 3.6 for illustration purposes, but for higher precision, the mean crossing method was used (Section 4.3).

This method is straightforward although the instantaneous changes in v_{dT} introduce discontinuities that can pose problems to the ODE solver at low error tolerance ($< 10^{-6}$). A second disadvantage is that the method is computationally expensive since the simulation time has to be sufficiently long for transient effects to become negligible, and the ODE model must be solved with a perturbation

at different timing for each data point on the PRC. Finally, the precision of each point computed is limited by the size of the time step used during the simulation. Realistically, the computation time required for time steps $< 1.0 \times 10^{-3}$ are not feasible and the mean crossing method is only able to improve precision to a certain extent.

Due to the computation cost of each PRC, reduction of the problem size is needed for the problem to be solved within a reasonable amount of time. The kinetic parameters for PER and TIM were assumed to be symmetric and lumped together to reduce the number of estimated system parameters from 38 to 23. The multiplicative factor m was to be estimated as well, bringing the total to 24.

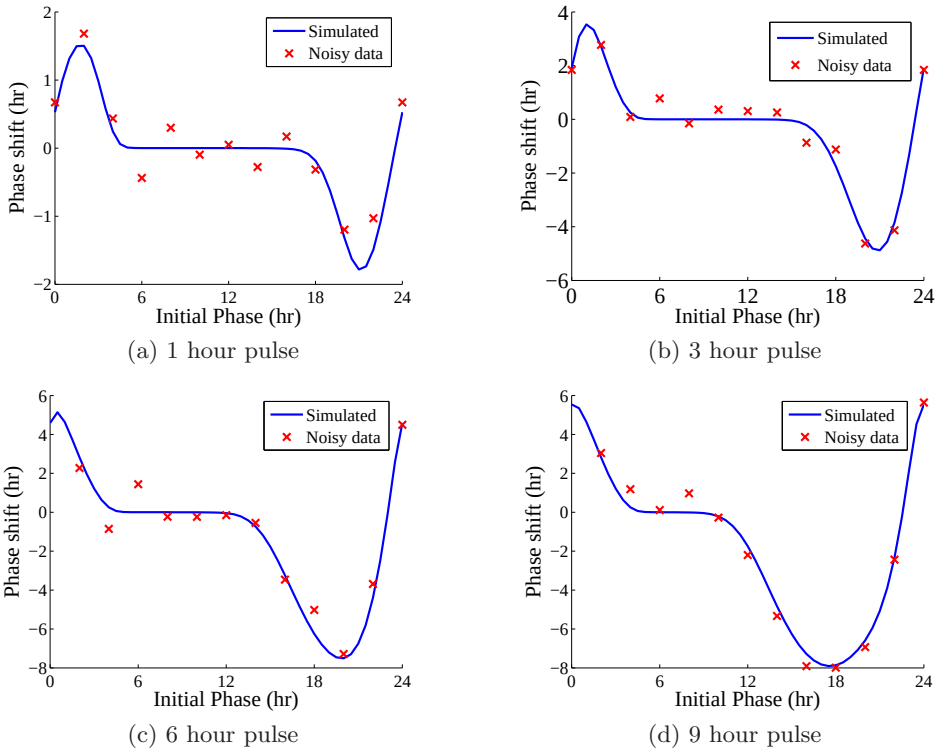
The symmetric system parameters from Leloup and Goldbeter [95] were used to generate the dataset. The dataset was composed of 4 PRCs for 1, 3, 6, and 9 hour pulses, each with 12 data points and corrupted using Gaussian noise with a variance of 0.25 hours^2. The measurement noise variance was selected based on variance estimated from the regression of *Drosophila* PRCs found in the PRC atlas [18]. Dual harmonic sinusoidal regression performed on ten PRCs gave estimated variance in the range of 0.15 to 0.35 and the mid-range value of 0.25 was selected.

As in the earlier section, the parameters were constrained between 0.1 and 10 times of their true values except for k_{d}, k_{dC} and k_{dN} which were constrained between 0 and 10 times of their true values, and n constrained between 1 and 10. An objective function (Φ) was constructed using the method of Maximum Likelihood to measure error in the PRC (ρ) and period (τ):

$$\Phi(\mathbf{p}) = \left(\sum_{j=1}^{N_{\mathrm{PRC}}} \sum_{i=1}^{N} \frac{1}{\sigma_{j,i}^2} \big(\hat{\rho}_j(T_i) - \rho_j(T_i)\big)^2 \right) + \left(\frac{1}{\sigma_{\tau,\mathrm{ave}}^2} (\hat{\tau}_{ave} - \tau)^2 \right). \tag{5.3}$$

The PRC is plotted over time T_i normalized to 24 hrs. Phase 0 of the PRC was fixed at the peak of *tim* mRNA levels.

The DE strategy used was DE/rand-to-best/1/bin with $F = 0.4$, $\lambda = 0.3$,



(a) 1 hour pulse

(b) 3 hour pulse

(c) 6 hour pulse

(d) 9 hour pulse

Figure 5.12: Comparison of simulated PRCs with data.

$Cr = 0.9$ and a population size of 200. Using the best set of estimates from four runs, the simulated PRC is compared to the data set in Figure 5.12, showing excellent agreement. However, the parameter estimates clearly show little agreement with the true values in Table 5.8.

With the failure to estimate parameters using PRC data, an alternative approach was considered. An *in-silico* dataset with 10% gaussian noise and 100 hourly samples was first generated using the model with symmetric parameters and used for parameter estimation. The estimated parameters and standard deviations computed from the FIM were then used as the *a priori* parameters in MAP estimation (Section 4.4.2) with PRC data. The objective function used was:

$$\Phi(\mathbf{p}) = \left(\sum_{j=1}^{N_{\text{PRC}}} \sum_{i=1}^{N} \frac{1}{\sigma_{j,i}^2} \left(\hat{\rho}_j(T_i) - \rho_j(T_i) \right)^2 \right) + \left(\sum_{k=1}^{N_{\mathbf{p}}} \frac{1}{\sigma_k^2} (\hat{p}_k - p_k)^2 \right). \tag{5.4}$$

The new objective function is similar to the earlier objective function (Equation

Parameters	True	Estimates	% Deviation
v_{s}	1.0	1.66271	66.27105
v_{m}	0.7	1.51875	116.96457
v_{d}	2.0	3.69752	84.87579
k_{s}	0.9	2.74233	204.70370
k_1	0.8	0.13574	83.03217
k_2	0.2	0.06620	66.90218
k_3	1.2	1.86589	55.49046
k_4	0.6	0.13922	76.79633
K_{m}	0.2	0.61541	207.70439
K_{I}	1.0	2.64700	164.69971
K_{d}	0.2	0.03835	80.82292
K_1	2.0	1.20347	39.82635
K_2	2.0	5.86582	193.29081
K_3	2.0	2.19736	9.86778
K_4	2.0	7.24289	262.14426
V_1	8.0	18.66104	133.26303
V_2	1.0	0.11284	88.71583
V_3	8.0	12.21116	52.63950
V_4	1.0	4.87749	387.74900
k_{d}	0.01	0.00033	96.73671
k_{dC}	0.01	0.09740	874.04818
k_{dN}	0.01	0.03930	292.98825
n	4	2.79261	30.18470
m	2	1.73652	13.17423

Table 5.8: Parameter estimates with PRC data.

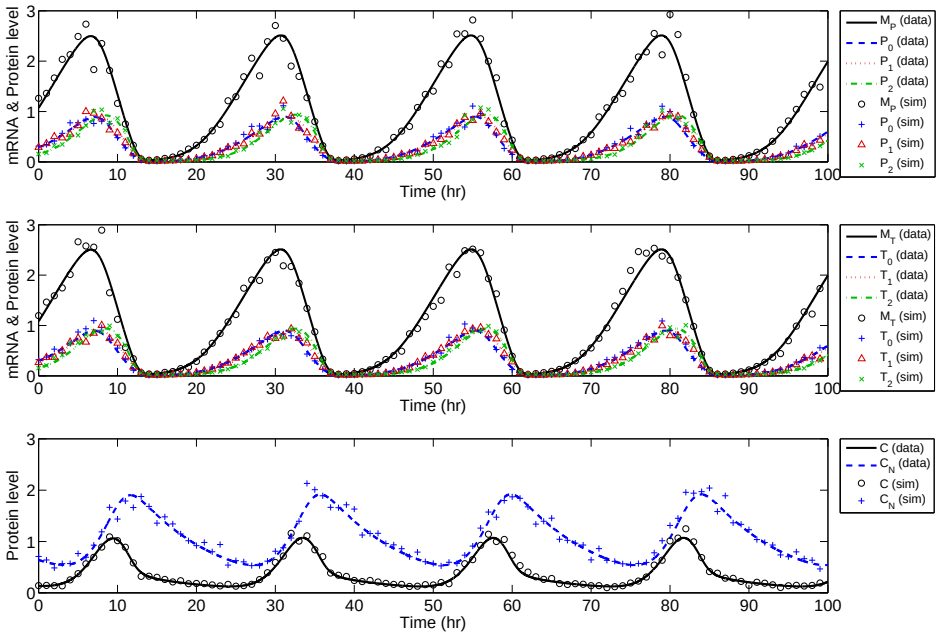


Figure 5.13: Comparison of best fit simulation with data for 10-state Goldbeter model using symmetric parameters. Parameter estimates are used as the *a priori* parameters in MAP estimation.

5.3) but with the period error term removed (since the parameters were fitted to the period in the *a prior* estimation) and augmented with an additional term to measure the difference between the prior values of parameters with the estimates.

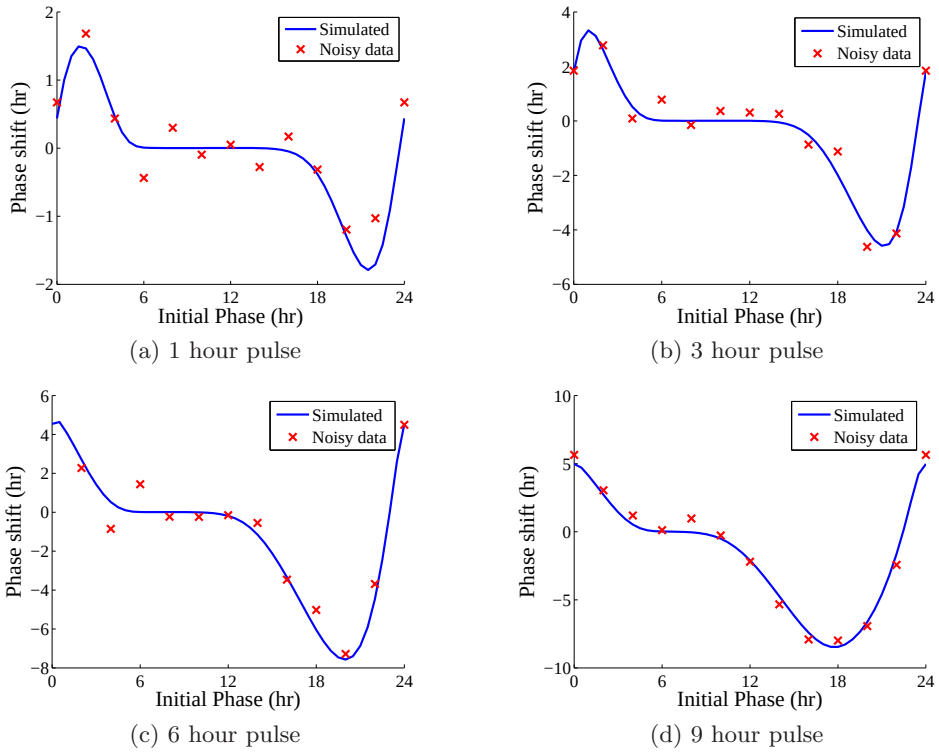
Table 5.9 shows the *a priori* estimates and the standard deviations computed using the FIM method, as well as the parameters estimated by MAP. Most of the estimates obtained *a priori* show excellent agreement with the true values except for unidentifiable parameters (K_2, V_2, K_4, V_4, k_{d}, k_{dC}, k_{dN}). As in Section 5.3.1, the Michaelis-Menten kinetic parameters for the backward reactions and the non-specific protein degradation are not identifiable. Figure 5.13 also shows good agreement the simulated system and data.

The MAP estimated parameters listed in Table 5.9 are very close to the MLE estimates, except for practically unidentifiable parameters (highlighted in bold) in MLE such as k_{dC} and k_{dN} with large CVs. This is not surprising since the *a priori* parameters already show excellent agreement with the true values and have correspondingly small standard deviations. For most parameters, there is

		MLE			MAP	
P.[1]	True	Estimates	% Dev.[2]	% CV	Estimates	% Dev.[2]
v_s	1.00	0.98128	1.87230	1.82460	0.97959	2.04084
v_m	0.70	0.70375	0.53591	4.13039	0.70438	0.62626
v_d	2.00	2.12640	6.32002	3.29286	2.13956	6.97804
k_s	0.90	0.94933	5.48115	2.42200	0.94596	5.10623
k_1	0.80	0.53869	**32.66347**	3.67959	0.53785	**32.76830**
k_2	0.2	0.19157	4.21341	6.94303	0.19085	4.57514
k_3	1.2	1.24170	3.47520	5.17352	1.23438	2.86467
k_4	0.6	0.64969	8.28092	5.82168	0.64894	8.15694
K_m	0.2	0.20522	2.61051	10.49064	0.20252	1.26104
K_I	1.0	0.99773	0.22662	1.30654	0.99763	0.23721
K_d	0.2	0.20197	0.98368	5.14350	0.19979	0.10467
K_1	2.0	1.47839	**26.08051**	12.85879	1.46375	**26.81226**
K_2	2.0	0.44724	**77.63825**	**50.42314**	0.44467	**77.76666**
K_3	2.0	2.27532	13.76610	19.75953	2.27638	13.81889
K_4	2.0	4.51212	**125.6061**	**214.1110**	3.94871	**97.43533**
V_1	8.0	6.735104	15.811204	9.479186	6.82149	14.73136
V_2	1.0	0.39803	**60.19683**	**53.68422**	0.38719	**61.28139**
V_3	8.0	9.44418	18.05221	17.65198	9.56228	19.52848
V_4	1.0	2.20848	**120.8484**	**200.6259**	2.70979	**170.9789**
k_d	0.01	0.01190	18.97986	107.88739	0.01436	**43.56158**
k_{dC}	0.01	1.3×10^{-10}	**99.99999**	4.3×10^{10}	0.00631	**36.89379**
k_{dN}	0.01	9.8×10^{-11}	**99.99999**	1.3×10^{10}	5.0×10^{-4}	**95.03372**
n	4.0	4.29796	7.44905	2.94440	4.28814	7.20360
m	2				1.93427	3.28640

[1] *Parameters*
[2] *Deviation*

Table 5.9: Parameter estimates using MLE and subsequent MAP.



(a) 1 hour pulse

(b) 3 hour pulse

(c) 6 hour pulse

(d) 9 hour pulse

Figure 5.14: Comparison of data with PRCs computed with MAP estimated parameters.

little or no improvement in the MAP estimates over the MLE estimates. The PRCs from the MAP estimates are also compared to the data in Figure 5.14, again showing excellent agreement.

Considering the failure of initial parameter estimation effort PRC data and lack of improvement in parameter estimates for the MAP estimation, it implies that the PRC dataset lacks information on the dynamics of the ODE model for parameter estimation. Due to the high computation cost of generating phase response measurements, size of the dataset used was kept small (48 data points) for the problem to be solved within reasonable time. This is in contrast with available experimental PRC datasets which are generally much larger, well in excess of a hundred data points. Implementation of a more computationally efficient method of PRC generation will allow the use of datasets with size comparable to experimental data and thus enable a more accurate assessment on the feasibility of using PRC data in parameter estimation. However, considering how extremely poor parameter estimates were able to produce excellent fits with

the PRC data in the initial estimation effort, it is unlikely that a much larger PRC dataset will produce much better estimates.

5.4 Computational Issues

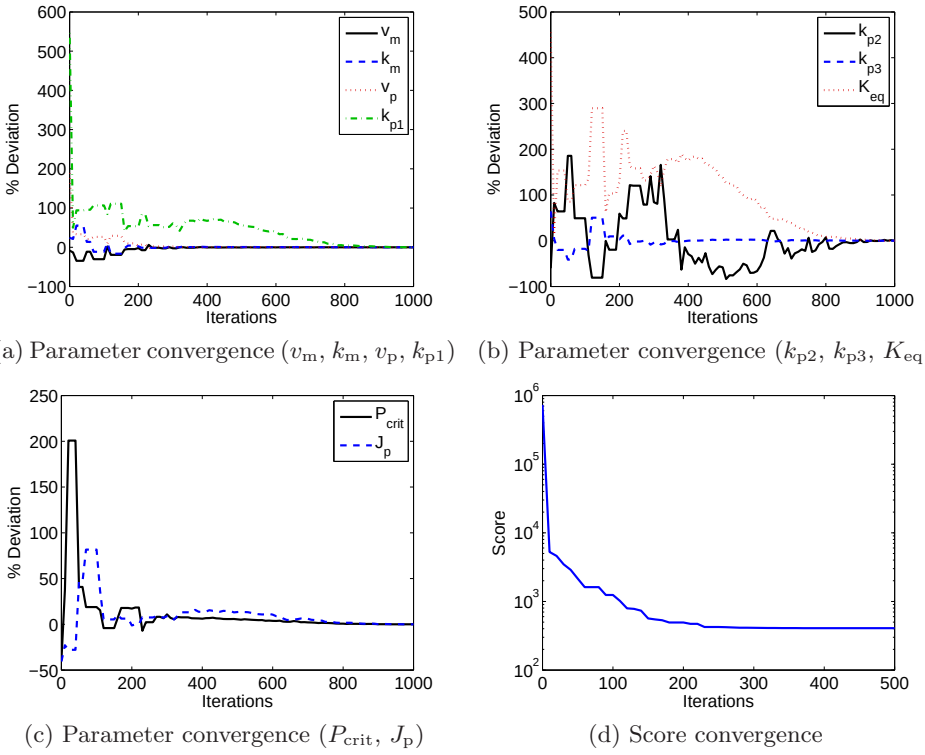
This section discusses some of the computation issues related to the application of the parameter estimation framework to the examples discussed earlier in this chapter.

5.4.1 Convergence

In this work, the parameter estimation program was executed in batch mode on the computational cluster and terminated when the maximum number of iterations (varies with problem size) was reached. During each run, evolution of the best objective function score was recorded and the final solution population was saved at program termination. Convergence was then determined by inspecting the convergence curve of the best objective function versus the iteration count. Convergence was considered to be achieved when the objective function fell below a threshold value and was improving by less than 10^{-5} on the average over 50 iterations. Although the optimum was not known *a priori*, a suitable threshold value was the objective function value of the true parameters. Since the datasets used were generated *in-silico* with known parameters, the true parameters gave a desirable fit to the noisy data. Experience from the examples showed that the converged solutions were 5-10% less compared to the threshold and this justifies the selection. In practice, this threshold value may differ based on the requirements of the user.

In the event that satisfactory convergence was not obtained, the population could be reloaded to continue from the prior search. Since the search algorithm is stochastic, four runs were performed on each set of data. The best solution from all the runs were then collected and compared to ensure consistent convergence, although this gives no guarantee that the global optimal solution is found.

Initially, the search convergence was based on observing the evolution of the



(a) Parameter convergence (v_{m}, k_{m}, v_{p}, k_{p1})

(b) Parameter convergence (k_{p2}, k_{p3}, K_{eq})

(c) Parameter convergence (P_{crit}, J_{p})

(d) Score convergence

Figure 5.15: Convergence of parameters and score compared for the 2-state Tyson model.

best set of parameter estimates. This was motivated by the desire to obtain accurate parameter estimates since we sought to recover the original parameters in the parameter estimation. It was found that parameters of the best solution continued to evolve while the objective function varied less than by 0.1%, as illustrated in Figure 5.15 for the 2-state Tyson model. Subsequent parameter identifiability analysis however showed that the slowest converging parameters (k_{p1}, k_{p2}, K_{eq}) tend to have the large CVs or are even unidentifiable. Thus, it was concluded that convergence based on parameter values was not efficient.

5.4.2 Parallelization

The code for the original parameter estimation program was a serial implementation. To solve the parameter estimation problem of the two state model, a run of 1500 iterations with a population size of 100 takes approximately 4 hours 20 minutes (260 minutes) to complete on a 2.66 Ghz Intel Core Duo PC. The time required is acceptable but to solve the five state model, a rough estimation of

the time required is:

$$260 \times 2 \times 2 \times 0.8 = 832 \text{ minutes or } 13 \text{ hours } 52 \text{ minutes.} \tag{5.5}$$

The first factor accounts for the doubling of population size and the second assumes a doubling in the number of iterations necessary to solve the problem. The third factor accounts for the difference in computation cost of the five state model objective function. Contrary to the typical expectation of a higher computation cost when solving a larger ODE system of equations, the computation cost for objective function evaluation of the five state model is actually lower compared to the two state model. The main reason is the complexity of the ODEs in the two state model (Equation 5.1), where a total of three equations are actually evaluated. In particular, one contains a square root function which is computationally expensive to evaluate.

Nevertheless, the computation time of the five state model is still more than three times that of the two state model. It is clear that the computation time required will continue to grow rapidly for any further increase in problem size such as the ten state model. Thus a decision was made to implement a parallel version of the parameter estimation in order to take advantage of high performance computing.

For a population based optimizer such as DE, the workload is *embarrassingly parallel* and thus parallel implementation is fairly straightforward by distributing the objective function evaluations during each iteration among the available processors. Using the parallel code, the parameter estimation of the two state (1500 iterations) and five state (3000 iterations) models only requires 5 minutes 10 seconds and 21 minutes, respectively. The speedup due to parallelization is 87 times. Using Amdahl's Law [99]

$$\text{Factor of Speedup} = \frac{1}{(1 - P_{\text{code}}) + \frac{P_{\text{code}}}{N_{\text{CPU}}}} , \tag{5.6}$$

where N_{CPU} is the number of processors, P_{code} is the proportion of parallelizable

code P_{code} can be computed. For a speedup factor of 87 and $N_{\text{CPU}} = 100$, P_{code} was found to be 0.998. Such a large proportion of parallelized computation is possible due to:

1. Embarrassingly parallel program structure.
2. Large computation cost of objective function evaluation compared to the DE search algorithm and communication overhead for parallel processing on a cluster.
3. The population size being an integer multiple of the number of processors available since the evaluation of objective function is not split between different processors.

In this work, the DE algorithm used generates a fixed number of trial solutions during each iteration, enabling an easy determination of a suitable number of processors for a given population size.

Chapter 6

Conclusions

In this work, a framework for parameter estimation of oscillatory systems was presented. A phase dependent objective function based on MLE was developed to capture the error in the shape and periodicity of the system states. Due to the nonlinear and nonconvex nature of parameter space, a global stochastic search algorithm was used.

The methodology was applied to three circadian rhythm models using *in-silico* data to study its efficacy. In all three examples, model simulation with the estimated parameters gave excellent agreement with the datasets. However, some of the parameter estimates obtained deviate considerably ($> 50\%$) from their true values and these can be attributed to the parameters being insensitive and thus not practically identifiable with the given datasets. The results obtained nevertheless show that the methodology was effective in solving the parameter estimation problem.

In the investigation on the effects of noise levels and sampling time on parameter identifiability in the first two examples, it was found that reducing noise by increasing replicates is more effective than a faster sampling rate in improving parameter identifiability. This is applicable to wet-lab experiments, where experiments can be easily limited by cost and available resources. If true replicates are possible in the experiment, they will be preferable to increasing the sampling time.

Parameter estimation with PRC data was attempted, but the results were unsatisfactory due to the lack of information in the PRC. Another approach was taken by using PRC data in MAP estimation to improve parameters obtained using the developed phase dependent objective function. The MAP estimation produced parameters very close to the initial estimates, though some parameters show slight improvements.

6.1 Future Directions

The next step is to validate the methodology using actual experimental data. Application of the methodology to practical modeling problems can be considered as the real test of its efficacy. However, the issue of mismatch between the model and the physical system (or plant-model mismatch) will be relevant when using wet-lab experimental data. Since models are not true depictions of the actual system, the models will then be evaluated based on the data fit and model parsimony. Usually, the simplest model with the best fit is selected, though a slightly poorer fit may be acceptable for a much simpler model.

The parameter estimation problems tackled in this work involve free running circadian systems. A possible avenue of further work is to compare the quality of parameter estimates from free running and entrained systems, which was used by Forger and Peskin [68]. The effect of different entrainment *zeitgeber* in terms of light to darkness ratio and circadian period on parameter identifiability can also be studied. If variations in the *zeitgeber* prove to have a substantial effect on the parameter identifiability, the use of *zeitgeber* in the design of experiments can be studied. One advantage offered by using entrained systems is that period estimation of the data is no longer necessary, thus eliminating errors resulting from comparing data points at the incorrect phases.

Another possible line of investigation is the parameter estimation of stochastic oscillatory models. Although deterministic ODEs are commonly used in modeling and analysis of cellular processes, they are not appropriate for processes that involve species with low copy count. Compared to ODE models,

stochastic models in the form of Stochastic Differential Equations or Chemical Master Equation are more accurate depictions of cellular networks. However, solving such models require much more computation effort. Since the parameter estimation framework developed in this work requires an accurate period estimation, a large number of simulations of the oscillation cycles may be required, thus compounding the computation cost. For circadian rhythm models, an alternative is to simulate only entrained systems and consequently avoid the necessity of period estimation.

References

[1] Kreutz C. and J. Timmer, Systems biology: Experimental design, FEBS Journal, 276, pp. 923-942. 2009.

[2] Ashyraliyev M., Y. Fomekong-Nanfack, J.A. Kaandorp and J.G. Blom, Systems biology: Parameter estimation for biochemical models, FEBS Journal, 276, pp. 886-902. 2009.

[3] Ljung L., System Identification - Theory for the User. Upper Saddle River, N.J.: PTR Prentice Hall, 2nd ed. 1999.

[4] van den Bos A., Parameter Estimation for Scientists and Engineers. Hoboken N.J.: John Wiley & Sons. 2007.

[5] Butcher E.C., E.L. Berg and E.J. Kunkel, Systems biology in drug discovery, Nature Biotechnology, 22, pp. 1253-1259. 2004.

[6] Auffray C., Z. Chen and L. Hood, Systems medicine: The future of medical genomics and healthcare, Genome Medicine, 1, p. 2. 2009.

[7] Kitano H., Systems Biology: A Brief Overview, Science, 295, pp. 1662-1664. 2002.

[8] Cedersund G. and J. Roll, Systems biology: Model based evaluation and comparison of potential explanations for given biological data, FEBS Journal, 276, pp. 903-922, 2009.

[9] Evans T., E.T. Rosenthal, J. Youngblom, D. Distel and T. Hunt, Cyclin: A protein specified by maternal mRNA in sea urchin eggs that is destroyed at each cleavage division, Cell, 33, p. 389-396. 1983.

[10] Dunlap J.C., Molecular bases for circadian clocks, Cell, 96, pp. 271-290. 1999.

[11] Hess B. and A. Boiteux, Oscillatory phenomena in biochemistry, Annual Review of Biochemistry, 40, pp. 237-258. 1971.

[12] Gerisch G., H. Fromm, A. Huesgen and U. Wick, Control of cell-contact sites by cyclic AMP pulses in differentiating *Dictyostelium* cells, Nature, 255, pp. 547-549. 1975.

[13] Konopka R. and S. Benzer, Clock Mutants of *Drosophila melanogaster*, Proc Natl Acad Sci U.S.A., 68, pp. 2112-2116. 1971.

[14] Hardin P.E., Essential and expendable features of the circadian time keeping mechanism, Current Opinion in Neurobiology, 16, pp. 686-692. 2006.

[15] Kulman S.J., S. R. Mackey and J.F. Duffy, Introductory workshop: Chronobiology concepts and nomenclature. In 72nd Cold Spring Harbor Symposium: Clocks and Rhythms. 2007.

[16] Pittendrigh C.S., Circadian systems: General perspective and entrainment. In Handbook of Behavioral Neurobiology: Biological Rhythms, edited by J. Aschoff, pp. 57-80 and 95-124, New York: Plenum Press. 1960.

[17] Dunlap J.C., J.J. Loros and P.J. DeCoursey (eds.), Chronobiology: Biological timekeeping. Sunderland, MA: Sinauer Associates, Inc. 2004.

[18] Johnson C.H., An Atlas of Phase Responses Curves for Circadian and Circatidal Rhythym. Nashville, USA: Department of Biology, Vanderbilt University. 1990.

[19] Hall J.C. and M. Rosbash, Genes and biological rhythms, Trends in Genetics, 3, pp. 185-191. 1987.

[20] Adams M.D., S.E. Celniker, R.A. Holt et al., The genome sequence of *Drosophila melanogaster*, Science, 287, pp. 2185-2195. 2000.

[21] Allada R., Circadian Clocks: A Tale of Two Feedback Loops, Cell, 112, pp. 284-286. 2003.

[22] Taghert P.H. and O.T. Shafer, Mechanisms of clock output in the *Drosophila* circadian pacemaker system, Journal of Biological Rhythms, 21, pp. 445-457. 2006.

[23] Zeng H., Z. Qian, M.P. Meyers and M. Rosbash, A light-entrainment mechanism for the *Drosophila* circadian clock, Nature, 380, pp. 129-135. 1996.

[24] Glossop N.R.J., L.C. Cyons and P.E. Hardin, Interlocked Feedback Loops Within the *Drosophila* Circadian Oscillator, Science, 286, pp. 766-768. 1999.

[25] Meyer P., L. Saez and M.W. Young, PER-TIM Interactions in Living *Drosophila* Cells: An Interval Timer for the Circadian Clock, Science, 311, pp. 226-229. 2006.

[26] Yu W., H. Zheng, J.H. Houl, B. Dauwalder and P.E. Hardin, PER-dependent rhythms in CLK phosphorylation and E-box binding regulate circadian transcription, Genes and Development, 20, pp. 723-733. 2006.

[27] Ceriani M.F., T.K. Darlington, D. Staknis, P. Mas, A.A. Petti, C.J. Weitz and S.A. Kay, Light dependent sequestration of TIMELESS by CRYPTOCHROME, Science, 285, pp. 553-556. 1999.

[28] Glossop N.R., J.H. Houl, H. Zheng, F.S. Ng, S.M. Dudek and P.E. Hardin, VRILLE feeds back to control circadian transcription of clock in the *Drosophila* circadian oscillator, Neuron, 37, pp. 249-261. 2003.

[29] Hardin P.E., The Circadian Timekeeping System of *Drosophila*, Current Biology, 15, pp. R714-722. 2005.

[30] Goldbeter A., A model for circadian oscillations in the *Drosophila* period protein (PER), Proc. R. Soc. Lond. B, 261, pp. 319-324. 1995.

[31] Leloup J-C. and A. Goldbeter, A Model for Circadian Rhythms in *Drosophila* Incorporating the Formation of a Complex between the PER and TIM Proteins, Journal of theoretical Biology, 13, pp. 70-87. 2005.

[32] Tyson J.J., C.I. Hong, D. Thron and B. Novak, A Simple Model of Circadian Rhythm Based on Dimerization and Proteolysis of PER and TIM, Biophysical Journal, 77, pp. 2411-2417. 1999.

[33] Scheper T.O., D. Klinkenberg, J. van Pelt and C. Pennartz, A model of molecular circadian clocks: Multiple mechanisms for phase shifting and a requirement for strong nonlinear interactions, Journal of Biological Rhythms, 14, pp. 213-220. 1999.

[34] Smolen P., D.A. Baxter and J.H. Byrne, A Reduced Model Clarifies the Role of Feedback Loops and Time Delays in the *Drosophila* Circadian Oscillator, Biophysical Journal, 83, pp. 2349-2359. 2002.

[35] Smolen P., P.E. Hardin, B.S. Lo, D.A. Baxter and J.H. Byrne, Simulation of *Drosophila* Circadian Oscillations, Mutations, and Light Responses by a Model with VRI, PDP-1, and CLK, Biophysical Journal, 86, pp. 2786-2802. 2004.

[36] Jirstrand M., Parameter estimation in biochemical reaction networks observer based prediction error minimization. In 8th International Conference on Systems Biology, Long Beach, USA. 2007.

[37] Koh G., D. Hsu and P.S. Thiagarajan, Composition of signaling pathway models and its application to parameter estimation. In ACM International Conference on Computational Biology (RECOMB) Poster Book, Singapore. 2008.

[38] Edgar T.F., D.M. Himmelblau and L.S. Lasdon, Optimization of Chemical Processes. New York, N.Y.: McGraw Hill, 2nd ed. 2001.

[39] Nelder J.A. and R. Mead, A simplex method for function minimization, The Computer Journal, 7, pp. 308-313. 1965.

[40] Lagarias J.C., J.A. Reeds, M.H. Wright and P.E. Wright, Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions, SIAM Journal of Optimization, 9, pp. 112-147. 1998.

[41] Wolfram Research Inc., Mathematica Edition: Version 7.0. Champaign, Illinois: Wolfram Research, Inc. 2008.

[42] Hoops S., S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes and U. Kummer, COPASI, a COmplex PAthway SImulator, Bioinformatics, 22, pp. 3067-3074. 2006.

[43] Herbert S.M., M. Hucka, A. Finney, C. Wellock, H. Bolouri, J. Doyle and H. Kitano, Next Generation Simulation Tools: The Systems Biology Workbench and BioSPICE Integration, Omics, 7, pp. 353-370. 2003.

[44] Hooke R. and T.A. Jeeves, Direct Search Solution of Numerical and Statistical Problems, Journal of ACM, 8, pp. 212-229. 1961.

[45] Conn A.R., N.I.M. Gould and P.L. Toint, LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), Springer Series in Computational Mathematics, vol. 17. Heidelberg, New York: Springer Verlag. 1992.

[46] Floudas, C. A., Deterministic Global Optimization. Dordrecht, The Netherlands: Kluwer Academic Publishers. 2000.

[47] Price K.V., R.M. Storn and J.A. Lampinen, Differential Evolution - A Practical Approach to Global Optimization. Berlin, Germany: Springer. 2005.

[48] Eiben A.E. and J.E. Smith, Introduction to Evolutionary Computing. Berlin, Germany: Springer. 2005.

[49] Biegler, L. T. and I. E. Grossmann, Retrospective on optimization. Computers and Chemical Engineering, 28, pp. 11691192. 2004.

[50] Grossmann I.E. and L.T. Biegler, Part II. Future Perspective on Optimization, Computers and Chemical Engineering, 28, pp. 1193-1218. 2004.

[51] Land A.H. and A.G. Doig, An Automatic Method for Solving Discrete Programming Problems, Econometrica, 28, pp. 497-520. 1960.

[52] Sahinidis N.V., A general purpose global optimization software package, Journal of Global Optimization, 8, pp. 201-205. 1996.

[53] Adjiman C.S., I.P. Androulakis, C.D. Maranas and C.A. Floudas, A global optimization method αBB for process design, Computers and Chemical Engineering, 20, p. S419-S424. 1996.

[54] Floudas C.A., I.G. Akrotirianakisa, S. Caratzoulasa, C.A. Meyera and J. Kallrathb, Global Optimization in the 21st century: Advances and challenges, Computers and Chemical Engineering, 29, pp. 1185-1202. 2005.

[55] Rinnooy Kan A.H.G. and G.T. Timmer, Stochastic Global Optimization Methods, Part 2: Multi Level Methods, Math Prog, 39, pp. 57-78. 1987.

[56] Mendes P. and D.B. Kell, Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation, Bioinformatics, 14, pp. 869-883. 1998.

[57] Goldberg D.E., Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley. 1989.

[58] Schwefel H-P., Evolution and Optimum Seeking. New York: Wiley & Sons. 1995.

[59] Kennedy J., S. Yu and R.C. Eberhart, Swarm Intelligence. San Franciso, CA.: Morgan Kaufmann. 2001.

[60] Dorigo M., V. Maniezzo and A. Colorni, Ant System: Optimization by a Colony of Cooperating Agents, IEEE Transactions on Systems, Man, and Cybernetics Part B, 26, p. 29-41. 1996.

[61] Laguna, M. and R. C. Martí, Scatter Search: Methodology and Implementations in C. Berlin, Germany: Springer. 2003.

[62] Storn R.M. and K.V. Price, Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012, International Computer Science Institute. March 1995.

[63] Egea J.A., M. Rodriguez-Fernandez, J.R. Banga and R. Martí Scatter Search for chemical and bio-process optimization, Journal of Global Optimization, 37, pp. 481-503. 2007.

[64] Kirkpatrick S., C. D. Gelatt and M. P. Vecchi, Optimization by simulated annealing, Science, 220, pp. 671-680. 1983.

[65] Cerny V., A thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, Journal of Optimization Theory and Applications, 45, pp. 41-51. 1985.

[66] Wenzel W. and K. Hamacher, Stochastic Tunneling Approach for Global Minimization of Complex Potential Energy Landscapes, Phys. Rev. Lett., 82, pp. 3003-3007. 1999.

[67] Glover F. and M. Laguna, Tabu Search. Norwell, MA.: Kluwer. 1997.

[68] Forger D.B. and C.S. Peskin, A detailed predictive model of the mammalian circadian clock. Proc. Nat. Acad. Sci., 100, pp. 14806-14811. 2003.

[69] Locke J.C.W., A.J. Millar and M.S. Turner, Modeling genetic networks with noisy and varied experimental data: The circadian clock in *Arabidopsis thaliana*, Journal of Theoretical Biology, 234, pp. 383-393. 2005.

[70] Locke J.C.W., M.M. Southern, L. Kozma-Bognár, V. Hibberd, P.E. Brown, M.S. Turner and A.J. Millar, Extension of a genetic network model by iterative experimentation and mathematical analysis. Molecular Systems Biology, 1. 2005.

[71] Zeilinger M.N., E.M. Farré, S.R. Taylor, S.A. Kay and F.J. Doyle III, A novel computational model of the circadian clock in *Arabidopsis* that incorporates PRR7 and PRR9. Molecular Systems Biology, 2. 2006.

[72] Bagheri N., M.J. Lawson, J. Stelling and F.J. Doyle III, Modeling the *Drosophila melanogaster* Circadian Oscillator via Phase Optimization. Journal of Biological Rhythms, 23, pp. 525-537. 2008.

[73] Tyson J.J. and B. Novak, Regulation of the Eukaryotic Cell Cycle: Molecular Antagonism, Hysteresis, and Irreversible Transitions, Journal of Theoretical Biology, 210, pp. 249-263. 2001.

[74] Ciliberto A., B. Novak and J.J. Tyson, Steady State and Oscillations in the p53/Mdm2 Network, Cell Cycle, 4, pp. 488-493. 2005.

[75] Chickarmane V., B. N. Kholodenko and H.M. Sauro, Oscillatory dynamics arising from competitive inhibition and multisite phosphorylation, Journal of theoretical Biology, 244, pp. 68-76. 2007.

[76] Varma A., M. Morbidelli and H. Wu, Parametric Sensitivity in Chemical Systems. New York: Cambridge University Press. 1999.

[77] Kramer M.A., H. Rabitz and J. Calo, Sensitivity analysis of oscillatory systems, Applied Mathematical Modeling, 8, pp. 328-340. 1984.

[78] Gunawan R. and F.J. Doyle III, Isochron-Based Phase Response Analysis of Circadian Rhythms, Biophysical Journal, 91, pp. 2131-2141. 2006.

[79] Zak D.E., J. Stelling and F.J. Doyle III, Sensitivity analysis of oscillatory (bio)chemical systems, Computers Chemical Engineering, 29, pp. 663-673. 2005.

[80] Johnson C.H., Phase Response Curves: What Can They Tell Us about Circadian Clocks. In Circadian Clocks from Cell to Human, edited by H. T. and K. Honma, pp. 209-249, Sapporo: Hokkaido Univ. Press. 1992.

[81] Hansel D., G. Mato and C. Meunier, Synchrony in Excitatory Neural Networks, Neural Computation, 7, pp. 307-337. 1995.

[82] Oppenheim A.V., A.S. Willsky and S. Hamid, Signals and Systems. Upper Saddle River, N.J.: PTR Prentice Hall, 2nd ed. 1996.

[83] Guckenheimer J. and P. Holmes, Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields. Applied Mathematical Sciences Vol. 42, New York: Springer-Verlag. 1983.

[84] Elowitz, M. B., A. J. Levine, E. D. Siggia and P. S. Swain, Stochastic Gene Expression in a Single Cell. Science, 297, pp. 11831186. 2002.

[85] Blake, W. J., M. Kaern, C. R. Cantor and J. J. Collins, Noise in eukaryotic gene expression. Nature, 422, pp. 633637. 2003.

[86] Banerjee, B., S. Balasubramanian, G. Ananthakrishna, T. V. Ramakrishnan and G. V. Shivashankar, Tracking Operator State Fluctuations in Gene Expression in Single Cells. Biophysical Journal, 86, pp. 30523059. 2004.

[87] Krishna, S., B. Banerjee, T. V. Ramakrishnan and G. V. Shivashankar, Stochastic simulation of the origins and implications of long-tailed distributions in gene expression. Proc Natl Acad Sci U.S.A., 102, pp. 47714776. 2005.

[88] Baker J.E., Reducing Bias and Inefficiency in the Selection Algorithm. In Proceedings of the Second International Conference on Genetic Algorithms and their Application, pp. 14-21, Hilldale. 1986.

[89] Storn R.M., Differential Evolution (DE) for Continuous Function Optimization. Jan 2009. http://www.icsi.berkeley.edu/ storn/code.html.

[90] Rodriguez-Fernandez M., J.A. Egea and J.R. Banga, Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems, BMC Bioinformatics, 7. 2006.

[91] Zak D.E., G.E. Gonye, J.S. Schwaber and F.J. Doyle III, Importance of Input Perturbations in Stochastic Gene Expression in the Reverse Engineering of Genetic Regulatory Networks: Insights From an Identifiability Analysis of an In Silico Network, Genome Research, 13, pp. 2396-2405. 2003.

[92] Cohen S.D. and A.C. Hindmarsh, CVODE, a stiff/nonstiff ODE solver in C, Computers in Physics, 10, pp. 138-143. 1996.

[93] Frigo M. and S.G. Johnson, The Design and Implementation of FFTW3, Proceedings of the IEEE, 93, pp. 216-231. 2005.

[94] The Mathworks Inc., MATLAB R2006b. Natick, Massachusetts: The Mathworks Inc. 2006.

[95] Leloup J-C. and A. Goldbeter, A Simple Model of Circadian Rhythm Based on Dimerization and Proteolysis of PER and TIM, Journal of Biological Rhythms, 13, pp. 70-87. 1998.

[96] Leloup J-C. and A. Goldbeter, Modeling the molecular regulatory mechanism of circadian rhythms in *Drosophila*, BioEssays, 22, pp. 84-93. 2000.

[97] Comas M., D.G.M. Beersma, K. Spoelstra and S. Daan, Phase and Period Response of the Circadian System of Mice (*Mus musculus* to Light Stimuli of Different Duration, Journal of Biological Rhythms, 21, pp. 362-372. 2006.

[98] Crosthwaite S.K., J.J. Loros and J.C. Dunlap, Light-Induced Resetting of a Circadian Clock Is Mediated by a Rapid Increase in frequency Transcript, Cell, 81, pp. 1003-1012. 1995.

[99] Amdahl G., Validity of the single processor approach to achieving large scale computing capabilities. In AFIPS Joint Computer Conferences, Atlantic City, New Jersey. 1967.

Appendix

Appendix A

FIM derivation

The definition of the FIM is

$$\mathrm{FIM} = \mathrm{E}\left\{\left[\frac{\partial}{\partial \mathbf{p}} \log f(\hat{\mathbf{y}}|\mathbf{p})\right]\left[\frac{\partial}{\partial \mathbf{p}} \log f(\hat{\mathbf{y}}|\mathbf{p})\right]^{\mathrm{T}}\right\} . \tag{A.1}$$

Assuming a Gaussian distribution for errors in $\hat{\mathbf{y}}$, we first define the Fisher score vector as:

$$\mathrm{FS} = \frac{\partial}{\partial \mathbf{p}} \log f(\hat{\mathbf{y}}|\mathbf{p}) . \tag{A.2}$$

Substituting in the formula for Gaussian distribution,

$$\mathrm{FS} = -\frac{\partial}{\partial \mathbf{p}}\left[\frac{N}{2}\log(2\pi) + \frac{1}{2}\log(|\mathbf{V}|) + \frac{1}{2}(\hat{\mathbf{y}}-\mathbf{y})^{\mathrm{T}}\mathbf{V}^{-1}(\hat{\mathbf{y}}-\mathbf{y})\right] \tag{A.3}$$

and since the first two terms are constants, the equation reduces to:

$$\mathrm{FS} = -\frac{\partial \mathbf{y}^{\mathrm{T}}}{\partial \mathbf{p}}\mathbf{V}^{-1}(\hat{\mathbf{y}}-\mathbf{y}) . \tag{A.4}$$

Substituting back into the FIM gives:

$$\begin{aligned}
\mathrm{FIM} &= \mathrm{E}\left\{\frac{\partial \mathbf{y}^{\mathrm{T}}}{\partial \mathbf{p}}\mathbf{V}^{-1}(\hat{\mathbf{y}}-\mathbf{y})\,(\hat{\mathbf{y}}-\mathbf{y})^{\mathrm{T}}\mathbf{V}^{-1}\frac{\partial \mathbf{y}}{\partial \mathbf{p}}\right\} \\
&= \frac{\partial \mathbf{y}^{\mathrm{T}}}{\partial \mathbf{p}}\mathbf{V}^{-1}\mathrm{E}\Big\{(\hat{\mathbf{y}}-\mathbf{y})(\hat{\mathbf{y}}-\mathbf{y})^{\mathrm{T}}\Big\}\mathbf{V}^{-1}\frac{\partial \mathbf{y}}{\partial \mathbf{p}} \\
&= \frac{\partial \mathbf{y}^{\mathrm{T}}}{\partial \mathbf{p}}\mathbf{V}^{-1}\mathbf{V}\mathbf{V}^{-1}\frac{\partial \mathbf{y}}{\partial \mathbf{p}} \\
&= \frac{\partial \mathbf{y}^{\mathrm{T}}}{\partial \mathbf{p}}\mathbf{V}^{-1}\frac{\partial \mathbf{y}}{\partial \mathbf{p}} .
\end{aligned}$$

Appendix B

Model equations for 10 state *Drosophila* circadian model

$$\frac{\mathrm{d}M_\mathrm{p}}{\mathrm{d}t} = v_\mathrm{sP}\frac{K_\mathrm{IP}^n}{K_\mathrm{IP}^n + C_\mathrm{N}^n} - v_\mathrm{mP}\frac{M_\mathrm{p}}{K_\mathrm{mp} + M_\mathrm{p}} - k_\mathrm{d}M_\mathrm{p}$$
$$\frac{\mathrm{d}P_0}{\mathrm{d}t} = k_\mathrm{sP}M_\mathrm{p} - V_\mathrm{IP}\frac{P_0}{K_\mathrm{1P} + P_0} + V_\mathrm{2P}\frac{P_1}{K_\mathrm{2P} + P_1} - k_\mathrm{d}P_0$$
$$\frac{\mathrm{d}P_1}{\mathrm{d}t} = V_\mathrm{1P}\frac{P_0}{K_\mathrm{1P} + P_0} - V_\mathrm{2P}\frac{P_1}{K_\mathrm{2P} + P_1} - V_\mathrm{3P}\frac{P_1}{K_\mathrm{3P} + P_1} + V_\mathrm{4P}\frac{P_2}{K_\mathrm{4P} + P_1} - k_\mathrm{d}P_1$$
$$\frac{\mathrm{d}P_2}{\mathrm{d}t} = V_\mathrm{3P}\frac{P_1}{K_\mathrm{3P} + P_1} - V_\mathrm{4P}\frac{P_2}{K_\mathrm{4P} + P_2} - k_3P_2T_2 + k_4C - v_\mathrm{dP}\frac{P_2}{K_\mathrm{dP} + P_2} - k_\mathrm{d}P_2$$
$$\frac{\mathrm{d}M_\mathrm{T}}{\mathrm{d}t} = v_\mathrm{sT}\frac{K_\mathrm{IT}^n}{K_\mathrm{IT}^n + C_\mathrm{N}^n} - v_\mathrm{mT}\frac{M_\mathrm{T}}{K_\mathrm{mT} + M_\mathrm{T}} - k_\mathrm{d}M_\mathrm{T}$$
$$\frac{\mathrm{d}T_0}{\mathrm{d}t} = k_\mathrm{sT}M_\mathrm{T} - V_\mathrm{IT}\frac{T_0}{K_\mathrm{1T} + T_0} + V_\mathrm{2T}\frac{T_1}{K_\mathrm{2T} + T_1} - k_\mathrm{d}T_0$$
$$\frac{\mathrm{d}T_1}{\mathrm{d}t} = V_\mathrm{1T}\frac{T_0}{K_\mathrm{1T} + T_0} - V_\mathrm{2T}\frac{T_1}{K_\mathrm{2T} + T_1} - V_\mathrm{3T}\frac{T_1}{K_\mathrm{3T} + T_1} + V_\mathrm{4T}\frac{T_2}{K_\mathrm{4T} + T_1} - k_\mathrm{d}T_1$$
$$\frac{\mathrm{d}T_2}{\mathrm{d}t} = V_\mathrm{3T}\frac{T_1}{K_\mathrm{3T} + T_1} - V_\mathrm{4T}\frac{T_2}{K_\mathrm{4T} + T_2} - k_3P_2T_2 + k_4C - v_\mathrm{dT}\frac{T_2}{K_\mathrm{dT} + T_2} - k_\mathrm{d}T_2$$
$$\frac{\mathrm{d}C}{\mathrm{d}t} = k_3P_2T_2 - k_4C - k_1C + k_2C_\mathrm{N} - k_\mathrm{dC}C$$
$$\frac{\mathrm{d}C_\mathrm{N}}{\mathrm{d}t} = k_1C - k_2C_\mathrm{N} - k_\mathrm{dN}C_\mathrm{N}$$