Phase Models and Computations for Oscillators

by

Önder Şuvak

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Electrical-Electronics Engineering

Koç University

July 2013

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Önder Şuvak

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

Prof. Alper Demir

Reading Committee:

Assoc. Prof. Alper T. Erdoğan

Assist. Prof. Emre Mengi

Prof. Varga Kalantarov

Assoc. Prof. Müştak E. Yalçın

Date:

To my family

ABSTRACT

Oscillators as key components of many natural and engineered systems have been a research focus for decades in many disciplines such as electronics and biology. The time keeping capability of autonomous oscillators and the synchronization of coupled oscillators are best described in terms of a scalar quantity, so-called the phase of an oscillator. Phase computations for perturbed and coupled oscillators and equations that describe phase dynamics have been quite useful in both electronics and biology in forming a rigorous understanding of oscillatory system behavior and designing oscillators that are least affected by undesired disturbances such as noise.

We first review the notion of isochrons, which forms the basis for the generalized phase notion for an oscillator that we cover in a rigorous manner. The notion of isochrons for oscillators has been first introduced by Arthur Winfree in 1974 and heavily utilized in mathematical biology in studying biological oscillators. Isochrons were instrumental in introducing a notion of generalized phase for an oscillator and form the basis for oscillator perturbation analysis formulations. Calculating the isochrons of an oscillator is a very difficult task. Except for some very simple planar oscillators, isochrons can not be calculated analytically and one has to resort to numerical techniques. Previously proposed numerical methods for computing isochrons can be regarded as brute-force, which become totally impractical for non-planar oscillators with dimension more than two. In this thesis, we present a precise and carefully developed theory and numerical techniques for computing local but quadratic approximations for isochrons. Previous work offers the theory and the numerical methods needed for computing only local linear approximations. Our treatment is general and applicable to oscillators with large dimension. We present examples for isochron computations, verify our results against exact calculations in a simple analytically calculable case, test our methods on complex oscillators.

We next present a unified theory of phase equations for autonomous oscillators through an assimilation of the work that has been done on oscillator analysis in both electronics and biology during the past seventy years. Based on the generalized oscillator phase notion that is founded on the theory of isochrons, we present a general framework for phase equations and derive in a unified manner a phase equation for perturbed oscillators that is exact but practically unusable, and practically useful ones that are based on linear (previously known in the literature) and quadratic (new, more accurate) isochron approximations. We discuss the utility of these phase equations in performing (semi) analytical phase computations and also describe simpler and more accurate schemes for numerical phase computations. Carefully run numerical experiments on several examples are presented which compare the accuracy of the various phase computation schemes and the phase equations described.

Biochemical oscillators perform crucial functions in cells, e.g., they set up circadian clocks. Phase computation techniques for continuous oscillators that are based on isochrons have been used for characterizing the behavior of various types of oscillators under the influence of perturbations such as noise. In this thesis, we also extend the applicability of these phase equations and phase computation schemes to biochemical oscillators as discrete, molecular systems. In particular, we describe techniques for computing the instantaneous phase of discrete, molecular oscillators for SSA (Stochastic Simulation Algorithm) generated sample paths. The impact of noise that arises from the discrete and random nature of the mechanisms that make up molecular oscillators can be characterized based on these phase computations.

Modeling and analysis studies of oscillators in electronics and biology seem to have progressed independently, without any cross-fertilization in between. Even though work on oscillator analysis in electronics did not directly make use of isochrons, similar concepts, models and perturbation analysis techniques, though using completely different terminology and formulations, have been developed in both disciplines. In this thesis, we reveal the connection between oscillator analysis work in these two seemingly disparate disciplines.

ÖZETÇE

Osilatörler, doğal ve mühendislik ürünü birçok sistemin önemli bileşenleri olarak, elektronik ve biyoloji gibi birçok disiplinde, onlarca yıldır bir araştırma odağı haline gelmişlerdir. Otonom osilatörlerin zaman referansı sağlama becerisi ve bağlaşımlı osilatörlerin senkronizasyonu, en iyi şekilde, osilatör fazı olarak adlandırılan bir değer ile betimlenir. Değişikliğe uğratılmış ve bağlaşımlı osilatörler için faz hesaplamaları ve faz dinamiğini ortaya koyan denklemler, elektronik ve biyolojide, salınımlı sistem davranışı ve gürültü gibi istenmeyen olgulardan en az şekilde etkilenen osilatörlerin tasarımı konusunda bir anlayış geliştirilmesini sağlamışlardır.

Öncelikle, osilatörler için genelleştirilmiş faz mefhumunu oluşturan isokron kavramını sunuyoruz. Osilatörler için isokron kavramı, 1974 yılında Arthur Winfree tarafından ilk olarak ortaya atılmıştır ve matematiksel biyolojide biyolojik osilatörleri incelemek üzere yoğun şekilde kullanılmıştır. İsokronlar, bir osilatör için genelleştirilmiş faz kavramını geliştirmede kullanılmışlardır ve osilatör sarsım analizi formülasyonlarının temelini oluşturmaktadırlar. Bir osilatörün isokronlarını hesaplamak çok güçtür. Bazı çok basit düzlemsel osilatörler dışında, isokronlar analitik olarak hesaplanamazlar ve bu durumda nümerik teknikler kullanılmak zorundadır. İsokronların hesaplanması için önceden önerilen nümerik metodlar, boyutu ikiden yüksek olan ve bu durumda düzlemsel olmayan osilatörler için tamamen kullanışsız hale gelir. Bu tezde, isokronların lokal fakat karesel yaklaşımlarını hesaplamak için kesin ve dikkatle geliştirilmiş bir teori ile nümerik teknikler sunuyoruz. Bu konuda, önceki yayınlarda isokronların ancak lokal doğrusal yaklaşımlarını hesaplamak için teori ve nümerik metodlar önerilmektedir. Geliştirdiğimiz metodlar geneldir ve yüksek boyuttaki osilatörlere uygulanabilmektedir. Bu tezde ayrıca, isokron hesaplamaları için örnekler sunuyoruz, basit ve analitik bir örnekte yaklaşık sonuçlarımızı kesin değerlerle karşılaştırıyoruz ve karmaşık osilatörlerde metodlarımızı test ediyoruz.

Elektronik ve biyoloji disiplinlerinde son 70 yılda osilatör analizi konusunda yapılmış çalışmalardan faydalanarak, otonom osilatörler için faz denklemleri üzerine birleştirici bir teori sunuyoruz. İsokron teorisine dayalı olan genelleştirilmiş faz kavramı temeline bağlı olarak, faz denklemleri için genel bir teori takdim ediyoruz ve birleştirici bir teorinin parçaları halinde, değişikliğe uğratılmış osilatörler için kesin fakat pratik anlamda kullanışsız bir faz denklemi ile isokronların doğrusal (literatürde önceden gösterildiği şekilde) ve karesel (yeni ve daha yüksek doğruluğa sahip) yaklaşımlarına dayalı kullanışlı faz denklemleri çıkarıyoruz. Bu faz denklemlerinin (yarı) analitik faz hesaplamaları için kullanımlarını tartışıyoruz ve nümerik faz hesaplamaları için daha basit ve daha yüksek doğruluğa sahip faz hesaplama yöntemlerini açıklıyoruz. Birkaç örnek üzerinde koşulmuş nümerik hesaplama deneyleri aracılığıyla, açıklanan değişik türdeki faz hesaplama yöntemlerinin ve faz denklemlerinin doğruluklarını karşılaştırıyoruz.

Biyokimyasal osilatörler hücrelerde değişik fonksiyonları yerine getirmektedirler. Örneğin, canlıların biyolojik saatlerini belirlemektedirler. Sürekli değerli osilatörler için isokronlara dayalı faz hesaplama teknikleri, gürültü gibi bazı sarsımlardan etkilenen değişik türdeki osilatörlerin davranışlarını karakterize etmek için kullanılmışlardır. Bu tezde, bu faz denklemlerinin ve faz hesaplama düzenlerinin kullanım alanını, ayrık değerli moleküler sistemler olan biyokimyasal osilatörleri içine alacak şekilde genişletiyoruz. Özellikle, ayrık değerli moleküler osilatörler için SSA (Stokastik Simülasyon Algoritması) ile oluşturulmuş örnek yollar üzerinde anlık faz hesaplamaları yapmayı sağlayan teknikleri açıklıyoruz. Moleküler osilatörleri oluşturan mekanizmaların ayrık değerli ve rastgele tabiatından kaynaklanan gürültünün etkileri, bu faz hesaplamaları aracılığıyla karakterize edilebilir.

Elektronik ve biyolojide, osilatörlerin modellenmesi ve analizi üzerine çalışmaların, birbirleriyle etkileşime girmeden bağımsız olarak yürütüldükleri anlaşılmaktadır. Osilatör analizi üzerine elektronik alanındaki çalışmalarda isokron kavramından doğrudan yararlanılmamasına rağmen, her iki disiplinde birbirinden tamamıyle farklı terminolojiler ve formülasyonlar kullanılmış olsa da benzer kavramlar, modeller ve sarsım analiz teknikleri geliştirilmiştir. Bu tezde, birbirinden ayrık gözüken bu iki disiplindeki osilatör analizi üzerine yapılan çalışmalardaki bağlantılar da ortaya çıkarılmaktadır.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my M.S. and Ph.D. supervisor Prof. Alper Demir, for the motivation and guidance he provided during my studies, on virtually every aspect of this work. Specifically, he kindly weighed my abilities and shaped the course of the study towards those avenues that would profit me in the most fruitful fashion, helping me express my findings explicitly and comprehensibly. His impact has been irrevocably imprinted.

Professors Alper T. Erdoğan and Emre Mengi were members of my Ph.D evaluation and reading committees. They both examined my work and articles thoroughly, coming up with insightful questions and wonderful suggestions. They patiently attended my presentations in the last three years of my Ph.D. I owe much to them for their invaluable contributions.

Professors Varga Kalantarov and Müştak E. Yalçın kindly accepted to be members of my reading committee. Within a limited period of time, they were able to see through my thesis, presenting their questions and suggestions. I would like to thank them both for their time and enthusiasm.

I would also like to thank Danny Elad and his Analog and Mixed Signal Technologies Group at IBM Haifa Research Laboratories in Israel for prividing me with a 3-month term as an IBM Great Minds 2012 Intern, during July-October 2012. The experience I had in such a world-class company was terrific.

TÜBİTAK awarded me with a BİDEB 2211 Ph.D. scholarship during 2008-2013. I would not have been able to complete my studies without this financial support. I gratefully acknowledge TÜBİTAK and wish they continue with this program for prospective academics.

A friend of more than twenty years standing, Y.İ., was very kind to offer his help whenever I was very much in need. Him and his support I can never forget.

Lastly, my family stood by me on my decision for a life in academics. I am going to continue to work hard so as to have our mutual sacrifices pay well eventually.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Oscillators

Oscillatory behavior is encountered in many types of systems including electronic, optical, mechanical, biological, chemical, financial, social and climatological systems. Carefully designed oscillators are intentionally introduced into many engineered systems to provide essential functionality for system operation. In electronic systems, oscillators are used to generate clock signals that are needed in the synchronization of operations in digital circuits and sampled-data systems. The periodic signal generated by an electronic oscillator is used as a carrier and for frequency translation of signals in all types of communication systems. Electromagnetic fields oscillating at a very high frequency (on the order of a peta Hz) are behind laser operation. Lasers generate monochromatic (with power concentrated at a single frequency or wavelength) light and are used in many applications in medicine, telecommunications, etc. Oscillatory behavior in biological systems is seen in population dynamics models (prey-predator systems), in neural systems [1], in the motor system, and in circadian rhythms [2]. Intracellular and intercellular oscillators of various types perform crucial functions in biological systems. Due to their essentialness, and intricate and interesting dynamical behavior, biological oscillations have been a research focus for decades. Genetic oscillators that are responsible for setting up the circadian rhythms have received particular attention [3]. Circadian rhythms are crucial for the survival of many species, and there are many health problems, including cancer, caused by the disturbance of these clocks in humans.

Oscillators in electronic and telecommunication systems are adversely affected by the presence of undesired disturbances in the system. Various types of disturbances such as noise affect the spectral and timing properties of the ideally periodic signals generated by oscillators, resulting in power spreading in the spectrum and jitter and phase drift in the time domain [4]. Unlike other systems which contain an implicit or explicit time reference, autonomously oscillating systems respond to noise in a peculiar and somewhat nonintuitive manner. Understanding the behavior of oscillators used in electronic systems in the presence of disturbances and noise has been a preoccupation for researchers for many decades [5]. The behavior of biological oscillators under various types of disturbances has also been the focus of a good deal of research work in the second half of the 20th century [1, 2, 6, 7].

1.2 Isochrons

The influence of perturbations and noise on oscillators can be analyzed and quantified best by referring to the sets called *isochrons*. Oscillators can be modeled as autonomous dynamical systems with a system of differential equations. Almost all oscillators that are useful in practice have stable periodic solutions, i.e., attracting limit cycles or periodic closed orbits that reside in a high-dimensional space. The notion of isochrons for oscillators that are associated with stable periodic solutions has been introduced and utilized in mathematical biology in studying biological oscillations [2, 8].

An isochron can be thought to be a set of synchronized points in the neighborhood of the attracting limit cycle of an oscillator. The notion of isochrons is intimately related to the notion of *asymptotic phase* for stable periodic solutions of oscillators [9, 10]: All of the points that reside on a particular isochron have the same asymptotic phase. For an N-dimensional oscillator, each isochron is an $(N-1)$-dimensional hypersurface. The isochrons of a periodic orbit foliate its domain of attraction, i.e., their union covers the neighborhood of the orbit [8, 1]. Isochrons were first introduced by Winfree [2], but it was Guckenheimer [11] who rigorously proved their existence and revealed their mathematical properties.

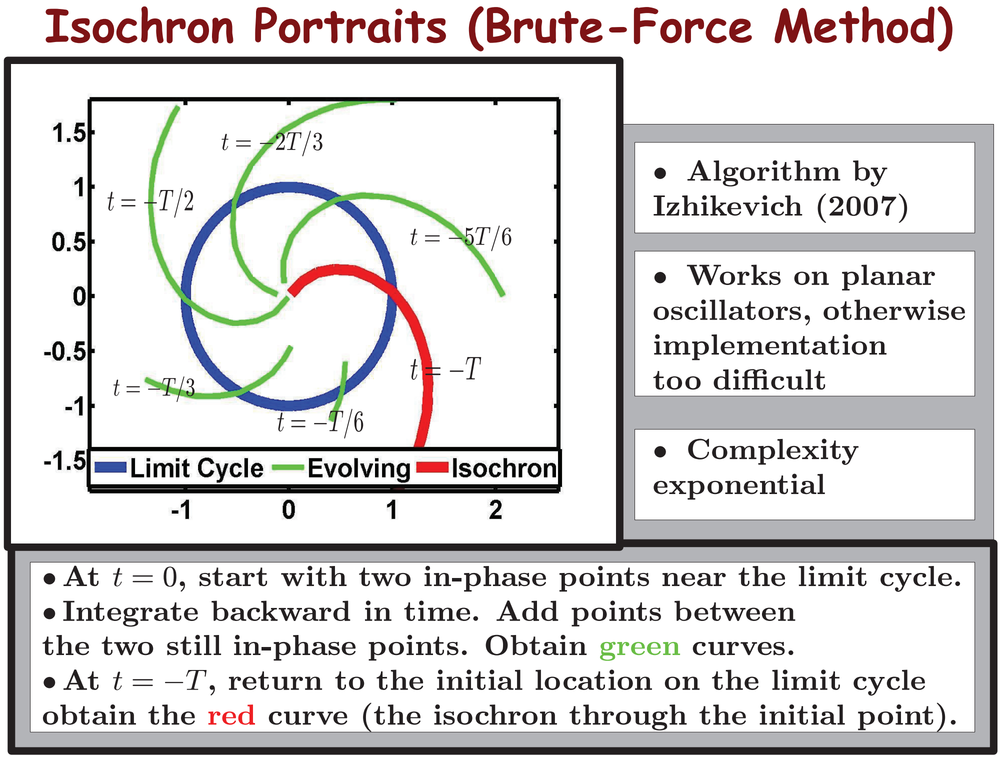
1.3 Isochrons, PPVs and PRCs

The isochron concept is crucial in introducing a notion of generalized phase for an oscillation and lies at the foundation of scalar (one-dimensional) phase domain models for oscillators that have been widely utilized in studying various forms of oscillation phenomena and synchronization of coupled oscillators [8, 1]. In mathematical biology, the *phase response curves* (PRCs) that are used in the perturbation analysis of oscillators in the phase domain are defined based on isochrons [1].

Understanding and characterizing the behavior of oscillators used in electronic systems in the presence of disturbances and noise has been a preoccupation of researchers for many decades [5]. The works on the modeling and analysis of oscillators in electronics and biology seem to have progressed independently, without any cross-fertilization in between. However, a comparative study of the literature on the topic from both disciplines shows that similar concepts, models and perturbation analysis techniques have been developed in both, but using completely different terminology and formulations. For instance, the so-called infinitesimal phase response curves (along state variables) that are used in studying the perturbation behavior of biological oscillators [1] turn out to be the entries of the so-called *perturbation projection vector* (PPV) that has been used in performing perturbation analysis for electronic oscillators [12]. Infinitesimal PRCs and PPVs are intimately related to isochrons [1], as we also show in this thesis.

1.4 Computing Isochrons: Previous Work

Due to their importance and utility in oscillator analysis, finding (calculating) the isochrons of periodic solutions for oscillators is crucial. However, this is a very difficult task [2, 1]. Except for some very simple two-dimensional oscillators, exact isochrons can not be calculated in a fully analytical manner and one has to resort to numerical techniques [8]. The main numerical method for computing isochrons [8, 1] works as follows: “Many” initial points in the domain of attraction “near” a point on the periodic orbit are chosen. Then, the differential equations that describe the oscillator are integrated backwards for “some” time to compute a collection of trajectories that end (not start, because of backward integration) at each of these initial points. The collection of the points from all of these trajectories at a particular time t approximately lie on the same isochron [1]. With this method, approximations for isochrons are computed point by point, in a brute-force manner. A compact, nice Matlab implementation (restricted to planar two-dimensional oscillators) of this technique that starts with only two initial points and adaptively introduces more points as necessary is provided by Izhikevich in [1]. An illustration of the method running on the limit cycle of a simple polar oscillator and computing a single isochron belonging to this periodic orbit is given in Figure 1.1. Notice how the worm-like baby isochron evolves into maturity through



Isochron Portraits (Brute-Force Method)

- Algorithm by Izhikevich (2007)
- Works on planar oscillators, otherwise implementation too difficult
- Complexity exponential

- At $t=0$, start with two in-phase points near the limit cycle.
- Integrate backward in time. Add points between the two still in-phase points. Obtain green curves.
- At $t=-T$, return to the initial location on the limit cycle obtain the red curve (the isochron through the initial point).

Figure 1.1: Brute-force method for isochron computation.

backward integration. After a full cycle of computations, we reach the original point, where now the “mature” isochron crosses the limit cycle. However, there are several shortcomings of this numerical method, the most important one being that it becomes totally impractical for non-planar oscillators with dimension more than two. In N dimensions, an isochron is an $N-1$-dimensional hypersurface: The computational cost of a method that constructs a hypersurface in a point by point manner obviously increases exponentially with the dimension of the hypersurface. Even if one is somehow willing to live with exponential complexity, the implementation of this method for non-planar oscillators seems to be a very challenging task.

1.5 Local Approximations for Isochrons

In order to attain a tractable and meaningful technique for computing isochrons, one idea is to find *local approximations* to isochrons in the neighborhood of a point on the periodic orbit of the oscillator [8]. A local approximation makes sense in practical applications, because useful oscillators usually stay close to their stable periodic orbits even when they are experiencing disturbances. The simplest local approximation for a hypersurface around a point is simply the tangent hyperplane, which corresponds to a first-order, linear approximation. The tangent hyperplane for an isochron at a point on the periodic orbit can be characterized by finding a vector that is normal to it. It turns out (as we show in this thesis) that the PPVs (or the infinitesimal PRCs along the state variables of the oscillator assembled into a vector) at each point on the periodic orbit are actually orthogonal to the tangent hyperplanes for the isochrons [8]. The PPVs and the infinitesimal PRCs can be computed by finding a periodic solution of the so-called *adjoint* equation (which we discuss in this thesis) along with a normalization condition [5, 12, 13, 14, 15, 1]. The linear, hyperplanar local approximations for isochrons (and hence the PPVs and infinitesimal PRCs) form the basis for deriving scalar phase domain differential equations for oscillators [5, 1]. These first-order accurate phase domain models have been used in various disciplines to study the behavior of oscillators under perturbations and noise and the synchronization of coupled oscillators.

1.6 Phase of an Oscillator

The dynamical behavior of autonomous and coupled oscillators is best described and analyzed in terms of a scalar quantity, called the *phase* of an oscillator. There are many notions and definitions of phase that appeared in the literature. The simplest and most straightforward phase definition is perhaps obtained when the equations of a two-dimensional, planar oscillator are expressed in polar coordinates, with amplitude and polar angle as the state variables. While it is useful in some cases to define the polar angle as the phase of an oscillator, this definition does not easily generalize to oscillators that are higher dimensional. In the general case, it is our conviction that the most useful, rigorous and precise definition of phase that appeared in the literature is the one that is based on the *isochrons* of an oscillator [2, 1, 8]. The phase of an oscillator that is defined based on isochrons is intimately related to time. In fact, the phase of an autonomous oscillator that is in periodic steady-state, or one that is in the domain of attraction of a limit cycle and hence in the process of converging to a periodic steady-state, is simply equal to time. This makes perfect sense, since an autonomous oscillator that is not experiencing any perturbations performs as an ideal, precise time keeper when it is in the domain of attraction of a limit cycle, even when it has not yet settled to a periodic steady-state solution. For perturbed oscillators, the actual phase deviates from time, capturing the degrading impact of disturbances on the time keeping ability and synchronization capability of the oscillator.

1.7 Oscillator Phase Models

Since phase serves as a crucial and useful quantity in compactly describing the dynamical behavior of an oscillator, one would be first of all interested in simply computing the phase of an oscillator that is experiencing perturbations. Furthermore, if the phase of a non-trivial, practical oscillator can somehow be computed or characterized in a semi or fully analytical manner, one can then draw concrete conclusions and obtain useful characterizations that can be used in practice in assessing the time keeping performance and the synchronization behavior of oscillators. Indeed, we observe in the literature that, in various disciplines, researchers have derived *phase equations* that compactly describe the dynamics of weakly perturbed oscillators [1, 5]. It appears that a phase equation for oscillators has first been derived by Malkin [16] in his work on the reduction of weakly perturbed oscillators to their phase models [1], and the same equation in different forms has been subsequently reinvented by various other researchers in several disciplines [17, 2, 5]. This phase equation has been used in mathematical biology to study circadian rhythms and coupled oscillators in models of neurological systems [2, 1], and in electronics for the analysis of phase noise and timing jitter in oscillators [18, 5]. The acclaimed phase equation is a nonlinear but *scalar* differential equation. As such, it represents the ultimate *reduced-order* model for a complex nonlinear dynamical system. Its scalar nature and the specific form of the nonlinearity in this equation makes it possible in some cases to solve, or characterize the solutions of, this equation in (semi) analytical form, e.g., in the investigation of synchronization of coupled oscillators [17, 1] and in characterizing phase noise in electronic oscillators with stochastic perturbations as models of electronic noise sources.

1.8 Novel Contributions and Outline of the Thesis

The concise list of the work accomplished in this thesis is as follows. We focus first on reviewing the concept of isochrons and developing better (i.e., quadratic) approximations for isochrons. As a second task, we improve the famous phase model [16, 17, 2, 19, 1, 14, 5] in terms of accuracy, laying the foundations of such a study on an isochron-based framework. In the course of developing new models, we also propose simpler phase computation schemes, as algebraic equations, which are very accurate and fast methods for transient phase simulations. The last task is applying the paradigms and models we have proposed to an intricate and interesting research problem, i.e. phase computations for discrete molecular oscillators.

This thesis has the following novel contributions. First, to the best of our knowledge, this is the first work that systematically treats the problem of computing quadratic approximations for isochrons in a general setting, *without* being restricted to planar oscillators with two state variables. As a matter of fact, we are not aware of any previous work on quadratic approximations of isochrons even for planar oscillators, even though most of the literature on oscillators concentrates on the planar, two-dimensional case. Our theory and the numerical algorithm proposed applies to any N-dimensional oscillator which is described by a system of differential equations that possesses a stable periodic solution for which isochrons are assumed to exist.

Second, we present a unified framework and rigorous theory for the phase definition, the phase equations and phase computation schemes for perturbed oscillators by subsuming the work that has been done both in mathematical biology (e.g., [16, 17, 2, 14, 8, 1] and others) and in electrical engineering (e.g., [18, 5], their references and many others). In doing so, we rederive the acclaimed phase equation that has been used for decades by putting it into context and emphasizing its origins and the exact nature of the approximations it incorporates. The phase equation derivations presented in [14] in particular, from the mathematical biology literature, form the basis for our treatment on a unified theory of phase equations.

Third, we derive new, more accurate phase equations based on the theory of local quadratic isochron approximations The second-order accurate phase model, founded upon the concept of quadratic isochron approximations, will be useful in analyzing the behavior of oscillators under stronger disturbances, injection or phase-locking with stronger injection/reference signals, and the synchronization of oscillators that are not weakly coupled.

Fourth, we derive and emphasize the utility of simpler and more accurate phase computation schemes that should replace phase equations when very accurate transient phase computations are necessary and analytical phase characterizations are not desirable.

Fifth, observing that the oscillator phase computation techniques that are developed are applicable to oscillators having continuously changing states, we extend the endeavor to molecular oscillators with states inherently changing in a discrete manner. Note that for this purpose, one needs to manipulate a discrete oscillator model appropriately and employ several intricate approximations so that the developed paradigms pertaining to oscillator phase computations can capture such models.

We have proposed the quadratic isochron concept, along with the theoretical characterization and the numerical method for its computation in [20] (a preliminary version of the findings were presented in [21]). The phase computation schemes have appeared also in [20]. The unified framework of phase computation methods based on the concept of isochrons, the revealing account on phase equations, and the actual derivation of the novel second-order phase equation were published in [22]. Our preliminary findings and results on discrete molecular oscillators appeared recently in a conference presentation [23], and in [24].

The outline of the thesis on a chapter basis is as follows. In Chapter 2, we will review the literature, separately for the two disciplines, electronics and biology, that have helped build a powerful background and foundations upon which a comprehensive oscillator phase theory was constructed.

Chapter 3 includes the basic mathematical background necessary to carry out the derivations in the following chapters. Particular focus is on paradigms and quantities that must be computed using the nonlinear autonomous Ordinary Differential Equation (ODE) describing an oscillator and also on Floquet theory [9].

In Chapter 4, we review the mathematical definition and necessary properties of the sets of points called isochrons. The local approximations of isochrons are also related in this chapter. The (already known) PPV or the infinitesimal PRCs actually characterize the linear approximations of isochrons [1, 8, 14, 13, 15, 5, 12, 25]. The idea can be conceived, however, that higher order isochron approximations may help improve the accuracy of phase computation methods. Accordingly, we introduce the (new) notion of quadratic approximations for the isochrons through a matrix function that we call $\mathbf{H}(t)$ and also propose a numerical method for its computation. Local isochron information for a simple oscillator and several intricate oscillators are computed through the methods stated. The validity of the proposed algorithm for the computation of quadratic approximations, i.e., $\mathbf{H}(t)$, is confirmed with the results provided in Chapter 4. The sections of Chapter 4 that are on the theoretical characterization and numerical computation of $\mathbf{H}(t)$ were proposed in [20].

Chapter 5 reviews the definition of oscillator phase (a scalar quantity) based on the isochrons of oscillators [1, 8, 14]. The phase computation problem is stated, as it can be formulated benefitting from [14]. This problem is to be solved in practice through approximate methods that in turn depend on local isochron approximations. As required for further investigation, Chapter 5 rigorously relates the PPV $\mathbf{v}_1(t)$ (or infinitesimal PRCs) and $\mathbf{H}(t)$, characterizing the linear and quadratic approximations of isochrons, respectively, to the scalar phase of an oscillator. The contents of this chapter expand on part of the work in [22].

In Chapter 6, the oscillator phase computation methods based on the isochrons of oscillators are described. Note that the phase of a noiseless or unperturbed oscillator is uninteresting, since the phase in this case is time itself. Therefore, we focus our attention on particularly noisy and perturbed oscillators. We describe phase computation methods that depend on the exact forms of isochrons and then, utilizing the concept of local isochron approximations, devise approximate methods. The methods come in two forms: phase equations and phase computation schemes. We also put these methods into a unified framework based on several attributes. Comments on the accuracies of the approximations involved and computational complexities entailed in the methods are also provided. Phase computation experiments on several intricate oscillators are presented. This chapter details the work in [22].

All phase computation theory and methods stated up to this point capture oscillators with states changing in a continuous manner. However, in molecular scale where state changes are in discrete form, some biochemical systems exhibit oscillatory behavior as well. Such systems prove to be very important in inter- and intra-cellular operations in living or synthetically composed organisms since they maintain the upkeep of vital phenomena, such as circadian rhythms. As any other oscillator, these molecular oscillators are afflicted by noise of various origins. It is necessary to quantify the effect of noise on the phase of molecular oscillators as well. However, the devised phase computation techniques for continuous oscillators do not directly apply to such molecular oscillators with discrete states. In fact, some subtleties have to be resolved before the designed methods of phase computation can be used to calculate the phase of discrete molecular oscillators. An account describing these subtleties, how we managed to resolve them, an illustration of the phase computation methods applied to several intricate molecular oscillators is given in Chapter 7, which extends the workshop presentation in [23], and the article [24].

Lastly, Chapter 8 concludes the thesis by suggesting some further research directions.

Chapter 2

PREVIOUS WORK

In this chapter, we present a brief literature review of the previous work on analysis of oscillator phase. The focus is on the work done in two disciplines, in particular, electronics and biology. We aim to illustrate separately in these two disciplines the similarity of the ideas, theoretical and computational paradigms that have rendered the challenging task of oscillator phase analysis a thriving and rewarding research preoccupation.

2.1 Previous Work in Biology

2.1.1 PRCs (Phase Response Curves)

PRCs (Phase Response or Resetting Curves) show how an autonomous oscillator responds to a perturbing stimulation in terms of phase [26, 2, 1, 14]. The perturbation in this sense is usually taken to be a pulse usually without offset (alternatively the upper half of a sinusoid can be used) of desired duration and magnitude that "pokes" the autonomous oscillator starting at a particular time along the period. The phase change incurred by the perturbation should be measured (the temporarily perturbed oscillator state should return to the attracting periodic orbit and the phase shift with respect to an originally unperturbed and thus impeccably periodic solution should be observed and recorded). It is generally observed that an oscillator responds differently (i.e., the phase of the oscillator is affected in a different manner) when such a pulse hits a particular state variable of the oscillator at different times. This phenomenon might be attributed to the autonomous and periodically time-variant nature of oscillators. If the phase change values on the oscillator are measured experimentally, recorded and plotted in a trace for a particular type of perturbation (e.g., for a pulse with a fixed duration and magnitude) hitting one of the state variables of the oscillator independently at consecutive discrete times along the period, one obtains a PRC for that perturbation and that state variable [26, 2, 1, 14].

PRCs have been used for quantifying the perturbation and noise susceptibility of oscillators in biology. PRCs yield valuable information as to the intervals along the period during which a state variable is most sensitive to perturbations and how much the phase of the oscillator is degraded accordingly.

2.1.2 Isochrons and Infinitesimal PRCs

One may well appreciate that given a mathematical model for an oscillator in the form of a differential equation, the PRCs can be produced computationally, though in a pain-staking manner. It is then desirable to lay more insight into the computational techniques used. For this purpose, Winfree discovered the isochrons [2], the sets of equiphase points associated with the periodic orbit of an oscillator. Though Winfree was not aware of the connection, the isochrons are founded upon a mathematical property that belongs to a majority of oscillators, called "asymptotic phase" [11, 9, 1]. Guckenheimer proved the existence and mathematical properties of isochrons [11]. Through isochrons, it is possible to rigorously define oscillator phase. Therefore, all computational techniques designed for calculating oscillator phase (and PRCs in that respect) naturally depend on the concept of isochrons.

Isochrons are very hard to compute numerically and the few techniques for calculating them are not really practicable for oscillators of high dimension [1]. Considering this difficulty, the computational quantities called *infinitesimal PRCs* were introduced [2, 1], a premise on which PRCs for desired types of pulses or any other perturbation can be computed through the techniques that were derived and to be stated shortly. In a sense, infinitesmall PRCs serve as a sort of impulse response which can then be used to compute response to other perturbations. We will also elucidate the connection between isochrons and infinitesimal PRCs.

We describe now how infinitesimal PRCs are calculated [2]. A particular state variable in the mathematical model of the oscillator is excited, at a particular timepoint along the period, with a narrow pulse, approximating in essence an impulse although the magnitude should be limited so as to make the oscillator states remain within the domain of attraction of the periodic orbit. More precisely, the magnitude of the pulse should be small enough so that the phase response changes linearly with this magnitude. Then, the phase change incurred by the perturbation is computed as described above, in the case of PRCs. This phase shift should then be scaled so that it becomes the phase shift due to an impulse of proper magnitude. These phase shifts computed as such at a set of discrete timepoints for a particular oscillator state variable constitutes the (inherently periodic) infinitesimal PRC for that variable.

It has been pointed out that the vector composed of the infinitesimal PRCs stacked together is the periodic solution of the adjoint LPTV (Linear Periodically Time Varying) system obtained through linearization of the original differential equation describing an oscillator [16, 1]. It can also be shown that if an isochron is linearized around the point on the periodic orbit that it passes through, the resulting hyperplane is orthogonal to this vector of infinitesimal PRCs corresponding to that point on the orbit [14, 1].

2.1.3 Phase Models in Biology

Having now all the necessary ingredients, the aim is to design a model or a method through which general PRCs (corresponding to particular types of perturbations) can be calculated making use of the infinitesimal PRCs. To the best of our knowledge it was Malkin in 1949, who designed the phase model for weakly perturbed oscillators [16]. Later, Kuramoto independently came up with virtually the same model [17]. Winfree, having the concept of isochrons at his disposal, developed again the same model through a geometrical point of view that he adopted [2]. It is to be emphasized that all these scientists made use of the infinitesimal PRCs in designing their models.

The works of Malkin [16] and Kuramoto [17] captured the phase computation of coupled oscillators as well. Later, Hoppensteadt and Izhikevich reviewed in their work the model developed independently by Malkin, Kuramoto, and Winfree [19]. Brown *et al.* outlined the derivation of the model particularly of Winfree, using the isochron concept [14].

Other interesting and useful findings based on the phase model formalism or related phenomenal insight include the case where the forcing perturbation is noise of stochastic nature. It has been shown in various biological disciplines that white noise forcing on oscillators causes phase diffusion and spreading around harmonics in the power spectrum [27, 28].

2.2 Previous Work in Electronics

2.2.1 ISFs (Impulse Sensitivity Functions) and PPV (Perturbation Projection Vector)

Culmination of the work of about six decades on oscillator phase analysis in electronics occurred in the 1990s. Parallels of the isochron concept and (infinitesimal) PRCs of biology emerged in the form of ISFs and the PPV.

The concept of isochrons and geometrical point of view in the development of oscillator phase models was missing in electronics until recently [25]. However, other paradigms have been in use. It has to be noted that computation of the so-called ISFs as described in [29] is exactly the same as for infinitesimal PRCs explained above. In this sense, we may rightfully state that ISFs in electronics and infinitesimal PRCs in biology are exactly the same quantity.

The PPV (introduced and utilized in [18, 5, 12]) is described to be exactly the vector that comprises the ISFs and is obtained by solving for the periodic solution of the adjoint LPTV system derived from the differential equation model of an oscillator [5, 12]. Therefore, the PPV is exactly the vector that emerges (the one that consists of the infinitesimal PRCs) in the phase models of Malkin [16], Kuramoto [17], and Winfree [2].

In [5], a method of numerical computation for the PPV is outlined, apparently based on the shooting algorithm [30]. A more accurate, harmonic balance based algorithm for again PPV calculation is given in [12].

2.2.2 Phase Models in Electronics

Having drawn the parallels between the computational paradigms in electronics and biology, it is not surprising now to state that the phase models developed in electronics [5] are extraordinarily similar to those in biology [16, 17, 2, 19, 1, 14]. Particularly, the seminal work of Demir *et al.* in [5] is partly inspired by the work of Kaertner [18], who makes use of Floquet theory [9] to describe the vector we have stated to be consisting of the ISFs [29], this vector coming to be called the celebrated PPV [12].

A rigorous characterization of phase noise through SDEs (Stochastic Differential Equations) forced by white noise inherent in electronic devices and related semi-analytical investigation based on stochastic calculus is given also in [5]. Phase diffusion and the pertaining diffusion of power into nearby frequencies in the power spectrum has been mathematically justified [5]. Recall that similar analytical findings are also reported in biology [27, 28].

2.3 Summary

We have pointed out the similarities and direct parallels between paradigms and models used in biology and electronics, in the study of oscillator phase. It appears that while the pertaining foundations in biology bear outstanding theoretical value (such as the concept of isochrons and accordingly the rigorous definition of phase), in electronics exquisite developments lead to reliable and efficient numerical calculation of pertaining computational quantities (such as the infinitesimal PRCs, or with aliases, ISFs or the PPV). The renowned phase model was developed independently by several scientists in the two disciplines. Both algebraic and geometrical (regarding the use of the isochron concept) points of view were taken in this development. Further semi-analytical insight into the phase phenomena of oscillators through this phase model revealed astounding discoveries that marked an alliance of experimental observations and theoretical findings (as in the case of inherent phase diffusion and spectral power density spreading around the harmonics).

Chapter 3

BACKGROUND AND PRELIMINARIES

In this chapter, we review a set of mathematical tools that will be beneficial in the following chapters, while presenting the revealing account on oscillator phase and its numerical computation. Section 3.1 includes some basic definitions and properties on mathematical models of oscillators. Section 3.2 provides a few properties of an amazingly helpful tool, called Floquet theory, which facilitates the mathematical treatment of oscillator models.

3.1 Oscillator Models

The dynamics of an oscillator is described by a system of autonomous ODEs (Ordinary Differential Equations):

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

where $\mathbf{x} \in \Re^N$ and $\mathbf{f} : \Re^N \to \Re^N$ is a nonlinear vector function.

Definition 3.1.1 (STF) *The* State Transition Function *(STF)* Φ *of the ODE in (3.1) is defined by*

$$\mathbf{x}(t) = \Phi(t, s, \mathbf{x}(s)). \tag{3.2}$$

Φ merely returns the solution of (3.1) at time t, given an initial condition $\mathbf{x}(s)$ at time s. The analytical form of Φ is not available except in some very simple cases, but it can be computed numerically.

Under certain conditions, (3.1) is the mathematical representation of an oscillator and has possibly more than one periodic solutions. Usually, only one of these periodic solutions is of interest. We denote this particular solution by $\mathbf{x}_\mathrm{s}(t)$ and the period of $\mathbf{x}_\mathrm{s}(t)$ by T.

Definition 3.1.2 (Limit Cycle or Periodic Orbit) *The set of points given by*

$$\gamma = \left\{ \mathbf{x} \in \Re^N \middle| \ \mathbf{x} = \mathbf{x}_\mathrm{s}(t), \forall t \in \Re^+ \right\} \tag{3.3}$$

is called the limit cycle *or the* periodic orbit *traced by* $\mathbf{x}_\mathrm{s}(t)$.

There are specifically two crucial properties that the limit cycles of the oscillators that are of interest to us possess, namely *asymptotic orbital stability* and *asymptotic phase* [9]:

Definition 3.1.3 (Asymptotic Orbital Stability) *Let* $\mathbf{x}_0(t)$ *be a solution of the ODE in (3.1) such that the distance between* γ*, the limit cycle as in Definition 3.1.2, and* $\mathbf{x}_0(t)$ *approaches zero as* $t \to +\infty$*. If there are such solutions around* γ*, then* γ *is said to possess* asymptotic orbital stability *[9]. The initial conditions* $\mathbf{x}_0(t=0)$*, for those solutions* $\mathbf{x}_0(t)$ *that ultimately run within an* ϵ*-tube of points on* γ*, reside in the set* $\mathcal{W}$*, the* domain of attraction*, i.e.,* $\mathbf{x}_0(0) \in \mathcal{W}$*. Clearly,* $\mathbf{x}_\mathrm{s}(t) \in \mathcal{W}, \forall t$*.*

Definition 3.1.4 (Asymptotic Phase) $\mathbf{x}_\mathrm{s}(t)$ *has the* asymptotic phase *property if for each solution* $\mathbf{x}_0(t)$*, such that* $\mathbf{x}_0(0) \in \mathcal{W}$*, there is a constant* $\alpha(\mathbf{x}_0(0))$ *(called the asymptotic phase), such that*

$$\lim_{t\to+\infty} [\mathbf{x}_0(t) - \mathbf{x}_\mathrm{s}(t + \alpha(\mathbf{x}_0(0)))] = 0. \tag{3.4}$$

3.2 Floquet Theory

Expanding (3.1) to first-order (linearization) around γ, we obtain

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{G}(t)\mathbf{y}, \tag{3.5}$$

with $\mathbf{G}(t) = \partial\mathbf{f}(\mathbf{x}_\mathrm{s}(t))/\partial\mathbf{x}_\mathrm{s}(t)$. (3.5) is an LPTV (Linear Periodically Time-Varying) system.

Definition 3.2.1 (STM for LPTV) *The* State Transition (Sensitivity) Matrix *(STM) of (3.5), from time* s *to time* t*, is defined as follows as the* Jacobian *of the state transition function* Φ *in Definition 3.1.1 evaluated on the periodic orbit represented by* $\mathbf{x}_\mathrm{s}(t)$*:*

$$\Upsilon(t,s) = \frac{\partial\mathbf{x}_\mathrm{s}(t)}{\partial\mathbf{x}_\mathrm{s}(s)} = \frac{\partial\Phi(t,s,\mathbf{x}_\mathrm{s}(s))}{\partial\mathbf{x}_\mathrm{s}(s)}. \tag{3.6}$$

$\Upsilon(t,0)$ is the matrix solution of (3.5) at time t, with the initial condition $\Upsilon(0,0) = \mathbf{I}_N$, where $\mathbf{I}_N$ is the $(N \times N)$ identity matrix. Note that $\Upsilon(t,0)$ is nonsingular for all time.

Remark 3.2.1 *It can be easily checked that* $\dot{\mathbf{x}}_\mathrm{s}(t)$*, dot denoting time derivative, solves (3.5) and* $\dot{\mathbf{x}}_\mathrm{s}(0)$ *is an eigenvector of* $\Upsilon(T,0)$ *(the* Monodromy Matrix*) corresponding to the eigenvalue 1.*

Through *Floquet Theory* [9], $\Upsilon(T, 0)$ can be expressed as follows

$$\Upsilon(t, s) = \sum_{i=1}^{N} \exp(\mu_i(t - s))\mathbf{u}_i(t)\mathbf{v}_i^{\mathsf{T}}(s), \tag{3.7}$$

where $\mathbf{u}_i(t)$ and $\mathbf{v}_j(t)$, for $1 \leq i, j \leq N$, are T-periodic Floquet vector functions and form two *bi-orthonormal sets*, i.e.,

$$\mathbf{v}_j^{\mathsf{T}}(t)\mathbf{u}_i(t) = \delta_{ij} \quad \forall t \in \Re \tag{3.8}$$

and the constants $\mu_i \in \mathbb{C}$ are called the *Floquet exponents*. It can be verified that the vector solutions of (3.5) are linear combinations of $\exp(\mu_i t)\mathbf{u}_i(t)$, $1 \leq i \leq N$.

Remark 3.2.2 *We set $\mu_1 = 0$ and $\mathbf{u}_1(t) = \dot{\mathbf{x}}_{\mathrm{s}}(t)$, in view of Remark 3.2.1.*

Remark 3.2.3 *All of the other Floquet exponents are assumed to be such that $\mathtt{Re}\{\mu_i\} < 0$ for $2 \leq i \leq N$. This condition, i.e., having a single Floquet exponent that is equal to zero with the rest having negative real parts, ensures that the limit cycle γ possesses the asymptotic orbital stability and the asymptotic phase properties [9].*

The adjoint form of (3.5) is given by

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = -\mathbf{G}^{\mathsf{T}}(t)\mathbf{z}. \tag{3.9}$$

The state transition matrix of (3.9) can be expressed as follows

$$\Upsilon^{\mathsf{T}}(s, t) = \sum_{i=1}^{N} \exp(-\mu_i(s - t))\mathbf{v}_i(s)\mathbf{u}_i^{\mathsf{T}}(t), \tag{3.10}$$

in terms of the Floquet components defined above. $\Upsilon^{\mathsf{T}}(t, 0)$ is the matrix solution of (3.9) at time t, with the initial condition $\Upsilon^{\mathsf{T}}(0, 0) = \mathbf{I}_N$. The vector solutions of (3.9) are linear combinations of $\exp(-\mu_i t)\mathbf{v}_i(t)$, $1 \leq i \leq N$. It follows that the scalar product of the general vector solution $\mathbf{z}(t)$ of (3.9) and the general vector solution $\mathbf{y}(t)$ of (3.5), i.e., $\mathbf{z}^{\mathsf{T}}(t)\mathbf{y}(t)$, is constant for all time.

Figure 3.1 summarizes the above account. Through linearization of the autonomous ODE in (3.1), the generic forms for the forward and adjoint LPTV equations are found as in (3.5) and (3.9), respectively. The relation between the solutions of these two equations is noted. Floquet theory provides more insight into the solutions than the immediate observations can supply. In Figure 3.1, we also remark the significance of $\mathbf{u}_1(t)$ and $\mathbf{v}_1(t)$ along with the biorthonormal sets formed by the solutions to (3.5) and (3.9).

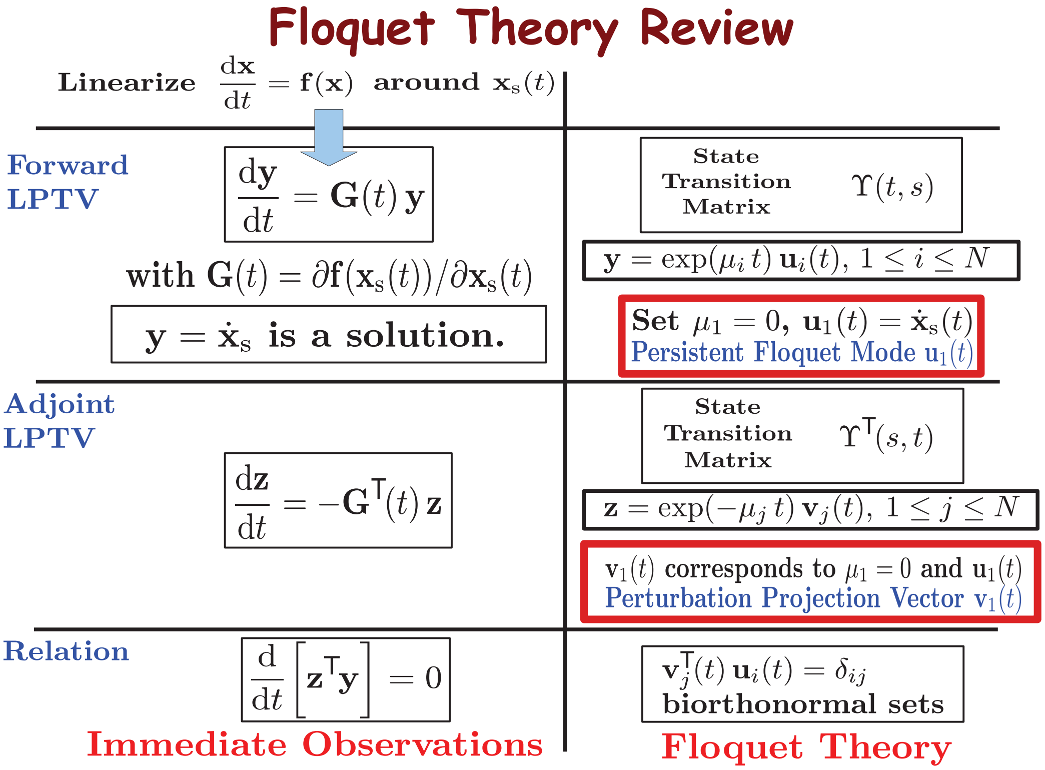

Floquet Theory Review
Linearize $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x})$ around $\mathbf{x}_s(t)$
Forward LPTV
$\frac{d\mathbf{y}}{dt} = \mathbf{G}(t)\,\mathbf{y}$
with $\mathbf{G}(t) = \partial\mathbf{f}(\mathbf{x}_s(t))/\partial\mathbf{x}_s(t)$
$\mathbf{y} = \dot{\mathbf{x}}_s$ is a solution.
State Transition Matrix $\Upsilon(t,s)$
$\mathbf{y} = \exp(\mu_i\,t)\,\mathbf{u}_i(t),\ 1 \le i \le N$
Set $\mu_1 = 0$, $\mathbf{u}_1(t) = \dot{\mathbf{x}}_s(t)$
Persistent Floquet Mode $\mathbf{u}_1(t)$
Adjoint LPTV
$\frac{d\mathbf{z}}{dt} = -\mathbf{G}^{\mathsf{T}}(t)\,\mathbf{z}$
State Transition Matrix $\Upsilon^{\mathsf{T}}(s,t)$
$\mathbf{z} = \exp(-\mu_j\,t)\,\mathbf{v}_j(t),\ 1 \le j \le N$
$\mathbf{v}_1(t)$ corresponds to $\mu_1 = 0$ and $\mathbf{u}_1(t)$
Perturbation Projection Vector $\mathbf{v}_1(t)$
Relation
$\frac{d}{dt}\left[\mathbf{z}^{\mathsf{T}}\mathbf{y}\right] = 0$
$\mathbf{v}_j^{\mathsf{T}}(t)\,\mathbf{u}_i(t) = \delta_{ij}$
biorthonormal sets
Immediate Observations
Floquet Theory

Figure 3.1: Floquet theory review.

Since $\exp(\mu_i t)\mathbf{u}_i(t)$, $1 \le i \le N$, solves the forward LPTV equation in (3.5), we obtain through substitution and manipulations

$$\frac{d\mathbf{u}_i(t)}{dt} - \mathbf{G}(t)\mathbf{u}_i(t) = -\mu_i\mathbf{u}_i(t) \tag{3.11}$$

which can be considered as an eigenvalue equation for the operator $\frac{d\cdot}{dt} - \mathbf{G}(t)$ with $-\mu_i$ as the eigenvalue and $\mathbf{u}_i(t)$ as the eigenfunction. Similarly, since $\exp(-\mu_i t)\mathbf{v}_i(t)$, $1 \le i \le N$, solves the adjoint LPTV equation in (3.9), we have

$$-\frac{d\mathbf{v}_i(t)}{dt} - \mathbf{G}^{\mathsf{T}}(t)\mathbf{v}_i(t) = -\mu_i\mathbf{v}_i(t). \tag{3.12}$$

These eigenvalue equations can be solved numerically (as both an eigenvalue and a periodic boundary value problem) to compute all of μ_i, $\mathbf{u}_i(t)$ and $\mathbf{v}_i(t)$ for $1 \le i \le N$.

Definition 3.2.2 *The PPV (Perturbation Projection Vector) is defined as the T-periodic vector solution of the adjoint LPTV equation in (3.9), i.e., $\mathbf{v}_1(t)$ corresponding to the Floquet* *exponent* $\mu_1 = 0$ *(*$\mathbf{u}_1(t) = \dot{\mathbf{x}}_s(t)$ *corresponds to the same exponent, see Remark 3.2.2), which satisfies the following* normalization condition

$$\mathbf{v}_1^{\intercal}(t)\mathbf{u}_1(t) = 1, \quad \forall t \in \Re, \tag{3.13}$$

due to the bi-orthonormality condition in (3.8).

The PPV is intimately related to isochrons and plays a pivotal role in deriving scalar phase domain models for oscillators [5]. The so-called infinitesimal phase response curves (PRCs) along state variables that are used in studying the perturbation behavior of biological oscillators [1] are in fact nothing but the entries of the PPV, which have been used in performing perturbation analysis for electronic oscillators [12, 25]. The adjoint LPTV equation in (3.9) that defines the PPV seems to have first appeared in Malkin's work (1949) on the reduction of weakly perturbed oscillators to their phase models [1]. The adjoint equation and the normalization condition in (3.13) has been used in mathematical biology in computing PRCs [1, 8, 14, 13, 15], and in electronics, in computing the PPV [5, 12, 25].

Chapter 4

ISOCHRONS OF OSCILLATORS AND THEIR LOCAL APPROXIMATIONS

Isochrons are the sets associated with a periodic orbit (or equivalently a limit cycle) of an oscillator. These sets were discovered and introduced first in mathematical biology by the two pioneers, Winfree (1974) [2] and Guckenheimer (1975) [11]. They have been beneficial in articulating a rigorous definition of oscillator phase [8, 1] and also in structuring numerical methods for oscillator phase computations [2, 8, 1, 5, 12].

In this chapter, we first review isochrons and their basic useful properties along with the difficulties encountered in their numerical calculations (Section 4.1). Computing local approximations of isochrons (local around the limit cycle) seems to be the best route to take in order to incorporate isochrons to numerical oscillator phase computation studies. Section 4.2 is then about the numerical computation of local isochron approximations. As we have related in previous chapters, linear approximations of isochrons are already known in the literature and have been extensively used in phase computations [1, 8, 14, 13, 15, 5, 12, 25]. We introduce the quadratic approximations of isochrons as a novel paradigm and also propose numerical methods for computing them, as explained in detail in Section 4.2. The local approximations for isochrons are going to be most helpful in deriving the practically usable oscillator phase computation methods of Chapter 6. In Section 4.3, we test the algorithms for numerical isochron computations on several oscillators.

4.1 Isochrons of Oscillators

In this section, we review the notion of isochrons for an oscillator, on which the phase computation schemes and the phase equations we discuss in Chapter 6 are founded. We provide a formal definition, summarize their basic properties (Section 4.1.1) and review computational techniques for characterizing the isochrons of an oscillator (Section 4.1.2).

4.1.1 Isochrons and Basic Properties

Isochrons of oscillators are associated with stable periodic solutions (i.e., limit cycles). Isochrons can be simply described as follows [8, 1] (see Figure 4.1): $\mathbf{x}_0 = \mathbf{x}(0)$ (on γ) and $\mathbf{y}_0$ are two initial points that generate the trajectory solutions $\mathbf{x}(t)$ (on γ) and $\mathbf{y}(t)$, respectively, such that $\mathbf{y}(t) \to \mathbf{x}(t)$ as $t \to \infty$. The set of all such points in $\mathcal{W}$ like $\mathbf{y}_0$ (satisfying the same condition) is called the *isochron* (denoted by η_0) of $\mathbf{x}_0$ [8, 1]. All points constituting the isochron of $\mathbf{x}_0$ are thought to be *synchronized* with $\mathbf{x}_0$, i.e., they are in *phase*. The formal definition of isochrons is as follows:

Definition 4.1.1 (Isochrons) *Let $\mathbf{x}_\eta(t)$ be a solution of (3.1) such that $\mathbf{x}_\eta(t_0) \in \mathcal{W}$. The isochron with time tag t_0, which is to be called η_{t_0}, is defined as the set*

$$\eta_{t_0} = \left\{ \mathbf{x}_\eta(t_0) \,\middle|\, \lim_{\tau \to +\infty} \Big[\mathbf{x}_\eta(\tau) - \mathbf{x}_{\mathrm{s}}(\tau) \Big] = 0 \right\} \tag{4.1}$$

where $\mathbf{x}_\eta(\tau) = \Phi(\tau, t_0, \mathbf{x}_\eta(t_0))$ and, $\mathbf{x}_{\mathrm{s}}(\tau) = \Phi(\tau, t_0, \mathbf{x}_{\mathrm{s}}(t_0))$ is the periodic solution.

An isochron comprises points having the same asymptotic phase [9, 10]. For an N-dimensional oscillator, each isochron is an $N-1$-*dimensional hypersurface*. The union of isochrons covers the neighborhood of the orbit [8, 1].

The following property of isochrons serves as the key in defining a generalized phase for an oscillator. It is related to the periodic behavior of any solution of (3.1) (not just $\mathbf{x}_{\mathrm{s}}(t)$) with its initial condition in $\mathcal{W}$: Any general solution $\mathbf{x}(t)$ of (3.1) with $\mathbf{x}(0) \in \mathcal{W}$ (which is not periodic if $\mathbf{x}(0)$ is not on γ) hits exactly the same isochron at subsequent points in time exactly T (the period on γ) apart from each other, albeit at different locations. The formal proof requires the following lemma:

Lemma 4.1.1 *By definition, the point $\mathbf{x}_{\mathrm{s}}(0)$ on the limit cycle γ resides on the isochron with time tag 0, i.e., $\mathbf{x}_{\mathrm{s}}(0) \in \eta_0$. Let $\mathbf{x}(t)$ be a solution of (3.1) with the initial condition $\mathbf{x}(0) \in \eta_0$, i.e., $\mathbf{x}(0)$ resides on the same isochron as $\mathbf{x}_{\mathrm{s}}(0)$. Then, for any t, both $\mathbf{x}(t)$ and the steady-state periodic solution $\mathbf{x}_{\mathrm{s}}(t)$ reside on the same isochron η_t, i.e. $\mathbf{x}(t), \mathbf{x}_{\mathrm{s}}(t) \in \eta_t$ [2].*

Proof: By definition, we have

$$\lim_{\tau \to +\infty} \Upsilon(\tau, 0, \mathbf{x}(0)) = \lim_{\tau \to +\infty} \Upsilon(\tau, 0, \mathbf{x}_{\mathrm{s}}(0)). \tag{4.2}$$

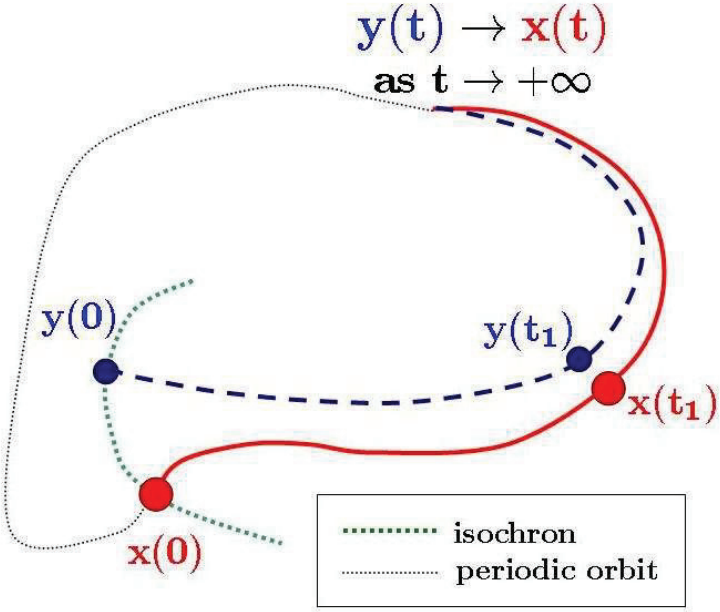


Figure 4.1: A stable, attracting periodic orbit and one of its isochrons.

Since

$$\Upsilon(\tau, t, \mathbf{x}_\mathrm{s}(t)) = \Upsilon(\tau, t, \Upsilon(t, 0, \mathbf{x}_\mathrm{s}(0))) \tag{4.3}$$

for $\tau > t$, we obtain

$$\Upsilon(\tau, t, \mathbf{x}_\mathrm{s}(t)) = \Upsilon(\tau, 0, \mathbf{x}_\mathrm{s}(0)). \tag{4.4}$$

Similarly, we have

$$\Upsilon(\tau, t, \mathbf{x}(t)) = \Upsilon(\tau, 0, \mathbf{x}(0)) \tag{4.5}$$

for $\tau > t$. Then, through (4.4), (4.5), and (4.2), we obtain

$$\lim_{\tau \to +\infty} \Upsilon(\tau, t, \mathbf{x}(t)) = \lim_{\tau \to +\infty} \Upsilon(\tau, t, \mathbf{x}_\mathrm{s}(t)) \tag{4.6}$$

which proves that $\mathbf{x}(t), \mathbf{x}_\mathrm{s}(t) \in \eta_t$. ■

The following theorem precisely characterizes the periodic behavior of a general solution:

Theorem 4.1.1 *Let $\mathbf{x}(t)$ be a solution of (3.1), with the initial condition $\mathbf{x}(0)$ residing on a particular isochron, i.e., $\mathbf{x}(0) \in \eta_0$. Then, $\mathbf{x}(nT) \in \eta_0$, for any integer n, i.e., the solution (3.1) hits the same isochron η_0 periodically, with the same period T of the limit cycle γ or the periodic solution $\mathbf{x}_s(t)$.*

Proof: By Lemma 4.1.1 and the hypothesis, it must be true that $\mathbf{x}(nT), \mathbf{x}_{\mathrm{s}}(nT) \in \eta_{nT}$. However, since $\mathbf{x}_{\mathrm{s}}(nT) = \mathbf{x}_{\mathrm{s}}(0)$, η_{nT} and η_0 must be the same isochron. ■

4.1.2 Computing Isochrons

Due to their importance and utility in oscillator analysis (as we later demonstrate in this thesis), finding (calculating) the isochrons of periodic solutions for oscillators is crucial. However, this is a very difficult task [2, 1]. Analytical methods are out of the question most of the time [8]. The main numerical method for computing isochrons [8, 1] is based on the property of isochrons captured by Theorem 4.1.1 above. A detailed description can be found in Chapter 1 and Section 4.3.2. Basically, the difficulty of implementation and the exponential complexity entailing point-by-point constructions of surfaces are the shortcomings of this method, which preclude its feasibility for oscillators other than planar ones.

4.2 Local Approximations of Isocrons

In this section, we review the linear approximations for isochrons (defined by the PPV $\mathbf{v}_1(t)$) and explain in detail the quadratic approximation, which furthermore require a matrix function $\mathbf{H}(t)$ for their characterization. Section 4.2.1 is an overview of this section on local approximations for isochrons. Section 4.2.2 below shows how the two functions $\mathbf{v}_1(t)$ and $\mathbf{H}(t)$ fully characterize the quadratic approximations of isochrons and how the particular formulation of the account helps us rigorously observe that indeed $\mathbf{v}_1(t)$ is the vector defining the hyperplanar (linear) approximations for isochrons on exactly the points on the limit cycle. Section 4.2.3 includes the theoretical analysis of the novel function $\mathbf{H}(t)$, and in Section 4.2.4 we propose a numerical method for the computation of $\mathbf{H}(t)$.

4.2.1 Overview

A tractable technique for isochron computations is based on *local approximations* around points on γ [8], since useful oscillators stay close to their limit cycles even when afflicted by perturbations. The linear approximation (i.e., hyperplane) of a surface (in this case an isochron) in the neighborhood of a point (on γ) is characterized by a vector (normal to the hyperplane). It turns out that the normal vector is the PPV $\mathbf{v}_1(t)$, the vector whose entries are the infinitesimal PRCs [8, 21]. Therefore, the linear approximation for the isochron η_t passing through $\mathbf{x}_\mathrm{s}(t)$ is given by the hyperplane equation

$$\mathbf{v}_1^\mathsf{T}(t)\,\mathbf{y} = 0 \tag{4.7}$$

where $\mathbf{y} = \mathbf{x} - \mathbf{x}_\mathrm{s}(t)$, i.e., the loci of points $\mathbf{x}$ satisfying (4.7) define a hyperplane tangent to η_t at $\mathbf{x}_\mathrm{s}(t)$. We will shortly show that quadratic approximations can be characterized by a quadric hypersurface equation given by

$$\mathbf{v}_1^\mathsf{T}(t)\,\mathbf{y} + \frac{1}{2}\,\mathbf{y}^\mathsf{T}\,\mathbf{H}(t)\,\mathbf{y} = 0 \tag{4.8}$$

where $\mathbf{H}(t) = \partial \mathbf{v}_1(t)/\partial \mathbf{x}_\mathrm{s}(t)$ is the PPV Jacobian, computed numerically through intricate BVP (Boundary Value Problem) formulations. (4.8) captures the linear case in (4.7) when the quadratic term in (4.8) is omitted. $\mathbf{v}_1(t)$ and $\mathbf{H}(t)$ are crucial quantities, and we will hereon assume that these are available numerically at a set of points along the limit cycle γ. The technically detailed and revealing explanations on local isochron approximations are shortly to be presented.

4.2.2 Linear and Quadratic Approximations for Isochrons

We now formulate the quadratic approximation of isochrons problem formally and in a precise manner. We first define a matrix $\mathbf{H}(t)$ as follows:

Definition 4.2.1 $\mathbf{H}(t) \in \Re^{N\times N}$ *is defined to be the Jacobian of the PPV, i.e.* $\mathbf{v}_1(t)$*, evaluated on the periodic orbit represented by* $\mathbf{x}_\mathrm{s}(t)$*:*

$$\mathbf{H}(t) = \frac{\partial \mathbf{v}_1(\mathbf{x}_\mathrm{s}(t))}{\partial \mathbf{x}_\mathrm{s}(t)}. \tag{4.9}$$

Note that, in the above definition, PPV $\mathbf{v}_1 \in \Re^N$ is formally considered to be a function of $\mathbf{x}_\mathrm{s}(t) \in \Re^N$, with $\mathbf{v}_1(t) = \mathbf{v}_1(\mathbf{x}_\mathrm{s}(t))$. The PPV is a periodic solution of the adjoint equation in (3.9) obtained by linearizing the oscillator equations in (3.1) around the periodic solution $\mathbf{x}_\mathrm{s}(t)$. Hence, $\mathbf{H}(t)$ defined as the Jacobian of the PPV $\mathbf{v}_1$ with respect to $\mathbf{x}_\mathrm{s}(t)$ is meaningful and nonzero. For notational simplicity, we use $\mathbf{v}_1(t)$ instead of $\mathbf{v}_1(\mathbf{x}_\mathrm{s}(t))$. Note that $\mathbf{H}(t)$ is also defined at every point of the periodic orbit, parameterized by time t.

We next state and prove one of the main results of this chapter in the thesis, which rigorously reveals the crucial roles of the PPV $\mathbf{v}_1(t)$ and $\mathbf{H}(t)$ in forming linear and quadratic approximations for isochrons:

Theorem 4.2.1 *Let $\mathbf{x}_\eta(t)$ and $\mathbf{x}_\mathrm{s}(t)$ be on the same isochron, which is η_t according to the notation given in Definition 4.1.1. The locus of the vectors $\mathbf{y}$ that satisfy*

$$\mathbf{v}_1^\intercal(t)\mathbf{y} + \frac{1}{2}\mathbf{y}^\intercal \mathbf{H}^\intercal(t)\mathbf{y} = 0 \tag{4.10}$$

with $\mathbf{y} = \mathbf{x}_\eta(t) - \mathbf{x}_\mathrm{s}(t)$ form the quadratic approximation for η_t around $\mathbf{x}_\mathrm{s}(t)$. (4.10) represents an $N-1$-dimensional quadric *surface that forms a second-order approximation (expansion) for the isochron η_t (an $N-1$-dimensional hypersurface) passing through $\mathbf{x}_\mathrm{s}(t)$.*

Proof: With $\mathbf{x}_\eta(t) = \mathbf{x}_\mathrm{s}(t) + \mathbf{y}$, the Taylor expansion of $\mathbf{x}_\eta(\tau)$ around $\mathbf{x}_\mathrm{s}(t)$ can be written as

$$\mathbf{x}_\eta(\tau) = \Phi(\tau, t, \mathbf{x}_\eta(t)) = \mathbf{x}_\mathrm{s}(\tau) + \mathbf{LIN} + \mathbf{QUAD} + \mathbf{R} \tag{4.11}$$

where $\mathbf{LIN} = \mathbf{LIN}(\tau, t)$ (see (4.14) below) and $\mathbf{QUAD} = \mathbf{QUAD}(\tau, t)$ (see (4.15) below) and $\mathbf{R} = \mathbf{R}(\tau, t) = O(c(\tau)||\mathbf{y}||^3)$ (for some function $c(\tau)$). Assuming that at $\tau = t$ the values $\mathbf{x}_\eta(t)$ and $\mathbf{x}_\mathrm{s}(t)$ are on the same isochron yields

$$\lim_{\tau\to+\infty} \Big[\mathbf{LIN} + \mathbf{QUAD} + \mathbf{R}\Big] = 0. \tag{4.12}$$

We assume that $||\mathbf{R}|| < \varepsilon||\mathbf{LIN} + \mathbf{QUAD}||$ for some $\varepsilon < 1$. Then with $\mathbf{Z} = \mathbf{LIN} + \mathbf{QUAD}$, we have

$$\begin{aligned} ||\mathbf{Z}|| = ||\mathbf{Z} + \mathbf{R} - \mathbf{R}|| &\leq ||\mathbf{Z} + \mathbf{R}|| + ||\mathbf{R}|| \\ &\leq ||\mathbf{Z} + \mathbf{R}|| + \varepsilon||\mathbf{Z}|| \end{aligned}$$

Hence $(1-\varepsilon)||\mathbf{Z}|| \leq ||\mathbf{Z} + \mathbf{R}|| \to 0$, which then implies due to (4.12)

$$\lim_{\tau\to+\infty} [\mathbf{LIN} + \mathbf{QUAD}] = 0 \tag{4.13}$$

It is clear that one may solve for $\mathbf{y}$ from (4.13) provided that the natures of the partial derivative expressions in the explicit forms for $\mathbf{LIN}$ and $\mathbf{QUAD}$ as in

$$\mathbf{LIN} = \frac{\partial \Phi(\tau, t, \mathbf{x}_\mathrm{s}(t))}{\partial \mathbf{x}_\mathrm{s}(t)} \mathbf{y} \tag{4.14}$$

and

$$\mathbf{QUAD} = \frac{1}{2} \frac{\partial^2 \Phi(\tau, t, \mathbf{x}_\mathrm{s}(t))}{\partial \mathbf{x}_\mathrm{s}(t) \partial \mathbf{x}_\mathrm{s}(t)} (\mathbf{y} \otimes \mathbf{y}) \tag{4.15}$$

are demystified. Notice that in (4.15) the second order partial derivative is an $N \times N^2$ matrix and $\mathbf{y} \otimes \mathbf{y}$ is a Kronecker product with size $N^2 \times 1$. Therefore, it will be better to resort to an analysis utilizing indices. We will focus on (4.14) first. Examining (3.6) and (3.7), a single entry of **LIN** is

$$LIN_j = \sum_{k=1}^{N}\sum_{i=1}^{N} \exp(\mu_i(\tau - t))u_{i,j}(\tau)v_{i,k}(t)y_k \tag{4.16}$$

where $u_{i,j}$ and $v_{i,k}$ are the jth and kth entries of $\mathbf{u}_i$ and $\mathbf{v}_i$ respectively. We now analyze (4.15). In view of (3.6) and (3.7),

$$\frac{\partial \Upsilon_{jk}(\tau,t)}{\partial x_{s,l}(t)} = \sum_{i=1}^{N} \exp(\mu_i(\tau - t))u_{i,j}(\tau)\frac{\partial v_{i,k}(t)}{\partial x_{s,l}(t)}, \tag{4.17}$$

where Υ_{jk} and $x_{s,l}$ are the (j,k)th and lth entries of Υ and $\mathbf{x}_\mathrm{s}$ respectively. Then, by (4.15)

$$QUAD_j = \frac{1}{2}\sum_{l=1}^{N}\sum_{k=1}^{N}\sum_{i=1}^{N} \exp(\mu_i(\tau - t))u_{i,j}(\tau)\frac{\partial v_{i,k}(t)}{\partial x_{s,l}(t)}y_k y_l. \tag{4.18}$$

For $\tau \to +\infty$, (4.16) and (4.18) reduce to

$$LIN_{j,\infty} = u_{1,j}(\tau)\left[\sum_{k=1}^{N} v_{1,k}(t)y_k\right] \tag{4.19}$$

and

$$QUAD_{j,\infty} = u_{1,j}(\tau)\left[\frac{1}{2}\sum_{l=1}^{N}\sum_{k=1}^{N}\frac{\partial v_{1,k}(t)}{\partial x_{s,l}(t)}y_k y_l\right], \tag{4.20}$$

respectively, since $\mathtt{Re}\{\mu_i\} < 0$ for $2 \le i \le N$. Since $u_{1,j}(\tau)$ cannot be all zero for $1 \le j \le N$, we may choose a j such that $u_{1,j}(\tau) \neq 0$ and then divide both (4.19) and (4.20) by this $u_{1,j}(\tau)$. Using Definition 4.2.1 in (4.20) and noting (4.19), (4.13) reduces to (4.10). ■

Theorem 4.2.1 above, one of the novel contributions of this thesis, precisely characterizes the quadratic approximations for isochrons and captures linear approximations (previously known in the literature) as a special case:

Corollary 4.2.1 *If the quadratic term is omitted in (4.10), we obtain the linear approximation for isochrons*

$$\mathbf{v}_1^\intercal(t)\mathbf{y} = 0, \tag{4.21}$$

which represents a hyperplane that is tangent to the isochron η_t passing through $\mathbf{x}_\mathrm{s}(t)$. The PPV $\mathbf{v}_1(t)$ is obviously the normal vector for this hyperplane.

In order to utilize the result in Theorem 4.2.1 to obtain a practical method for computing quadratic approximations for isochrons, one needs the means to compute the PPV $\mathbf{v}_1(t)$ and the matrix $\mathbf{H}(t)$ defined in Definition 4.2.1. The computation of $\mathbf{v}_1(t)$ is well covered in the literature. In the next two sections, we develop the theory and numerical methods needed to compute $\mathbf{H}(t)$.

4.2.3 Theoretical Characterization of $\mathbf{H}$

In order to characterize $\mathbf{H}(t)$, we first derive a matrix differential equation that it satisfies, then we formulate a BVP based on this equation, and we analyze the properties of this BVP.

Matrix Differential Equation for $\mathbf{H}(t)$

One of the main results and novel contributions of the thesis is captured in the following theorem that derives an equation for $\mathbf{H}(t)$:

Theorem 4.2.2 $\mathbf{H}(t)$ *in (4.9) satisfies (with* $\Psi(t) = \mathbf{H}(t)$*)*

$$\frac{\mathrm{d}\Psi(t)}{\mathrm{d}t} + \Psi(t)\mathbf{G}(t) + \mathbf{G}^{\mathsf{T}}(t)\Psi(t) = -\mathbf{M}(t), \tag{4.22}$$

where the entries of the $(N \times N)$*-sized square matrix* $\mathbf{M}(t)$ *are given as in*

$$M_{il}(t) = \sum_{j=1}^{N} \frac{\partial G_{ji}(t)}{\partial x_{s,l}(t)} v_{1,j}(t). \tag{4.23}$$

Proof: We begin by computing

$$\frac{\partial}{\partial \mathbf{x}_{\mathrm{s}}(0)} \left[\frac{\mathrm{d}\mathbf{v}_1(t)}{\mathrm{d}t} = -\mathbf{G}^{\mathsf{T}}(t)\mathbf{v}_1(t) \right], \tag{4.24}$$

noting that $\mathbf{v}_1(t)$ satisfies the adjoint LPTV equation in (3.9). The left-hand side of (4.24), i.e. $\partial[\dot{\mathbf{v}}_1(t)]/\partial\mathbf{x}_{\mathrm{s}}(0)$, is

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{H}(t)\frac{\partial \mathbf{x}_{\mathrm{s}}(t)}{\partial \mathbf{x}_{\mathrm{s}}(0)}\right] = \left[\frac{\mathrm{d}\mathbf{H}(t)}{\mathrm{d}t} + \mathbf{H}(t)\mathbf{G}(t)\right]\frac{\partial \mathbf{x}_{\mathrm{s}}(t)}{\partial \mathbf{x}_{\mathrm{s}}(0)}, \tag{4.25}$$

where we have used (4.9) and noted the significance of Definition 3.2.1 for (3.5), because simply

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{\partial \mathbf{x}_{\mathrm{s}}(t)}{\partial \mathbf{x}_{\mathrm{s}}(0)}\right] = \mathbf{G}(t)\frac{\partial \mathbf{x}_{\mathrm{s}}(t)}{\partial \mathbf{x}_{\mathrm{s}}(0)}. \tag{4.26}$$

The right-hand side of (4.24), which is $\partial[-\mathbf{G}^{\mathsf{T}}(t)\mathbf{v}_1(t)]/\partial\mathbf{x}_{\mathrm{s}}(0)$, has two terms to be computed, i.e. $-\mathbf{G}^{\mathsf{T}}(t)\left[\partial\mathbf{v}_1(t)/\partial\mathbf{x}_{\mathrm{s}}(0)\right]$ and $-\left[\partial\mathbf{G}^{\mathsf{T}}(t)/\partial\mathbf{x}_{\mathrm{s}}(0)\right]\mathbf{v}_1(t)$. The first one is simply

$$-\mathbf{G}^{\mathsf{T}}(t)\frac{\partial \mathbf{v}_1(t)}{\partial \mathbf{x}_{\mathrm{s}}(0)} = -\mathbf{G}^{\mathsf{T}}(t)\mathbf{H}(t)\frac{\partial \mathbf{x}_{\mathrm{s}}(t)}{\partial \mathbf{x}_{\mathrm{s}}(0)}. \tag{4.27}$$

The second one can be written as a matrix $\mathbf{S}(t,0)$ whose entries are

$$\begin{aligned} S_{ik}(t,0) &= -\sum_{j=1}^{N} \frac{\partial G_{ji}(t)}{\partial x_{s,k}(0)} v_{1,j}(t) \\ &= -\sum_{j=1}^{N} \left[\sum_{l=1}^{N} \frac{\partial G_{ji}(t)}{\partial x_{s,l}(t)} \frac{\partial x_{s,l}(t)}{\partial x_{s,k}(0)} \right] v_{1,j}(t) \\ &= -\sum_{l=1}^{N} \left[\sum_{j=1}^{N} \frac{\partial G_{ji}(t)}{\partial x_{s,l}(t)} v_{1,j}(t) \right] \frac{\partial x_{s,l}(t)}{\partial x_{s,k}(0)} \qquad (4.28) \\ &= -\sum_{l=1}^{N} M_{il}(t) \frac{\partial x_{s,l}(t)}{\partial x_{s,k}(0)}, \qquad (4.29) \end{aligned}$$

where we have used (4.23) in the transition from (4.28) to (4.29). Therefore, simply

$$\mathbf{S}(t,0) = -\mathbf{M}(t)\frac{\partial \mathbf{x}_{\mathrm{s}}(t)}{\partial \mathbf{x}_{\mathrm{s}}(0)}. \qquad (4.30)$$

Observing that we have (4.25) for the left-hand side and the sum of (4.27) and (4.30) for the right-hand side of (4.24), (4.24) can be expressed as

$$\left[\frac{\mathrm{d}\mathbf{H}(t)}{\mathrm{d}t} + \mathbf{H}(t)\mathbf{G}(t) = -\mathbf{G}^{\mathsf{T}}(t)\mathbf{H}(t) - \mathbf{M}(t)\right]\frac{\partial \mathbf{x}_{\mathrm{s}}(t)}{\partial \mathbf{x}_{\mathrm{s}}(0)}. \qquad (4.31)$$

Since $\partial\mathbf{x}_{\mathrm{s}}(t)/\partial\mathbf{x}_{\mathrm{s}}(0)$ is nonsingular (see Definition 3.2.1), we conclude after simple manipulations that $\mathbf{H}(t)$ satisfies (4.22). ■

Equation (4.22) is an instance of the *continuous-time Lyapunov matrix differential equation.* The general form for its solution is given as follows [31] (a proof is given in Appendix Section C.1):

Lemma 4.2.1 *$\Psi(t)$, the general solution of (4.22), is expressed as*

$$\Psi(t) = \Upsilon^{\mathsf{T}}(t,0)\Psi(0)\Upsilon(0,t) - \int_0^t \Upsilon^{\mathsf{T}}(t,\tau)\mathbf{M}(\tau)\Upsilon(\tau,t)d\tau, \qquad (4.32)$$

where $\Upsilon^{\mathsf{T}}(s,t)$ and $\Upsilon(t,s)$ are the STMs given in (3.10) and (3.7), respectively.

Before we use (4.22) in order to formulate a BVP for $\mathbf{H}(t)$, we state two important and obvious properties of $\mathbf{H}(t)$:

- As it is defined on the periodic orbit $\mathbf{x}_{\mathrm{s}}(t)$, parameterized by time t, $\mathbf{H}(t)$ is T-periodic just like $\mathbf{u}_1(t)$, PPV $\mathbf{v}_1(t)$ and $\mathbf{x}_{\mathrm{s}}(t)$.

- $\mathbf{H}(t)$, as well as $\mathbf{M}(t)$ defined in (4.23), is symmetric.

In all, $\mathbf{H}(t)$ is a symmetric and T-periodic solution of (4.22).

Boundary Value Problem for $\mathbf{H}(0)$

We now formulate a periodic BVP based on (4.22), using a shooting formulation. Let us first define $\mathbf{H}_{\mathrm{g}}(t)$ to be a generic periodic solution for (4.22). Supposing $\mathbf{H}_{\mathrm{g}}(0)$ is known and (4.22) is solved from $t = 0$ backwards in time (due to the instability of the equation when solved forward in time), we are certain to obtain $\mathbf{H}_{\mathrm{g}}(-T) = \mathbf{H}_{\mathrm{g}}(0)$. Then, through the form in (4.32), we obtain after some manipulations

$$\begin{aligned} &\Upsilon^{\mathsf{T}}(-T,0)\mathbf{H}_{\mathrm{g}}(0)\Upsilon(0,-T) - \mathbf{H}_{\mathrm{g}}(0) \\ - &\int_{0}^{-T} \Upsilon^{\mathsf{T}}(-T,\tau)\mathbf{M}(\tau)\Upsilon(\tau,-T)d\tau = 0. \end{aligned} \tag{4.33}$$

Equation (4.33) is an instance of the *discrete Lyapunov matrix equation* [31]. If (4.33) can be solved for $\mathbf{H}_{\mathrm{g}}(0)$, it can be used as the correct initial condition to obtain a periodic solution for (4.22).

Solutions of the BVP

If (4.33) had a unique solution, it would give us $\mathbf{H}(0)$, and we would immediately be resorting to numerical methods for solving (4.33). However, (4.33) has infinitely many solutions as we demonstrate next. In analyzing the uniqueness/existence of the solutions of (4.33), the basis set that is established with the following lemma will be most useful:

Lemma 4.2.2 *Any* $N \times N$ *matrix* $\mathbf{K}$ *can be uniquely expressed as a linear combination of the set of rank-one matrices* $\{\mathbf{v}_r(t)\mathbf{v}_p^{\mathsf{T}}(t), r = 1 \ldots N, p = 1 \ldots N\}$ *for any* t *as follows*

$$\mathbf{K} = \sum_{r=1}^{N}\sum_{p=1}^{N} a_{rp}(t)\ \mathbf{v}_r(t)\mathbf{v}_p^{\mathsf{T}}(t), \tag{4.34}$$

with

$$a_{rp}(t) = \mathbf{u}_r^{\mathsf{T}}(t)\mathbf{K}\mathbf{u}_p(t). \tag{4.35}$$

In other words, the N^2 *rank-one matrices in the above set form a basis for* $N \times N$ *matrices.*

Proof: Substituting (4.35) into (4.34), we obtain

$$\mathbf{K} = \left[\sum_{r=1}^{N} \mathbf{v}_r(t)\mathbf{u}_r^{\mathsf{T}}(t)\right] \mathbf{K} \left[\sum_{p=1}^{N} \mathbf{u}_p(t)\mathbf{v}_p^{\mathsf{T}}(t)\right] = [\mathbf{I}_N]\,\mathbf{K}\,[\mathbf{I}_N] \tag{4.36}$$

due to the biorthonormality property in (3.8). ■

Now we are equipped with all the tools to state and prove a theorem about the nature of the solutions of (4.33):

Theorem 4.2.3 *The infinitely many solutions of the BVP in (4.33) are in the form*

$$\mathbf{H}_{\mathrm{g}}(0) = c_{11}\mathbf{v}_1(0)\mathbf{v}_1^{\intercal}(0) + \sum_{\substack{r,\ p\ not \\ both\ 1}} c_{rp}\mathbf{v}_r(0)\mathbf{v}_p^{\intercal}(0) \tag{4.37}$$

where c_{rp}, for r and p not both 1, are fixed, and c_{11} is an arbitrary scalar constant that marks the single degree of freedom.

Proof: By substituting (3.7), (3.10) and (4.37) into (4.33) and multiplying both sides with $\mathbf{u}_r^{\intercal}(0)$ from the left and with $\mathbf{u}_p(0)$ from the right, we obtain the following after some manipulations

$$\begin{aligned} &[\lambda_r\lambda_p - 1]\{c_{rp}\} \\ &= [\lambda_r\lambda_p]\left\{\int_0^{-T} \exp(\mu_r\tau)\mathbf{u}_r^{\intercal}(\tau)\mathbf{M}(\tau)\mathbf{u}_p(\tau)\exp(\mu_p\tau)d\tau\right\}. \end{aligned} \tag{4.38}$$

for $1 \leq r, p \leq N$ where $\lambda_i = \exp(\mu_i T)$. Note that (4.38) represents a set of N^2 equations. Let us focus on the equation obtained when both $r = 1$ and $p = 1$ in (4.38). Since $\lambda_1 = 1$ for $\mu_1 = 0$, the left-hand side of (4.38) is zero even if $c_{11} \neq 0$. Equation (4.33) will not have a solution unless the right-hand side of (4.38) is also equal to zero for $r = p = 1$. Using the identity in (4.39) of Theorem 4.2.4 (to be explained shortly), the right-hand side of (4.38), for $r = p = 1$, can be shown to be identically equal to zero as well. Thus, a solution exists for (4.38) and hence (4.33), but it is not unique. Equation (4.38) is trivially satisfied for $r = p = 1$ and does not impose a constraint on c_{11} while all of the other coefficients c_{rp} in the decomposition in (4.37) are fully determined by the other $(N^2 - 1)$ equations in (4.38). The multiple solutions of (4.33) are marked by a single degree of freedom, c_{11} in (4.37). ■

The BVP equation in (4.33) does not determine the coefficient of the $\mathbf{v}_1(0)\mathbf{v}_1^{\intercal}(0)$ component of $\mathbf{H}(0)$ in a decomposition using the basis given in Lemma 4.2.2, so (4.33) is not sufficient by itself to determine and compute $\mathbf{H}(0)$. The coefficient of the $\mathbf{v}_1(0)\mathbf{v}_1^{\intercal}(0)$ component of $\mathbf{H}(0)$ is determined based on a *normalization condition* given by the following theorem:

Theorem 4.2.4 *It follows from the normalization condition (for the PPV $\mathbf{v}_1(t)$) in (3.13) that*

$$\mathbf{H}(t)\mathbf{u}_1(t) = -\mathbf{G}^{\mathsf{T}}(t)\mathbf{v}_1(t) \tag{4.39}$$

and

$$\mathbf{u}_1^{\mathsf{T}}(t)\mathbf{H}(t)\mathbf{u}_1(t) = -\mathbf{u}_1^{\mathsf{T}}(t)\mathbf{G}^{\mathsf{T}}(t)\mathbf{v}_1(t). \tag{4.40}$$

Proof: (4.39) is obtained simply through

$$\mathbf{H}(t)\mathbf{u}_1(t) = \frac{\partial \mathbf{v}_1(t)}{\partial \mathbf{x}_{\mathrm{s}}(t)} \frac{\mathrm{d}\mathbf{x}_{\mathrm{s}}(t)}{\mathrm{d}t} = \frac{\mathrm{d}\mathbf{v}_1(t)}{\mathrm{d}t} \tag{4.41}$$

noting that $\mathbf{u}_1(t) = \dot{\mathbf{x}}_{\mathrm{s}}(t)$ and $\mathrm{d}\mathbf{v}_1(t)/\mathrm{d}t = -\mathbf{G}^{\mathsf{T}}(t)\mathbf{v}_1(t)$. ■

With (4.40) and based on Lemma 4.2.2, we can determine the coefficient of the $\mathbf{v}_1(0)\mathbf{v}_1^{\mathsf{T}}(0)$ component as follows:

$$c_{11} = -\mathbf{u}_1^{\mathsf{T}}(0)\mathbf{G}^{\mathsf{T}}(0)\mathbf{v}_1(0). \tag{4.42}$$

In Section 4.2.4 below, we will combine the BVP equation in (4.33) and the *normalization condition* in (4.40) to obtain an *augmented* BVP formulation that will enable us to develop a rigorous numerical method for computing $\mathbf{H}(t)$.

4.2.4 Numerical Method for Computing $\mathbf{H}$

In the numerical scheme to be described, it is taken for granted that the periodic solution $\mathbf{x}_{\mathrm{s}}(t)$ along with $\mathbf{u}_1(t) = \dot{\mathbf{x}}_{\mathrm{s}}(t)$ and the PPV $\mathbf{v}_1(t)$ have been computed through either the shooting method [5] or harmonic balance [12]. We describe here a numerical method for computing $\mathbf{H}(t)$.

It is possible to augment the BVP formulation in (4.33) in order to embed the information in (4.40) so that $\mathbf{H}(0)$ out of the infinitely many solutions of (4.33) can be "hand-picked", followed with the computation of $\mathbf{H}(t)$ for a single period. However, we devise a different scheme here which has better numerical properties. We first define $\mathbf{H}_{\mathrm{d}}(t)$, i.e., the deflated $\mathbf{H}(t)$:

Definition 4.2.2 *The deflated form of* $\mathbf{H}(t)$ *is*

$$\mathbf{H}_{\mathrm{d}}(t) = \mathbf{H}(t) - \left[\mathbf{u}_1^{\mathsf{T}}(t)\mathbf{H}(t)\mathbf{u}_1(t)\right] \mathbf{v}_1(t)\mathbf{v}_1^{\mathsf{T}}(t). \tag{4.43}$$

$\mathbf{H}_{\mathrm{d}}(t)$ *is essentially* $\mathbf{H}(t)$ *with the* $\mathbf{v}_1(t)\mathbf{v}_1^{\mathsf{T}}(t)$ *component extracted.*

Our numerical scheme proceeds as follows:

- Construct an augmented BVP for $\mathbf{H}_{\mathrm{d}}(0)$.
- With $\mathbf{H}_{\mathrm{d}}(0)$, compute $\mathbf{H}_{\mathrm{d}}(t)$ for an entire period.
- Add the missing component $[\mathbf{u}_1^{\mathsf{T}}(t)\mathbf{H}(t)\mathbf{u}_1(t)]\ \mathbf{v}_1(t)\mathbf{v}_1^{\mathsf{T}}(t)$ to $\mathbf{H}_{\mathrm{d}}(t)$ to obtain $\mathbf{H}(t)$.

We explain each of the steps above in sequence:

Augmented BVP for Computing $\mathbf{H}_{\mathrm{d}}(0)$

Definition 4.2.3 *The integral in (4.32) is denoted by*

$$\mathbf{INTG}(t,0) = -\int_0^t \Upsilon^{\mathsf{T}}(t,\tau)\mathbf{M}(\tau)\Upsilon(\tau,t)d\tau. \tag{4.44}$$

The computation of $\mathbf{INTG}(t,0)$ is accomplished by numerically solving the differential equation in (4.22) with a zero initial condition.

Lemma 4.2.3 *Omitting the t dependence for all terms, with (4.43) and (4.39), we obtain*

$$\mathbf{H}_{\mathrm{d}}\mathbf{u}_1 = (\mathbf{I}_N - \mathbf{v}_1\mathbf{u}_1^{\mathsf{T}})\left[-\mathbf{G}^{\mathsf{T}}\mathbf{v}_1\right]. \tag{4.45}$$

It follows that

$$\mathbf{u}_1^{\mathsf{T}}\mathbf{H}_{\mathrm{d}}\mathbf{u}_1 = 0. \tag{4.46}$$

Proof:

$$\begin{aligned}\mathbf{H}_{\mathrm{d}}\mathbf{u}_1 &= (\mathbf{H} - \mathbf{v}_1\mathbf{u}_1^{\mathsf{T}}\mathbf{H}\mathbf{u}_1\mathbf{v}_1^{\mathsf{T}})\,\mathbf{u}_1 \\ &= (\mathbf{I}_N - \mathbf{v}_1\mathbf{u}_1^{\mathsf{T}})\,[\mathbf{H}\mathbf{u}_1] \qquad (4.47)\\ &= (\mathbf{I}_N - \mathbf{v}_1\mathbf{u}_1^{\mathsf{T}})\left[-\mathbf{G}^{\mathsf{T}}\mathbf{v}_1\right], \qquad (4.48)\end{aligned}$$

where the identity $\mathbf{H}\mathbf{u}_1 = -\mathbf{G}^{\mathsf{T}}\mathbf{v}_1$ of (4.39) is used to obtain (4.48) from (4.47). (4.46) is true because in $\mathbf{H}_{\mathrm{d}}$, the $\mathbf{v}_1\mathbf{v}_1^{\mathsf{T}}$ component is missing. ■

The following theorem statement provides all that is necessary to implement the augmented BVP scheme for computing $\mathbf{H}_{\mathrm{d}}(0)$.

Theorem 4.2.5 *There exist certain nontrivial constants $a \neq \pm 1$, b, c, and d such that $\mathbf{H}'$, as given by*

$$\mathbf{H}' = \begin{bmatrix} \mathbf{H}_{\mathrm{d}}(0) & \mathbf{v}_1(0) \\ \mathbf{v}_1^{\intercal}(0) & b \end{bmatrix}, \tag{4.49}$$

is the unique solution of the matrix equation

$$\mathbf{A}\mathbf{H}'\mathbf{A}^{\intercal} - \mathbf{E}\mathbf{H}'\mathbf{E}^{\intercal} + \mathbf{Q} = 0, \tag{4.50}$$

which is an instance of the generalized discrete Lyapunov matrix algebraic equation *[32] with*

$$\mathbf{A} = \begin{bmatrix} \Upsilon^{\intercal}(-T,0) & 0 \\ \mathbf{u}_1^{\intercal}(0) & 0 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \mathbf{I}_N & d\,\mathbf{v}_1(0) \\ a\,\mathbf{u}_1^{\intercal}(0) & c \end{bmatrix}, \tag{4.51}$$

and

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}, \tag{4.52}$$

where

$$\begin{aligned}
\mathbf{Q}_{11} &= \mathbf{INTG}(-T,0) + (2d + bd^2)\mathbf{v}_1(0)\mathbf{v}_1^{\intercal}(0) && (4.53)\\
\mathbf{Q}_{12} &= a\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0) + (ad + c + bcd)\mathbf{v}_1(0) && \\
&- \Upsilon^{\intercal}(-T,0)\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0) && (4.54)\\
\mathbf{Q}_{21} &= [\mathbf{Q}_{12}]^{\intercal} && (4.55)\\
\mathbf{Q}_{22} &= 2ac + bc^2 && (4.56)
\end{aligned}$$

noting that in (4.54), the identity in (4.45) can be used to compute $\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0)$.

Proof: We compute $\mathbf{A}\mathbf{H}'\mathbf{A}^{\intercal}$ and $\mathbf{E}\mathbf{H}'\mathbf{E}^{\intercal}$. Blockwise we have

$$\mathbf{A}\mathbf{H}'\mathbf{A}^{\intercal} = \begin{bmatrix} (\mathbf{A}\mathbf{H}'\mathbf{A}^{\intercal})_{11} & (\mathbf{A}\mathbf{H}'\mathbf{A}^{\intercal})_{12} \\ (\mathbf{A}\mathbf{H}'\mathbf{A}^{\intercal})_{21} & (\mathbf{A}\mathbf{H}'\mathbf{A}^{\intercal})_{22} \end{bmatrix} \tag{4.57}$$

and

$$\mathbf{E}\mathbf{H}'\mathbf{E}^{\intercal} = \begin{bmatrix} (\mathbf{E}\mathbf{H}'\mathbf{E}^{\intercal})_{11} & (\mathbf{E}\mathbf{H}'\mathbf{E}^{\intercal})_{12} \\ (\mathbf{E}\mathbf{H}'\mathbf{E}^{\intercal})_{21} & (\mathbf{E}\mathbf{H}'\mathbf{E}^{\intercal})_{22} \end{bmatrix}, \tag{4.58}$$

where

$$\begin{aligned}
\left(\mathbf{AH'A}^{\mathsf{T}}\right)_{11} &= \Upsilon^{\mathsf{T}}(-T,0)\mathbf{H}_{\mathrm{d}}(0)\Upsilon(0,-T) && (4.59)\\
\left(\mathbf{AH'A}^{\mathsf{T}}\right)_{12} &= \Upsilon^{\mathsf{T}}(-T,0)\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0) && (4.60)\\
\left(\mathbf{AH'A}^{\mathsf{T}}\right)_{21} &= \left[\left(\mathbf{AH'A}^{\mathsf{T}}\right)_{12}\right]^{\mathsf{T}} && (4.61)\\
\left(\mathbf{AH'A}^{\mathsf{T}}\right)_{22} &= \mathbf{u}_1^{\mathsf{T}}(0)\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0) && (4.62)
\end{aligned}$$

and

$$\begin{aligned}
\left(\mathbf{EH'E}^{\mathsf{T}}\right)_{11} &= \mathbf{H}_{\mathrm{d}}(0) + (2d+bd^2)\mathbf{v}_1(0)\mathbf{v}_1^{\mathsf{T}}(0) && (4.63)\\
\left(\mathbf{EH'E}^{\mathsf{T}}\right)_{12} &= a\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0) + (ad+c+bcd)\mathbf{v}_1(0) && (4.64)\\
\left(\mathbf{EH'E}^{\mathsf{T}}\right)_{21} &= \left[\left(\mathbf{EH'E}^{\mathsf{T}}\right)_{12}\right]^{\mathsf{T}} && (4.65)\\
\left(\mathbf{EH'E}^{\mathsf{T}}\right)_{22} &= a^2\mathbf{u}_1^{\mathsf{T}}(0)\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0) + 2ac + bc^2 && (4.66)
\end{aligned}$$

respectively.

In view of (4.59)-(4.62), (4.63)-(4.66), and (4.53)-(4.56), (4.50) is expressed explicitly as

$$\begin{aligned}
&\mathbf{AH'A}^{\mathsf{T}} - \mathbf{EH'E}^{\mathsf{T}} + \mathbf{Q}\\
&= \begin{bmatrix} (\mathbf{LHS})_{11} & (\mathbf{LHS})_{12} \\ (\mathbf{LHS})_{21} & (\mathbf{LHS})_{22} \end{bmatrix} = 0 && (4.67)
\end{aligned}$$

where

$$\begin{aligned}
(\mathbf{LHS})_{11} &= \Upsilon^{\mathsf{T}}(-T,0)\mathbf{H}_{\mathrm{d}}(0)\Upsilon(0,-T)\\
&\quad - \mathbf{H}_{\mathrm{d}}(0) + \mathbf{INTG}(-T,0) && (4.68)\\
(\mathbf{LHS})_{12} &= 0 && (4.69)\\
(\mathbf{LHS})_{21} &= 0 && (4.70)\\
(\mathbf{LHS})_{22} &= (a^2-1)\mathbf{u}_1^{\mathsf{T}}(0)\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0) && (4.71)
\end{aligned}$$

Setting $(\mathbf{LHS})_{11} = 0$ in (4.68) complies with $\mathbf{H}_{\mathrm{d}}(0)$ being a solution of the original BVP. Setting $(\mathbf{LHS})_{22} = 0$ in (4.71), with $a \neq \pm 1$ ensures that $\mathbf{u}_1^{\mathsf{T}}(0)\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0) = 0$, as it should be. ■

The augmented BVP formulation given above essentially enforces $\mathbf{u}_1^{\mathsf{T}}(0)\mathbf{H}_{\mathrm{d}}(0)\mathbf{u}_1(0) = 0$ as given by (4.46) of Lemma 4.2.3, ensuring the removal of the $\mathbf{v}_1(0)\mathbf{v}_1^{\mathsf{T}}(0)$ component to obtain the deflated $\mathbf{H}_{\mathrm{d}}(0)$.

Computing $\mathbf{H}_{\mathrm{d}}(t)$ *for One Period*

In order to compute $\mathbf{H}_{\mathrm{d}}(t)$ with the initial condition $\mathbf{H}_{\mathrm{d}}(0)$ known, we need a modified differential equation for $\mathbf{H}_{\mathrm{d}}(t)$, which does not satisfy (4.22):

Theorem 4.2.6 *The deflated* $\mathbf{H}_{\mathrm{d}}(t)$ *satisfies*

$$\begin{aligned} & \frac{\mathrm{d}\mathbf{H}_{\mathrm{d}}(t)}{\mathrm{d}t} + \mathbf{H}_{\mathrm{d}}(t)\mathbf{G}(t) + \mathbf{G}^{\intercal}(t)\mathbf{H}_{\mathrm{d}}(t) \\ = & -\left[\mathbf{M}(t) - \left[\mathbf{u}_1^{\intercal}(t)\mathbf{M}(t)\mathbf{u}_1(t)\right]\mathbf{v}_1(t)\mathbf{v}_1^{\intercal}(t)\right]. \end{aligned} \tag{4.72}$$

Adding the missing component to obtain $\mathbf{H}(t)$

We compute $\mathbf{H}(t)$ from $\mathbf{H}_{\mathrm{d}}(t)$ using

$$\mathbf{H}(t) = \mathbf{H}_{\mathrm{d}}(t) - \left[\mathbf{u}_1^{\intercal}(t)\mathbf{G}^{\intercal}(t)\mathbf{v}_1(t)\right]\mathbf{v}_1(t)\mathbf{v}_1^{\intercal}(t) \tag{4.73}$$

based on (4.40) and (4.43).

Implementation Notes

$\mathbf{x}_{\mathrm{s}}(t)$, $\mathbf{u}_1(t)$, $\mathbf{v}_1(t)$ are computed using either the shooting or harmonic balance method. $\mathbf{M}(t)$ in (4.23) is computed with the symbolically available second-order derivatives of $\mathbf{f}(\mathbf{x})$ with respect to $\mathbf{x}$, through the use of Matlab's symbolic toolbox. For the solution of the augmented BVP equation (Section 4.2.4), $\mathbf{INTG}(-T, 0)$ in (4.44) and $\Upsilon^{\intercal}(-T, 0)$ are computed by integrating (backwards in time, using trapezoidal discretization) (4.22) (with zero initial condition) and (3.9) (with the identity matrix as the initial condition) respectively. In order to compute $\mathbf{INTG}(-T, 0)$ and solve (4.50) [31, 4], we use variants of the Bartels-Stewart algorithm [32, 33] as implemented in the control systems toolbox of Matlab that are based on SLICOT [34]. $\mathbf{H}_{\mathrm{d}}(0)$, as yielded by the augmented BVP, makes the initial condition for (4.72) to compute $\mathbf{H}_{\mathrm{d}}(t)$ for one period (again through backward integration and the trapezoidal scheme). $\mathbf{H}(t)$ is finally computed for one period through (4.73). The computational cost incurred is $\mathcal{O}\big(K\,N^3\big)$ where N is the number of state variables and K is the number of time steps used per period for discretization. The cubic complexity (due to dense numerical linear algebra) is not as overwhelming as the exponential one incurred by the brute-force method for computing isochrons. Quadratic isochron approximations for oscillators with hundreds of state variables can be computed with reasonable resources in this manner.

4.3 Results for Local Isocron Approximation Computations

We first verify the correctness of the theory and the numerical methods we have presented for quadratic isochron approximations and our implementation by comparing the numerical results we obtain for a simple planar oscillator with analytical isochron characterizations. We then present results for the Van der Pol oscillator and more complex biological oscillators for which analytical calculations are not possible.

4.3.1 Simple Analytical Oscillator

This simple planar oscillator is described in polar coordinates with

$$\begin{aligned} \dot{r} &= 1 - r \\ \dot{\theta} &= r \end{aligned} \tag{4.74}$$

The only limit cycle (stable periodic solution) of (4.74) is described by $r = 1$ (the unit circle), with the initial condition $(r_0 = 1, \theta_0 = 0)$, corresponding to $\mathbf{x}_{\mathrm{s}}(t) = (\cos(t), \sin(t))$ in Cartesian coordinates. It can be shown that the system in (4.74) has isochrons described by [2]

$$\theta + r - 1 = t_c, \quad \text{with } t_c \in [0, 2\pi), \tag{4.75}$$

where an isochron corresponding to a particular t_c passes through $\mathbf{x}_{\mathrm{s}}(t_c)$ on the limit cycle (see Appendix Section A.1 for the justification). A number of these isochrons are shown in Figure 4.2 along with the limit cycle.

For this simple planar oscillator, $\mathbf{u}_1(t)$, the PPV $\mathbf{v}_1(t)$, $\mathbf{INTG}(t, 0)$ in (4.44), the STM $\Upsilon(t, s)$ and the matrix $\mathbf{H}(t)$ can all be analytically calculated, and thus this oscillator also provides an excellent problem for testing the implementation. With $\mathbf{v}_1(t)$ and $\mathbf{H}(t)$, linear and quadratic approximations for the isochrons can be computed using (4.21) and (4.10), as shown in Figure 4.3 along with the exact isochrons in Figure 4.2 based on (4.75). The quadratic approximation follows the exact isochron much more closely compared with the linear approximation, especially when further away from the limit cycle.

For this case, $\mathbf{x}_{\mathrm{s}}(t)$, $\mathbf{v}_1(t)$, and $\mathbf{H}(t)$ are analytically calculable and purely sinusoidal quantitites (which makes harmonic balance [5, 12] the most suitable method for the numerical computations of $\mathbf{x}_{\mathrm{s}}(t)$ and then $\mathbf{v}_1(t)$). Since the trapezodidal discretization scheme (a second-order scheme) is used to compute $\Upsilon^{\intercal}(-T, 0)$, $\mathbf{INTG}(-T, 0)$, and $\mathbf{H}(-T)$ as described

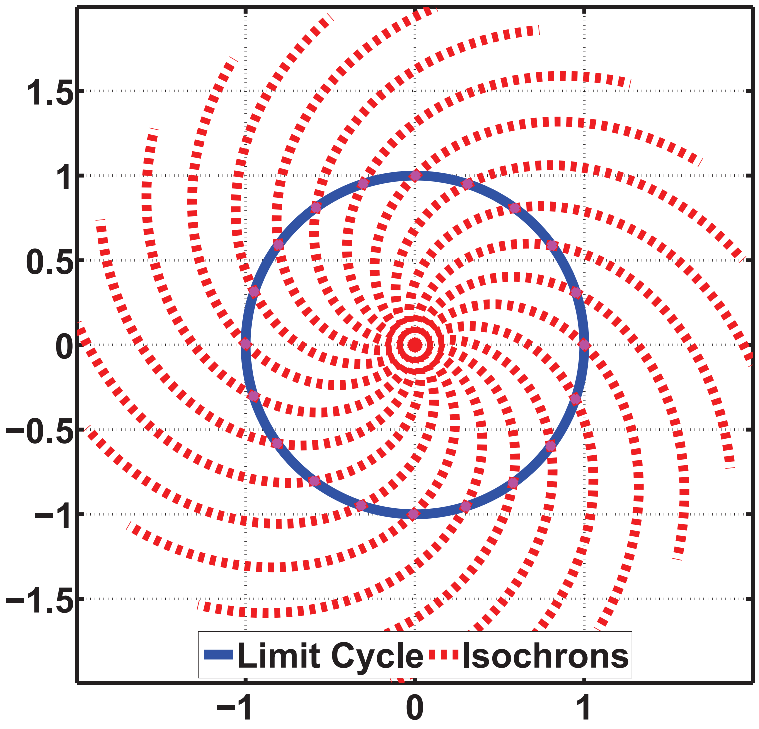


Figure 4.2: Limit cycle and isochron portrait for the simple analytical oscillator.

in Section 4.2.4, we expect the error due to discretization to decrease quadratically as the time step is reduced. A plot of the relative 2-norm error between the exact and computed $\mathbf{H}(-T)$ as a function of the time step used for discretization (uniform in this case) is shown in Figure 4.4. In this plot, an error curve that shows ideal quadratic decrease as a function of time step is also included, which completely coincides with the error curve obtained by our numerical scheme. These results, comparing the numerically computed $\mathbf{H}(t)$ against the analytically calculated $\mathbf{H}(t)$ for the simple oscillator, bear witness to the correctness of our theory and the numerical methods for computing $\mathbf{H}(t)$.

4.3.2 Van der Pol Oscillator

The particular version of the Van der Pol oscillator we consider is described as follows in Cartesian coordinates

$$\begin{aligned}\dot{x} &= y \\ \dot{y} &= -x + y - y\,x^2.\end{aligned} \tag{4.76}$$

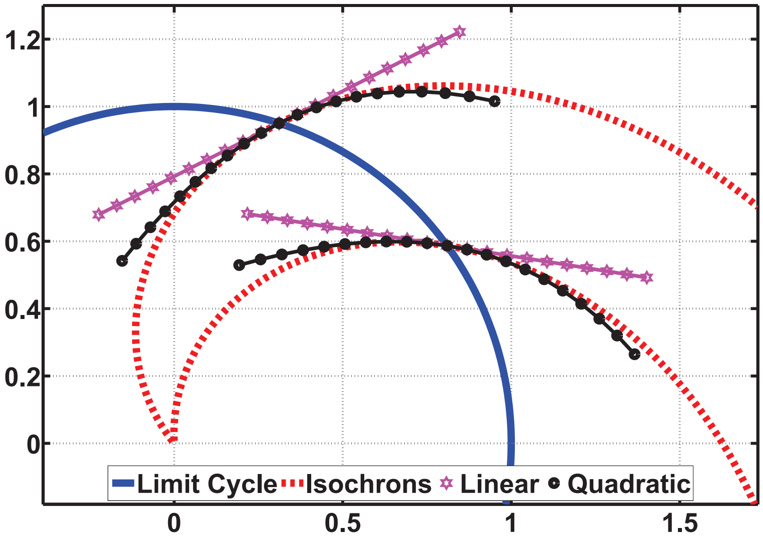


Figure 4.3: Isochrons and their approximations for the simple analytical oscillator.

The periodic solution $\mathbf{x}_s(t)$, the PPV $\mathbf{v}_1(t)$, or the matrix $\mathbf{H}(t)$ is not analytically calculable in this case. We rely on the well-established numerical methods for computing $\mathbf{x}_s(t)$ and $\mathbf{v}_1(t)$, and our proposed numerical method for $\mathbf{H}(t)$. However, since (4.76) is a planar oscillator, a brute-force numerical scheme can alternatively be used to compute its complete isochron portrait. Apart from this brute-force scheme explained below (also see Chapter 1 for an intuitive account), see Appendix Section A.2 for an alternative scheme that can yield partial information about the local behavior of the exact isochrons of an oscillator in virtually any dimension.

It can be shown that if two points $\mathbf{x}_1(0)$ and $\mathbf{x}_2(0)$, belonging to two different solutions $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ of an oscillating system, are initially on the same isochron, then at any time $t = t_c$, $\mathbf{x}_1(t_c)$ and $\mathbf{x}_2(t_c)$ will also be on the same isochron. This is also true for $t_c < 0$. Since γ is an attracting limit cycle (when oscillator equations are integrated forward in time) in view of our assumptions, γ becomes a "repelling" limit cycle when oscillator equations are integrated backward in time. This means that when $\mathbf{x}_1(0)$ and $\mathbf{x}_2(0)$ are chosen to be very close to some point $\mathbf{x}_s(t_p)$ on γ (so that it may be assumed $\mathbf{x}_1(0)$ and $\mathbf{x}_2(0)$ are on the same isochron as $\mathbf{x}_s(t_p)$), at $t = t_c < 0$, $\mathbf{x}_1(t_c)$ and $\mathbf{x}_2(t_c)$ will be further away from each other but be on the same isochron as $\mathbf{x}_s(t_p - t_c)$. Adaptively inserting new points between $\mathbf{x}_1(t_c)$ and $\mathbf{x}_2(t_c)$ and integrating the oscillator equations backward in time,

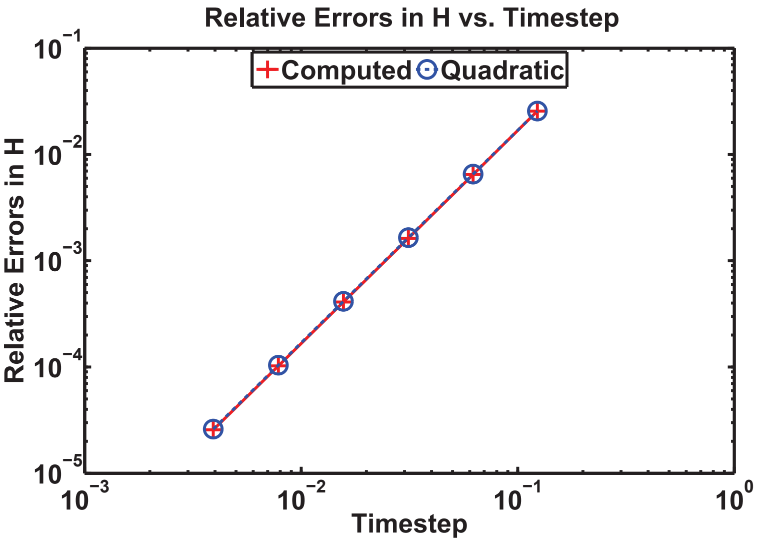


Figure 4.4: Relative error between numerically computed and analytical $\mathbf{H}(-T)$.

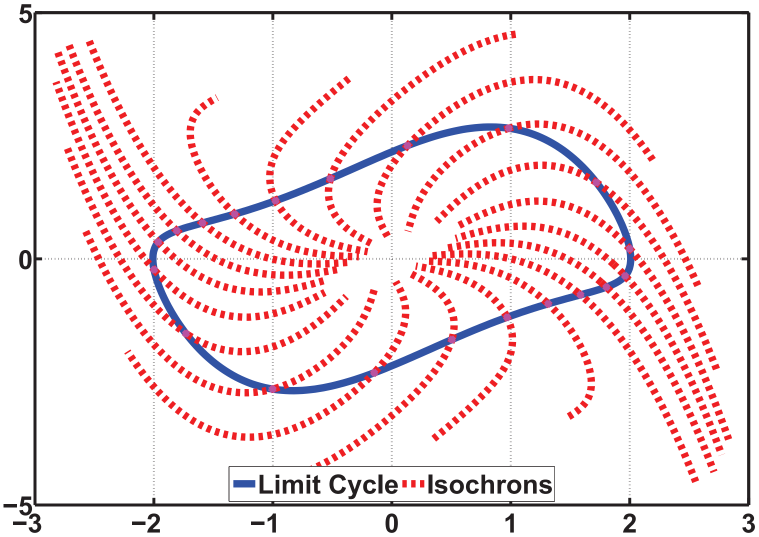


Figure 4.5: Limit cycle and isochron portrait for the Van der Pol oscillator.

it is possible to obtain a somewhat reasonably accurate depiction for the isochron passing through $\mathbf{x}_s(t_p - T) = \mathbf{x}_s(t_p)$ when $t = -T$ is reached. Izhikevich provides a compact Matlab code for this technique in [1], which works reasonably well although it uses forward Euler in backward integration. We have improved this code to utilize the trapezoidal scheme and to provide more detailed and explanatory illustrations. Our experience shows that utmost care must be taken in any case when using Izhikevich's method even for very simple planar

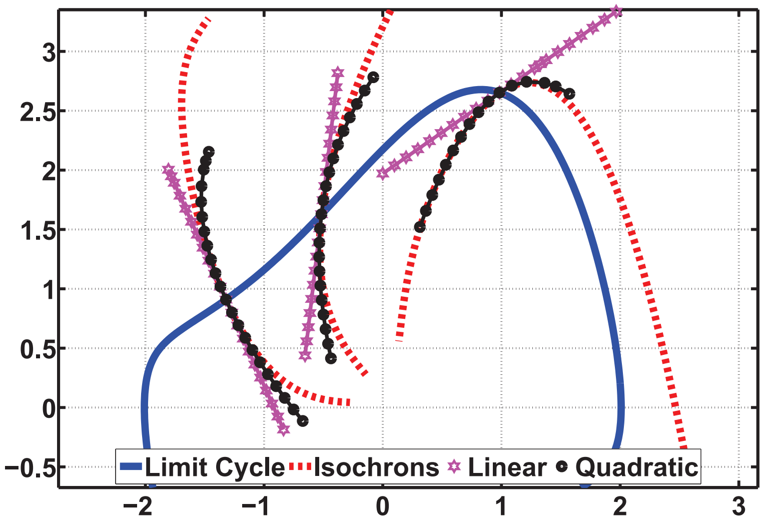


Figure 4.6: Isochrons and their approximations for the Van der Pol oscillator.

oscillators, since backward integration around attracting limit cycles of oscillatory, especially stiff, systems is by nature unstable. Figure 4.5 shows the isochron portrait for the Van der Pol oscillator of (4.76) computed with this technique. In Figure 4.6, several isochrons along with their corresponding linear and quadratic approximations computed using the techniques described in this thesis are shown. As we observe in this figure, the quadratic approximation (computed using the techniques proposed in this thesis) fits the real isochron (computed through the method of Izhikevich [1] described above) extraordinarily well for one isochron, and in general, quadratic approximations are much more accurate than the linear ones, especially when further away from the limit cycle.

4.3.3 Mammalian Circadian Oscillator with 7 State Variables

Biological oscillators (based on genetic networks) that are responsible for setting up the circadian (24-hour) rhythms are crucial for the survival of many species, and there are many health problems caused by the disturbance of these clocks in humans. We now apply the techniques proposed in the thesis to a model of the mammalian circadian clock described in [35]. Since this oscillator model has seven state variables, it is not possible to visualize its isochrons in a graph. Instead, we choose a different method to demonstrate the properties of an isochron of this oscillator, after computing $\mathbf{x}_s(t)$, $\mathbf{v}_1(t)$, and $\mathbf{H}(t)$ using the schemes described in the thesis. In Figure 4.7, the (normalized) distance between the linear and

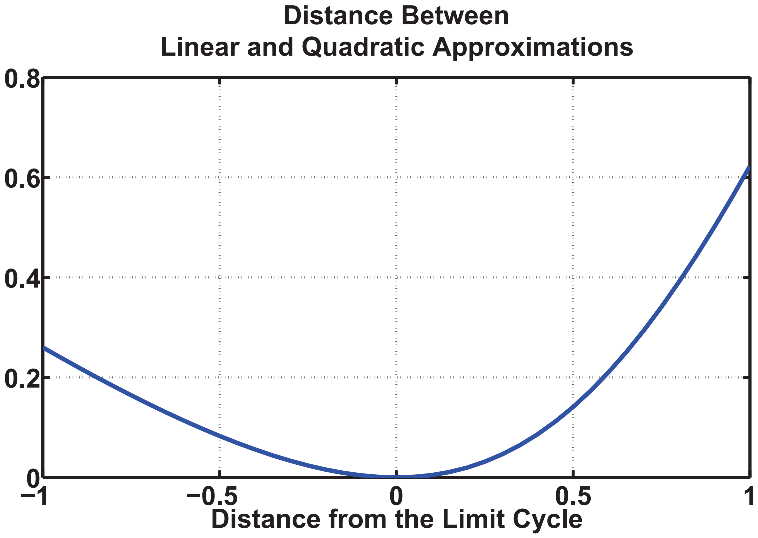


Figure 4.7: Distance between the linear and quadratic isochron approximation.

quadratic approximations for a particular isochron is shown as one moves away from the limit cycle in a certain direction in the seven-dimensional space. We deduce that the quadratic approximation will be much better in approximating the exact isochron. There is no other method which can give us a better approximation than the quadratic approximation we are able to compute. The brute-force method [1] is totally out of the question.

4.3.4 Drosophila Circadian Oscillator with 25 State Variables

Finally, we present results obtained on a model of the Drosophila circadian rhythm described in [36] in order to indicate the computational resources required with our current implementation of the numerical methods based on dense numerical linear algebra. The Drosophila circadian model in [36] has only 15 state variables. However, this model contains a pure time delay operator. By modeling this delay operator with a 10th order Pade approximation, we were able to generate an oscillator model with 25 state variables. We computed the matrix $\mathbf{H}(t)$ for this oscillator in 5 seconds of CPU time on a PC running Linux with a 2.6 GHz Opteron processor. The computation was done with an implementation of our numerical methods in Matlab using dense numerical linear algebra. The time discretization was done using 1000 points. Recalling that the computational complexity for this implementation is $\mathcal{O}(K\,N^3)$, where K is the number of timepoints and N is the number of state variables,

$\mathbf{H}(t)$ for an oscillator with 200 state variables can be computed in under one hour with 1000 time points. We should note here that it is difficult to come by biological oscillator models with more than tens of state variables. In fact, we were not able to find a biological oscillator model with more than 20-30 state variables in the model databases available on the Internet. As such, we believe that an implementation of the numerical schemes proposed in this thesis based on dense numerical linear algebra can be used to compute quadratic isochron approximations for all practical biological oscillator models.

4.4 Summary

In this chapter, we reviewed the isochron concept for autononous oscillators, stated and proved some of the crucial properties of the sets of in-phase points called isochrons. Comments were made on the difficulty entailed in the exact numerical calculation of isochrons. The need to resort to local approximations was then pointed out. After stating that the linear approximation is indeed a hyperplane determined by the point on the limit cycle, which the exact isochron crosses, and the vector called the PPV (or infinitesimal PRCs or ISFs stacked into a vector). We then proposed that quadratic, instead of linear, approximations should locally provide a higher level of accuracy. It was shown that another quantity, i.e. a matrix function $\mathbf{H}(t)$, is necessary to characterize the proposed quadratic approximations for isochrons. Then, a theoretical characterization of $\mathbf{H}(t)$ was accomplished, and a reliable method for its numerical computation was designed. Several tests on a simple polar oscillator (with analytical expressions for any attribute available) and other intricate oscillators were run, and the numerical method was shown to be robust.

Chapter 5

PHASE OF AN OSCILLATOR

We now define and discuss a generalized notion of *instantaneous phase*, uniformly applicable to oscillators that are (1) autonomous and in periodic steady-state, i.e., tracing a limit cycle, (2) autonomous but not in steady-state, i.e., not have settled to a limit cycle yet. In this chapter, we concentrate on the generalized instantaneous phase definition for oscillators that are autonomous, i.e., that are not experiencing any perturbations. We later, in Chapter 6, extend the phase definition to non-autonomous oscillators, i.e., ones that are experiencing perturbations, hence not in steady-state, but that stay in the domain of attraction of a particular limit cycle. The phase definition we discuss shortly is based on the notion of isochrons described in Chapter 4 and the key periodic behavior property captured by Theorem 4.1.1.

The outline of this chapter is as follows. In Section 5.1, we formulate the generalized period definition by the use of the isochron concept. The instantaneous oscillator phase definition is given in Section 5.2. Isochrons are established as the level sets of instantaneous phase in Section 5.3. The oscillator phase computation problem is rigorously formulated in Section 5.4. Chapter 6 describes numerical methods for computationally solving this problem. Since approximate methods for oscillator phase computation are the ones that are practically usable as we reveal in again Chapter 6, local approximations of isochrons have to be incorporated into these methods via the functions $\mathbf{v}_1(t)$ and $\mathbf{H}(t)$ (both explained in Chapter 4). However, first we have to reveal the connection between the instantaneous oscillator phase and these two functions. This is done in Section 5.5.

Most of the insight and intuitive explanations in Sections 5.1 through 5.4 are inspired by the work in [14].

5.1 Generalized Period

Prior to the definition of phase, we first introduce, as the formulation in [14] suggests, a *generalized periodicity* notion applicable to the solutions of autonomous (unperturbed) oscillators which are not necessarily periodic, i.e., which have not yet settled to a periodic limit cycle. The simple periodicity notion of Section 3.1 only applies to autonomous oscillators that have already settled to a limit cycle, i.e., that are in periodic steady-state.

Definition 5.1.1 (Generalized Period) *Let $\mathbf{x}(t)$ be a solution of (3.1) such that the initial condition $\mathbf{x}(0) \in \eta_0$ but not on the limit cycle, i.e., $\mathbf{x}(0) \neq \mathbf{x}_s(0)$. Hence, $\mathbf{x}(t)$ is not periodic. With the set $\mathcal{T}$ as*

$$\mathcal{T} = \left\{ \mathcal{T}_i \,\middle|\, \Phi(\mathcal{T}_i, 0, \mathbf{x}(0)) \in \eta_0 \right\}, \tag{5.1}$$

the generalized period *of $\mathbf{x}(t)$ is defined as the smallest positive number in $\mathcal{T}$.*

Based on Theorem 4.1.1, the generalized period defined above is equal to T, the “familiar” period on the limit cycle γ, for any solution of the autonomous oscillator equations in (3.1) with an initial condition in the domain of attraction, i.e., $\mathbf{x}(0) \in \mathcal{W}$.

5.2 Generalized Instantaneous Phase

We now precisely and formally, following the formulation in [14], define a generalized instantaneous phase, denoted by $\hat{t}$, that has units of time, for every point $\mathbf{x} \in \mathcal{W}$ in the domain of attraction of a limit cycle γ as well as at all of the points on the limit cycle itself:

Definition 5.2.1 (Phase in Units of Time) *$\hat{t}(\mathbf{x}) : \Re^N \rightarrow \Re$ is defined at a particular point $\mathbf{x} \in \mathcal{W}$ as follows:*

1. *If $\mathbf{x} \in \gamma$ (on the limit cycle), i.e., $\mathbf{x} = \mathbf{x}_\mathrm{s}(t^*)$:*

$$\hat{t}(\mathbf{x}) = \hat{t}(\mathbf{x}_\mathrm{s}(t^*)) = t^*. \tag{5.2}$$

2. *If $\mathbf{x} \in \mathcal{W} - \gamma$ (in the domain of attraction but not on the limit cycle), the phase $\hat{t}(\mathbf{x})$ is again computed through (5.2), but we maintain in this case that $\mathbf{x} \in \eta_{t^*}$ and $\mathbf{x}_\mathrm{s}(t^*) \in \eta_{t^*}$.*

With the above definition, the phase of a point on the limit cycle is simply defined to be the time tag t of that point in the periodic steady-state solution $\mathbf{x}_{\mathrm{s}}(t)$. For all other points $\mathbf{x}$ in the domain of attraction but not on the limit cycle, the phase is defined to be the time tag of another point $\mathbf{x}_{\mathrm{s}}(t^*)$ that (a) resides exactly on the limit cycle, and, (b) is on the same isochron as the point $\mathbf{x}$.

The fact that the phase $\hat{t}$ is defined above as a function of solely points $\mathbf{x}$ in the state-space, i.e., $\hat{t}(\mathbf{x}) : \Re^N \to \Re$, enables us to treat the independent time variable t as an implicit parameter, i.e., $\hat{t}(\mathbf{x}(t)) = \hat{t}(\mathbf{x})$. One of the benefits of this is that once we have the whole isochron portrait of an oscillator in the domain of attraction of a limit cycle, any point $\mathbf{x}$ in $\mathcal{W}$ can be associated with a phase value. The fact that $\mathbf{x}$ may be an instantaneous point in the trajectory of a particular solution of (3.1), i.e., $\mathbf{x}(t)$, becomes implicit and, in a sense, irrelevant with this definition. From this perspective, the phase of a point $\mathbf{x}$ is always the same value, and this fact is consistent with the uniqueness of the solutions of (3.1) given an initial condition. Any two solutions, i.e., trajectories, of (3.1) that intersect at some point $\mathbf{x}(t)$ become the same trajectory afterwards and hence have the same instantaneous phase.

Let $\mathbf{x}(t)$ be a solution (i.e., a trajectory) of (3.1) with the initial condition $\mathbf{x}(0) \in \eta_0$, i.e., $\mathbf{x}(0)$ resides on the isochron η_0 that passes through $\mathbf{x}_{\mathrm{s}}(0)$ on the limit cycle γ. Due to Lemma 4.1.1, both $\mathbf{x}(t)$ and the steady-state periodic solution $\mathbf{x}_{\mathrm{s}}(t)$ reside on the same isochron η_t, i.e. $\mathbf{x}(t), \mathbf{x}_{\mathrm{s}}(t) \in \eta_t$ for all t. Now, as a direct result of Definition 5.2.1, we can deduce that

$$\hat{t}(\mathbf{x}(t)) = t \tag{5.3}$$

for $0 \le t \le T$, which means that the instantaneous phase $\hat{t}$ for an autonomous oscillator that is not experiencing any perturbations is simply equal to time t, even if the oscillator is not in periodic steady-state. For $t > T$, $\hat{t} = t - \lfloor \frac{t}{T} \rfloor T$ where $\lfloor \cdot \rfloor$ rounds its argument to the closest integer less than or equal to its argument. Here, $\hat{t}$ is a discontinuous quantity as a function of time, since it wraps between 0 and the period T. It makes sense to define an unwrapped version of $\hat{t}$ which is continuous, i.e., $\hat{t}(\mathbf{x}(t)) = t$ for all $t \ge 0$. With this unwrapped version, the phase of an oscillator which is not experiencing any perturbations, is simply equal to time t. This in a sense means that an oscillator not experiencing any perturbations performs as an ideal, precise time keeper, when it is in the domain of attraction of a limit cycle, even when it has not yet settled to the periodic steady-state solution. Later in the thesis, we will discuss the case of perturbed oscillators, for which (5.3) does not hold anymore.

5.3 Isochrons as Level Sets of Generalized Phase

A straightforward but profoundly significant consequence of the phase definition in Definition 5.2.1 is the following: The isochrons in the domain of attraction of a limit cycle simply become the level sets of the generalized phase, which is formally captured by the following lemma (also suggested in [14]):

Lemma 5.3.1 *Isochrons are the level sets of the phase $\hat{t}$, i.e., the isochron η_{t^*} with time tag t^* is given by*

$$\eta_{t^*} = \left\{ \mathbf{x} \;\middle|\; \hat{t}(\mathbf{x}) = t^* \right\} \tag{5.4}$$

Proof: It follows directly from Definition 5.2.1. ■

In (5.4), if we let t^* vary in the range $[0, T)$, it is possible to pick out all of the level sets of $\hat{t}$, i.e., all of the isochrons. Also note that $\hat{t} = t^* + n\,T$, for integer n, are all associated with the same level set, i.e., the same isochron η_{t^*}.

5.4 Oscillator Phase Computation Problem

Oscillator phase computation is interesting only when the phase $\hat{t}$ deviates from time t, i.e., when the oscillator is afflicted by perturbations or noise. The phase computation problem to be defined shortly is to be solved either exactly or approximately via the methods to be described in Chapter 6.

Let us assume we have a perturbed oscillator and that at the outset the oscillator state lies on the limit cycle for convenience (see Figure 5.1). We call the perturbed oscillator trajectory as $\mathbf{x}(t)$. We will elaborate more on the differential equation that $\mathbf{x}(t)$ satisfies and the implications of this model along with the exact and approximate methods (that are induced by the mathematical oscillator model) to solve for the instantaneous phase in Chapter 6. If the perturbations or noise affecting the oscillator are somehow shut down right at the outset, the oscillator will trace the limit cycle, i.e., the mathematical expression for the trajectory solution will be $\mathbf{x}_{\mathrm{s}}(t)$ (recall that we start at a point on the limit cycle and this point can always be set as $\mathbf{x}_{\mathrm{s}}(0)$).

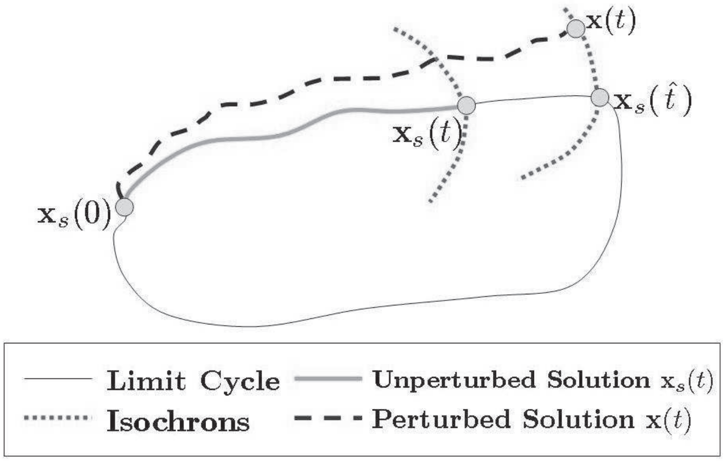


Figure 5.1: Oscillator phase computation problem ($\mathbf{x}(t)$ and $\mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t)))$ are in-phase).

The essence of the phase computation problem is indeed computation of the instantaneous phase of the perturbed oscillator solution $\mathbf{x}(t)$, i.e., $\hat{t}(\mathbf{x}(t))$. This value of this instantaneous phase is related to the unperturbed solution $\mathbf{x}_\mathrm{s}(t)$, as the following theorem shows (also stated in [14]).

Theorem 5.4.1 *Let $\mathbf{x}(t)$ be a solution of the perturbed oscillator in question. Then, at any instant t^*, $\mathbf{x}(t^*)$ and $\mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t^*)))$ are on the same isochron as depicted in Figure 5.1.*

Proof: Examining Definition 5.2.1 for $\hat{t}$, we observe that

$$\hat{t}(\mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t^*)))) = \hat{t}(\mathbf{x}(t^*)), \tag{5.5}$$

which proves that $\mathbf{x}(t^*)$ and $\mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t^*)))$ are on the same level set of $\hat{t}$. By Lemma 5.3.1, isochrons are the level sets of $\hat{t}$. Hence, the claim. ■

Theorem 5.4.1 states that $\mathbf{x}(t)$ and $\mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t)))$ are in phase at any t. Hence, it makes sense to approximate quantities involving $\mathbf{x}(t)$ by expansions around an in-phase point on γ, which is $\mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t)))$. This is how we approximately keep track of the phase of a solution $\mathbf{x}(t)$ that may never even once cross the limit cycle. We define the difference between the perturbed solution $\mathbf{x}(t)$ and the point $\mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t)))$ on the limit cycle as

$$\mathbf{y} = \mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t))) \tag{5.6}$$

and call it the *orbital deviation*.

The next section concretizes the link between the functions ($\mathbf{v}_1(t)$ and $\mathbf{H}(t)$) characterizing the local approximations of isochrons and the instantaneous phase $\hat{t}$. Based on the results to be stated, approximations involving expansions around an in-phase point on γ, i.e., approximations of the perturbed or noisy $\mathbf{x}(t)$ around the in-phase $\mathbf{x}_s(\hat{t}(\mathbf{x}(t)))$, will be possible, and this is the mainspring of the derivation of nearly all the approximate phase computation methods to be described in Chapter 6.

5.5 Phase Gradient and Hessian

Having defined the generalized phase $\hat{t}$ and established isochrons as the level sets of $\hat{t}$, we exploit the connection between the local (linear and quadratic) isochron approximations that were described in Section 4.2 and the phase $\hat{t}$ in order to determine the *gradient* and the *Hessian* of $\hat{t}$ evaluated on the limit cycle γ. The phase gradient ($\mathbf{v}_1(t)$ as will be proved next) and Hessian ($\mathbf{H}(t)$) on the limit cycle will be essential in deriving approximate phase equations for oscillators later in our treatment. The isochron-PPV connection has been rigorously established also in [25].

Theorem 5.5.1 *1. The gradient of phase $\hat{t}$ (defined as a column vector here) evaluated on γ is given by the PPV $\mathbf{v}_1(t)$:*

$$\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_s(t)} = \mathbf{v}_1(t). \tag{5.7}$$

2. The Hessian of phase $\hat{t}$ evaluated on γ is given by the matrix $\mathbf{H}(t)$ in (4.10):

$$\nabla^2_{\mathbf{x}}\, \hat{t}(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_s(t)} = \mathbf{H}(t). \tag{5.8}$$

Since the Hessian of a function is simply equal to the Jacobian of its gradient, it follows that the matrix $\mathbf{H}(t)$ is simply equal to the Jacobian of the PPV evaluated on the limit cycle γ.

Proof:

1. Since the isochron η_t passing through $\mathbf{x}_s(t)$ is a level set for the phase $\hat{t}$, the gradient $\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x}_s(t))$ must be normal to the tangent plane for η_t at $\mathbf{x}_s(t)$. Due to (4.21), $\mathbf{v}_1(t)$ is

normal to η_t at $\mathbf{x}_\mathrm{s}(t)$. Therefore, we may declare $\nabla_\mathbf{x}\, \hat{t}(\mathbf{x}_\mathrm{s}(t)) = c\, \mathbf{v}_1(t)$, for some $c \neq 0$. Then,

$$\begin{aligned} \left.\frac{\mathrm{d}\hat{t}(\mathbf{x})}{\mathrm{d}t}\right|_{\mathbf{x}_\mathrm{s}(t)} &= [\nabla_\mathbf{x}\, \hat{t}(\mathbf{x}_\mathrm{s}(t))]^\intercal \frac{\mathrm{d}\mathbf{x}_\mathrm{s}(t)}{\mathrm{d}t} \\ &= [c\, \mathbf{v}_1(t)]^\intercal\, \mathbf{u}_1(t). \end{aligned} \tag{5.9}$$

The last expression in (5.9) is obviously equal to c due to the normalization condition in (3.13), but it is also equal to 1 due to (5.3). Therefore, $c = 1$. Hence, the claim. [25] contains an alternative derivation.

2. See Section 4.2.

■

The approximate first-order and second-order phase equations for perturbed oscillators that we will later derive in the thesis will make use of the phase gradient and Hessian that we have established above.

5.6 Summary

In this chapter, a rigorous definition of oscillator phase is established based on the isochron concept, and the phase computation problem is formulated. We first remarked the generalized period notion for oscillators. Following the phase definition, isochrons were then shown to be the level sets of the phase previously defined. Upon these premises, it was then possible to formulate and state the oscillator phase computation problem. Considering the high computational costs of isochron computations, it is easy to deduce that phase computation methods depending on exact isochron information will be computationally expensive. Therefore, we foresee that practically usable methods must make use of isochron approximations, characterized by $\mathbf{v}_1(t)$ and $\mathbf{H}(t)$. The last task in this chapter was constructing the link between oscillator phase and these two functions.

Chapter 6

PHASE EQUATIONS AND PHASE COMPUTATION SCHEMES FOR OSCILLATORS

In the previous chapter, we have formally stated the oscillator phase computation problem. The methods to be described in this chapter are designed to solve this problem either exactly or approximately. These methods come in two flavors: There are *phase equations* in the form of ODEs and *phase computation schemes* as algebraic equations, all to be solved for the instantaneous phase $\hat{t}$ of a trajectory of perturbed oscillator states. The phase equations are theoretically intricate, both in derivation and analysis, and also amenable to semi-analytical investigation. An exact phase equation (reviewed in Section 6.1 in line with the explanations provided in [14]), although itself practically not usable, facilitates the derivation of the already known, acclaimed and extensively used first-order phase equation, employing linear approximations for isochrons, (of Section 6.2 and extensively analyzed in [16, 1, 2, 17, 18, 5, 14, 25]) and the novel second-order phase equation, making use of quadratic approximations for isochrons (of Section 6.3). The phase computation schemes are simpler, computational complexitywise more costly but more accurate with respect to the equations, as explained in Section 6.4. An overview of these phase computation methods, along with a unified classification framework characterized by the approximations employed in the attributes of isochrons (see Chapter 4) and orbital deviations (see Section 5.4) is provided in Section 6.5, where we also state suggestions to the prospective user, in terms of the computational complexities incurred with each method and where it would be advantageous to favor a higher order method. Section 6.6 includes phase computation results on several intricate oscillators, obtained with the proposed methods.

6.1 Exact Phase Equation

Having discussed isochrons, introduced the formal definition of the generalized instantaneous phase and also stated formally the oscillator phase computation problem, we are now fully equipped to embark on deriving phase equations for autonomous (Section 6.1.1) and perturbed oscillators (Section 6.1.2). A concise and rigorous treatment of phase equations based on isochrons, similar to the one described in this section, is given in [14], though with certain gaps in the derivations. Here, we start with the derivations in [14] and provide a detailed and revealing account based on the rigorous definition of generalized phase discussed in Chapter 5.

6.1.1 Autonomous Unperturbed Oscillators

We first consider an autonomous oscillator described by (3.1) that is not experiencing any perturbations but not necessarily in periodic steady-state. We assume an initial condition $\mathbf{x}(0)$ for (3.1) that is in the domain of attraction $\mathcal{W}$ of a limit cycle γ associated with the periodic steady-state solution $\mathbf{x}_{\mathrm{s}}(t)$, i.e., $\mathbf{x}(0) \in \mathcal{W}$. Without loss of generality, we assume that $\mathbf{x}(0) \in \eta_0$, i.e., $\mathbf{x}(0)$ resides on the isochron η_0 that passes through $\mathbf{x}_{\mathrm{s}}(0)$ on the limit cycle γ. Then, for any $\mathbf{x}(0) \in \mathcal{W}$ and $\mathbf{x}(t)$, we can derive

$$\begin{aligned} \frac{\mathrm{d}\hat{t}(\mathbf{x})}{\mathrm{d}t} &= [\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x}(t))]^{\mathsf{T}}\, \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} && (6.1)\\ &= [\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x}(t))]^{\mathsf{T}}\, \mathbf{f}(\mathbf{x}(t)) = 1 && (6.2)\end{aligned}$$

using the chain rule for derivatives and (3.1). The equality to 1 follows from (5.3) that was derived in Section 5.2. We should point out here that (6.2) holds point-wise for t (an observation reported in [14]), i.e., it is irrelevant whether $\mathbf{x}(t)$ is in fact a solution trajectory obtained by solving (3.1) with a particular initial condition or any other set of points $\mathbf{x}$ indexed by t.

The exact phase equation for autonomous, unperturbed oscillators is given by

$$\frac{\mathrm{d}\hat{t}(\mathbf{x})}{\mathrm{d}t} = [\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x}(t))]^{\mathsf{T}}\, \mathbf{f}(\mathbf{x}(t)) = 1, \quad \hat{t}(\mathbf{x}(0)) = 0 \tag{6.3}$$

where we assume, without loss of generality, that $\mathbf{x}(0) \in \eta_0$. The solution of (6.3) above is simply

$$\hat{t} = t \tag{6.4}$$

as previously identified in (5.3) in Section 5.2. If (6.3) above is evaluated on the limit cycle γ it takes the form

$$[\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x}_{\mathrm{s}}(t))]^{\mathsf{T}}\, \mathbf{f}(\mathbf{x}_{\mathrm{s}}(t)) = \mathbf{v}_1(t)^{\mathsf{T}}\mathbf{u}_1(t) = 1 \tag{6.5}$$

6.1.2 Perturbed Oscillators

We now concentrate on perturbed oscillators described by

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{x}(t), t). \tag{6.6}$$

where $\mathbf{b}(\mathbf{x}(t), t)$ is the perturbation vector, dependent on both the state vector $\mathbf{x}$ and explicitly on time t. We assume that the solutions of the perturbed oscillator in (6.6) never leave the domain of attraction $\mathcal{W}$ for a particular limit cycle γ. In other words, if the perturbation $\mathbf{b}(\mathbf{x}(t), t)$ is turned off at some point, the oscillator returns back to the same attracting limit cycle eventually. Next, we derive the form of the exact phase equation for the perturbed oscillator in (6.6) as captured by the following theorem:

Theorem 6.1.1 *The following is an exact equation describing the time evolution of $\hat{t}$ for the perturbed oscillator described by (6.6):*

$$\frac{\mathrm{d}\hat{t}(\mathbf{x}(t))}{\mathrm{d}t} = 1 + \left[\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x}(t))\right]^{\intercal} \mathbf{b}(\mathbf{x}(t), t), \quad \hat{t}(0) = 0 \tag{6.7}$$

Proof: (6.7) is obtained through

$$\left[\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x}(t))\right]^{\intercal} \left[\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{x}(t), t)\right], \tag{6.8}$$

i.e., the scalar product of the phase gradient $[\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x}(t))]$ and (6.6). Noting that (6.2) holds point-wise at all t (with t as an implicit parameter) irrespective of the fact that $\mathbf{x}(t)$ is a solution trajectory for an autonomous or perturbed oscillator, (6.7) follows. A similar derivation of the exact phase equation was given in [14]. ■

Eqn. (6.7) above is an exact equation for $\hat{t}$, but $\nabla_{\mathbf{x}}\, \hat{t}(\mathbf{x}(t))$ above is the phase gradient at $\mathbf{x}(t)$ in $\mathcal{W}$ which is not necessarily on the limit cycle γ. As shown in Theorem 5.5.1, the phase gradient *on* the limit cycle is given by the PPV, but the computation of its value at other points in $\mathcal{W}$ requires the complete isochron portrait of the oscillator. As discussed before, computing the isochrons of an oscillator is a very difficult task. Still, one can use the phase Hessian $\mathbf{H}(t)$ on the limit cycle to compute an approximation for the phase gradient at other points in $\mathcal{W}$. In fact, later we indirectly make use of this fact in deriving approximate, second-order phase equations for perturbed oscillators.

We are now in a position to derive approximate phase equations for perturbed oscillators that are useful in practice. Particularly, we are going to make use of Theorem 5.4.1 for the purpose.

6.2 First-Order Phase Equation

In this section, we start with the exact phase equation in (6.7) and, through some approximations, derive a *first-order phase equation.* We call this equation a first-order equation, because, as we show in this section, it is founded on first-order, i.e., linear approximations for the isochrons of an oscillator.

In order to justify the approximations we will apply on the exact phase equation in (6.7), we first state two assumptions.

Assumption 6.2.1 (Weak Perturbation) *We assume that the perturbation vector* $\mathbf{b}(\mathbf{x}(t), t)$ *in (6.6) is of small magnitude for all* t*, so that the orbital deviation in (5.6) stays small, i.e., the oscillator does not wander too far away from its limit cycle, which is made precise below.*

Assumption 6.2.2 (Linear Orbital Deviation Dynamics) *We assume that the orbital deviation in (5.6) is of small magnitude so that it always (approximately) resides in an hyperplane that is defined by the linear approximation for the isochron* $\eta_{\hat{t}(\mathbf{x}(t))}$ *in (4.21) passing through* $\mathbf{x}_{\mathrm{s}}(\hat{t}(\mathbf{x}(t)))$*. More precisely, the orbital deviation is approximately in the linear subspace spanned by the Floquet eigenmodes* $\mathbf{u}_i(\hat{t}(\mathbf{x}(t)))$*,* $i = 2 \ldots N$ *of (3.5) excluding the persistent mode, i.e.,*

$$\mathbf{y} \approx \sum_{i=2}^{N} c_i \, \mathbf{u}_i(\hat{t}(\mathbf{x}(t))), \tag{6.9}$$

which follows from (3.8) and (4.21).

We now derive the first-order phase equation from the exact equation in (6.7) by employing two approximations: The phase gradient $\nabla_{\mathbf{x}} \hat{t}(\mathbf{x}(t))$ and the state dependent perturbation $\mathbf{b}(\mathbf{x}(t), t)$ at $\mathbf{x}(t)$ are approximated with the ones evaluated at $\mathbf{x}_{\mathrm{s}}(\hat{t}(\mathbf{x}(t)))$, exactly on the limit cycle, as follows.

$$\nabla_{\mathbf{x}} \, \hat{t}(\mathbf{x}(t)) \approx \nabla_{\mathbf{x}} \, \hat{t}(\mathbf{x}_{\mathrm{s}}(\hat{t})) = \mathbf{v}_1(\hat{t}), \tag{6.10}$$

$$\mathbf{b}(\mathbf{x}(t), t) \approx \mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}), t). \tag{6.11}$$

where (6.10) follows from (5.7). The above approximations are motivated by the assumption that the orbital deviation in (5.6) is small, and hence $\mathbf{x}(t)$ and $\mathbf{x}_{\mathrm{s}}(\hat{t}(\mathbf{x}(t)))$ are close to each other. The approximations in (6.10) and (6.11) can be considered as zeroth order Taylor's series expansions for the phase gradient and the state dependent perturbation around

$\mathbf{x}_{\mathrm{s}}(\hat{t}(\mathbf{x}(t)))$ on the limit cycle. Later below, we will prove that the above approximations are completely consistent with Assumption 6.2.2. By substituting (6.10) and (6.11) into the exact phase equation in (6.7), we arrive at the first-order phase equation

$$\frac{\mathrm{d}\hat{t}}{\mathrm{d}t} = 1 + \mathbf{v}_1^{\mathsf{T}}(\hat{t})\,\mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}), t), \quad \hat{t}(\mathbf{x}(0)) = 0, \tag{6.12}$$

where $\hat{t} = \hat{t}(\mathbf{x}(t))$. We observe that in (6.12), $\mathbf{v}_1^{\mathsf{T}}(\hat{t})\,\mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}), t)$ is the coefficient of the component along $\mathbf{u}_1(\hat{t})$, the persistent mode, of the perturbation vector $\mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}), t)$.

The *first-order phase equation* in (6.12) is in fact the acclaimed phase equation we mentioned in Chapter 1 that has been used in various disciplines as a *reduced* model for weakly perturbed or noisy oscillators [16, 1, 2, 17, 18, 5, 14, 25].

We will shortly prove the consistency of the first-order phase equation with Assumption 6.2.2 and demonstrate that it is founded on linear isochron approximations. The first-order phase equation along with some of the results discussed in this section have appeared previously in the literature in various forms. However, we delve on the topic in order to have a unified treatment of first and second-order phase equations and to put the acclaimed first-order phase equation into context and emphasize its origins and approximate nature.

The following lemma states that if the state dependent perturbation is first evaluated on the limit cycle and then projected onto the persistent Floquet mode $\mathbf{u}_1$, then the solution of the perturbed oscillator equations is simply equal to a time/phase-shifted version of the periodic steady-state solution $\mathbf{x}_{\mathrm{s}}(t)$, where the phase shift is exactly characterized by the generalized phase $\hat{t}$ that is the solution of the first-order phase equation in (6.12).

Lemma 6.2.1 *The solution $\hat{t}$ of the first-order phase equation in (6.12) satisfies the following equation:*

$$\frac{\mathrm{d}\mathbf{x}_{\mathrm{s}}(\hat{t})}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}_{\mathrm{s}}(\hat{t})) + \left[\mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}), t)\right]\mathbf{u}_1(\hat{t}) \tag{6.13}$$

where $\hat{t} = \hat{t}(\mathbf{x}(t))$.

Proof: Noting that

$$\frac{\mathrm{d}\mathbf{x}_{\mathrm{s}}(\hat{t})}{\mathrm{d}\hat{t}} = \mathbf{f}(\mathbf{x}_{\mathrm{s}}(\hat{t})) = \mathbf{u}_1(\hat{t}) \tag{6.14}$$

we can write

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}_{\mathrm{s}}(\hat{t})}{\mathrm{d}t} &= \left[\frac{\mathrm{d}\mathbf{x}_{\mathrm{s}}(\hat{t})}{\mathrm{d}\hat{t}}\right]\left[\frac{\mathrm{d}\hat{t}(\mathbf{x}(t))}{\mathrm{d}t}\right] \\ &= \left[\mathbf{u}_1(\hat{t})\right]\left[1 + \left[\mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}), t)\right]\right] \end{aligned} \tag{6.15}$$

where (6.12) was used. Eqn. (6.13) immediately follows from (6.15). ■

The above result may seem surprising at first: The first-order phase equation is an approximate equation, but its solution $\hat{t}$ somehow satisfies the perturbed oscillator equations in (6.13) exactly. One can resolve this dilemma easily by simply noting that the projected perturbation $\left[\mathbf{v}_1^\mathsf{T}(\hat{t})\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)\right]\mathbf{u}_1(\hat{t})$ in (6.13) is in a particular form which makes sure that the perturbed oscillator never leaves its limit cycle, i.e., the orbital deviation in (5.6) is always zero. As such, the approximations in (6.10) and (6.11) that involve zeroth order expansions around the limit cycle, which are used in deriving the first-order phase equation, become exact. However, for general perturbations that are not projected onto the persistent Floquet mode on the limit cycle, the perturbed oscillator will leave its limit cycle, in which case, the first-order phase equation will become approximate. The accuracy of the first-order phase equation is then determined by the degree Assumption 6.2.2 is violated. Below, we make these arguments more precise.

Next, we proceed to derive an equation for the orbital deviation. We first approximate the perturbed oscillator equations in (6.6) through first and zeroth order Taylor expansions of $\mathbf{f}(\mathbf{x}(t))$ and $\mathbf{b}(\mathbf{x}(t),t)$, respectively, around $\mathbf{x}_\mathrm{s}(\hat{t})$ and obtain

$$\frac{\mathrm{d}\mathbf{x}_\mathrm{s}(\hat{t})}{\mathrm{d}t} + \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}_\mathrm{s}(\hat{t})) + \mathbf{G}(\hat{t})\mathbf{y} + \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t) \tag{6.16}$$

where $\mathbf{y} = \mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})$ is the orbital deviation. The linearization, i.e., first-order expansion, of $\mathbf{f}(\mathbf{x}(t))$ around $\mathbf{x}_\mathrm{s}(\hat{t})$ is justified based on Assumption 6.2.2. We then perform a decomposition of the perturbation vector $\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)$ evaluated on the limit cycle using the Floquet basis as follows

$$\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t) = \sum_{i=1}^{N}\left[\mathbf{v}_i^\mathsf{T}(\hat{t})\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)\right]\mathbf{u}_i(\hat{t}) \tag{6.17}$$

which follows from the fact that $\mathbf{u}_i(\hat{t})$ and $\mathbf{v}_j(\hat{t})$ are bi-orthonormal. We can now precisely characterize the orbital deviation as below:

Theorem 6.2.1 *With the approximations in (6.16), the orbital deviation* $\mathbf{y} = \mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})$ *satisfies*

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{G}(\hat{t})\mathbf{y} + \left[\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t) - \left[\mathbf{v}_1^\mathsf{T}(\hat{t})\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)\right]\mathbf{u}_1(\hat{t})\right] \tag{6.18}$$

with the solution

$$\mathbf{y} = \sum_{i=2}^{N}\left[\frac{-\mathbf{v}_i^\mathsf{T}(\hat{t})\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)}{\mu_i}\right]\mathbf{u}_i(\hat{t}). \tag{6.19}$$

Note that the second term on the right hand side of (6.18) is $\mathbf{b}(\mathbf{x}_s(\hat{t}), t)$ with its component along the persistent mode $\mathbf{u}_1(\hat{t})$ extracted.

With Theorem 6.2.1 (lengthy proof included in Appendix Section C.2), we have thus shown that the first-order phase equation in (6.12) is consistent with Assumption 6.2.2.

6.3 Second-Order Phase Equation

We now derive a novel, more accurate *second-order phase equation* for perturbed oscillators. This equation is called second-order, because it is founded on second-order, i.e., quadratic, approximations for the isochrons of an oscillator. Many of the derivation steps for the second-order phase equation prove to be very similar to those that were carried out for the first-order phase equation. In this case, we will make use of the phase Hessian $\mathbf{H}(t)$ in (4.10) and (5.8), as well as the phase gradient $\mathbf{v}_1(t)$, evaluated on the limit cycle γ.

6.3.1 Normalization Condition for Phase Hessian $\mathbf{H}(t)$

A comprehensive treatment of the phase Hessian $\mathbf{H}(t)$, a periodic and symmetric matrix function, can be found in Section 4.2. We state here a normalization condition for $\mathbf{H}(t)$ that connects with the normalization condition in (3.13) for the PPV $\mathbf{v}_1(t)$: simply $\mathbf{H}(t)\mathbf{u}_1(t) = -\mathbf{G}^\intercal(t)\mathbf{v}_1(t)$ (see Lemma 4.2.4 for the justification).

6.3.2 Generalized Bi-Orthonormal Bases

In deriving and verifying the first-order phase equation in Section 6.2, we have employed zeroth order expansions on the limit cycle and hence used the bi-orthonormal Floquet modes in (3.8) that are valid only exactly on the limit cycle in order to perform decompositions for the perturbation and the orbital deviation. For the second-order phase equation, we will be employing higher-order expansions around the limit cycle, and hence we will need to perform decompositions of the perturbation and the orbital deviations around, and not exactly on, the limit cycle. Thus, we need to establish a generalized set of bi-orthonormal basis vectors that is also valid at points around the limit cycle. This is what we do next. We derive a generalized, approximately bi-orthonormal set starting with $\mathbf{u}_i(t)$ and $\mathbf{v}_j(t)$, for $1 \leq i, j \leq N$ as follows: We recall that $\mathbf{u}_i(t)$ and $\mathbf{v}_j(t)$ are derived from the forward and adjoint LPTV equations in (3.5) and (3.9) that are obtained by linearizing the oscillator equations around the limit cycle characterized by the periodic steady-state solution $\mathbf{x}_\mathrm{s}(t)$. As such, $\mathbf{u}_i$ and $\mathbf{v}_j$ are in fact explicit functions of $\mathbf{x}_\mathrm{s}(t)$ and can be expressed as $\mathbf{u}_i(\mathbf{x}_\mathrm{s}(t))$ and $\mathbf{v}_j(\mathbf{x}_\mathrm{s}(t))$. Their dependence on t is actually implicit, through $\mathbf{x}_\mathrm{s}(t)$. The following theorem establishes an approximately bi-orthonormal set of basis vectors that are also valid around the limit cycle:

Theorem 6.3.1 *Let us expand $\mathbf{u}_i(\mathbf{x}_\mathrm{s}(t)+\mathbf{y})$ and $\mathbf{v}_j(\mathbf{x}_\mathrm{s}(t)+\mathbf{y})$ (where $\mathbf{y}$ is of small magnitude, accounting for the displacement, i.e., orbital deviation, from γ) into a first-order Taylor series around $\mathbf{x}_\mathrm{s}(t)$ and obtain*

$$\begin{aligned}\mathbf{u}_i(\mathbf{x}_\mathrm{s}(t)+\mathbf{y}) &\approx \mathbf{u}_i(t)+\tfrac{\partial \mathbf{u}_i(t)}{\partial \mathbf{x}_\mathrm{s}(t)}\mathbf{y}\\ \mathbf{v}_j(\mathbf{x}_\mathrm{s}(t)+\mathbf{y}) &\approx \mathbf{v}_j(t)+\tfrac{\partial \mathbf{v}_j(t)}{\partial \mathbf{x}_\mathrm{s}(t)}\mathbf{y}\end{aligned} \tag{6.20}$$

The approximate forms of $\mathbf{u}_i(\mathbf{x}_\mathrm{s}(t)+\mathbf{y})$ and $\mathbf{v}_j(\mathbf{x}_\mathrm{s}(t)+\mathbf{y})$ in (6.20) for $1 \le i,j \le N$ form two sets of approximately bi-orthonormal basis vectors. More precisely,

$$\left[\mathbf{v}_i(t)+\frac{\partial \mathbf{v}_i(t)}{\partial \mathbf{x}_\mathrm{s}(t)}\mathbf{y}\right]^\mathsf{T}\left[\mathbf{u}_i(t)+\frac{\partial \mathbf{u}_i(t)}{\partial \mathbf{x}_\mathrm{s}(t)}\mathbf{y}\right] \approx \delta_{ij} \tag{6.21}$$

approximately holds if the second-order term

$$\mathbf{y}^\mathsf{T}\left[\frac{\partial \mathbf{v}_j(t)}{\partial \mathbf{x}_\mathrm{s}(t)}\right]^\mathsf{T}\left[\frac{\partial \mathbf{u}_i(t)}{\partial \mathbf{x}_\mathrm{s}(t)}\right]\mathbf{y} \tag{6.22}$$

is negligible compared with the first-order terms in $\mathbf{y}$.

Proof: We first obtain

$$\mathbf{v}_j^\mathsf{T}(t)\frac{\partial \mathbf{u}_i(t)}{\partial \mathbf{x}_\mathrm{s}(t)}+\mathbf{u}_i^\mathsf{T}(t)\frac{\partial \mathbf{v}_j(t)}{\partial \mathbf{x}_\mathrm{s}(t)}=0 \tag{6.23}$$

for $1 \le i,j \le N$, which directly follows from

$$\frac{\partial}{\partial \mathbf{x}_\mathrm{s}(t)}\left[\mathbf{v}_j^\mathsf{T}(t)\mathbf{u}_i(t)=\delta_{ij}\right] \tag{6.24}$$

that is based on (3.8). If we expand the left-hand-side of (6.21), make use of (3.8) and (6.23), and omit (6.22), we obtain (6.21). ■

Next, we use the bi-orthonormal bases established by Theorem 6.3.1 in order to decompose a state dependent perturbation. Let $\mathbf{b}(\mathbf{x}(t),t)$ be an $N\times 1$ vector, explicitly depending on both $\mathbf{x}$ and t, where $\mathbf{x}(t)=\mathbf{x}_\mathrm{s}(t)+\mathbf{y}$. We first expand $\mathbf{b}(\mathbf{x}(t),t)$ into a first-order Taylor series around $\mathbf{x}_\mathrm{s}(t)$

$$\mathbf{b}(\mathbf{x}_\mathrm{s}(t)+\mathbf{y},t)\approx \mathbf{b}(\mathbf{x}_\mathrm{s}(t),t)+\frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(t),t)}{\partial \mathbf{x}_\mathrm{s}(t)}\mathbf{y} \tag{6.25}$$

and decompose it using the bases established in Theorem 6.3.1 as below

$$\mathbf{b}(\mathbf{x}_\mathrm{s}(t),t) + \frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(t),t)}{\partial \mathbf{x}_\mathrm{s}(t)}\mathbf{y} = \sum_{i=1}^{N} c_i \left[\mathbf{u}_i(t) + \frac{\partial \mathbf{u}_i(t)}{\partial \mathbf{x}_\mathrm{s}(t)}\mathbf{y}\right] \tag{6.26}$$

where the coefficients c_i are given by

$$c_i = \left[\mathbf{v}_i(t) + \frac{\partial \mathbf{v}_i(t)}{\partial \mathbf{x}_\mathrm{s}(t)}\mathbf{y}\right]^\intercal \left[\mathbf{b}(\mathbf{x}_\mathrm{s}(t),t) + \frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(t),t)}{\partial \mathbf{x}_\mathrm{s}(t)}\mathbf{y}\right] \tag{6.27}$$

We note here that, as stated in Theorem 6.3.1, the quadratic term (in $\mathbf{y}$) above is assumed to be negligible compared with the first-order terms.

In order to use the general approximate procedure described above for projections not exactly on but close to γ, we need the partial derivatives of the Floquet modes $\mathbf{u}_i(\mathbf{x}_\mathrm{s}(t))$ and $\mathbf{v}_j(\mathbf{x}_\mathrm{s}(t))$ with respect to $\mathbf{x}_\mathrm{s}(t)$ in (6.20). We already have $\partial \mathbf{v}_1(t)/\partial \mathbf{x}_\mathrm{s}(t) = \mathbf{H}(t)$ as the phase Hessian on the limit cycle. $\partial \mathbf{u}_1(t)/\partial \mathbf{x}_\mathrm{s}(t)$ is also readily available: Since $\mathbf{u}_1(t) = \dot{\mathbf{x}}_\mathrm{s}(t) = \mathbf{f}(\mathbf{x}_\mathrm{s}(t))$, the partial derivative of $\mathbf{u}_1(t)$ with respect to $\mathbf{x}_\mathrm{s}(t)$ is given by

$$\frac{\partial \mathbf{u}_1(t)}{\partial \mathbf{x}_\mathrm{s}(t)} = \frac{\partial \mathbf{f}(\mathbf{x}_\mathrm{s}(t))}{\partial \mathbf{x}_\mathrm{s}(t)} = \mathbf{G}(t). \tag{6.28}$$

The partial derivatives with respect to $\mathbf{x}_\mathrm{s}(t)$ of modes other than $\mathbf{u}_1(\mathbf{x}_\mathrm{s}(t))$ and $\mathbf{v}_1(\mathbf{x}_\mathrm{s}(t))$ are not available. We recall that the other Floquet modes $\mathbf{u}_i(\mathbf{x}_\mathrm{s}(t))$ and $\mathbf{v}_j(\mathbf{x}_\mathrm{s}(t))$ (for $i \neq 1$ and $j \neq 1$) themselves are not available either, since they are not needed eventually in performing computations. However, their existence is used in the derivations.

Now, if we set $i = j = 1$ in (6.21) we obtain

$$\left[\mathbf{v}_1(t) + \mathbf{H}(t)\mathbf{y}\right]^\intercal \left[\mathbf{u}_1(t) + \mathbf{G}(t)\mathbf{y}\right] \approx 1 \tag{6.29}$$

which holds if we note (4.39) and omit the second-order term $\mathbf{y}^\intercal\mathbf{H}(t)\mathbf{G}(t)\mathbf{y}$.

Now, all the tools are available for carrying out reasonably accurate projections around γ. We will be resorting to projections onto the perturbed persistent mode $\mathbf{u}_1(\mathbf{x}_\mathrm{s}(t) + \mathbf{y})$ to account for the phase of our perturbed oscillator.

6.3.3 Derivation of the Second-Order Phase Equation

In deriving the second-order phase equation, we still uphold Assumption 6.2.1, but the phase equation we eventually derive will be more accurate than the first-order phase equation for perturbations of similar magnitude, or equivalently, for the same accuracy level, we will be able to handle oscillators which are not as weakly disturbed. For the second-order case, Assumption 6.2.2 will however be replaced with the following:

Assumption 6.3.1 (Quadratic Orbital Deviation Dynamics) *We assume that the orbital deviation $\mathbf{y} = \mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t)))$ is of small magnitude so that it (approximately) resides in a quadric hypersurface that is defined by the quadratic approximation for the isochron $\eta_{\hat{t}(\mathbf{x}(t))}$ in (4.10) passing through $\mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t)))$.*

We now obtain the second-order phase equation from the exact equation in (6.7) by employing two approximations: The phase gradient $\nabla_\mathbf{x}\, \hat{t}(\mathbf{x}(t))$ and the state dependent perturbation $\mathbf{b}(\mathbf{x}(t), t)$ at $\mathbf{x}(t)$ are both expanded into a first-order Taylor series around $\mathbf{x}_\mathrm{s}(\hat{t}(\mathbf{x}(t)))$ on the limit cycle

$$\nabla_\mathbf{x}\, \hat{t}(\mathbf{x}(t)) \approx \mathbf{v}_1(\hat{t}) + \mathbf{H}(\hat{t})\, \mathbf{y}, \tag{6.30}$$

$$\mathbf{b}(\mathbf{x}(t), t) \approx \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t) + \frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\, \mathbf{y}. \tag{6.31}$$

where (6.30) follows from (5.8) and the fact that the Hessian of a quantity is equal to the Jacobian of its gradient. We note here that, in deriving the first-order phase equation in Section 6.2, the expansions above were done to zeroth order. From this perspective, one can refer to the phase equations being derived as zeroth and first-order. However, we have chosen the naming convention currently in use in order to reflect the isochron approximation order these phase equations correspond to. Essentially, the first-order expansion for the phase gradient in (6.30) corresponds to a quadratic isochron approximation.

By substituting (6.30) and (6.31) into the exact phase equation in (6.7), we obtain the second-order phase equation:

$$\begin{aligned} \frac{\mathrm{d}\hat{t}}{\mathrm{d}t} &= 1 + \\ &\left[\mathbf{v}_1(\hat{t}) + \mathbf{H}(\hat{t})\, \mathbf{y}\right]^\intercal \left[\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t) + \frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\, \mathbf{y}\right] \end{aligned} \tag{6.32}$$

with $\hat{t}(\mathbf{x}(0)) = 0$ as the initial condition. If we compare the second-order phase equation above with the first-order one in (6.12), we observe that the right-hand-side of (6.32) has extra terms which involve the orbital deviation $\mathbf{y}$. These terms indeed help increase the accuracy of the second-order equation over the first-order one, but the offshoot is that now a differential equation for the orbital deviation $\mathbf{y}$ itself is indispensibly necessary, so that the two equations, (6.32) and the one describing the dynamics of $\mathbf{y}$, which is to be derived shortly, can be simultaneously solved in a coupled manner for both $\hat{t}(\mathbf{x}(t))$ and $\mathbf{y}$. The first-order phase equation in (6.12) does not have any terms that involve the orbital deviation $\mathbf{y}$, and hence it can be solved by itself without requiring a coupled solution for $\mathbf{y}$ using (6.18).

Finally we note that, recalling (6.26) and (6.27), the second term on the right-hand side of (6.32) constitute the coefficient c_1 of the component along $\mathbf{u}_1(\mathbf{x}_\mathrm{s}(\hat{t})+\mathbf{y}) \approx \mathbf{u}_1(\hat{t})+\mathbf{G}(\hat{t})\,\mathbf{y}$, of the perturbation vector $\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t})+\mathbf{y},t) \approx \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t) + \left[\partial\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)/\partial\mathbf{x}_\mathrm{s}(\hat{t})\right]\,\mathbf{y}$. In contrast with the first-order equation in (6.12) where $\mathbf{u}_1(\hat{t})$ is the persistent mode, here, $\mathbf{u}_1(\hat{t})+\mathbf{G}(\hat{t})\mathbf{y}$ is employed as an approximation for the persistent mode off the limit cycle γ.

6.3.4 Orbital Deviation Equation and Consistency

Next, we derive a differential equation for the orbital deviation $\mathbf{y}$, show that the second-order phase equation is consistent with Assumption 6.3.1 and it is indeed founded on quadratic isochron approximations. Our treatment here follows the flow in Section 6.2 used for the first-order case.

The following lemma states that if the state dependent perturbation is first expanded into a first-order Taylor series around the limit cycle and then projected (using the generalized bases we have established in Section 6.3.2) onto the persistent mode $\mathbf{u}_1$, then the solution of the perturbed oscillator equations is simply equal to a time/phase-shifted version of the periodic steady-state solution $\mathbf{x}_\mathrm{s}(t)$, where the phase shift is exactly characterized by the generalized phase $\hat{t}$ that is the solution of the second-order phase equation in (6.32).

Lemma 6.3.1 *The solution $\hat{t}$ of the second-order phase equation in (6.32) satisfies the following equation:*

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}_\mathrm{s}(\hat{t})}{\mathrm{d}t} &= \mathbf{f}(\mathbf{x}_\mathrm{s}(\hat{t}))+ \\ &\left[\mathbf{v}_1(\hat{t})+\mathbf{H}(\hat{t})\,\mathbf{y}\right]^\intercal \left[\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)+\frac{\partial\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)}{\partial\mathbf{x}_\mathrm{s}(\hat{t})}\,\mathbf{y}\right]\,\mathbf{u}_1(\hat{t})\end{aligned} \tag{6.33}$$

Proof: As in the proof of Lemma 6.2.1, we note that $\mathrm{d}\mathbf{x}_\mathrm{s}(\hat{t})/\mathrm{d}\hat{t} = \mathbf{f}(\mathbf{x}_\mathrm{s}(\hat{t})) = \mathbf{u}_1(\hat{t})$, and evaluate $\left[\mathrm{d}\mathbf{x}_\mathrm{s}(\hat{t})/\mathrm{d}\hat{t}\right]\left[\mathrm{d}\hat{t}/\mathrm{d}t\right]$. ■

We now proceed to derive an equation for the orbital deviation. We approximate the perturbed oscillator equations in (6.6) through second and first order (in contrast with the first and zeroth order expansions in Section 6.2) Taylor expansions of $\mathbf{f}(\mathbf{x}(t))$ and $\mathbf{b}(\mathbf{x}(t), t)$, respectively, around $\mathbf{x}_\mathrm{s}(\hat{t})$ and obtain

$$\begin{aligned}&\frac{\mathrm{d}\mathbf{x}_\mathrm{s}(\hat{t})}{\mathrm{d}t} + \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} \\ &= \mathbf{f}(\mathbf{x}_\mathrm{s}(\hat{t})) + \mathbf{G}(\hat{t})\,\mathbf{y} + \frac{1}{2}\frac{\partial \mathbf{G}(\hat{t})}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\,(\mathbf{y}\otimes\mathbf{y}) \\ &+ \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t) + \frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\,\mathbf{y}\end{aligned} \tag{6.34}$$

where $\frac{\partial \mathbf{G}(\hat{t})}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}$ represents a $N \times N^2$ matrix and $\otimes$ denotes the Kronecker product making $\mathbf{y}\otimes\mathbf{y}$ an $N^2 \times 1$ vector. More precisely:

$$\left[\frac{1}{2}\frac{\partial \mathbf{G}(\hat{t})}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\,(\mathbf{y}\otimes\mathbf{y})\right]_j = \frac{1}{2}\sum_{i=1}^{N}\sum_{l=1}^{N}\frac{\partial G_{ji}(\hat{t})}{\partial x_{s,l}(\hat{t})} y_i y_l. \tag{6.35}$$

We note that two additional, higher-order terms are present in (6.34) compared to (6.16), which correspond to the additional terms in the second-order phase equation on top of the ones in the first-order equation in (6.12). We can now precisely characterize the orbital deviation as below:

Theorem 6.3.2 *With the approximations in (6.34), the orbital deviation* $\mathbf{y} = \mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})$ *approximately satisfies*

$$\begin{aligned}\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} &= \mathbf{G}(\hat{t})\mathbf{y} + \frac{1}{2}\frac{\partial \mathbf{G}(\hat{t})}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\,(\mathbf{y}\otimes\mathbf{y}) \\ &+ \left[\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t) + \frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\,\mathbf{y}\right] \\ &- \left[\mathbf{v}_1(\hat{t}) + \mathbf{H}(\hat{t})\,\mathbf{y}\right]^\mathsf{T}\left[\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t) + \frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\,\mathbf{y}\right] \\ &\left[\mathbf{u}_1(\hat{t}) + \mathbf{G}(\hat{t})\,\mathbf{y}\right]\end{aligned} \tag{6.36}$$

Note that in view of (6.26) and (6.27), the last term on the right-hand side of (6.36) is basically a deflated form of $\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}) + \mathbf{y}, t) \approx \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t) + \left[\partial\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)/\partial\mathbf{x}_\mathrm{s}(\hat{t})\right]\mathbf{y}$*, with the component along* $\mathbf{u}_1(\mathbf{x}_\mathrm{s}(\hat{t}) + \mathbf{y}) \approx \mathbf{u}_1(\hat{t}) + \mathbf{G}(\hat{t})\,\mathbf{y}$*, the persistent mode off* γ *by* $\mathbf{y}$*, extracted.*

Proof: Subtracting (6.33) from (6.34), we obtain (6.36) with the addition of the following term to the right-hand side

$$\left[\mathbf{v}_1(\hat{t}) + \mathbf{H}(\hat{t})\,\mathbf{y}\right]^\mathsf{T}\left[\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t) + \frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\,\mathbf{y}\right]\left[\mathbf{G}(\hat{t})\,\mathbf{y}\right] \tag{6.37}$$

which is a higher-order term due to the presence of an additional $\mathbf{y}$ in $\mathbf{G}(\hat{t})\,\mathbf{y}$, and hence deemed negligible compared with the other terms. ■

The next theorem captures the consistency of the orbital deviation equation derived above:

Theorem 6.3.3 *The orbital deviation* $\mathbf{y}$ *that satisfies (6.36) is consistent with Assumption 6.3.1, i.e., it (approximately) resides in a quadric hypersurface that is defined by the quadratic isochron approximations in (4.10).*

Proof: This is only a proof sketch. A full version of the proof is included in Appendix Section C.2. We multiply (6.36) from the left by $\left[\mathbf{v}_1(\hat{t}) + \mathbf{H}(\hat{t})\mathbf{y}\right]^\intercal$. Then, assuming $\mathrm{d}\mathbf{y}/\mathrm{d}t \approx \mathrm{d}\mathbf{y}/\mathrm{d}\hat{t}$, omitting higher order terms, and through several manipulations from the phase Hessian theory of Section 4.2, we show that the instantaneous value of $\mathbf{y}$ that solves (6.36) satisfies (4.10) with t replaced by $\hat{t}$. ■

6.3.5 Simplified Second-Order Phase Equation

If we compare the first-order phase equation in (6.12) and the associated orbital deviation equation (6.18) with the second-order versions in (6.32) and (6.36), we observe two key distinctions:

- The coupling between the first-order equations are one way, i.e., the orbital deviation equation in (6.18) requires the solution of the phase equation in (6.12), however, the converse is not true, i.e., the phase equation can be solved by itself without coupling it to the solution of (6.18). This is a key benefit, and the main reason behind the popularity of the first-order phase equation as it represents a *significantly-reduced-order* nonlinear model (from many equations to one) for an oscillator. On the other hand, the coupling between the second-order equations for the phase and the orbital deviations is two way, requiring a coupled solution.

- The phase equations in both cases are nonlinear, i.e., the right-hand-sides of both the first-order phase equation in (6.12) and the second-order equation in (6.32) are nonlinear in $\hat{t}$. On the other hand, the first-order orbital deviation equation is linear in $\mathbf{y}$ whereas the second-order one is nonlinear, i.e., the right-hand-side of (6.36) is a quadratic function of $\mathbf{y}$ if the third-order quantity in the last term is ignored.

Having to solve nonlinear coupled equations simultaneously for the phase and orbital deviation is the price one pays in exchange for improved accuracy. One may argue that this high a price is not justified. On the other hand, it is indeed possible to reach a compromise between first and second order phase equations as follows, thanks to the phase equation theory and approximation framework developed herein. The description of orbital deviation dynamics by a set of equations as large as the original system for an oscillator is actually what leads one to question the usefulness of the second order phase equation. One may apply model order reduction techniques to the nonlinear (quadratic) orbital deviation equations, but these methods are not yet well developed. Recalling that such reductions for linear systems are well-established, one may replace the quadratic orbital deviation equations with a set of linear ones, thus trading off some of the accuracy gained through our formulations for the applicability of these reduction techniques. We present then the simplified (and reducible) second order phase macromodel as

$$\begin{aligned}\frac{\mathrm{d}\hat{t}}{\mathrm{d}t} = 1 + &\left[\mathbf{v}_1(\hat{t})\right]^{\intercal}\left[\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t) + \frac{\partial \mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\,\mathbf{y}\right] + \\ &\mathbf{b}^{\intercal}(\mathbf{x}_\mathrm{s}(\hat{t}),t)\,\mathbf{H}(\hat{t})\,\mathbf{y}\end{aligned} \tag{6.38}$$

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{G}(\hat{t})\mathbf{y} + \left[\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t) - \left[\mathbf{v}_1^{\intercal}(\hat{t})\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}),t)\right]\mathbf{u}_1(\hat{t})\right] \tag{6.39}$$

Above, (6.39) is exactly the same as (6.18), from the first order phase equation theory. (6.38) is obtained from (6.32), by omitting the quadratic term in $\mathbf{y}$. See also Appendix Section A.4 for a procedure that suggests the derivation of perhaps a practically usable and scalar phase equation that is in theory more accurate than again the scalar first-order equation of Section 6.2.

6.4 Phase Computation Schemes

The phase computation schemes are simpler methods to compute the instantaneous phase of a perturbed oscillator. Unlike the phase equations, the schemes must have a priori available the perturbed oscillator solution, i.e., solution of (6.6), for which they are used to compute the phase. The brute-force scheme to be described below in Section 6.4.1 aims to compute on which exact isochron a point in the domain of attraction resides. This scheme directly makes use of the asymptotic phase property. The linear and quadratic phase computation schemes (of Sections 6.4.2 and 6.4.3) compute, on the other hand, respectively the linear and quadratic isochron approximations on which a point in $\mathcal{W}$ lies. In view of these explanations, it is not really required for phase computations through the schemes to have oscillator states residing in a perturbed oscillator trajectory, i.e., the schemes treat individual points independently one by one. Some implementation notes and subtleties involved with the schemes are given in Section 6.4.4.

6.4.1 Brute-Force Phase Computations for Perturbed Oscillators

The brute-force method for phase computations is directly based on Definition 3.1.4 for asymptotic phase, Definition 4.1.1 for isochrons, and the connection between asymptotic phase and isochrons. In this method, the perturbed solution $\mathbf{x}(t)$ of (6.6) and the unperturbed solution $\mathbf{x}_{\mathrm{s}}(t)$ (of (6.6) with perturbation $\mathbf{b}(t)$ set to zero) are computed and recorded at a set of time points. Then, for each one of these time points, we determine the isochron on which the perturbed solution $\mathbf{x}(t)$ resides and compute the corresponding phase shift with respect to the unperturbed solution $\mathbf{x}_{\mathrm{s}}(t)$ as follows: At a particular time point t_0, $\mathbf{x}(t_0)$ is not necessarily and not generally on γ and also will have registered a phase shift $\alpha(t_0)$ (a constant value since t_0 is a particular time point) with respect to $\mathbf{x}_{\mathrm{s}}(t_0)$. The unperturbed ODEs for the oscillator (in (3.1) with the perturbation set to zero) are solved from $t = t_0$ with initial condition $\mathbf{x}(t_0)$ for "some" time (generally for a great many number of periods) in order to obtain the steady-state periodic solution $\mathbf{x}^*(t)$, which will have become "almost" periodic for $t \gg t_0$. In other words, $\mathbf{x}^*(t)$ eventually settles to tracing the limit cycle γ as time progresses and after all of the transients have died. The phase shift between $\mathbf{x}^*(t)$ and the unperturbed periodic solution $\mathbf{x}_{\mathrm{s}}(t)$ at some $t \gg t_0$ will be $\alpha(t_0)$. After the long simulation, $\alpha(t_0)$ can be calculated through various methods. For instance, the phase shift between $\mathbf{x}^*(t)$ and $\mathbf{x}_{\mathrm{s}}(t)$ can be easily computed by calculating the first-harmonic Fourier series coefficients for these periodic waveforms based on either the numerically evaluated Fourier integral or DFT/FFTs depending on whether these waveforms have been uniformly sampled as a function of time. Then, the phase $\hat{t}(\mathbf{x}(t_0))$ is found through $\hat{t}(\mathbf{x}(t_0)) = t_0 + \alpha(t_0)$. We stress here that in order to obtain the transient phase trajectory for the perturbed solution, the above computation has to be repeated for all of the recorded time points of the perturbed solution $\mathbf{x}(t)$.

The brute-force method for phase computations described above does not involve any approximations for the isochrons, and can ideally yield very accurate results. However, this method is hampered by the execrable need to simulate (i.e., solve) the ODEs for the oscillator for very long (ideally infinite) intervals of time. Since inaccuracies may arise due to finite simulation time and the residual transients that have not died completely, and as a result of round-off error accumulation when ODEs are numerically solved for very long time intervals, the brute-force method is not only extremely inefficient but may also yield grossly inaccurate results. Still, it can be used to test the performances and accuracies of alternative methods in carefully run experiments on simple examples, at the expense of the predicament just stated.

6.4.2 Phase Computations with Linear Isochron Approximations

In this new scheme we propose for phase computations, the perturbed solution $\mathbf{x}(t)$ is first computed by numerically integrating the set of ODEs in (6.6). At every time point during this numerical integration, we simply use the linear isochron approximations in order to determine on which isochron the perturbed solution $\mathbf{x}(t)$ resides and hence determine its phase $\hat{t} = \hat{t}(\mathbf{x}(t))$. We do this by solving the following equation (that describes linear isochrons) for $\hat{t}$

$$\mathbf{v}_1^{\mathsf{T}}(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_{\mathrm{s}}(\hat{t})\right] = 0 \tag{6.40}$$

that is based on (4.21) and $\mathbf{y} = \mathbf{x}(t) - \mathbf{x}_{\mathrm{s}}(\hat{t})$. In solving the above equation for $\hat{t}$, we are essentially searching for that point $\mathbf{x}_{\mathrm{s}}(\hat{t})$ on the limit cycle γ which is in-phase with $\mathbf{x}(t)$ using the linear isochron approximations. We note here that even though the above equation is based on linear isochron approximations, it is a nonlinear equation in terms of $\hat{t}$. We use Newton's method to numerically solve this equation for $\hat{t}$ at every time point during the numerical integration of (6.6) for $\mathbf{x}(t)$. In applying Newton's method to (6.40), we need the Jacobian (in this case a scalar) of the equations with respect to $\hat{t}$, which can be determined as follows by applying (3.9)

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\hat{t}} \left[\mathbf{v}_1^{\mathsf{T}}(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_{\mathrm{s}}(\hat{t})\right]\right] \\
= \quad & -\mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{G}(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_{\mathrm{s}}(\hat{t})\right] - \mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{u}_1(\hat{t}) \\
= \quad & -\mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{G}(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_{\mathrm{s}}(\hat{t})\right] - 1,
\end{aligned} \tag{6.41}$$

noting that $\mathbf{x}(t)$ in this context is merely a constant and that we have used (3.13). For the Newton's method to converge reliably, a close enough initial guess for $\hat{t}$ is needed as well. Due to the assumption on the initial condition of (6.6), we have $\mathbf{x}(0) = \mathbf{x}_\mathrm{s}(0)$ and therefore $\hat{t}(\mathbf{x}(0)) = 0$. In solving (6.40) at time point t for $\hat{t}(\mathbf{x}(t))$, we simply use $\hat{t}(\mathbf{x}(t - \Delta t))$ as the initial guess, i.e., the value of $\hat{t}$ from the previous time point that has just been computed, which should be sufficient in most cases. If it is needed, one can always use more elaborate extrapolation schemes that involve more than one previous time point in order to generate a better initial guess for $\hat{t}$.

We note here that since the quantities $\mathbf{x}_\mathrm{s}(t)$ and $\mathbf{v}_1(t)$ are available only at discrete time points as they are numerically computed, interpolations will be necessary when carrying out Newton's method explained above.

6.4.3 Phase Computations with Quadratic Isochron Approximations

The phase computation scheme we have proposed in Section 6.4.2 can be modified in a straightforward manner to use the quadratic isochron approximations instead of the linear ones so that its accuracy is improved. One would simply replace (6.40) with

$$\begin{aligned} & \mathbf{v}_1^\intercal(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right] \\ + \; & \frac{1}{2} \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right]^\intercal \mathbf{H}(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right] = 0. \end{aligned} \tag{6.42}$$

that is based on the quadratic isochron approximation in (4.10) with $\mathbf{y}(\hat{t}) = \mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})$. In solving the scalar nonlinear equation above for $\hat{t}$ using Newton's method, we need the following derivative

$$\begin{aligned} & \frac{\mathrm{d}}{\mathrm{d}\hat{t}} \Bigg[\mathbf{v}_1^\intercal(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right] \\ & \quad + \frac{1}{2} \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right]^\intercal \mathbf{H}(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right] \Bigg] \\ = \; & -\mathbf{v}_1^\intercal(\hat{t}) \mathbf{G}(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right] - \mathbf{v}_1^\intercal(\hat{t}) \mathbf{u}_1(\hat{t}) \\ & -\mathbf{u}_1^\intercal(\hat{t}) \mathbf{H}(\hat{t}) \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right] \\ & + \frac{1}{2} \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right]^\intercal \frac{\mathrm{d}\mathbf{H}(\hat{t})}{\mathrm{d}\hat{t}} \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right] \end{aligned} \tag{6.43}$$

$$= -1 + \frac{1}{2} \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right]^\intercal \frac{\mathrm{d}\mathbf{H}(\hat{t})}{\mathrm{d}\hat{t}} \left[\mathbf{x}(t) - \mathbf{x}_\mathrm{s}(\hat{t})\right] \tag{6.44}$$

Above, the first two terms arise similarly as in (6.41); note that the transition from (6.43) to (6.44) is possible through (3.13) and (4.39), and again the fact that $\mathbf{H}(t)$ is symmetric.

Also, the differential term in (6.44) can be calculated through the fact that $\mathbf{H}(t)$ satisfies (4.22) so that we have

$$\frac{\mathrm{d}\mathbf{H}(\hat{t})}{\mathrm{d}\hat{t}} = -\mathbf{G}^{\intercal}(\hat{t})\mathbf{H}(\hat{t}) - \mathbf{H}(\hat{t})\mathbf{G}(\hat{t}) - \mathbf{M}(\hat{t}). \tag{6.45}$$

This scheme as well requires interpolations on all of $\mathbf{x}_{\mathrm{s}}(t)$, $\mathbf{v}_1(t)$, and $\mathbf{H}(t)$ when implementing Newton's method as these quantities are computed numerically and available at a set of discrete time points.

6.4.4 Implementation Notes for the Approximate Phase Computation Schemes

The new phase computation schemes of Sections 6.4.2 and 6.4.3 may suffer failures originating from one main issue if the Newton's methods for numerically solving (6.40) and (6.42) are not implemented carefully: Isochrons foliate the domain of attraction of the limit cycle and one and only one isochron passes through a particular point in $\mathcal{W}$. This is required for all of the points in the domain of attraction to have a well-defined, unique phase. Well behaved oscillators that have the asymptotic phase property obey this condition. However, in some regions in the domain of attraction, different isochrons can come close to each other and result in a situation where (6.42) possesses multiple solutions that are close to each other. If the bare, standard Newton's method is used in this case to solve (6.42), the derivative (Jacobian) may become quite small due to the other nearby solutions and cause problems. Since the desired solution is still the one that is closest to the initial guess, a modified (with limiting or damping, or a simple homotopy) Newton's method can rectify the problem here. We also note that (6.40) and (6.42) always have multiple solutions that are separated from each other with integer multiples of the period of oscillation, but these rarely cause problems in Newton's method as the undesired solutions are far away from the desired one and the initial guess used.

One can also encounter other problems in solving (6.40) and (6.42) due to relatively mild inaccuracies of linear and quadratic isochron approximations. If exact isochrons are considered, one and only one isochron passes through every point in the domain of attraction of a limit cycle. However, this is not necessarily true for the linear or quadratic approximations of isochrons. Given a point in the domain of attraction, there may not exist any isochron approximation passing through it even though there may be one that is close to it. In this case, (6.40) and (6.42) will not have solutions where the LHS evaluates to zero. In such a situation, a sensible algorithm should find the value of $\hat{t}$ such that the absolute value of the LHS of (6.40) or (6.42) is minimized, i.e., compute the $\hat{t}$ corresponding to the closest isochron approximation. Here, an algorithm that searches for a minimum as opposed to a zero-crossing makes more sense. We should recall here that (6.40) and (6.42) are both *scalar* nonlinear equations with a single scalar unknown $\hat{t}$, a much easier problem to solve than a multi-dimensional nonlinear equation. One can use a variety of algorithms for solving nonlinear scalar equations, ranging from bisection search to minimization approaches to Newton's method, some of which are not practical (or do not make sense) in the multi-dimensional case.

Finally, we should note that if the numerical solutions of (6.40) and (6.42) fail to produce sensible results despite the utilization of a sophisticated algorithm for finding the zero-crossing or the extremum, it most likely means that the oscillator solution is moving too far away from the limit cycle for the kinds of perturbations considered rendering the linear and quadratic isochron approximations inaccurate for phase computations.

6.5 Overview of Phase Computation Schemes and Phase Equations for Perturbed Oscillators

We now present a classification framework for the phase equations (exact, first-order, second-order, simplified second-order) discussed so far in the thesis (in Sections 6.1 through 6.3) and also put them into context with other phase computation schemes for perturbed oscillators that were described in Section 6.4.

As discussed previously, the phase equations that were derived in this thesis can serve as reduced-order or reduced-complexity models for perturbed oscillators. As such, they are used to *replace* the original equations for perturbed oscillators in (6.6) in analytical as well as numerical studies of various phenomena such as phase noise and timing jitter in electronic oscillators [5] and the synchronization of coupled oscillators in neurological models [1]. The great utility of phase equations in *analytical* studies arises from two distinct properties:

1) The phase equations explicitly involve a scalar quantity or concept called *phase*, that is related to time, which makes it very convenient to define and analyze the time keeping performance and synchronization of oscillators.

2) The phase equations are in a much simpler form when compared with the original oscillator equations. For instance, the first-order phase equation described in Section 6.2 is a scalar nonlinear differential equation, and the simplified second-order phase equation described in Section 6.3.5 reduces the original set of nonlinear differential equations for the oscillator to a scalar nonlinear differential equation and a set of linear differential equations.

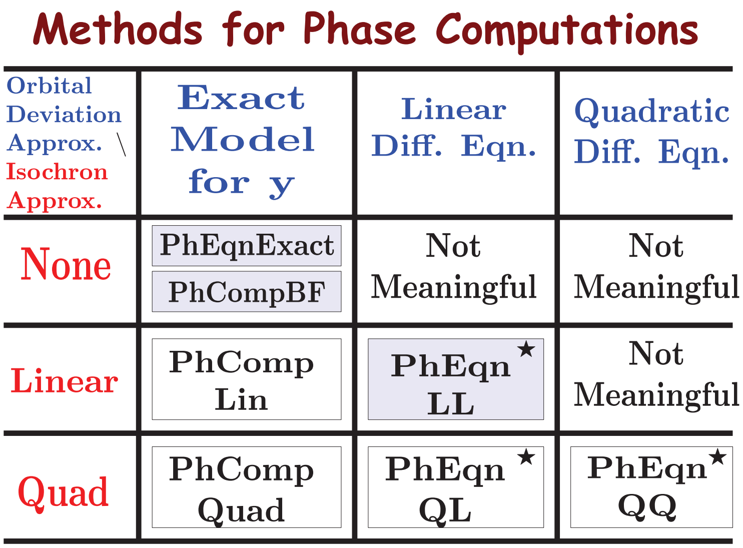
On the other hand, there exist certain applications, where numerical simulations of oscillation phenomena are performed and a technique for transient phase computations is needed (i.e., the phase of an oscillator given a particular perturbation waveform needs to be computed). In this case, the model reduction aspect of phase equations is not needed, and indeed, at the expense of having available a priori the numerical solution of (6.6) for the given perturbation, one can perform more accurate phase computations through the schemes that were explained in detail in Section 6.4, which will be appropriately acronymized next.

In all of the three phase computation schemes, one first numerically solves the original oscillator equations in (6.6) for a given perturbation to obtain the perturbed solution $\mathbf{x}(t)$. Then, the phase $\hat{t}(\mathbf{x}(t))$ that corresponds to $\mathbf{x}(t)$ is computed by simply determining the isochron on which $\mathbf{x}(t)$ resides. The three schemes differ from each other in the sense whether they employ approximations for the isochrons in this determination, and if they do, whether the approximation is linear or quadratic.

PhCompBF *Brute-Force Phase Computation Scheme with No Approximations for Isochrons*: See Section 6.4.1 for the detailed explanation. This method finds the exact isochron on which a point in $\mathcal{W}$ resides.

PhCompLin *Phase Computation Scheme based on Linear Approximations for Isochrons*: See Section 6.4.2. This method finds the linear isochron approximation that a particular point lies on.

PhCompQuad *Phase Computation Scheme based on Quadratic Approximations for Isochrons*: This technique is very similar to the scheme **PhCompLin** (see Section 6.4.3). With this method, the quadratic isochron approximation that accommodates a particular point is computed.



Methods for Phase Computations

Orbital Deviation Approx. \ Isochron Approx.	Exact Model for y	Linear Diff. Eqn.	Quadratic Diff. Eqn.
None	PhEqnExact PhCompBF	Not Meaningful	Not Meaningful
Linear	PhComp Lin	PhEqn ★ LL	Not Meaningful
Quad	PhComp Quad	PhEqn ★ QL	PhEqn★ QQ

Figure 6.1: Unified framework of phase computation methods.

6.5.1 *Classification of Phase Computation Schemes and Phase Equations*

We are now in a position to put into context the phase equations (explained in Sections 6.1 through 6.3) and the phase computation schemes (of Section 6.4). The classification framework to be presented reveals that basically two intertwined approximations are at work. First, isochrons may be approximated linearly (as in the model of Section 6.2 and **PhCompLin**) or quadratically (as in the model in Section 6.3, the simplified model of Section 6.3.5, and **PhCompQuad**). Second, the orbital deviation approximation can be not employed at all (as in **PhCompLin** and **PhCompQuad**, and this is the reason that the computation schemes need the perturbed solution to be available), or linear (as in the models of Section 6.2 and Section 6.3.5), or quadratic (as in the model of Section 6.3).

We now classify the phase computation schemes and the phase equations based on the two kinds of approximations described above. The phase computation schemes introduced in Section 6.4 were labeled as *PhComp(IsoApp)* where *IsoApp* was set to either *Lin* or *Quad* to indicate whether linear or quadratic isochron approximations were used, and to *BF* to indicate that the isochrons were computed in a brute-force manner without employing any approximations. The phase equations are now labeled as *PhEqn(IsoApp)(OrbApp)*, where

IsoApp is set to L or Q indicating that *(L)inear* or *(Q)uadratic* isochron approximations are being used, whereas *OrbApp* is set to L or Q to indicate the nature of the approximations employed for the orbital deviation:

PhEqnLL : First-order phase equation in (6.12) with the associated (linear) orbital deviation equation in (6.18).

PhEqnQQ : Second-order phase equation in (6.32) with the associated (quadratic) orbital deviation equation in (6.36).

PhEqnQL : The simplified second-order phase equation in (6.38) and the associated (linear) orbital deviation equation in (6.39).

The classifications of the phase computation schemes and phase equations are summarized in Figure 6.1. We note that the phase computation schemes, **PhCompLin** and **PhCompQuad** are new and were introduced in Section 6.4.2 and Section 6.4.3, respectively. The phase macromodels **PhEqnQQ** and **PhEqnQL** are new and introduced in Section 6.3 and Section 6.3.5, respectively. The new methods have unshaded background whereas the ones previously known in the literature have shaded backgrounds in Figure 6.1. The starred phase equations do not need to have the perturbed solution available to compute the phase shift.

6.5.2 Choice of Phase Computation Method

The choice depends on both the accuracy of the method, which is a direct consequence of the isochron approximation employed, and also the computational complexity incurred.

First, we comment on accuracy. Particularly in the case of phase equations, we will have to decide whether to employ linear (first order) or quadratic (second order) approximations. The simple rule of thumb is as follows: We compute the phase $\hat{t}$ and $\mathbf{y}$ with a second order method. If the component of $\mathbf{y}$ along $\mathbf{v}_1(\hat{t})$ (this component is always zero through a first order method) turns out to be significant on average (by some pre-determined measure), then we stick with the second order method. If not, a first order method can be used. A more detailed account on this issue is given in Appendix Section A.3.

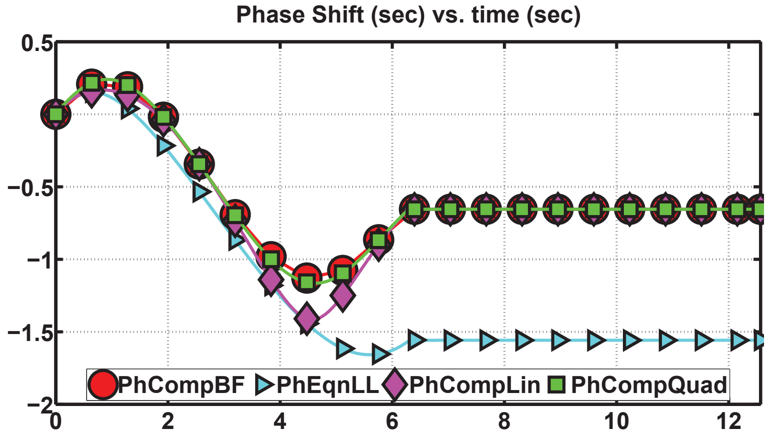


Figure 6.2: Phase computation schemes for the simple oscillator.

Second, we comment on complexity. Let us denote by N the number of states, K the number of timepoints along a single period, and L the number of timepoints along the interval of phase computation. We assume $\mathbf{x}_{\mathrm{s}}(t)$ and $\mathbf{v}_1(t)$ are computed exploiting sparsity in $\mathcal{O}(N\,K)$ time, whereas $\mathbf{H}(t)$ (not sparse) takes $\mathcal{O}(N^3\,K)$ (see Section 4.2.4). Perturbed solution computation is $\mathcal{O}(N\,L)$. We assume that matrix vector multiplications and linear system solutions with sparse matrices can be done in linear time. Then we have: **PhCompBF** incurs $\mathcal{O}(n_{\mathrm{per}}\,K\,N\,L + L\,K\log_2\,K)$ (n_{per}, ideally infinite, is the number of periods to simulate before making sure the solution settles to γ, and the logarithmic term is due to FFT), **PhCompLin** $\mathcal{O}(N\,L)$, **PhCompQuad** $\mathcal{O}(N^2\,L)$, **PhEqnLL** $\mathcal{O}(N\,L)$, and the multi-dimensional **PhEqnQQ** and **PhEqnQL** both $\mathcal{O}(N^3\,L)$. We observe that the phase computation problem poses a trade-off between accuracy and speed. A lengthy and detailed explanation on the derivation of these complexities is provided in Appendix Section B.1.

6.6 Results for Phase Computation Methods

We now compare the phase computation schemes and the phase equations (both already known and new) based on the carefully computed results of the brute-force method **PhCompBF**. In all figures, the resulting phase shift $\alpha(t) = \hat{t}(\mathbf{x}_p(t)) - t$ due to perturbations as described are shown. We present results for a simple polar oscillator and two electronic oscillators.

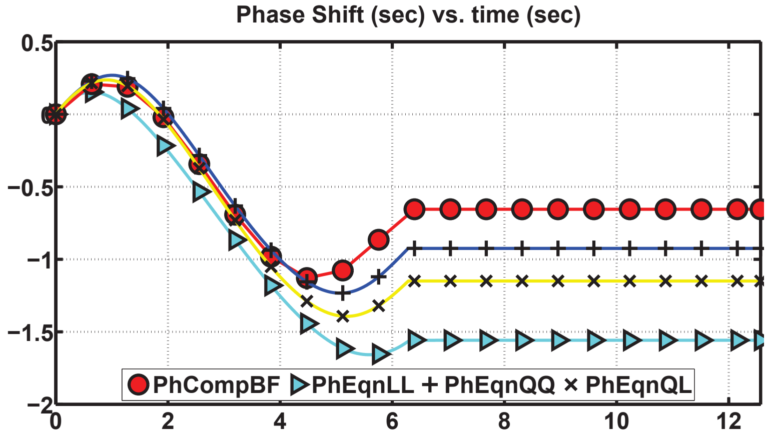


Figure 6.3: Phase equations for the simple oscillator.

6.6.1 Simple Oscillator

This simple planar oscillator is described in polar coordinates with the equations $\dot{r} = 1 - r$ and $\dot{\theta} = r$. The only limit cycle (stable periodic solution) is given by $r = 1$ (the unit circle), with the initial condition $(r_0 = 1, \theta_0 = 0)$, corresponding to $\mathbf{x}_s(t) = (\cos(t), \sin(t))$ in Cartesian coordinates. The phase of this system around $r = 1$ is given by $\hat{t}(r, \theta) = \theta + r - 1$ [2] (see Appendix Section A.1 for analytical derivations on this oscillator). We may thus use the exact phase equation **PhEqnExact** in (6.7), but **PhCompBF** complies well with **PhEqnExact** for this oscillator. Therefore, we compare the accuracies of approximate methods against **PhCompBF**, as for the other oscillators in this section.

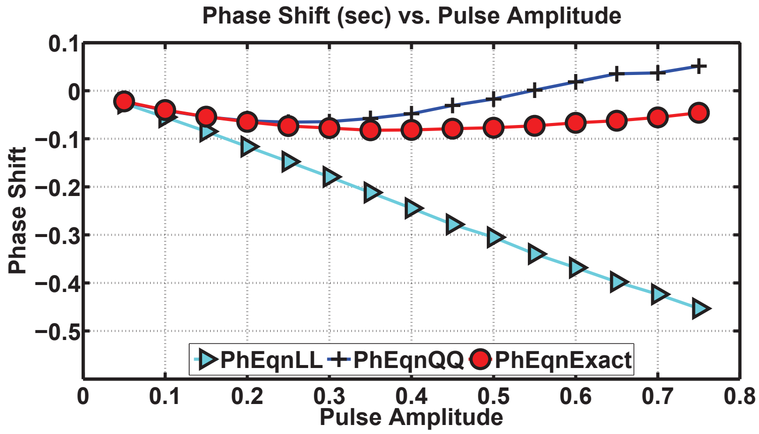


Figure 6.4: Pulse perturbations on the simple oscillator.

We first perturb the simple oscillator with $b_1(t) = 0.5\ \exp(-0.1\ t)$ for 1.5 periods, i.e. for $0 \leq t \leq 1.5T$. Note that $b_1(t)$ is the first entry of the perturbation vector $\mathbf{b}(t)$ in (6.6), and $b_2(t) = 0$. For $1.5T < t \leq 2T$, $b_1(t) = 0$. Therefore, the phase shift must exhibit changes in the interval $0 \leq t \leq 1.5T$ and must remain constant for $1.5T < t \leq 2T$. In Figure 6.2, where we compare phase computation schemes and **PhEqnLL** with **PhCompBF**, it is observed that **PhCompQuad** and **PhCompBF** are almost a perfect match. **PhCompLin** deviates from **PhCompQuad** only a little but it is also extraordinarily accurate. The acclaimed model **PhEqnLL** is not as accurate. In Figure 6.3, the phase equations are compared against **PhCompBF**. The proposed macromodels **PhEqnQQ** and **PhEqnQL** are both more accurate than **PhEqnLL**, and **PhEqnQQ** appears to be the most accurate model among the phase equations. Also note that none of the phase equations in Figure 6.3 are more accurate than the phase computation schemes of Figure 6.2.

The other simulation concerning the simple oscillator involves pulse-shaped perturbations. We test the accuracy of **PhEqnLL** and **PhEqnQQ** against that of **PhEqnExact** (computable in this case and interchangable with **PhCompBF**), by perturbing the oscillator with pulses of a fixed duration and varying magnitudes. The duration is two (about one third of T), i.e., a pulse of magnitude A (A varies between 0.05 and 0.75) afflicts the oscillator for $0 \leq t \leq 2$ and afterwards the oscillator runs free. We compute the cumulative phase shift that the oscillator suffers, with **PhEqnExact**, **PhEqnLL** and **PhEqnQQ**. The results are depicted in Figure 6.4. It is observed that **PhEqnLL** differs significantly from **PhEqnExact** as the magnitude of the pulse increases. **PhEqnQQ** follows **PhEqnExact** much more closely and is still accurate for larger pulse magnitudes, as expected, though in the extreme case (when $A = 0.75$) **PhEqnQQ** error is significant as well.

6.6.2 LC Oscillator

Figure 6.5 depicts an LC oscillator with the following set of differential equations describing its operation.

$$\begin{aligned} -C\frac{\mathrm{d}v(t)}{\mathrm{d}t} &= \frac{v(t)}{R} + i(t) + S\tanh\left(\frac{G_n}{S}v(t)\right) + b(t) \\ L\frac{\mathrm{d}i(t)}{\mathrm{d}t} &= v(t) \end{aligned} \tag{6.46}$$

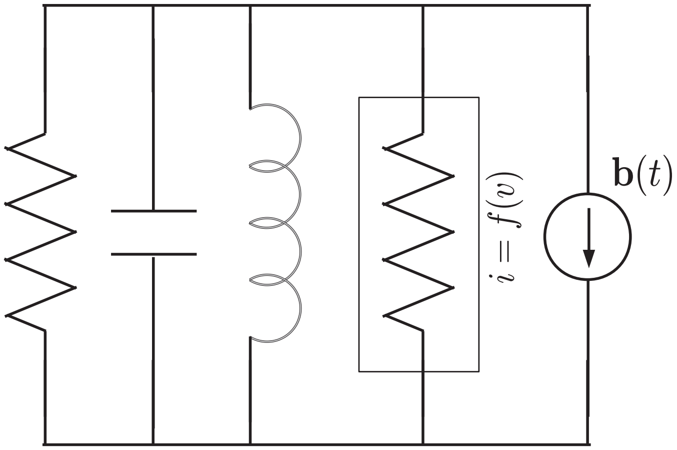


Figure 6.5: LC oscillator.

$b(t)$ is the perturbation. When $b(t) = 0$, the circuit is a free-running oscillator. The parameters are given as $L = 4.869 \times 10^{-7}/(2\pi)$ H, $C = 2 \times 10^{-12}/(2\pi)$ F, $R = 100$ ohms, $S = 1/R$ and $G_n = -1.1/R$. The LC circuit resonates at 1 GHz. The set of parameters is borrowed from [37].

We perturb this oscillator with $b(t) = 3 \times 10^9 \times C \times \exp(-0.05\, t \,/\, 10^{-9})$, for $0 \leq t \leq 1.5T$ and then the perturbation is shut off. In Figure 6.6, it is observed that both **PhCom-**

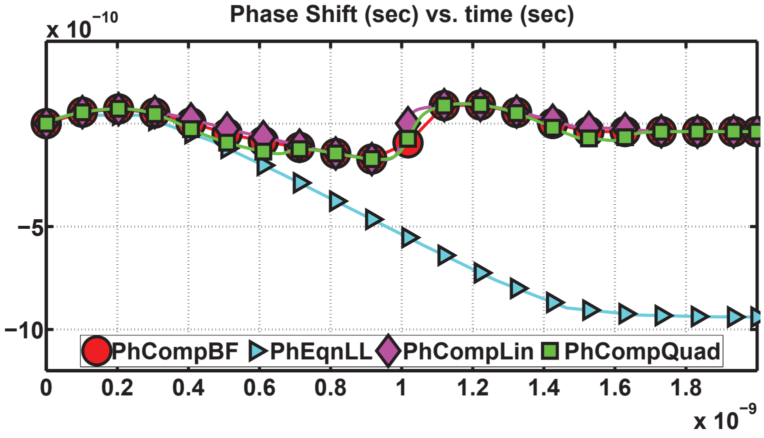


Figure 6.6: Phase computation schemes for the LC oscillator.

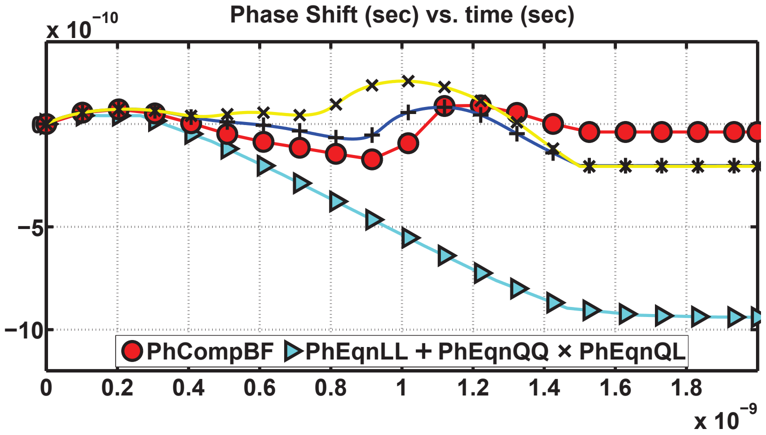


Figure 6.7: Phase equations for the LC oscillator.

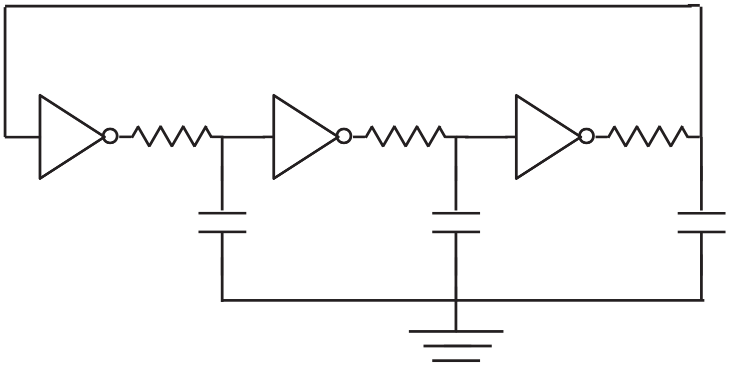


Figure 6.8: 3-stage ring oscillator.

pLin and **PhCompQuad** follow **PhCompBF** fairly closely, whereas **PhEqnLL** deviates significantly from these three. Findings for the phase equations (in Figure 6.7) show that **PhEqnQQ** and **PhEqnQL** are closer to **PhCompBF**, with **PhEqnQQ** tracing a more accurate curve. **PhEqnLL** is again observed to drift away from the other three.

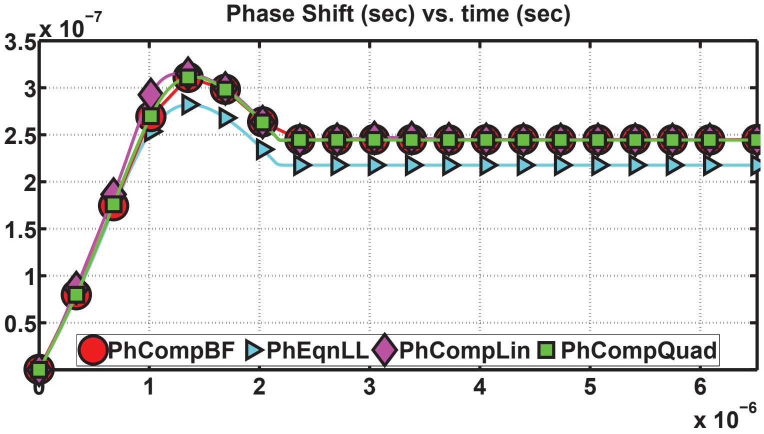


Figure 6.9: Phase computation schemes for the ring oscillator.

6.6.3 Ring Oscillator

Figure 6.8 depicts a three-stage ring oscillator with the differential equations

$$\begin{aligned}
-C_1\frac{\mathrm{d}v_1(t)}{\mathrm{d}t} &= \frac{v_1(t)}{R_1} - \frac{\tanh(G_{m3}\,v_3(t))}{R_1} \\
-C_2\frac{\mathrm{d}v_2(t)}{\mathrm{d}t} &= \frac{v_2(t)}{R_2} - \frac{\tanh(G_{m1}\,v_1(t))}{R_2} \\
-C_3\frac{\mathrm{d}v_3(t)}{\mathrm{d}t} &= \frac{v_3(t)}{R_3} - \frac{\tanh(G_{m2}\,v_2(t))}{R_3}
\end{aligned} \tag{6.47}$$

having identical stages with $C = 2$ nF, $R = 1$ kohms, and $G_m = -5$, resulting in an oscillation frequency of $1/T = 153498$ Hz. The parameter set is borrowed from [37].

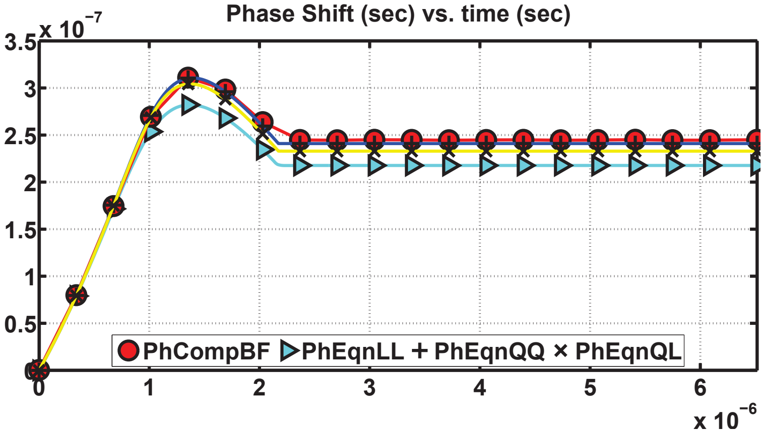


Figure 6.10: Phase equations for the ring oscillator.

We perturb this oscillator with $b_1(t) = 1.1 \times 10^5 \times C_1 \times \exp(-0.19\,t\,/\,(6.5147 \times 10^{-6}))$ for $0 \le t \le T/3$ and then the perturbation is shut off. Note that $b_2(t) = b_3(t) = 0$ in (6.6). Figure 6.9 shows that **PhCompQuad** follows **PhCompBF** just a little more closely than **PhCompLin**. **PhEqnLL** is somewhat off the target. In Figure 6.10, **PhEqnQQ**, **PhEqnQL** and **PhEqnLL** are sorted in order of decreasing accuracy, as expected.

6.7 Summary

In this chapter, we have made use of the isochron concept and the local (both linear and quadratic) approximations of isochrons (explained in Chapter 4) in order (a) to review and explicate in a precise manner known and (b) to construct and propose novel phase computation methods (both phase equations and phase computation schemes). These methods are all designed to solve the oscillator phase computation problem stated in Section 5.4. Phase equations can be used when the capability is needed to delve into semi-analytical investigation about phase perturbations and noise. Phase computation schemes provide accurate and fast transient phase simulation results. Phase equations can also be used for transient phase simulations. Analyses and tests in this chapter reveal that the phase equations provide faster but less accurate transient simulation results, compared to the schemes. However, the phase computation schemes appear to yield extraordinarily accurate results in many cases.

Chapter 7

PHASE COMPUTATIONS AND PHASE MODELS FOR DISCRETE MOLECULAR OSCILLATORS

This chapter is aimed to apply the phase computation techniques of Chapter 6 to discrete molecular biochemical oscillators. These proposed methods are not applicable to such oscillators in a straightforward manner. On the contrary, several manipulations and approximations have to be taken into account. We will shortly review molcular oscillators, provide information on several technical points. Then, after a literature review on previous work, we will proceed to explain how our methods can be applied to extract useful phase information from noisy biochemical oscillators. An intuitive presentation of the results will follow.

7.1 Introduction

7.1.1 Oscillators in Biological Systems

Biological oscillations are observed in population dynamics models, in neural systems [1], and in circadian rhythms [2]. Genetic oscillators are responsible for setting up the circadian rhythms [3]. Circadian rhythms are crucial for the survival of many species, therefore their impairment causes many health problems [38, 39]. For instance, working night shifts has been recently listed as a probable cause of cancer by the World Health Organization [40, 41, 42]. A milestone in synthetic biology is the work in [43] reporting on a genetic regulatory network called the repressilator, essentially a synthetic genetic oscillator.

In addition to studies in electronics analyzing the effects of perturbations and noise on oscillators (i.e., degraded spectral and timing properties) [4, 5], a good deal of research work for several decades has focused on the behavior of biological oscillators under various types of disturbances [1, 2, 6, 7].

7.1.2 Phase Computations for Discrete Oscillators

We have proposed in Section 4.2 a numerical method for the computation of quadratic approximations for the isochrons of oscillators, and along with linear approximations, which were previously covered in the literature, we have used these paradigms to establish the theory on the phase computation methods proposed in Chapter 6. In Section 6.2, we have reviewed the derivation of the first-order phase equation, with a formulation based on the isochron-theoretic oscillator phase. The first-order phase equation is based on the linear approximations for isochrons [2, 14, 1]. On top of this, in Section 6.3 we have also made use of again the quadratic isochron approximations of Section 4.2 to derive a novel second-order phase equation that is more accurate than the linear. The phase computation schemes of Section 6.4 are algebraic equations that can be used to solve for oscillator phase in a simple direct manner. However, the phase equations and phase computation schemes discussed above are founded on continuous oscillators described by differential equations. Therefore, these models and techniques do not directly facilitate the analysis of molecular oscillators with discrete-space models. In this chapter, we present a methodology, enabling the application of these continuous phase models and phase computation schemes on biological oscillators modeled in a discrete manner at the molecular level. This chapter details and expands on our contributions over this methodology.

We now summarize the workflow followed in the methodology and also give an outline of the chapter. Section 7.2 provides background information describing how the discrete model of the oscillator is transformed into a continuous, differential equation model through a limiting process based on the assumption that the concentration of molecular species in the model of the oscillator are large so that discrete effects are negligible [44, 45, 46, 47, 48, 49]. Section 7.2.5 reviews Gillespie's Stochastic Simulation Algorithm (SSA) [45], a Monte Carlo type of simulation scheme for discrete-state molecular systems. Reaction events in an SSA sample path are the most crucial ingredients in translating the continuous-state formalism on oscillator phase for use on molecular oscillators.

In Section 7.3, we provide a brief literature review of the approaches taken in the phase noise analysis of oscillators. Several seminal articles in the literature [50, 51, 28, 27, 52, 5, 53] are categorized according to three classification schemes in particular: the nature of the oscillator model used, the nature of the analysis method, and the phase definition adopted. We also classify in Section 7.3 the approach proposed in this chapter within the same framework.

Section 7.4 actually describes the novel contribution of this chapter, i.e., how discrete-state oscillator phase computation is accomplished using the paradigms of phase equations and phase computation schemes. Based on the continuous model developed in Section 7.2, a continuous phase model is constructed in a straightforward manner using the phase modeling techniques mentioned before. The continuous phase model is then discretized. In this discretization, the noise sources that are associated with the phase model are represented as a cumulation of the events that occur in the discrete, molecular level model of the oscillator. Through this two-way continuous-discrete transformation mechanism, we are able to perform phase computations for discrete, molecular oscillators based on the phase model theory for continuous oscillators. Moreover, a one-to-one comparison with full SSA (Stochastic Simulation Algorithm) [45] based simulations is possible, since the noise sources in the phase computation are synthesized from exactly the same events in the SSA simulation. If this technique is employed to run phase simulations for intricate discrete, molecular oscillators and the results are compared to the ones from full discrete simulations using SSA, it is expected that, for oscillators with a large number of molecules for every species, the transient phase waveforms should be quite accurate, since the oscillator that involves a large number of molecular species operates close to its continuous-deterministic limit, with small deviations. As such, the phase model constructed from the continuous-limit model of the oscillator is accurate. However, in many biological molecular oscillators, the number of molecules can be quite small. Such oscillators deviate too much from their continuous limit, and hence phase computations via continuous first-order phase models based on linear isochron approximations become inaccurate. This observation in fact prompted our work on the quadratic approximation theory and computational techniques for the isochrons of oscillators. With phase computation schemes based on quadratic isochron approximations, more accurate phase computations for discrete oscillators even with few molecules can be performed. The deviations from the continuous-deterministic limit is much better captured with quadratic (instead of linear) isochron approximations.

Section 7.5 explains how and where molecular oscillator models can be obtained to test the proposed algorithms, which types of information are obtained from the models in preparation for oscillator phase analysis, numerical implementation details for the proposed phase computation methods, and in this section are also derived the computational complexities for these methods. Section 7.6 provides performance results for the phase computation methods running on inricate molecular oscillators. The results are as expected, i.e., phase equations are quite accurate and fast for oscillators in a larger volume with big molecule numbers for the species, but they lose accuracy when a smaller volume is considered and noise effects become pronounced. Phase computation schemes are always very accurate, even in smaller volumes, but they are not as fast as the equations. Section 7.7 concludes the chapter and suggests some future research directions.

7.2 Modeling and simulation of discrete molecular oscillators

In this section we review, after giving preliminary information (Section 7.2.1), some crucial paradigms in the modeling of discrete molecular oscillators: a model that is the complete probabilistic characterization of a discrete system, known as the Chemical Master Equation or CME (Section 7.2.2), a continuous deterministic approximation to the CME in the form of the Reaction Rate Equation or RRE (Section 7.2.3), and the steps that let us proceed to a continuous stochastic model, the Chemical Langevin Equation (CLE), from again the CME (Section 7.2.4). Also a descriptive review of the SSA algorithm of Gillespie [45] for the simulation of molecular models is provided in Section 7.2.5.

7.2.1 Preliminaries

We first describe a mathematical model for an autonomous, discrete molecular oscillator based on a stochastic chemical kinetics formalism [44, 45, 46, 47, 48, 54]. We consider N molecular species denoted by $\mathsf{S}_1, \mathsf{S}_2, \ldots, \mathsf{S}_N$. Let $\mathbf{X}$ be the stochastic vector $[X_1, X_2, \ldots, X_N]^\mathsf{T}$ where X_i is the number of molecules of species S_i in the reaction chamber (i.e., a cell). The M reactions taking place among these molecular species are denoted by $\mathsf{R}_1, \mathsf{R}_2, \ldots, \mathsf{R}_M$. Let $a_j(\mathbf{X})$ denote the *propensity* [45, 47] of reaction j, i.e., the probability that one R_j reaction will occur somewhere in the system in the next infinitesimal time interval $[t, t + \mathrm{d}t)$ is given by $a_j(\mathbf{X})\,\mathrm{d}t$, i.e.,

$$\mathbb{P}\left(\mathsf{R}_j \text{ occurs in } [t, t + \mathrm{d}t)\right) = a_j(\mathbf{X})\,\mathrm{d}t \tag{7.1}$$

Let s_{ji} denote the change in the number of molecules of species S_i as a result of one R_j reaction. We define the stoichiometry vector $\mathbf{s}_j$

$$\mathbf{s}_j = [s_{j1}, s_{j2}, \ldots, s_{jN}]^\mathsf{T} \tag{7.2}$$

for reaction R_j, and the $N \times M$ stochiometry matrix [47]

$$\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_M] \tag{7.3}$$

7.2.2 Chemical master equation

The following derivation follows closely that outlined in [47]. Let us take a note of the events $\mathbf{X}(t + \mathrm{d}t) = \mathbf{x}$, $\mathbf{X}(t) = \mathbf{x} - \mathbf{s}_j$ and $\mathbf{X}(t) = \mathbf{x}$, where $\mathrm{d}t$ is an infinitesimal time element. Through several manipulations making use of these events and taking the limit as $\mathrm{d}t \to 0$ [47], we obtain

$$\begin{aligned} &\frac{\mathrm{d}\,\mathbb{P}\left(\mathbf{x}, t\right)}{\mathrm{d}t} = \\ &\sum_{j=1}^{M} \left[a_j(\mathbf{x} - \mathbf{s}_j)\,\mathbb{P}\left(\mathbf{x} - \mathbf{s}_j, t\right) - a_j(\mathbf{x})\,\mathbb{P}\left(\mathbf{x}, t\right)\right] \end{aligned} \tag{7.4}$$

where $\mathbb{P}\left(\mathbf{x}, t\right)$ denotes the probability that the system is at state $\mathbf{x}$ at time t. The above is known as the CME [47, 48, 49, 54]. If we enumerate all the (discrete) state configurations $\mathbf{X}$ can be in as $\mathtt{C}_1, \mathtt{C}_2, \ldots, \mathtt{C}_{ns}$ and define

$$p_i(t) = \mathbb{P}\left(\mathbf{x} = \mathtt{C}_i, t\right) \tag{7.5}$$

$$\mathbf{p}(t) = [p_1(t), p_2(t), \ldots, p_{ns}(t)]^\mathsf{T} \tag{7.6}$$

then, the CME in (7.4) can be written as

$$\frac{\mathrm{d}\,\mathbf{p}(t)}{\mathrm{d}t} = \mathbf{Q}\,\mathbf{p}(t) \tag{7.7}$$

where $\mathbf{Q}$ is a constant square matrix with dimension $ns \times ns$, known as the *transition rate matrix* [48, 49]. The above is a linear system of homogeneous ODEs, but the number of state configurations ns is possibly huge. It is usually not practically feasible to construct and solve (7.7). CME in (7.4) and (7.7) above corresponds to a homogeneous, continuous-time Markov chain model [48, 49, 54]. The state transitions of this Markov chain are highly structured and compactly described by the list of the reactions as in the CME. The CME provides the ultimate probabilistic characterization for a discrete molecular oscillator. It was shown that the solution of the CME converges to a unique stationary distribution. For a discrete molecular oscillator with a limit cycle, this stationary probability distribution takes the form of a "probability crater" for a planar system with two species [55].

7.2.3 From the stochastic CME to the deterministic rate equations

If we multiply both sides of CME in (7.4) with $\mathbf{x}$ and sum over all $\mathbf{x}$, we obtain, as shown especially in [47, 44],

$$\frac{\mathrm{d}\,\mathbb{E}\left[\mathbf{X}(t)\right]}{\mathrm{d}t} = \sum_{j=1}^{M} \mathbf{s}_j \; \mathbb{E}\left[a_j(\mathbf{X}(t))\right] \tag{7.8}$$

We note here that $\mathbb{E}\left[a_j(\mathbf{X}(t))\right] \neq a_j(\mathbb{E}\left[\mathbf{X}(t)\right])$ unless $a_j(\mathbf{x})$ is a linear function of $\mathbf{x}$. Thus, in general, (7.8) can not be solved for $\mathbb{E}\left[\mathbf{X}(t)\right]$ since the term $a_j(\mathbb{E}\left[\mathbf{X}(t)\right])$ involves higher-order moments of $\mathbf{X}(t)$ [47]. However, if we assume that the fluctuations of $\mathbf{X}(t)$ around its mean $\mathbb{E}\left[\mathbf{X}(t)\right]$ is negligible and thus can perform a crude moment closure scheme, i.e., if $\mathbb{E}\left[\mathbf{X}(t)\right] = \mathbf{X}(t)$, then (7.8) simplifies to

$$\frac{\mathrm{d}\,\mathbf{X}(t)}{\mathrm{d}t} = \sum_{j=1}^{M} \mathbf{s}_j \; a_j(\mathbf{X}(t)) = \mathbf{S}\;\mathbf{a}(\mathbf{X}(t)) \tag{7.9}$$

where $\mathbf{S}$ is the stoichiometry matrix defined in (7.3) and

$$\mathbf{a}(\mathbf{X}(t)) = \left[a_1(\mathbf{X}(t)), a_2(\mathbf{X}(t)), \ldots, a_M(\mathbf{X}(t))\right]^{\intercal} \tag{7.10}$$

is an $M \times 1$ column vector of reaction propensities evaluated at $\mathbf{X}(t)$. The above system of deterministic ODEs in (7.9) is known as the RRE [47, 44].

7.2.4 From CME to Langevin model

The derivations in this section have been particularly borrowed from [46]. If we assume that the reaction propensities $a_j(\mathbf{X}(t))$ for $j = 1, \ldots, M$ are constant in $[t, t + \mathrm{d}t)$ (known as the *leap condition*) [46, 47], then the number of the times reactions fire in $[t, t + \tau)$ are independent Poisson random variables [46, 47, 48, 49, 54] with mean and variance equal to $a_j(\mathbf{x}(t))\,\tau$, denoted by $\mathcal{P}_j(a_j(\mathbf{x}(t))\,\tau)$ for $j = 1, \ldots, M$. Hence, we can write

$$\mathbf{X}(t + \tau) = \mathbf{X}(t) + \sum_{j=1}^{M} \mathcal{P}_j(a_j(\mathbf{X}(t))\,\tau)\;\mathbf{s}_j \tag{7.11}$$

If we further assume that $a_j(\mathbf{x}(t))\,\tau \gg 1$, then $\mathcal{P}_j(a_j(\mathbf{x}(t))\,\tau)$ can be approximated with Gaussian random variables:

$$\mathcal{P}_j(a_j(\mathbf{x}(t))\,\tau) \approx a_j(\mathbf{x}(t))\,\tau + \sqrt{a_j(\mathbf{x}(t))\,\tau}\,\mathcal{N}_j(0,1) \tag{7.12}$$

where $\mathcal{N}_j(0,1)$ for $j = 1, \dots, M$ are independent Gaussian random variables with zero mean and unity variance [46, 47, 48, 49, 54]. Incorporating (7.12) into (7.11), we recognize the (forward) Euler discretization of the following *stochastic differential equation* (SDE), known as a *Langevin equation* [46, 47, 48, 54]:

$$\frac{\mathrm{d}\,\mathbf{X}(t)}{\mathrm{d}t} = \mathbf{S}\,\mathbf{a}(\mathbf{X}(t)) + \mathbf{S}\,\mathbb{D}\left(\left[\sqrt{\mathbf{a}(\mathbf{X}(t))}\right]\right)\xi(t) \tag{7.13}$$

where $\xi(t)$ denotes an $M \times 1$ vector of independent white stationary Gaussian processes with unity (two-sided) spectral density, and

$$\mathbb{D}\left(\left[\sqrt{\mathbf{a}(\mathbf{X}(t))}\right]\right) = \begin{bmatrix} \sqrt{a_1(\mathbf{X}(t))} & 0 & \cdots & \cdots & 0 \\ 0 & \sqrt{a_2(\mathbf{X}(t))} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & \sqrt{a_M(\mathbf{X}(t))} \end{bmatrix} \tag{7.14}$$

denotes the diagonal $M \times M$ matrix function shown in (7.13). We note here that if the stochastic, fluctuation term (known as the *diffusion* term) above is omitted, we obtain the RREs in (7.9). We note here that, with the Langevin model, the stochastic fluctuations in the oscillator are captured by the second term in the right hand side in (7.13). This term represents an *additive* noise in the model. By zeroing this additive noise term, we are able to obtain the mean, deterministic dynamics of the oscillator as the solution of the RREs in (7.9). On the other hand, in the discrete, Markov chain model of the oscillator, the mean, deterministic behavior of the system and the stochastic fluctuations are not separable from each other [46, 47, 48, 54].

7.2.5 Stochastic simulation algorithm (SSA)

Even though the CME in (7.4) and (7.7) provides the ultimate probabilistic characterization for a discrete molecular oscillator, its solution is most often not practical due to the huge number of possible state configurations. As a result, one most often performs a stochastic simulation of the continuous-time Markov chain that models the oscillator and generates a sample path or a realization for the state vector $\mathbf{X}(t)$ as a function of time t. This kind of a simulation can be performed with a technique called the SSA, proposed in Gillespie's seminal work [45]. In the original SSA algorithm [45], the computational cost per reaction event (due to the generation of a random variable from a dynamic discrete probability distribution) is $\mathcal{O}(M)$ in the number of reactions M. The cost per reaction event can be reduced to $\mathcal{O}(\log M)$ by using a binary tree for random selection of reactions [56], and to $\mathcal{O}(1)$ under certain conditions [57]. One also has to consider the fact that the time gap between reactions tends to shrink as the number of reactions M, the number of species N, and the number of molecules of every species increases. This means that the total computational cost of SSA for a given time period increases as a result [44]. On the other hand, if the numbers of molecules of all of the species are very large, discrete stochastic simulation of a discrete molecular oscillator in the sense of SSA may be unnecessary [47, 44]. In this case, the fluctuations around the deterministic limit cycle will be small, and the continuous Langevin model in (7.13) may be adequate. As the number of molecules increase, the reaction propensities $a_j(\mathbf{X}(t))$ become larger, and the fluctuation term in the Langevin model in (7.13) become less and less pronounced in comparison with the drift term, since the magnitude of the drift term is proportional to the reaction propensities whereas the fluctuation term is proportional to their square root [46, 47, 48].

Molecular models, their nature (as discrete or continuous, and as stochastic or deterministic), and the algorithms to solve these models are summarized in Figure 7.1. The approximation that leads us from the discrete stochastic CME to the continuous stochastic CLE is the Gaussian approximation to Poisson random variables and accordingly the τ-leap approximation. Similarly, infinite volume approximation takes us from the CLE to the continuous deterministic RRE. Sample paths in line with the CME can be generated through SSA. CLE is a type of stochastic differential equation, so it can be solved via appropriate algorithms. Solution of the RRE requires algorithms designed for ordinary differential equations (ODEs) [46, 47, 48].

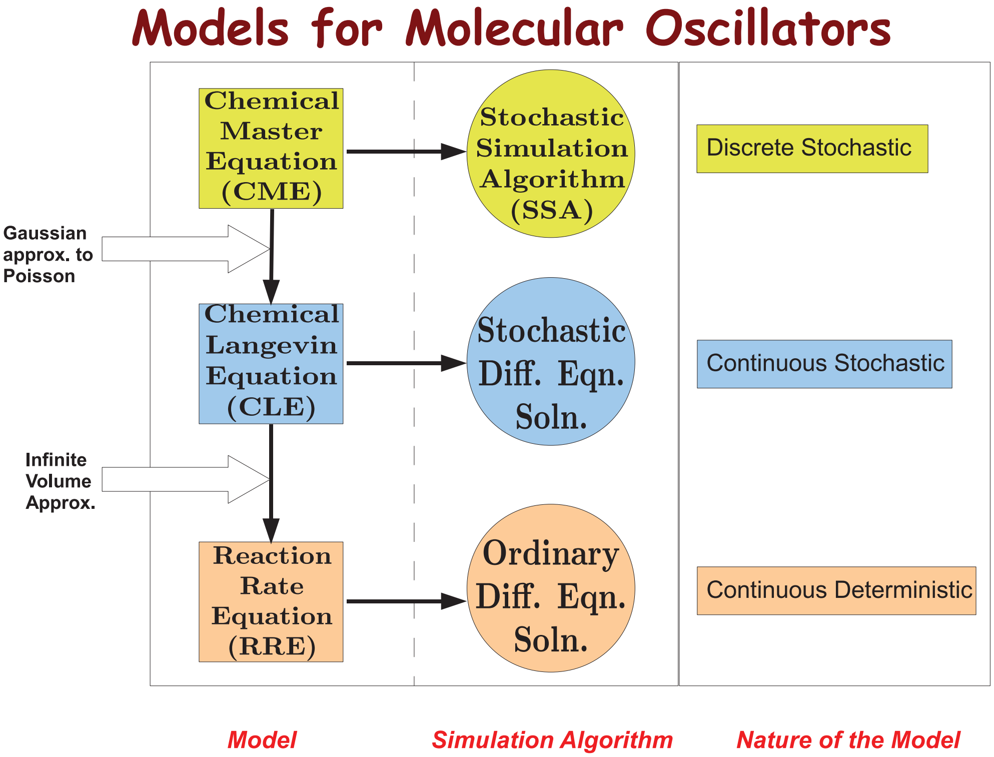

Models for Molecular Oscillators
Chemical Master Equation (CME)
Stochastic Simulation Algorithm (SSA)
Discrete Stochastic
Gaussian approx. to Poisson
Chemical Langevin Equation (CLE)
Stochastic Diff. Eqn. Soln.
Continuous Stochastic
Infinite Volume Approx.
Reaction Rate Equation (RRE)
Ordinary Diff. Eqn. Soln.
Continuous Deterministic
Model
Simulation Algorithm
Nature of the Model

Figure 7.1: Summary of molecular models and corresponding algorithms.

7.3 Related work

A classification scheme for categorizing previous work, pertaining to the phase noise analysis of biochemical oscillators, can be described as follows.

First, we note that there are basically two types of models for inherently noisy biochemical oscillators, i.e., discrete and continuous-state. CME describes the probabilistic evolution of the states of an oscillator, and it is referred to as the most accurate characterization for discrete molecular oscillators. Through approximations, one derives from CME the CLE, a continuous-state noisy model. CLE can be used to extract crucial information about the continuous-state system that is an approximate representation of its discrete-state ancestor. We note here that, in oscillator phase noise analyses, mostly the continuous-state model has been utilized [51, 28, 27, 52, 5, 53, 50].

Second, the nature of the phase noise analyses conducted can be considered in two categories, i.e., semi-analytical techniques and sample path-based approaches. Semi-analytical techniques have been developed, in particular, for the stochastic characterization of phase diffusion in oscillators [50, 51, 28, 27, 52, 5, 53]. In biology, CLE has been used as a tool in illustrating and quantifying the phase diffusion phenomena [50, 51, 28, 27, 53]. Characterization and computations pertaining to phase diffusion in electronic oscillators were carried out through a stochastic phase equation and the probabilistic evolution of its solutions [5], noting that the phase equation used was derived from an SDE (a Stochastic Differential Equation describing a noisy electronic oscillator) that corresponds to the CLE for biochemical oscillators. In all, these semi-analytical techniques are based on the continuous-state model of an oscillator. Regarding sample path-based approaches, one may recall that, in discrete state, SSA is used to generate sample paths, whose ensemble obeys the CME. In continuous state, CLE can in turn be used to generate sample paths. A recent study [52] illustrates derivations of the crucial findings presented in [28, 27, 5] and adopts an approach for phase diffusion constant computation, based on the transient phase computation of CLE-generated sample paths in an ensemble.

Third, oscillator phase can be defined via two different methods. There are the Hilbert transform-based and the isochron-based definitions. The phase computation based on the Hilbert transform [58] takes the evolution of a single state variable within a sample path to compute the phases of all time points in the whole sample path. The Hilbert transform-based phase computation technique can be used to compute the phase of any oscillatory waveform, without any information as to where this waveform came from. The oscillatory waveform could belong to one of the state variables of an oscillator generated with a simulation. This method has been utilized in [52, 50] for phase computations of sample paths. The isochron-theoretic phase (recall that an isochron portrait belongs to a limit cycle of the deterministic RRE) makes use of all of the state variables and equations for an oscillator. The isochron-based phase definition assigns a phase value to the points in the state space of the oscillator, making phase a property of the whole oscillator, not a property of just a certain state variable or a waveform obtained with a simulation of the oscillator [59, 22]. Note that even though there appears to be empirical evidence [52, 50] that there is a correspondence between the Hilbert transform-based and isochron-based phase definitions, a precise connection has not been worked out in the literature.

The hybrid phase computation techniques proposed in this chapter apply to discrete-state models and particularly the SSA generated sample paths of these models, based on the isochron-theoretic oscillator phase definition. Our approach is hybrid because isochrons are obtained based on the continuous model but the phase traces are computed for the sample paths generated by an SSA simulation that is based on the discrete model for an oscillator. This hybrid approach targets moderately noisy oscillators, within a container of not too large or small volume, consequently with not too high or low molecule numbers for the species in the system, respectively.

7.4 Phase Computations based on Langevin Models

There exists a well developed theory and numerical techniques for phase characterizations of oscillators with continuous-space models based on differential and stochastic differential equations. As described in Section 7.2.3 and Section 7.2.4, continuous models in the form of differential and stochastic differential equations can be constructed in a straightforward manner for discrete molecular oscillators. Thus, one can in principle apply the previously developed phase models and computation techniques (explained in Chapter 6) to these continuous models.

The outline of this section is as follows: After presenting the preliminaries (Section 7.4.1), the phase computation problem is introduced (Section 7.4.2). The methods in Section 7.4.3 (phase models in the form of ordinary differential equations) and in Section 7.4.4 (phase computation schemes that involve the numerical solution of certain algebraic equations) are designed to numerically solve the phase computation problem of Section 7.4.2.

7.4.1 Preliminaries

For a molecular oscillator, we observe that the deterministic RREs in (7.9) are indeed in the form of the autonomous ODEs as in (3.1). Moreover, we assume that these RREs have a stable periodic solution $\mathbf{x}_{\mathrm{s}}(t)$ (with period T) that represents a periodic orbit or limit cycle. Then, all the information in Chapter 3 concerning the state transition function, linearization of (3.1) into a forward LPTV form, the existence of the adjoint LPTV system, the Floquet functions (both forward and adjoint), particularly $\mathbf{u}_1(t)$ and $\mathbf{v}_1(t)$ applies directly to these RREs. A slight subtlety that must be pointed out is the form of the matrix function $\mathbf{G}(t)$

in the case of RREs:

$$\mathbf{G}(t) = \left.\frac{\mathrm{d}\,\mathbf{S}\,\mathbf{a}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{s}}(t)} \tag{7.15}$$

Moreover, it must be noted that the isochron theory of Chapter 4 including the approximations (linear and quadratic) and therefore existence of the phase Hessian $\mathbf{H}(t)$ (see Section 4.2) is taken for granted in the case of (7.9) as well.

7.4.2 Phase Computation Problem

The phase computation problem for oscillators (as adapted from Section 5.4 to the case in hand) can be stated as follows. It is observed in Figure 7.2 that assuming an SSA sample path and the periodic RRE solution start at the same point on the limit cycle (note that the two are in-phase initially), the two trajectories may end up on different isochrons instantaneously at $t = t_0$ (i.e., the two traces at this instant are out of phase). However, according to the properties of isochrons, there is always a point on the limit cycle that is in-phase with a particular point near the limit cycle. Therefore, the existence of $\mathbf{x}_{\mathrm{s}}(\hat{t})$ in-phase with the instantaneous point $\mathbf{x}_{\mathrm{ssa}}(t_0)$ is guaranteed. We call then the time argument $\hat{t}$ of $\mathbf{x}_{\mathrm{s}}(\hat{t})$ the instantaneous *phase* of $\mathbf{x}_{\mathrm{ssa}}(t_0)$ [8, 1, 2]. All methods described below in this section are designed to numerically compute this phase value.

7.4.3 Phase Equations based on Langevin Models

In this section, oscillator phase models in the form of ODEs as applies to discrete molecular oscillators are described. See Sections 6.1 through 6.3 for the phase equation theory in the case of continuous oscillators. We will now explain how to apply these models to discrete oscillator phase computation.

First-order phase equation based on linear isochron approximations

The first-order phase equation based on linear isochron approximations can be derived from the continuous Langevin model in (7.13) using the theory and numerical techniques described in Section 6.2, which takes the form

$$\frac{\mathrm{d}\hat{t}}{\mathrm{d}t} = 1 + \mathbf{v}^{\mathsf{T}}(\hat{t})\;\mathbf{S}\;\mathbb{D}\left(\left[\sqrt{a_{1:M}\big(\mathbf{x}_{\mathrm{s}}(\hat{t})\big)}\right]\right)\xi(t), \quad \hat{t}(0) = 0, \tag{7.16}$$

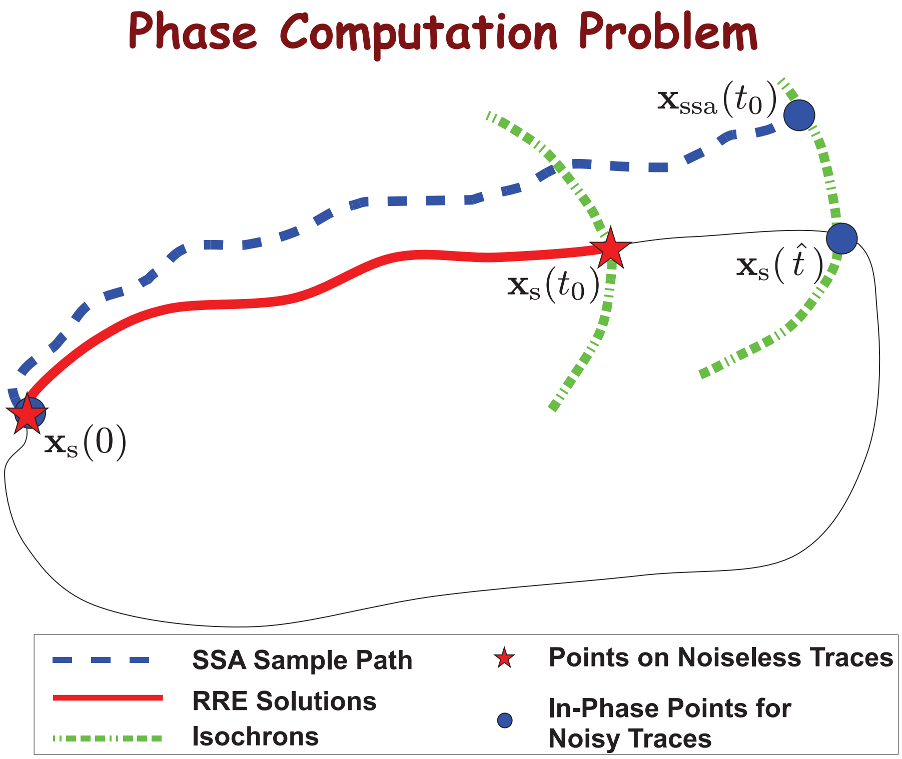

Phase Computation Problem
$\mathbf{x}_{\text{ssa}}(t_0)$
$\mathbf{x}_{\text{s}}(t_0)$
$\mathbf{x}_{\text{s}}(\hat{t})$
$\mathbf{x}_{\text{s}}(0)$
SSA Sample Path
RRE Solutions
Isochrons
Points on Noiseless Traces
In-Phase Points for Noisy Traces

Figure 7.2: Phase computation problem for continuous oscillators.

where $\hat{t}$ represents the total phase of the oscillator (in units of time) and $\mathbf{v}(t)$ is the PPV discussed above. $\mathbf{x}_{\text{s}}(\hat{t})$, the periodic solution $\mathbf{x}_{\text{s}}(t)$ evaluated at the perturbed phase $\hat{t}$, represents possibly a good approximation for the solution of the Langevin equation in (7.13) provided that the perturbed oscillator does not wander off too far away from the deterministic limit cycle represented by $\mathbf{x}_{\text{s}}(t)$.

The phase $\hat{t}$ defined above and the phase equation in (7.16), capture the deviations (from the periodic steady-state) of the perturbed oscillator only along the limit cycle, i.e., phase deviations. A perturbed oscillator also exhibits orbital deviations away from its deterministic limit cycle. Moreover, for a discrete, molecular oscillator, the deterministic periodic solution $\mathbf{x}_{\text{s}}(t)$ is merely the solution of its continuous and deterministic limit when the number of molecules are assumed to be very large. As such, the solution of a discrete molecular oscillator may exhibit large fluctuations around this continuous and deterministic limit. Thus, $\mathbf{x}_{\text{s}}(\hat{t})$ may not serve as a good approximation in such a case. In order to truly assess the quality of $\mathbf{x}_\mathrm{s}\big(\hat{t}\big)$ as an approximation in a meaningful manner, we need to compare it with a sample path solution of the discrete, Markov chain model that can be generated with an SSA simulation. However, a one-to-one comparison of $\mathbf{x}_\mathrm{s}\big(\hat{t}\big)$ based on the solution of the phase equation in (7.16) and a sample path obtained with an SSA simulation is not straightforward. In solving (7.16), one would normally generate sample paths for the independent white stationary Gaussian processes denoted by $\xi(t)$. In an SSA simulation, sample paths are generated as described in Section 7.2.5. If done so, a one-to-one comparison between a sample path from an SSA simulation and $\mathbf{x}_\mathrm{s}\big(\hat{t}\big)$ would not make sense. In order to make this sample path based comparison meaningful, we use the same discrete random events that are generated in an SSA simulation in order to synthesize the sample paths for the independent white stationary Gaussian processes $\xi(t)$ in the numerical simulation of (7.13). More precisely, we proceed as follows. We numerically compute the solution of (7.16) in parallel and synchronous with an SSA simulation. We discretize the SDE in (7.16) using time steps that are dictated by the reaction occurrence times in the SSA simulation. Assuming that the last reaction has just occurred at time t, the next reaction will occur at time $t+\tau$ and it will be the jth reaction, we form the update equation for $\hat{t}$ as follows

$$\hat{t}(t+\tau) = \hat{t}(t) + \tau + \mathbf{v}^\intercal\big(\hat{t}(t)\big)\, \mathbf{S}\, \left[\mathbf{e}_j - \mathbf{a}\big(\mathbf{x}_\mathrm{s}\big(\hat{t}(t)\big)\big)\, \tau\right] \tag{7.17}$$

where $\mathbf{e}_j$ is the $M \times 1$ unit vector with the jth entry set to 1 and the rest of the entries set to 0, and

$$\mathbf{a}\big(\mathbf{x}_\mathrm{s}\big(\hat{t}\big)\big) = \left[a_1\big(\mathbf{x}_\mathrm{s}\big(\hat{t}\big)\big), a_2\big(\mathbf{x}_\mathrm{s}\big(\hat{t}\big)\big), \cdots, a_M\big(\mathbf{x}_\mathrm{s}\big(\hat{t}\big)\big)\right]^\intercal \tag{7.18}$$

is an $M \times 1$ column vector of reaction propensities evaluated at $\mathbf{x}_\mathrm{s}\big(\hat{t}\big)$. The form of the update rule above in (7.17) can be deduced by examining (7.12) where we have approximated a Poisson random variable with a Gaussian one. With (7.17) above, the sample paths for the white Gaussian processes $\xi(t)$ in (7.13) (and hence the Wiener processes as their integral) are being generated as a cumulation of the individual events, i.e., reactions, that occur in the SSA simulation of the oscillator at a discrete, molecular level. In the update rule (7.17), we subtract $\mathbf{a}\big(\mathbf{x}_\mathrm{s}\big(\hat{t}(t)\big)\big)\, \tau$ from $\mathbf{e}_j$ that represents an individual reaction event in order to make the synthesized $\xi_j(t)$ zero mean. The mean, deterministic behavior of the oscillator is captured by the first drift term on the right hand side of (7.13) which is used in the computation of the periodic steady-state solution $\mathbf{x}_\mathrm{s}(t)$ and the PPV $\mathbf{v}(t)$. Thus, the mean behavior is already captured, and that is why, it needs to be subtracted in (7.17). We can now compare $\mathbf{x}_{\mathrm{s}}(\hat{t})$ and the SSA generated sample path in a one-to-one manner in order to assess the quality of $\mathbf{x}_{\mathrm{s}}(\hat{t})$. We should note here that the SSA simulation that is run in parallel and synchronous with the solution of the phase equation in (7.16) is necessary only for a meaningful sample path based comparison. One would normally not run an SSA simulation but simply generate sample paths for the Gaussian processes $\xi(t)$ and numerically solve (7.16) with an appropriate technique and generate a sample path for the phase $\hat{t}$. In this case, we would not be synthesizing $\xi(t)$ as a cumulation of reaction events from SSA, but instead directly as white Gaussian processes.

Figure 7.3 summarizes the phase equations (as opposed to the phase computation schemes, to be introduced later) approach for oscillator phase computations. An SSA sample path is generated. Then, the reaction events in the SSA sample path are recorded. This information, along with limit cycle and isochron approximations computed from the RRE, are fed into phase equations (the first-order phase equation in (7.16) has been given as an example in Figure 7.3), which in turn yield the phase $\hat{t}$.

In (7.17), we evaluate the reaction propensities at $\mathbf{x}_{\mathrm{s}}(\hat{t})$, on the solution of the system projected onto the limit cycle represented by $\mathbf{x}_{\mathrm{s}}(t)$. However, the oscillator also experiences orbital fluctuations and rarely stays on its limit cycle. Based on linear isochron approximations, we can in fact compute an approximation for the orbital fluctuations as well by solving the following equation (see Section 6.2)

$$\begin{aligned}\frac{\mathrm{d}\,\mathbf{Y}}{\mathrm{d}t} =&\mathbf{G}(\hat{t})\;\mathbf{Y} + \mathbf{S}\,\mathbb{D}\left(\left[\sqrt{a_{1:M}\left(\mathbf{x}_{\mathrm{s}}(\hat{t})\right)}\right]\right)\,\xi(t)\\ &-\left[\mathbf{v}^{\mathsf{T}}(t)\,\mathbf{S}\,\mathbb{D}\left(\left[\sqrt{a_{1:M}\left(\mathbf{x}_{\mathrm{s}}(\hat{t})\right)}\right]\right)\,\xi(t)\right]\mathbf{u}(\hat{t})\end{aligned} \tag{7.19}$$

With the orbital fluctuation computed by solving the above linear system of differential equations, we can form a better approximation for the solution of the oscillator:

$$\mathbf{X}(t) \approx \mathbf{x}_{\mathrm{s}}(\hat{t}) + \mathbf{Y}(t) \tag{7.20}$$

Then, one can evaluate the reaction propensities at $\mathbf{x}_{\mathrm{s}}(\hat{t}) + \mathbf{Y}(t)$ instead of $\mathbf{x}_{\mathrm{s}}(\hat{t})$ in (7.16), (7.17) and (7.19), in order to improve the accuracy of phase computations. One can further improve accuracy, by replacing $\mathbf{G}(\hat{t})$ in (7.19) with

$$\mathbf{G}(\hat{t}) = \left.\frac{\mathrm{d}\,\mathbf{S}\,\mathbf{a}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{s}}(\hat{t})+\mathbf{Y}(t)} \tag{7.21}$$

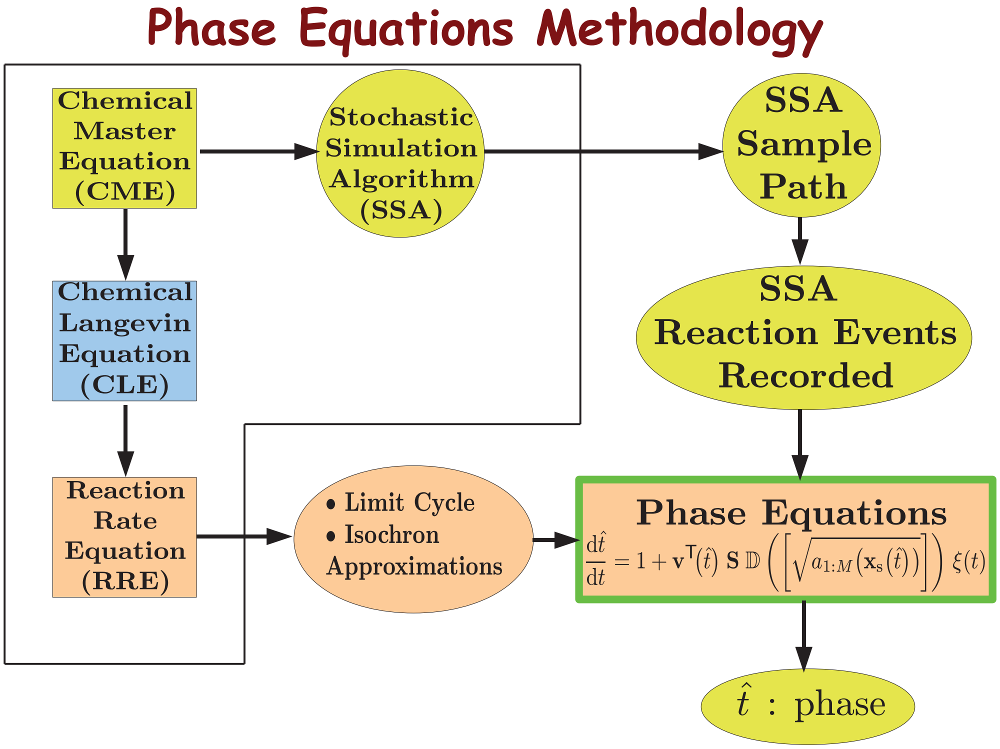

Phase Equations Methodology
Chemical Master Equation (CME)
Stochastic Simulation Algorithm (SSA)
SSA Sample Path
Chemical Langevin Equation (CLE)
SSA Reaction Events Recorded
Reaction Rate Equation (RRE)
• Limit Cycle
• Isochron
Approximations
Phase Equations
$\frac{d\hat{t}}{dt} = 1 + \mathbf{v}^{\mathsf{T}}(\hat{t})\ \mathbf{S}\ \mathbb{D}\left(\left[\sqrt{a_{1:M}(\mathbf{x}_s(\hat{t}))}\right]\right)\ \xi(t)$
$\hat{t}$: phase

Figure 7.3: Phase computations through phase equations methodology.

Still, the equations in (7.16) and (7.19) are both based on linear isochron approximations. Phase and orbital deviation equations based on quadratic approximations for isochrons will provide even better accuracy, which we discuss next.

Second-order phase equation based on quadratic isochron approximations

The second-order phase equation based on quadratic isochron approximations can be derived from the continuous Langevin model in (7.13) using the theory and numerical techniques described in Section 6.3, which takes the form

$$\begin{gathered}\frac{d\hat{t}}{dt} = 1 + \left[\mathbf{v}(\hat{t}) + \mathbf{H}(\hat{t})\ \mathbf{Y}\right]^{\mathsf{T}}\ \mathbf{S}\ \mathbb{D}\left(\left[\sqrt{a_{1:M}(\mathbf{X}(t))}\right]\right)\ \xi(t) \\ \hat{t}(0) = 0,\end{gathered} \tag{7.22}$$

with

$$\begin{aligned}\frac{\mathrm{d}\mathbf{Y}}{\mathrm{d}t} = \mathbf{G}(\hat{t})\,\mathbf{Y} + & \frac{1}{2}\frac{\partial \mathbf{G}(\hat{t})}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}\,(\mathbf{Y}\otimes\mathbf{Y}) \\ & + \mathbf{S}\,\mathbb{D}\left(\left[\sqrt{a_{1:M}(\mathbf{X}(t))}\right]\right)\,\xi(t) \\ & - \left\{\left[\mathbf{v}(\hat{t}) + \mathbf{H}(\hat{t})\,\mathbf{Y}\right]^\intercal \mathbf{S}\,\mathbb{D}\left(\left[\sqrt{a_{1:M}(\mathbf{X}(t))}\right]\right)\,\xi(t)\right\} \\ & \left[\mathbf{u}(\hat{t}) + \mathbf{G}(\hat{t})\,\mathbf{Y}\right]\end{aligned} \tag{7.23}$$

where $\mathbf{X}(t) = \mathbf{x}_\mathrm{s}(\hat{t}) + \mathbf{Y}(t)$, $\frac{\partial \mathbf{G}(\hat{t})}{\partial \mathbf{x}_\mathrm{s}(\hat{t})}$ represents an $N \times N^2$ matrix, and $\otimes$ denotes the Kronecker product making $\mathbf{Y}\otimes\mathbf{Y}$ an $N^2 \times 1$ vector.

With quadratic approximations for the isochrons of the oscillator, the phase computations based on (7.22) and (7.23) will be more accurate. We can assess the accuracy of the results obtained with these equations again by numerically solving them in synchronous fashion with an SSA simulation while synthesizing the white Gaussian processes $\xi(t)$ as a cumulation of the reaction events in SSA, as described above.

7.4.4 Phase Computation Schemes based on Langevin Models and SSA Simulations

With the phase equations based on linear and quadratic isochron approximations described in Section 7.4.3, we can compute the phase of an oscillator without having to run SSA simulations based on its discrete, molecular model. We note here again that the SSA simulations described in Section 7.2.5 were necessary only when a one-to-one comparison between the results of phase computations based on phase equations and SSA simulations was required. On the other hand, more accurate phase computations can be attained if they are based on, i.e., use information, from SSA simulations. In this hybrid scheme, we run an SSA simulation based on the discrete, molecular model of the oscillator. For points (in the state-space) on the sample path generated by the SSA simulation, we compute a corresponding phase by essentially determining the isochron on which the point in question lies. Here, one can either employ no approximations for the isochrons or perform phase computations based on linear or quadratic isochron approximations. In Section 6.4, we have established the theory for these types of approximate phase computation schemes based on linear and quadratic isochron approximations.

The brute-force phase computations without isochron approximations, which we call **PhCompBF** in short, aims to compute the phase difference between two individual given

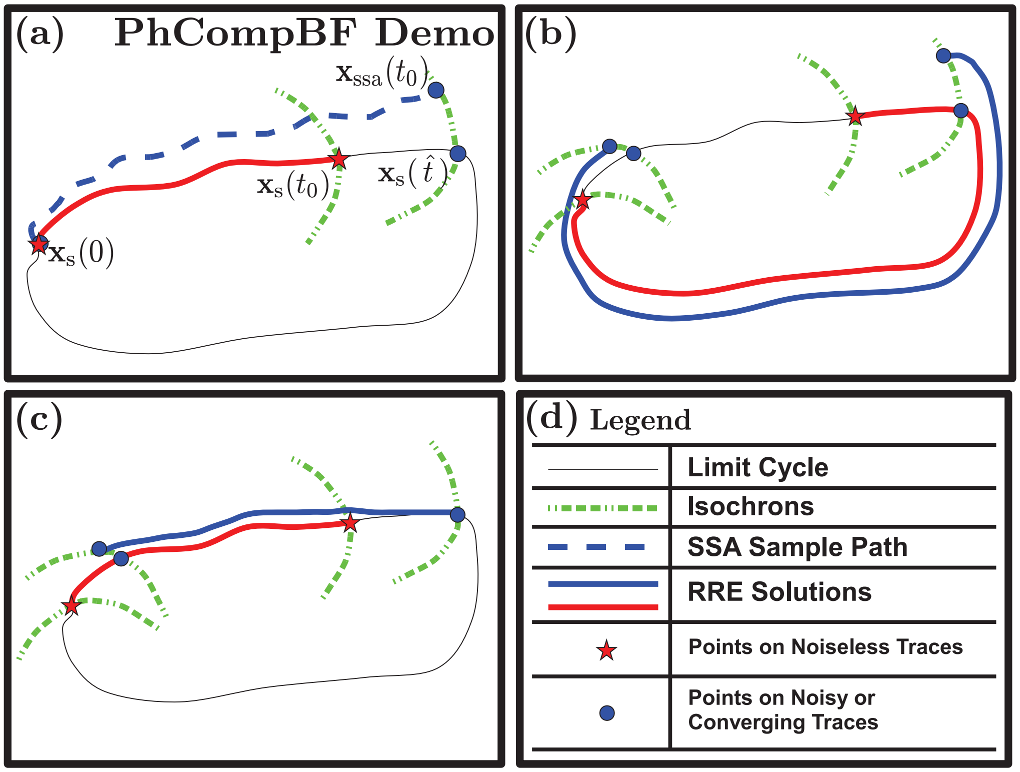

(a) PhCompBF Demo
$\mathbf{x}_{\mathrm{ssa}}(t_0)$
$\mathbf{x}_{\mathrm{s}}(t_0)$
$\mathbf{x}_{\mathrm{s}}(\hat{t})$
$\mathbf{x}_{\mathrm{s}}(0)$
(b)
(c)
(d) Legend
Limit Cycle
Isochrons
SSA Sample Path
RRE Solutions
Points on Noiseless Traces
Points on Noisy or Converging Traces

Figure 7.4: Brute-Force Phase Computation Scheme (**PhCompBF**).

points, based on the isochron-theoretic phase definition with respect to the periodic solution $\mathbf{x}_{\mathrm{s}}(t)$ tracing the limit cycle. This method is computationally costly (see Section 6.4.1), as the following explanation based on Figure 7.4 will reveal. An SSA sample path is computed and the instantanous phase of $\mathbf{x}_{\mathrm{ssa}}(t_0)$ is desired to be found. Note that t_0 is a particular value in time. For this purpose, in the transition from Figure 7.4 (a) to Figure 7.4 (b), all noise is switched off and RRE solutions (trajectories in state space) starting from $\mathbf{x}_{\mathrm{s}}(t_0)$ (star on the limit cycle) and $\mathbf{x}_{\mathrm{ssa}}(t_0)$ (circle off the limit cycle) in Figure 7.4 (a) are computed. We can compute the phase shift between these two traces only when the off-cycle solution converges as in Figure 7.4 (c), that is we will have to integrate RRE for this solution until it becomes approximately periodic in the time domain. In this plot, the illustration has been prepared such that the convergence to the limit cycle takes one period or so, but this may not always be the case. Indeed, ideally this process takes infinite time. This is why the brute-force method is costly. Eventually, the phase shift between the two trajectories can be computed and added to instantaneous time t_0, to compute the phase $\hat{t}$.

The phase computation based on isochron approximations and SSA simulations proceeds as follows (adapted from Sections 6.4.2 and 6.4.3): Let $\mathbf{x}_{\mathrm{ssa}}(t)$ be the sample path for the state vector of the oscillator that is being computed with SSA. We either solve

$$\mathbf{v}^{\intercal}\big(\hat{t}\big)\left[\mathbf{x}_{\mathrm{ssa}}(t)-\mathbf{x}_{\mathrm{s}}\big(\hat{t}\big)\right]=0 \tag{7.24}$$

based on linear isochron approximations or

$$\begin{aligned}\mathbf{v}^{\intercal}\big(\hat{t}\big)\,\left[\mathbf{x}_{\mathrm{ssa}}(t)-\mathbf{x}_{\mathrm{s}}\big(\hat{t}\big)\right]+\\ \frac{1}{2}\left[\mathbf{x}_{\mathrm{ssa}}(t)-\mathbf{x}_{\mathrm{s}}\big(\hat{t}\big)\right]^{\intercal}\mathbf{H}\big(\hat{t}\big)\,\left[\mathbf{x}_{\mathrm{ssa}}(t)-\mathbf{x}_{\mathrm{s}}\big(\hat{t}\big)\right]=0\end{aligned} \tag{7.25}$$

based on quadratic isochron approximations for the phase $\hat{t}$ that corresponds to $\mathbf{x}_{\mathrm{ssa}}(t)$. The above computation needs to be repeated for every time point t of interest. Above, for $\mathbf{x}_{\mathrm{ssa}}(t)$, we essentially determine the isochron (in fact, a linear or quadratic approximation for it) that passes through both the point $\mathbf{x}_{\mathrm{s}}\big(\hat{t}\big)$ on the limit cycle and $\mathbf{x}_{\mathrm{ssa}}(t)$. The phase of $\mathbf{x}_{\mathrm{s}}\big(\hat{t}\big)$, i.e., $\hat{t}$, is then the phase of $\mathbf{x}_{\mathrm{ssa}}(t)$ as well since they reside on the same isochron. An illustration of the linear scheme is given in Figure 7.5. In this plot, we are looking for an isochron whose linear approximation goes through $\mathbf{x}_{\mathrm{ssa}}(t_0)$, and this is the isochron of the point $\mathbf{x}_{\mathrm{s}}(\hat{t}_{\mathrm{lin}})$. Notice that the linear approximation (the straight line in Figure 7.5) is tangent to the isochron of $\mathbf{x}_{\mathrm{s}}(\hat{t}_{\mathrm{lin}})$ at exactly $\mathbf{x}_{\mathrm{s}}(\hat{t}_{\mathrm{lin}})$. $\hat{t}_{\mathrm{lin}}$ then is the phase computed by

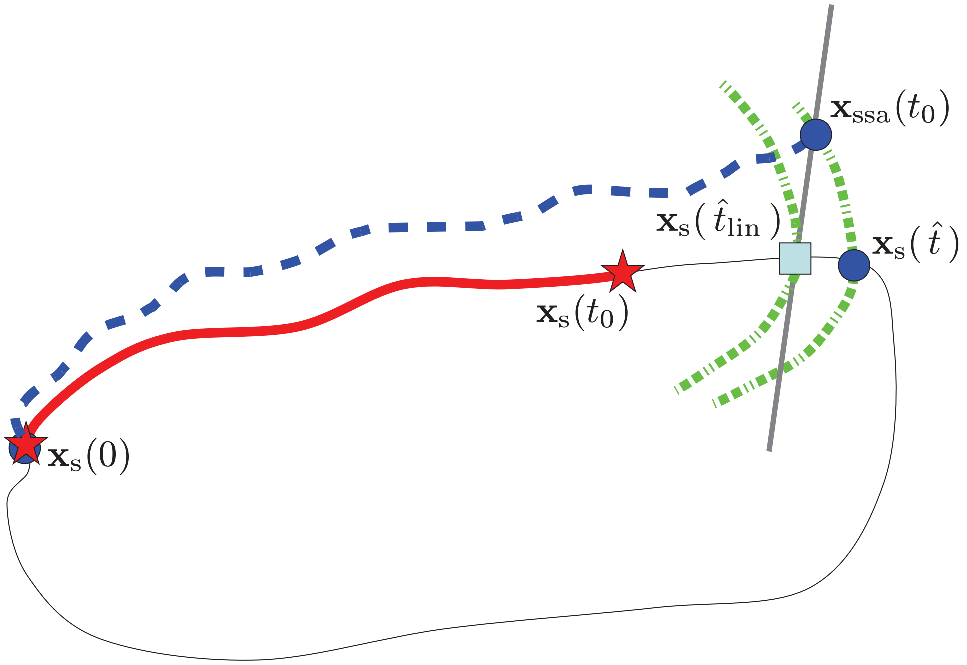

$\mathbf{x}_{\text{ssa}}(t_0)$
$\mathbf{x}_{\text{s}}(\hat{t}_{\text{lin}})$
$\mathbf{x}_{\text{s}}(\hat{t})$
$\mathbf{x}_{\text{s}}(t_0)$
$\mathbf{x}_{\text{s}}(0)$

Figure 7.5: Phase computation scheme depending on linear isochron approximations.

the linear scheme. Notice that there is some difference between the exact solution $\hat{t}$ and the approximate $\hat{t}_{\text{lin}}$. This difference is certain to shrink if generally isochrons are locally closer to being linear. For more accurate but still approximate solutions, the quadratic scheme can be used.

We should note here that, even though $\mathbf{x}_{\text{ssa}}(t)$ above is computed with an SSA simulation based on the discrete model of the oscillator, the steady-state periodic solution $\mathbf{x}_{\text{s}}(\hat{t})$, the phase gradient $\mathbf{v}(\hat{t})$ and the Hessian $\mathbf{H}(\hat{t})$ (i.e., all of the information that is used in constructing the isochron approximations) are computed based on the continuous, RRE model of the oscillator. The phase computation schemes we describe here can be regarded as *hybrid* techniques that are based both on the continuous, RRE and the discrete, molecular model of the oscillator. On the other hand, the phase computation schemes discussed in Section 7.4.3 based on phase equations are completely based on the continuous, RRE and Langevin models of the oscillator. Figure 7.6 explains the ingredients that the phase compu-

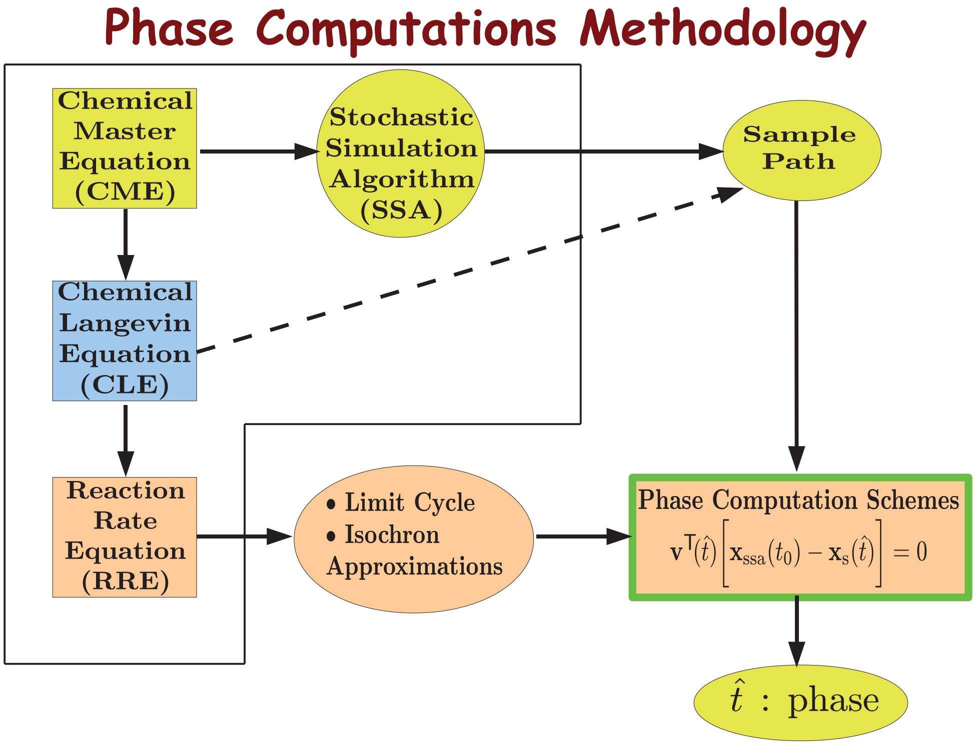

Phase Computations Methodology
Chemical Master Equation (CME)
Stochastic Simulation Algorithm (SSA)
Sample Path
Chemical Langevin Equation (CLE)
Reaction Rate Equation (RRE)
• Limit Cycle
• Isochron Approximations
Phase Computation Schemes
$\mathbf{v}^{\mathsf{T}}(\hat{t})\left[\mathbf{x}_{\text{ssa}}(t_0) - \mathbf{x}_{\text{s}}(\hat{t})\right] = 0$
$\hat{t}$: phase

Figure 7.6: Phase computation schemes methodology.

tation schemes utilize. An SSA sample path is generated (note that alternatively a sample path may be generated through the CLE). From the RRE model, limit cycle information ($\mathbf{x}_{\text{s}}(t)$) and isochron approximations ($\mathbf{v}(t)$ and $\mathbf{H}(t)$) are computed. All this information is fed into the phase computation schemes (in Figure 7.6 we have given the expression for the linear scheme for convenience, as this is the method likely to be preferred due to its lower complexity despite its inferior accuracy as compared to the quadratic scheme) and then finally the phase $\hat{t}$ is found.

7.5 Oscillator Models, Numerical Methods, and Implementation Notes

This section briefly describes where suitable oscillator models can be found particularly on the internet and how these models can be modified when possible (Section 7.5.1), how the obtained ODE models can be handled computationally (Section 7.5.2), a description of the numerical methods used in the simulations (Section 7.5.3), and the computational costs that they incur (Section 7.5.4).

7.5.1 Biochemical Oscillator Models

Oscillator models for analysis can be found from multiple resources on the web. Models generally come in two separate forms, described briefly as follows.

Models of the first type are translated directly from actual biochemical reactions. Propensities of the reactions are functions of a reaction rate parameter and appropriate algebraic expressions of molecule numbers associated with the reacting species. As such, the propensities are always positive. Moreover, the volume parameter (associated with the container or the cell accommodating the species) can easily be incorporated into the propensity functions. Volume of the cell implies the level of noisiness in the sample path simulations, i.e., basically, the more voluminous a cell, the more the number of each reacting species, and then the closer the sample path solution to the ensemble average. Therefore, one may rightfully declare that every different value for the volume parameter defines a new oscillator to be analyzed, although the mechanism of the reactions and the pattern for the propensities remain the same for a pre-determined setting.

Models of the second type are provided directly as ODE (Ordinary Differential Equation) models. In some cases, the propensity functions are difficult to handle, and it is not obvious how the crucial volume parameter can be incorporated into the equations. Then, it happens that analysis of these oscillators is a little restricted, not having the capability to adjust the level of noisiness in a correct and reliable manner. However, in all, the simulations can be carried out for the value of the volume implied by the ODE model.

As to where oscillator models can be found on the web, there are multiple alternatives. [60] is the website for a simulator, in which particularly models from [61] have been modified in appropriate form to be analyzed. We have benefitted extensively from the models we have obtained from these references, as most of them are models of the first type described above. One of the other alternatives is obtaining ODE models (models of the second type stated above) from online repositories such as [62, 63, 64] and manipulate them via appropriate software toolboxes [65, 66].

7.5.2 Information Computed from the ODE Model and SSA

Oscillator models are approximated by ODEs in the deterministic sense, through procedures already explained in the previous sections. Our purpose before handling a sample path generated by SSA is to have available in hand some crucial computational quantities that will help compute the phase along the sample path. All these crucial quantities will be computed using the ODE model. A shooting type of formulation [30] is preferred to obtain the periodic solution, more particularly a number of discrete timepoints for $\mathbf{x}_{\mathrm{s}}(t)$ along a single period. The shooting method solves this boundary value problem efficiently even for large systems of ODEs [30]. A further key benefit is that by-products of the shooting method can be utilized in solving for $\mathbf{v}(t)$, namely the PPV or the phase gradient [5]. On top of $\mathbf{x}_{\mathrm{s}}(t)$ and $\mathbf{v}(t)$ and using again the by-products of these computations, $\mathbf{H}(t)$, the phase Hessian, can be obtained through the algorithm proposed in Section 4.2.4. Now, SSA simulations for the sample paths of the noisy molecular oscillator can be performed [45], and these sample paths are analyzed in terms of phase with the following numerical methods. It should be recalled, however, that during the SSA simulation, also pieces of information have to be stored at each reaction event, conveying which reaction was chosen randomly to be simulated and what were the propensity function values at that particular instant.

7.5.3 Phase Simulations

In this section, we provide details concerning the numerical aspects of the proposed phase computation methods.

The brute-force scheme (**PhCompBF**) (described in Section 7.4.4, as adapted from Section 6.4.1) is basically run for all of the timepoints in an SSA-generated sample path, and it is very costly in terms of computation. If $\mathbf{x}_{\mathrm{ssa}}(t_0)$ is a timepoint in the sample path (naturally at where a state change takes place) the RRE is integrated with this initial condition at $t = 0$ for a long time so that this deterministic solution settles to the limit cycle in continuous time. The solution of the RRE with the initial condition $\mathbf{x}_{\mathrm{s}}(t_0)$ at t=0 can be readily computed, this is a shifted version of the periodic solution $\mathbf{x}_{\mathrm{s}}(t)$ that is available. If the phase shift between the two solutions is computed, this shift is the phase shift of the sample path $\mathbf{x}_{\mathrm{ssa}}$ at $t = t_0$ (see Section 6.4.1). Since one generally does not know the phase value at the very first timepoint of an SSA sample path, the brute-force scheme is mandatory in computing this phase value and providing the initial condition, on which all of the other approximate phase computation schemes and equations can operate.

The approximate phase computation schemes (again described in Section 7.4.4, as adapted from Sections 6.4.2 and 6.4.3) consist of solving the algebraic equation in (7.24) or (7.25), depending on whether linear or quadratic approximations are respectively preferred to be used, and they are also run for all points in the SSA sample path. Benefitting from the scalar nature of these equations, the bisection method is used extensively in their numerical solution. Details and subtleties involved with these schemes (of considerably less computational load compared to **PhCompBF**) are provided in Section 7.4.4.

Phase equations, described in Section 7.4.3 (as adapted from Sections 6.2 and 6.3), are in this context stochastic differential equations, operating on the recorded reaction events of an SSA sample path. The specific discretization scheme applied to the first order phase equation is explained in detail in Section 7.4.3. This discretization scheme can be easily extended to the second order phase equation of Section 7.4.3.

We will denote each method analyzed and used in generating results by some abbreviations, for ease of reference. The brute-force scheme explained above is denoted by **PhCompBF**, the scheme depending on linear isochron approximations (summarized by (7.24)) by **PhCompLin**, and that depending on quadratic in (7.25) by **PhCompQuad**. The first order phase equation of (7.16) is denoted by **PhEqnLL** (the first L for linear isochron approximations and the second L for linear orbital deviation approximations). The second order phase equation of (7.22) and (7.23) is denoted by **PhEqnQQ** (Q for both type of approximations, isochron and orbital deviation). We prefer to use instead of **PhEqnQQ** a simpler, but numerically more reliable, version of the second order equation. This simpler version is described by the equations (7.22) and (7.19). (7.19) is the orbital deviation equation belonging to the first order phase equation theory. In turn, we denote this simpler model by **PhEqnQL**.

7.5.4 Analysis of Computational Complexities

In this section, we analyze the computational costs of phase computation schemes and phase equations. Note that a more detailed explanation including the derivation of the computational complexities in this section is given in Appendix Section B.2. Most of the complexities in this section are different compared to those in Section 6.5.2, since we are using different algorithms in implementation, with the exception of **PhCompBF**.

Let us denote by N the number of states in an oscillator, K the number of timepoints along a single period, L the number of total timepoints along the interval where a phase computation method is run.

Preliminary statements on computational complexities are as follows. We assume as well-known complexities that $\mathbf{x}_\mathrm{s}(t)$, $\mathbf{G}(t)$ (assumed to be sparse), $\mathbf{u}(t)$ and $\mathbf{v}(t)$ are computable along a single period in $\mathcal{O}(N\,K)$ time. The computation of $\mathbf{H}(t)$ (which is usually not sparse) upon the stated quantities takes $\mathcal{O}(N^3\,K)$ time. We assume that if a matrix is sparse, then matrix vector multiplications and solving a linear system of equations involving this matrix can be done in linear time.

For **PhCompBF**, in order to compute the phase of a point $\mathbf{x}_\mathrm{ssa}(t_0)$, we have to integrate the RRE with initial condition $\mathbf{x}_\mathrm{ssa}(t_0)$ for an ideally infinite number, namely n_per, of periods, so that the states vector can be assumed more or less to be tracing the limit cycle. If FFT (Fast Fourier Transform) properties are used to compute the phase shift between periodic waveforms, the overall complexity of PhCompBF can be shown to amount to $\mathcal{O}(n_\mathrm{per}\,K\,N\,L + L\,K\log_2\,K)$.

The approximate phase computation schemes consist of solving the algebraic equations in (7.24) or (7.25) (depending on whether the linear or quadratic scheme is preferred). The bisections method is used to solve these equations. In order to compute the phase value of a particular timepoint, an interval has to be formed. In forming such an interval, we start with an interval, of length d_min and centered around the phase value of the previous timepoint, and double this length value until the interval is certain to contain the phase solution. The allowed maximum interval length is denoted by d_max. Then, the bisections scheme starts to chop down the interval until a tolerance value d_tol for the interval length is reached. The **PhCompLin** computational complexity can be shown to be

$$\mathcal{O}\left(\;N\;L\;\log_2\left\lceil\frac{d_\mathrm{max}^2}{d_\mathrm{tol}\,d_\mathrm{min}}\right\rceil\;\right) \tag{7.26}$$

and **PhCompQuad** complexity is

$$\mathcal{O}\left(\;N^2\;L\;\log_2\left\lceil\frac{d_\mathrm{max}^2}{d_\mathrm{tol}\,d_\mathrm{min}}\right\rceil\;\right) \tag{7.27}$$

based on the explanations above.

Table 7.1: Computational complexities for the phase computation schemes.

Scheme	Computational Complexity
PhCompBF	$\mathcal{O}(n_{\text{per}}\, K\, N\, L + L\, K \log_2\, K)$
PhCompQuad	$\mathcal{O}\left(\quad N^2\; L\; \log_2 \left\lceil \frac{d_{\max}^2}{d_{\text{tol}}\, d_{\min}} \right\rceil \quad\right)$
PhCompLin	$\mathcal{O}\left(\quad N\; L\; \log_2 \left\lceil \frac{d_{\max}^2}{d_{\text{tol}}\, d_{\min}} \right\rceil \quad\right)$

Table 7.2: Computational complexities for the phase equations.

Equation	Complexity (best)	Complexity (worst)
PhEqnLL	$\mathcal{O}(M\, L + N\, L)$	$\mathcal{O}(N\, M\, L)$
PhEqnQL	$\mathcal{O}(N^2\, L + M\, L)$	$\mathcal{O}(N^2\, L + N\, M\, L)$

The computational complexity expressions for all of the phase computation schemes are summarized in Table 7.1.

Phase equation solution complexities depend (in extreme conditions) mainly on the stoichiometric matrix **S** being sparse (few nonzero entries per row) or totally dense. Note that in realistic problems **S** is observed to be usually sparse. These stated respective conditions lead us to come up with best and worst case complexities. As such, **PhEqnLL** complexity in the best and worse case can be shown to be $\mathcal{O}(M\, L + N\, L)$ and $\mathcal{O}(N\, M\, L)$, respectively. **PhEqnQL** complexities are $\mathcal{O}(N^2\, L + M\, L)$ (best case) and $\mathcal{O}(N^2\, L + N\, M\, L)$ (worst case). Complexities for the phase equations are summarized in Table 7.2.

The essence of the above analyses is that there is a trade-off between accuracy and computational complexity. For mildly noisy oscillators, the phase equations should remain somewhat close to the results of the golden reference **PhCompBF** and the other approximate phase computation schemes, which imitate **PhCompBF** very successfully with much less computation times. For more noisy oscillators, we should expect the phase computation schemes to do still well, although the phase equations will compute some inaccurate results very fast. **PhCompBF** is always very slow.

7.6 Results

We now present results obtained with the proposed methods for oscillator phase computations on several intricate molecular oscillators. Accuracy demonstrations and computational speed-up figures will be given with respect to **PhCompBF**, the brute-force scheme, which we accept as the golden reference for oscillator phase computations. Section 7.6.1 below, in which we analyze the brusselator, contains details pertaining to the general flow of the phase computations and the preparatory procedures for all the methods. Section 7.6.2 and Section 7.6.3 are brief sections illustrating the performance of the methods for oscillators called the oregonator and the repressilator, respectively. All simulations were run on a computer with an Intel i7 processor at 3.07 GHz and accommodating 6 GB of memory.

7.6.1 Brusselator

The Brusselator is a theoretical model for a type of autocatalytic reaction. The Brusselator actually describes a type of chemical clock, and the Belousov-Zhabotinsky (BZ in short) reaction is a typical example [61]. The model below in (7.28) has been largely adapted from [60], which is based on [61].

$$\begin{aligned} \mathsf{A} &\overset{k_1}{\to} \mathsf{X} \\ \mathsf{B}+\mathsf{X} &\overset{k_2}{\to} \mathsf{R}+\mathsf{Y} \\ \mathsf{Y}+2\mathsf{X} &\overset{k_3}{\to} 3\mathsf{X} \\ \mathsf{X} &\overset{k_4}{\to} \mathsf{S} \end{aligned} \tag{7.28}$$

Parameter values in (7.28) are: $k_1 = 0.025\,\mathrm{s}^{-1}$, $k_2 = 1\,\mathrm{s}^{-1}\,\mathrm{mL}$, $k_3 = 1\,\mathrm{s}^{-1}\,(\mathrm{mL})^2$, and $k_4 = 0.01\,\mathrm{s}^{-1}$. Volume is set to 250 mL. Molecule numbers of A, B, R, and S are held constant.

Several models and quantities must be derived from the reactions in (7.28) before moving onto phase analysis. The stoichiometric matrix in this case reads

$$\mathbf{S} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \tag{7.29}$$

where the first row is for the species X and the second is for Y. The columns each denote the changes in molecule numbers as a reaction takes place, e.g., column one is for the first reaction in (7.28). Let us also call X the random process denoting the instantaneous molecule number for the species X, similarly Y is for Y in the same fashion. Then, the random process vector $\mathbf{X} = [\, X\, Y\,]^{\intercal}$ concatenates these numbers for convenience. The propensity functions for the reactions can be written as

$$\begin{aligned}
a_1(\mathbf{X}) &= k_1\, A \\
a_2(\mathbf{X}) &= \frac{k_2\, B\, X}{\Omega} \\
a_3(\mathbf{X}) &= \frac{k_3\, Y\, X\, (X-1)}{\Omega^2} \\
a_4(\mathbf{X}) &= k_4\, X
\end{aligned} \tag{7.30}$$

where Ω denotes the volume parameter. Using (7.30), the CME for the Brusselator can be derived in line with (7.4) as

$$\begin{aligned}
\frac{\mathrm{d}\mathbb{P}\,(X, Y; t)}{\mathrm{d}t} = -\, & \Bigg[k_1\, A + \frac{k_2\, B\, X}{\Omega} \\
& + \frac{k_3\, Y\, X\, (X-1)}{\Omega^2} + k_4\, X \Bigg]\, \mathbb{P}\,(X, Y; t) \\
& + k_1\, A\, \mathbb{P}\,(X-1, Y; t) \\
& + \frac{k_2\, B\, (X+1)}{\Omega}\, \mathbb{P}\,(X+1, Y-1; t) \\
& + \frac{k_3\, (Y+1)\, (X-1)\, (X-2)}{\Omega^2} \\
& \mathbb{P}\,(X-1, Y+1; t) \\
& + k_4\, (X+1)\, \mathbb{P}\,(X+1, Y; t)
\end{aligned} \tag{7.31}$$

Now it is possible to derive the CLE as in (7.13)

$$\begin{aligned}\frac{\mathrm{d}X}{\mathrm{d}t} =& \Bigg[k_1\,A - \frac{k_2\,B\,X}{\Omega} \\ &+ \frac{k_3\,Y\,X\,(X-1)}{\Omega^2} - k_4\,X\Bigg] \\ &+ \Bigg[\sqrt{k_1\,A}\,\xi_1(t) - \sqrt{\frac{k_2\,B\,X}{\Omega}}\,\xi_2(t) \\ &+ \sqrt{\frac{k_3\,Y\,X\,(X-1)}{\Omega^2}}\,\xi_3(t) \\ &- \sqrt{k_4\,X}\,\xi_4(t)\Bigg] \qquad (7.32)\\ \frac{\mathrm{d}Y}{\mathrm{d}t} =& \Bigg[\frac{k_2\,B\,X}{\Omega} \\ &- \frac{k_3\,Y\,X\,(X-1)}{\Omega^2}\Bigg] \\ &+ \Bigg[\sqrt{\frac{k_2\,B\,X}{\Omega}}\,\xi_2(t) \\ &- \sqrt{\frac{k_3\,Y\,X\,(X-1)}{\Omega^2}}\,\xi_3(t)\Bigg]\end{aligned}$$

It is easy to extract from (7.32) the RRE in (7.9) as

$$\begin{aligned}\frac{\mathrm{d}X}{\mathrm{d}t} =& \Bigg[k_1\,A - \frac{k_2\,B\,X}{\Omega} \\ &+ \frac{k_3\,Y\,X\,(X-1)}{\Omega^2} - k_4\,X\Bigg] \\ \frac{\mathrm{d}Y}{\mathrm{d}t} =& \Bigg[\frac{k_2\,B\,X}{\Omega} \\ &- \frac{k_3\,Y\,X\,(X-1)}{\Omega^2}\Bigg]\end{aligned} \qquad (7.33)$$

In preparation for phase analysis, some computational quantities have to be derived from (7.33).

The phase analysis of a continuous oscillator (modeled by nonlinear systems of ODEs such as an RRE) depends on linearizations around the steady-state periodic waveform $\mathbf{x}_\mathrm{s}(t)$ solving the RRE. $\mathbf{x}_\mathrm{s}(t)$ for the Brusselator in (7.28) is given in Figure 7.7. $\mathbf{x}_\mathrm{s}(t)$ has been computed for a whole period (with the actual approximate value for the period $T = 1000\,\mathrm{s}$) through the shooting method [30]. The species A, B, R, and S, with their molecule numbers constant, should be excluded from the machinery of the shooting method for it to work.

In fact, $\mathbf{x}_\mathrm{s}(t)$ computation is enough preparation for running the brute-force scheme **PhCompBF** as will be demonstrated next. Recalling that we aim to solve for the possibly

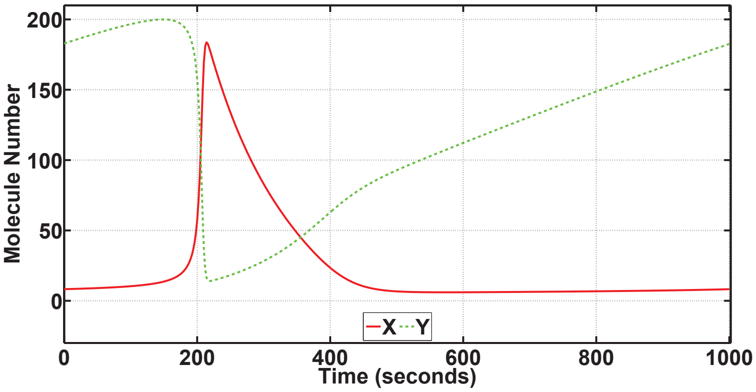


Figure 7.7: $\mathbf{x}_s(t)$ for the Brusselator.

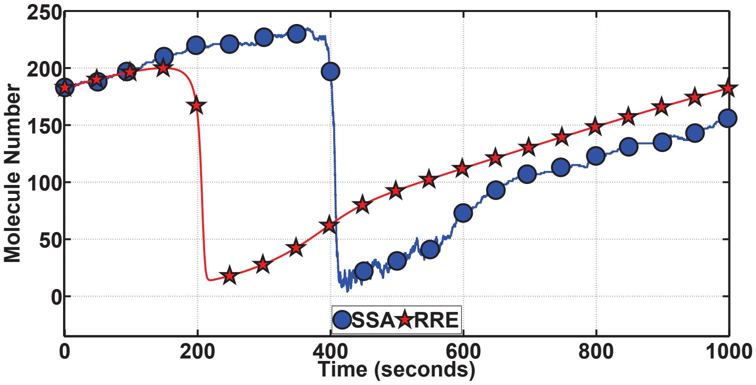


Figure 7.8: An SSA-generated sample path (species Y) for the Brusselator.

constantly changing phase along individual SSA-generated sample paths, we run the SSA algorithm to generate the sample path given in Figure 7.8. In this plot, the SSA simulation result and the unperturbed $\mathbf{x}_s(t)$ have been plotted on top of each other, for only species Y, for illustration purposes. It must be noted that both $\mathbf{x}_s(t)$ and the SSA sample path start initially at the same state on the limit cycle, therefore the star and the circle are on top of each other at $t = 0$ s. Due to isochron-theoretic oscillator phase theory, the initial relative phase, or the initial phase shift of the SSA sample path with respect to $\mathbf{x}_s(t)$ is zero.

In Figure 7.8, we would like to solve eventually for the time-evolving relative phase shift of the SSA sample path, for now with **PhCompBF**. This means solving for the phase shift for the visited states in the sample path, denoted by circles in the figure, and preferably for all the states in between the circles along the path as well. **PhCompBF** requires running a particular type of simulation for computing the relative phase shift of each visited state. We will demonstrate the method shortly, but let us comment on how much information can be gained by inspecting only the plot in Figure 7.8. The SSA simulation suggests that the system continually introduces noise, so that everything about the system appears noisy, the phase, the amplitude, etc. Phase is a particular quantity that helps quantify the effect of noise on an autonomously oscillating system. One may easily guess that the relative phase shift of the SSA sample path is always changing along the interval of simulation. It is not obvious at all how to compute this phase shift at particular points in time in Figure 7.8. Perhaps, one may argue that the sudden decrease that should take place at about $t = 200\,\mathrm{s}$ for the unperturbed $\mathbf{x}_{\mathrm{s}}(t)$, appears about 200 s in time later for the SSA path. However, this is only an educated guess and an approximate value. Also, that the stars and circles appear very close to each other for example in between 600 s and 1000 s does not directly help invoke the isochron-theoretic phase theory to deduce that the phase shift along this interval is close to zero. Recalling that Figure 7.8 depicts only species Y, one has to inspect also the other species to arrive at such a conclusion. It is also needless to state as a reminder that for two states to have the same relative phase, having the two states equal to each other is a sufficient but not necessary condition, again due to isochron theory. In all, accurately what happens to the phase shift along the interval is still obscure. As a side note, one should also note that without the perfectly periodic $\mathbf{x}_{\mathrm{s}}(t)$, it is awfully difficult to guess the period T, inspecting only a long SSA sample path. Relevant theory for noisy oscillators suggests that inspecting the zero-crossings of a whole ensemble of long and mildly noisy SSA sample paths yields information related to the period and phase diffusion constant of an oscillator, in a brute-force manner [5].

In order to demostrate **PhCompBF**, we have first plotted both the SSA sample path and the limit cycle (the closed curve traced over and over by $\mathbf{x}_{\mathrm{s}}(t)$) in 2-D state space as in Figure 7.9. As stated earlier, the star and the circle are initially coincident. Then, as time progresses, $\mathbf{x}_{\mathrm{s}}(t)$ just traces the limit cycle, but the SSA sample path $\mathbf{x}_{\mathrm{ssa}}(t)$ runs berserk. At $t_0 = 600\,\mathrm{s}$, we have again indicated where the two traces end up. The SSA path at this time is off the limit cycle. Since we do not have exact isochron information, it is not possible to compute the phase $\hat{t}$ value that makes $\mathbf{x}_{\mathrm{ssa}}(t_0 = 600\,\mathrm{s})$ and $\mathbf{x}_{\mathrm{s}}(\hat{t})$ in-phase, i.e., on the same isochron. If we could find this $\hat{t}$ value, then $\alpha(t_0 = 600\,\mathrm{s}) = \hat{t} - 600$ would be the

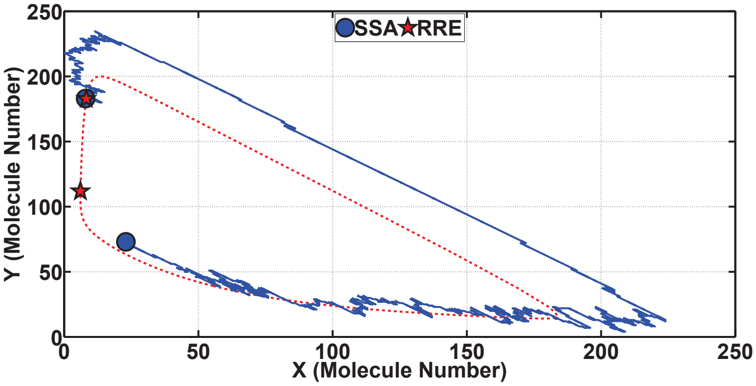


Figure 7.9: Limit cycle and SSA sample path for the Brusselator.

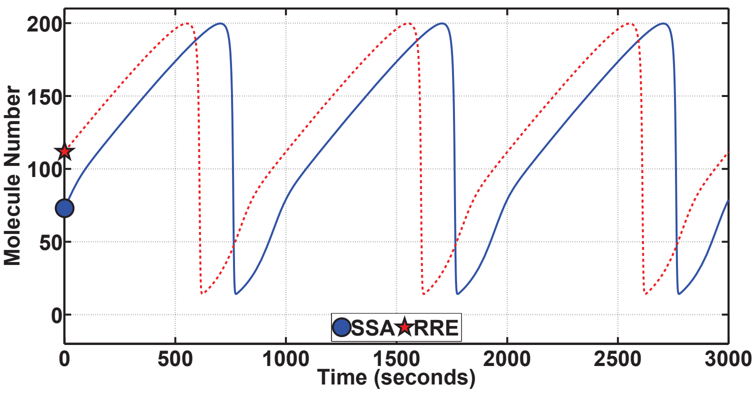


Figure 7.10: **PhCompBF** for the Brusselator.

sought phase shift value.

The value of the phase shift α can, however, be computed through a possibly long, ideally infinitely long, simulation, in line with the theory of asymptotic phase (a theory on intimate terms with isochrons). The following is the essence of **PhCompBF**. One takes in Figure 7.9 the states $\mathbf{x}_{\mathrm{ssa}}(t_0 = 600\,\mathrm{s})$ (the circle on the SSA path) and $\mathbf{x}_{\mathrm{s}}(t_0 = 600\,\mathrm{s})$ (the star on the limit cycle) and feeds them as initial conditions to the RRE in (7.9) and then simulates both traces for some time. The result is the two traces in Figure 7.10. In this plot, again only the species Y is demonstrated. The circular marker (along with the corresponding star) has been put only at the beginning of the simulation in Figure 7.10 to note the fact that only the initial value belongs to the SSA sample path. After this initial time, both traces are part of separate RRE solutions. Incorporation of these two new simulated traces into the plot of Figure 7.9

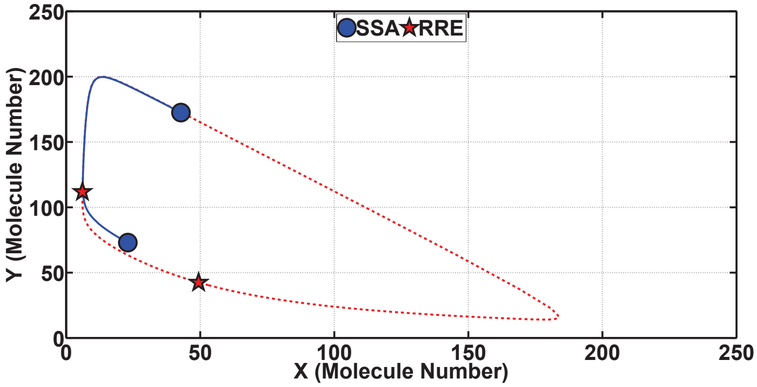


Figure 7.11: **PhCompBF** in state space for the Brusselator.

would be as follows (see Figure 7.11): The plot starting with the circle in Figure 7.10 (with both of the two states) would be a curve in the state space of Figure 7.9 starting from the circle off the limit cycle but gradually converging to it. Meanwhile, the plot starting from the star in Figure 7.10 would resume tracing the limit cycle in Figure 7.9 from again the star. Then, as shown in Figure 7.11, the two simulated plots are observed to be tracing the limit cycle after simulating long enough in time, the star of the unperturbed path always leading the circle of the initially perturbed path (but notice that during the simulation for both traces in Figure 7.11 all perturbations or noise are removed). Observe in Figure 7.11 that the star has went ahead to make the rightmost turn on the limit cycle, travelling clockwise, whereas the circle is still way behind. However, all along this simulation of Figure 7.11, the instantaneous phase shift between the two traces has remained the same. As the simulation goes on along the limit cycle, the circle (originating from the SSA simulation) and the star (of the unperturbed $\mathbf{x}_s(t)$) would appear sometimes near, and sometimes far away from each other. This effect is due to particularly the varying velocity along the limit cycle, all determined by the dynamic properties of the RRE. The constant difference in time between the circle and star is the phase shift $\alpha(t_0 = 600\,\mathrm{s})$ that we aim to compute. Notice that in the state space of Figure 7.9 and Figure 7.11, time is only an implicit parameter. Therefore, we have to inspect plots of the type in Figure 7.10 to obtain the desired phase shift value.

For some oscillators (as determined by the dynamics of the RRE again), a state off the limit cycle converges fast to begin tracing quickly an almost periodic curve, as in the case in hand. Almost two periods is enough to deduce the phase shift between the two curves.

After RRE simulations, the phase shift can be computed using Fourier transforms.

One question that may arise is why we are particularly using the traces belonging to the species Y to compute phase shifts in Figure 7.10. Indeed, it follows from the theory that phase is a scalar-valued property of the whole system, therefore investigating phase shifts over non-constant periodic molecule numbers for any species in a system would yield the same phase shift value. In this case, employing Y is only a matter of choice.

Notice that this brute-force scheme is carried out to compute the relative phase shift of the SSA sample path at only $t_0 = 600\,\mathrm{s}$. The phase shift for each state along the sample path can be computed one by one through the just outlined **PhCompBF**.

It has already been stated that **PhCompBF** is almost the golden reference for phase computations but also that the method is very time-consuming. It was for this reason that new methods depending on isochron and orbital deviation approximations were proposed. Particularly, two quantities are necessary for characterizing isochron approximations: the phase gradient $\mathbf{v}(t)$ and the phase Hessian $\mathbf{H}(t)$. These are depicted for the Brusselator respectively in Figure 7.12 and Figure 7.13. Recall that $\mathbf{v}(t)$ is a vector function, but $\mathbf{H}(t)$ is a matrix function. Therefore, only the phase Hessian diagonals have been plotted in Figure 7.13.

Phase computation schemes are fairly easy to comprehend geometrically. Regarding for example the limit cycle depicted in Figure 7.9, there are both a hyperplane (accounting for the linear isochron approximation) and a quadric surface (for quadratic approximation) associated with each point on the limit cycle. Equations for these characterizations are given in (7.24) and (7.25), respectively. A phase computation scheme aims to solve for that point on the limit cycle whose linear or quadratic isochron approximation passes through a given point, for example the stated point denoted by the circle off the limit cycle in Figure 7.9, $\mathbf{x}_{\mathrm{ssa}}(t_0 = 600\,\mathrm{s})$. Notice that **PhCompBF** is also a variant of these phase computation schemes, but in this case not the isochron approximations but the exact isochrons themselves associated with points on the limit cycle are used.

The geometrical interpretations of phase equations, on the other hand, are not easy to visualize. As stated in previous sections, phase equations are differential equations involving orbital deviation in addition to isochron approximations. Phase computation schemes are expected to be more costly but then more accurate with respect to phase equations. Phase

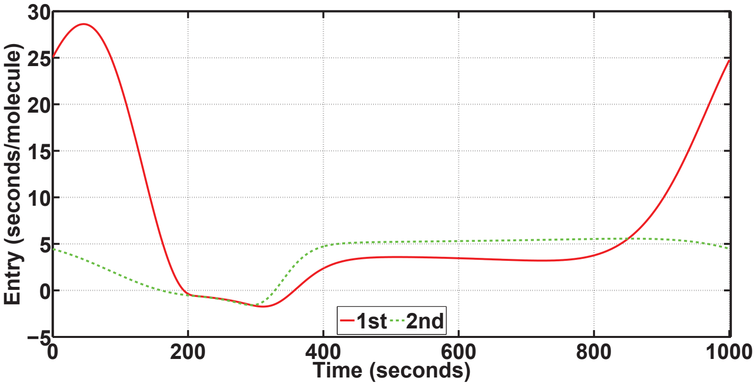


Figure 7.12: Phase Gradient for the Brusselator.

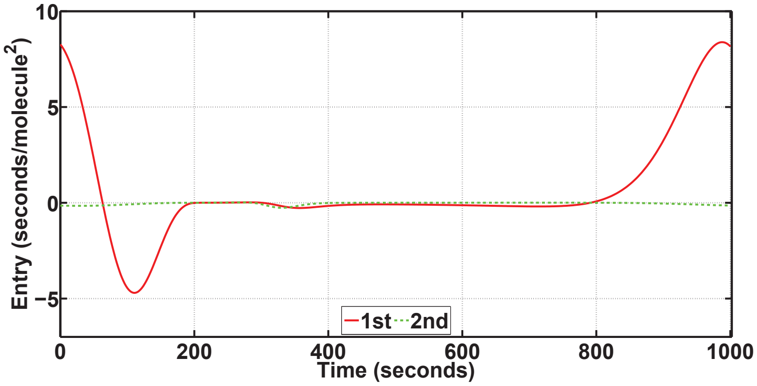


Figure 7.13: Phase Hessian Diagonals for the Brusselator.

equations, as they are differential equations and need to be discretized, suffer from local truncation errors and global errors, whereas this is not the case for the schemes that are in the form of algebraic equations. An approximate phase computation scheme may deviate from the golden reference (**PhCompBF** result) at times (particularly if the noisy state is too far off the limit cycle), but the scheme (if carefully designed) does not suffer from accumulation of global errors and its phase results are expected to be almost always very close to that of **PhCompBF**.

Computational complexity-wise the phase equations are indeed very fast. This makes the phase equations a feasible and accurate choice for the phase computations of less noisy oscillators, possibly with a dense grid of timepoints in an SSA sample path and high molecule numbers for every species in the system (especially in a container of large volume), deviating not much from their limit cycles.

The numerical procedures associated with the schemes render them more costly in computational complexity with respect to the equations. Therefore, one may rightfully contend that the phase computation schemes are tailored to fit phase computations for moderately noisy oscillators in small volume, with low molecule numbers for each species and possibly a sparse grid of timepoints in an SSA sample path.

We now check the performance of the phase computation methods for this oscillator, on a sample path that lasts about 1000 s, with the period about the same as that. The results are depicted in Figure 7.14. **PhCompBF** takes about 138 minutes. Speed-up of the methods on this duration are as follows: **PhCompLin** 56x, **PhEqnLL** 8583x, and **PhEqnQL** 2257x. The phase equations are most of the time sharing a common accuracy level, not disregarding the apparent attempt of **PhEqnQL** to come closer to **PhCompBF** around 400-600 s. **PhCompLin** is slower than the equations but almost as accurate as can be.

7.6.2 Oregonator

In this section, we present phase computation results for a well-known and studied biochemical oscillator, the oregonator [61]. This realistic oscillator accurately models the Belousov-Zhabotinsky reaction, an autocatalytic reaction that serves as a classical example of non-equilibrium thermodynamics. The molecular reactions model, adapted mostly from [60], is

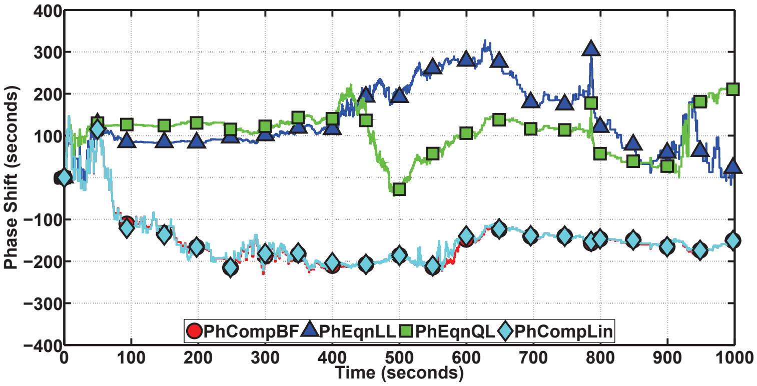


Figure 7.14: Phase computation methods on the Brusselator.

given as follows. Names of the reactants have been simplified for convenience.

$$\begin{aligned} \mathsf{A}+\mathsf{Y} &\overset{k_1}{\to} \mathsf{X}+\mathsf{R} \\ \mathsf{X}+\mathsf{Y} &\overset{k_2}{\to} 2\,\mathsf{R} \\ \mathsf{A}+\mathsf{X} &\overset{k_3}{\to} 2\,\mathsf{X}+2\,\mathsf{Z} \\ 2\,\mathsf{X} &\overset{k_4}{\to} \mathsf{A}+\mathsf{R} \\ 2\,\mathsf{B}+2\,\mathsf{Z} &\overset{k_5}{\to} \mathsf{Y} \end{aligned} \tag{7.34}$$

In (7.34), the propensity functions, employing also the volume of the container, can easily be derived. Parameter values are: $k_1 = 0.005\,\mathrm{s}^{-1}\,\mathrm{mL}$, $k_2 = k_3 = k_4 = 1\,\mathrm{s}^{-1}\,\mathrm{mL}$, and $k_5 = 1.25 \times 10^{-4}\,\mathrm{s}^{-1}\,(\mathrm{mL})^3$. Molecule numbers for the reactants A, B, and R are held constant. For this model, the volume initially is set to 12,000 mL. In this case, noise will not have considerable effect on a sample path. Then, we set the volume to 3,200 mL in order to obtain a moderately noisy oscillator. Later on, we will halve the value of the volume parameter, resulting in a very noisy oscillator, and the performance of the phase computation methods will be demonstrated for this latter case as well.

With the volume as 12000 mL, the performance of the phase computation methods on a particular sample path of length 4×10^4 s (the period is about 4.43×10^4 s) is depicted in Figure 7.15. **PhCompBF** simulation takes 502 minutes, with two periods of RRE computations before setting out to compute the phase shift values. There are a total of 8114 timepoints on the sample path. As the volume is decreased, the number of timepoints per unit time will reduce. The speed-up of the methods over **PhCompBF** are: **PhCompLin**

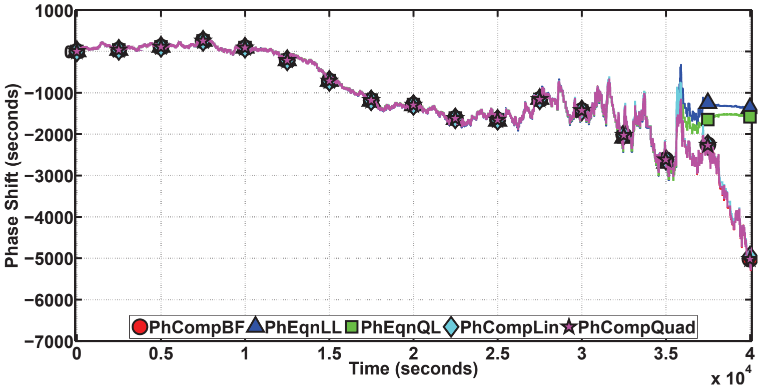


Figure 7.15: Phase computation methods on the Oregonator (volume=12000 mL).

70x, **PhEqnLL** 10733x, **PhCompQuad** 46x, and **PhEqnQL** 2791x. It is observed that all the methods for a good part of the sample path stick to the **PhCompBF** result. However, towards the end the phase equations (with **PhEqnQL** a little more accurate compared to **PhEqnLL**) begin accumulating global errors, Otherwise, they are exquisitely fast all the time and accurate at the beginning until they start deviating from the golden reference. The phase computation schemes are not as fast as the equations, but they are always accurate in this simulation.

We have also tested the phase computation methods on a sample path, with the volume set to 3200 mL. Figure 7.16 illustrates the results. The simulation interval length (5×10^4 s) is a little more than the period (about 4.37×10^4 s). The simulation for **PhCompBF** took 242 minutes, and there are 2981 timepoints in total. The observed speed-ups were: **PhCompLin** 70x, **PhEqnLL** 13971x, **PhCompQuad** 51x, and **PhEqnQL** 3203x. It is observed that the phase equations are really fast, keeping track of the exact phase though not very closely, whereas the computation schemes, though not as fast, are almost a perfect match for the exact phase in terms of accuracy.

We then set volume to 1600 mL, resulting in a noisier oscillator. We expect the phase equations results to deviate much more from the exact one, and the computation schemes to still do well. Again for a sample path (of length 5×10^4 s with the period 4.3×10^4 s), the **PhCompBF** simulation now takes 76 minutes. There are 1033 timepoints. Speed-ups with the methods are: 12637x (**PhEqnLL**), 74x (**PhCompLin**), and 44x (**PhCompQuad**). **PhEqnQL** apparently suffers from numerical problems for such a noisy oscillator, and the

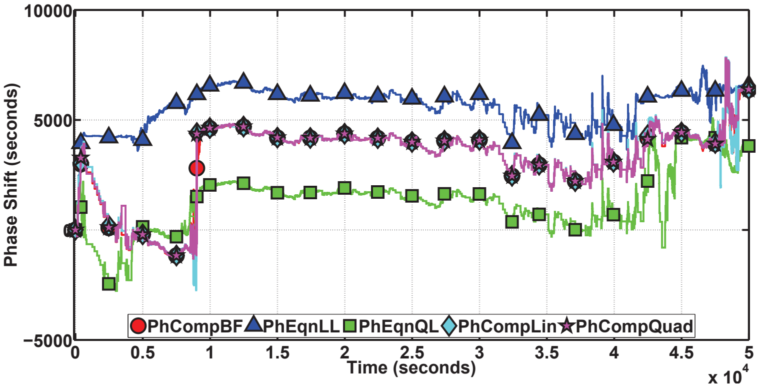


Figure 7.16: Phase computation methods on the Oregonator (volume=3200 mL).

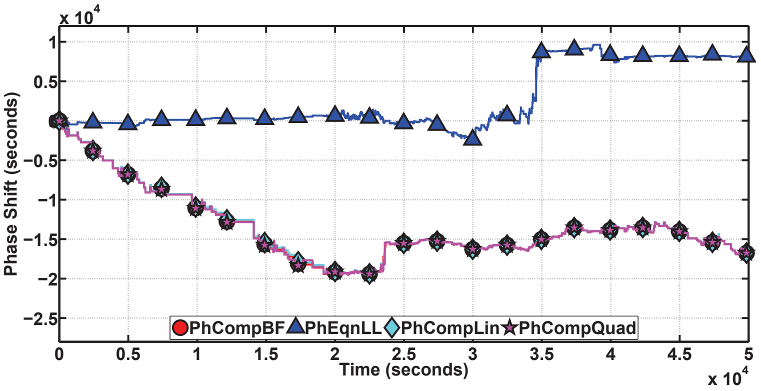


Figure 7.17: Phase computation methods on the Oregonator (volume=1600 mL).

result for this method is not included. In Figure 7.17, we observe in line with our expectations that although **PhEqnLL** is again very fast, the result it produces is almost unacceptably inaccurate, whereas both the computation schemes maintain their relative speed-ups (as compared to the less noisy version) along with their accuracies.

7.6.3 Repressilator

The Repressilator is a synthetic genetic regulatory network, designed from scratch and implemented in Escherichia coli using standard molecular biology methods [43]. Its development is a milestone in synthetic biology. We have obtained the model as an SBML file in XML format [62, 63, 64]. We have used the libSBML [65] and SBMLToolbox [66] libraries to interpret the model and incorporate it to our own manipulation and simulation toolbox for

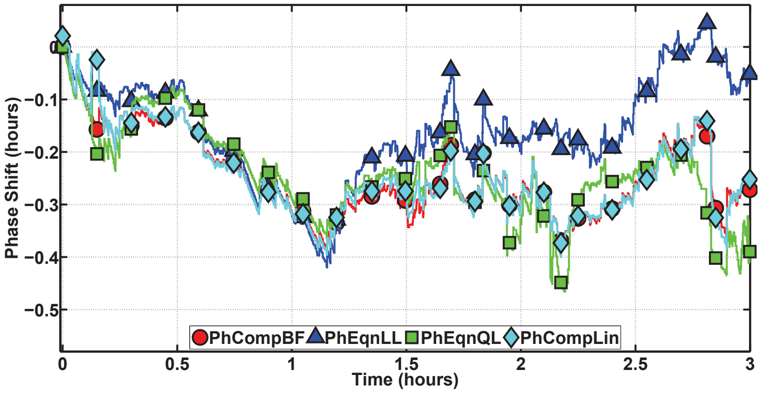


Figure 7.18: Phase computation methods on the Repressilator.

phase computations. The period of the continuous oscillator obtained from the model is about 2.57 h. A sample path running for about 3 h was generated, and the phase methods were applied. The results are in Figure 7.18. **PhCompBF** takes about 76 minutes. Speed-ups obtained with the methods are: **PhCompLin** 58x, **PhEqnLL** 7601x, and **PhEqnQL** 1994x. It appears in Figure 7.18 that **PhEqnLL** towards the end of the simulation has started to accumulate a global error. **PhEqnQL** looks a little more accurate. Again **PhCompLin** is, excepting a few minor intervals, the most accurate.

7.7 Conclusions and future work

The phase computation methods described in this chapter basically target three classes of discrete molecular oscillators. First, the continuous phase models, based on the information obtained from the oscillator model in the continuous-state limit (i.e., basically the limit cycle and isochron approximations), are acceptably accurate for discrete molecular oscillators with a large number of molecules for each species, in a big volume. Indeed, we have shown in this chapter that the phase equations serve this purpose well. Second, for oscillators with very few molecules for each species in a small volume, a new phase concept needs to be developed, without resorting to continuous limit approximations. This one is as yet an unsolved problem. Third, there are systems in between the two classes just stated, with moderate number of molecules, for which the continuous phase concept is still useful but requires a hybrid approach with combined use of both discrete and continuous models for acceptable accuracy (note that the phase computation schemes are tailored to concretize this hybrid approach), and this is where the contribution of this chapter should be placed.

As yet, the described methods benefit extensively from continuous state-space approximations derived from the molecular descriptions of such oscillators, and the assumed most accurate brute-force scheme shares this aspect. A future direction furthering this study can be described as follows, in line with the necessity of handling the second class of oscillators stated above. A proper phase model theory (not relying on continuous limit approximations) for discrete-space oscillators modeled with Markov chains needs to be developed. We believe that such a discrete phase model theory can be developed based on *cycle representations* for Markov chains [67, 55, 68]. We made progress also on this problem. We have developed a theory that precisely characterizes the phase noise of a single cycle in a continuous-time Markov chain. We were able to show that the phase noise theory we have developed for a single cycle in fact reduces to the previously developed continuous-space phase noise theory in the limit. We are currently working on extending this discrete phase noise theory to many cycles, i.e., to a *cycle decomposition* of a continuous-time Markov chain.

Chapter 8

CONCLUSIONS

8.1 Summary

The notion of isochrons can almost be presumed the most reliable concept, on which the oscillator phase definition can be depended. Information on the isochron portrait of a particular oscillator is crucial, when the oscillator is perturbed or noisy and thus its phase has to be quantified for analysis.

The exact computation of isochrons is very costly (see Chapter 1, Section 4.3.2, and also Appendix Section A.2). Therefore, one must resort to local approximations for isochrons. The linear approximations, defined by the celebrated PPV or the infinitesimal PRCs, are already known in the literature. However, heavily perturbed oscillators with states wandering away from their limit cycles are a challenge in analysis and demand more accurate information on their isochrons.

In this thesis, we have developed the notion of quadratic approximations for the isochrons of oscillators and have also proposed a numerical method for their numerical computation (see Section 4.2).

Furthermore, in this thesis, we have developed a unified framework of oscillator phase computations that is applicable virtually to any oscillator from an arbitrary discipline that can be described in a nonlinear autonomous ODE model (see Chapter 6). This framework subsumes the pertaining studies carried out in the last few decades on oscillator phase computations and is founded upon the concept of isochrons. Local (linear and quadratic) approximations of isochrons enable the development of practically usable phase models.

The notion of phase models in the form of differential equations, i.e., phase equations, is already known in the literature, and quite a substantial amount of work has been done in the analysis of such models, several of them have proved themselves tenacious enough to be transported to commercialization. We have successfully put these phase models into the context of our framework (see Section 6.2).

On top of the isochron concept (of Chapter 4) and our recent work on the quadratic local approximations of isochrons, we have developed second-order phase equations that are locally more accurate with respect to the already known phase model (see Section 6.3). This model will help analysis of oscillators that are affected by more than weak perturbations. Accordingly, analysis of more than weakly coupled oscillators will be possible thanks to this new model.

The phase equations are amenable to further semi-analytical investigation of oscillator phase. However, in essence, they are differential equations, and through discretizations and due to related numerical analysis considerations, the phase equations suffer from accumulation of global errors, when we use them for transient phase simulations. In Section 6.4, we have described, as another flavor of phase models, simpler phase computation schemes in the form of algebraic equations. These schemes, being more accurate than the phase equations (for they do not employ orbital deviation approximations) and due to their nature, are tailored to fit the transient simulation of oscillator phase.

As an application of the various phase models reviewed or proposed in this thesis, in Chapter 7 we have first worked several approximations on the inherently noisy discrete oscillator models and then utilized the phase computation methods to calculate the phases of individual SSA-generated sample paths of these oscillators. Based on the simulation results, it appears that the phase equations are very fast and quite accurate on less noisy oscillators. The phase computation schemes are not as fast as the equations but they retain their exquisite accuracies even on highly noisy molecular oscillators.

8.2 Future Work

It must first be noted that any future work that should in the natural course of research benefit from and utilize the methods proposed in this thesis would aim somehow to explore and gain deeper insight into oscillatory phenomena in various disciplines, including electronics and biology. Below, we outline some future research directions that could be taken in both adhering to this emphasis and enhancing the potential capabilities of the framework proposed.

- We have in this thesis mainly discussed the perturbations affecting the states of an oscillator. It is possible that the parameters determining the very nature of oscillators could be perturbed. Such type of perturbations are common particularly in biology, but they have drawn attention in electronics as well. A study unifying the analyses of parameter perturbations on oscillator phase founded upon isochrons seems to be missing in the literature.

- Injection locking and coupled oscillator analysis together with the phase noise analysis of PLLs (Phase Locked Loops) are important aspects of synchronization that deserve substantial research attention. The models developed in this thesis can perhaps be adapted to capture analysis of the diverse phenomena that take place in such applications.

- Scalar phase equations are especially important since they determine the ultimate reduced model of noisy oscillators, capturing the phase fluctuations only. As a more accurate model than PhEqnLL (of Section 6.2), the scalar model whose derivation is outlined in Appendix Section A.4 can perhaps inspire further semi-analytical research and therefore some more interesting findings.

- As briefly explained in Section 7.7, application of the proposed phase computation methods (developed for continuous models) to discrete molecular oscillators might be taking away some of the accuracy, and in doing this, one might not be exploring the full diversity of the research problem in hand. Therefore, a phase model theory directly treating discrete oscillators must be developed.

- In [52], it is suggested that an outright method of phase computation through Hilbert's transform yields correct results. However, considering the relative intricacies and computational costs of the methods reviewed and proposed in this thesis and Hilbert's transform, we hesitate in conceding to the validity of the approach in [52] employing the latter. It has to be examined thoroughly whether or not a direct method such as Hilbert's transform can indeed yield the sought phase results, and one has to ensure either the validity of such an approach or reveal if the particular method has resulted in theoretically correct values due to the special structure of the oscillating system analyzed.

In all of the possible research problems above, the work in this thesis can guide the workflow which eventually defines the approach to solution, and furthermore the techniques to be developed in solving these problems can perhaps be adapted from the methods developed herein.

Appendix A

MORE ON ISOCHRON AND PHASE COMPUTATIONS

A.1 Analytical Derivations for Exact Isochrons

We demonstrate (as a justification of the isochron expression stated in Sections 4.3.1 and 6.6.1) analytical information about a special polar oscillator, whose equations are given in polar coordinates as

$$\begin{aligned} \dot{r} &= 1 - r \\ \dot{\theta} &= r \end{aligned} \tag{A.1}$$

Notice that for the initial conditions $(r_0, \theta_0) = (1, 0)$ the oscillator in (A.1) traces the unit circle. Therefore, the unit circle is a limit cycle for this oscillator. Through the transformation

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\Big[r\ \exp(j\theta)\Big] &= \frac{\mathrm{d}}{\mathrm{d}t}\Big[\underbrace{r\ \cos(\theta)}_{x} + j\ \underbrace{r\ \sin(\theta)}_{y}\Big] \\ &= \left[\frac{1}{r}\frac{\mathrm{d}r}{\mathrm{d}t} + j\frac{\mathrm{d}\theta}{\mathrm{d}t}\right]\Big[r\ \exp(j\theta)\Big] \\ &= \left[\frac{x}{r}\frac{\mathrm{d}r}{\mathrm{d}t} - y\frac{\mathrm{d}\theta}{\mathrm{d}t}\right] + j\left[\frac{y}{r}\frac{\mathrm{d}r}{\mathrm{d}t} + x\frac{\mathrm{d}\theta}{\mathrm{d}t}\right] \end{aligned} \tag{A.2}$$

(A.1) can be written in Cartesian coordinates as

$$\begin{aligned} \dot{x} &= \tfrac{x}{\sqrt{x^2+y^2}}\left[1 - \sqrt{x^2+y^2}\right] - y\sqrt{x^2+y^2} \\ \dot{y} &= \tfrac{y}{\sqrt{x^2+y^2}}\left[1 - \sqrt{x^2+y^2}\right] + x\sqrt{x^2+y^2} \end{aligned} \tag{A.3}$$

The oscillator in (A.3) then traces the unit circle for the initial conditions $(x_0, y_0) = (1, 0)$.

In order to compute the phase of points in the domain of attraction of the unit circle, we have to solve the PDE

$$\begin{aligned} &\frac{\partial \hat{t}}{\partial x}\left[\frac{x}{\sqrt{x^2+y^2}}\left[1 - \sqrt{x^2+y^2}\right] - y\sqrt{x^2+y^2}\right] + \\ &\frac{\partial \hat{t}}{\partial y}\left[\frac{y}{\sqrt{x^2+y^2}}\left[1 - \sqrt{x^2+y^2}\right] + x\sqrt{x^2+y^2}\right] = 1 \end{aligned} \tag{A.4}$$

with the boundary condition

$$\hat{t}(x,y) = \arctan\left(\frac{y}{x}\right) \quad \text{on the unit circle} \tag{A.5}$$

where arctan is the four-quadrant version yielding values between 0 and 2π. Solving the problem given by (A.4) and (A.5) can be very difficult, both analytically and computationally.

Winfree in [2] demonstrates an alternative method of analytically solving this problem. It can be deduced that the isochron portrait of this oscillator has polar symmetry, so in polar coordinates we have the valid form $\hat{t}(r,\theta) = \theta - g(r)$ for some function $g(r)$. We then immediately have

$$\frac{\mathrm{d}\hat{t}}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\Big[\theta - g(r)\Big] = 1 \tag{A.6}$$

The following is then obtained.

$$\frac{\mathrm{d}g}{\mathrm{d}r} = \frac{\dot{\theta} - 1}{\dot{r}} = \frac{r-1}{1-r} = -1 \tag{A.7}$$

It is observed now that $\hat{t}(r,\theta) = \theta + r - c$ for some constant c. This constant turns out to be one since $\hat{t}(1,\theta) = \theta$. So we have

$$\begin{aligned} \hat{t}(r,\theta) &= \theta + r - 1 \\ \hat{t}(x,y) &= \arctan\left(\tfrac{y}{x}\right) + \sqrt{x^2+y^2} - 1 \end{aligned} \tag{A.8}$$

as equivalent forms for the phase. It can be easily checked that (A.8) satisfies the problem in (A.4) and (A.5). The resulting portrait of isochrons (level sets of the phase $\hat{t}$) is illustrated in Figure 4.2.

A.2 Alternative Exact Isochron Computations

We outline in this section a procedure that may yield useful but partial information about the local behavior of isochrons, as suggested in Section 4.3.2. Isochrons are hypersurfaces, most of the time with nonlinear behavior changing according to position. Note that we do not intend to compute isochrons exactly but employ only local approximations (linear and quadratic) around points on the limit cycle. It is plausible that if a perturbed solution traces not so close to the limit cycle, then phase computation methods depending on local approximations may not yield very accurate results. One may then wish to see how nonlinear an isochron really is around points on γ, by observing how the real isochron looks like. A method for computing isochrons in a brute-force manner is originally described in [1], thoroughly explained with an improved implementation and results herein (see Chapter 1 for intuitive explanation and Section 4.3.2 for a more technical account). It is stated that this method is not feasible for oscillators other than planar. Therefore, one is then compelled to devise a method that can give relatively useful partial information about an isochron, for oscillators with arbitrary dimension.

A method that may serve the purpose stated (gaining partial information about isochron nonlinearities) can be described as follows. Let us have two distinct points $\mathbf{x}_1$ and $\mathbf{x}_2$, which we somehow know are on the same isochron. Such two points can be generated for example by picking a point $\mathbf{x}_1$ that is in $\mathcal{W}$ but not on γ, and then using this $\mathbf{x}_1$ as an initial condition for the ODE in (3.1) and integrating (3.1) exactly an integer number of periods backward or forward in time to obtain $\mathbf{x}_2$ (see Theorem 4.1.1 for the validity of this fact). The essence of the current method is then to pick a number of initial points (for example along the line segment joining $\mathbf{x}_1$ and $\mathbf{x}_2$) and through some optimization method iteratively update these points in between, so that at the end (by some chosen tight tolerance) the path from $\mathbf{x}_1$ to $\mathbf{x}_2$ through these points forms a geodesic (shortest path) on the very isochron accommodating all the stated points. The varying curvature along this path might then somehow yield some partial information about the nonlinearity of the isochron, if carefully analyzed.

The details of the optimization problem stated are briefly as follows. This problem indeed amounts to minimizing the discretized form

$$\begin{aligned} &dist(\mathbf{z}_1, \ldots, \mathbf{z}_R) = \\ &\Delta u \sum_{i=0}^{R} \underbrace{\exp(c\left|\hat{t}(\mathbf{z}_i) - \hat{t}(\mathbf{x}_1)\right|)}_{\text{weight}} \left[\sum_{j=1}^{N} \left[\frac{z_{i+1,j} - z_{i,j}}{\Delta u} \right]^2 \right] \end{aligned} \tag{A.9}$$

where $\Delta u = 1/(R+1)$ and we set $\mathbf{z}_0 = \mathbf{x}_1$ and $\mathbf{z}_{R+1} = \mathbf{x}_2$. *dist* is a sum of weighted squares of distances between consecutive points along the path from $\mathbf{x}_1$ to $\mathbf{x}_2$. The minimization is done on values of the points in between, i.e., $\mathbf{z}_1, \ldots, \mathbf{z}_R$. The weight in (A.9) is especially chosen to force the points $\mathbf{z}_1, \ldots, \mathbf{z}_R$ onto the same isochron with $\mathbf{x}_1$ and $\mathbf{x}_2$. c must be a big positive real number. $\hat{t}(\mathbf{z}_i)$ and $\hat{t}(\mathbf{x}_1)$ can be computed through **PhCompBF**. The gradient information required by the unconstrained optimization algorithm can be analytically computed in part, but the part concerning the weight must be computed by finite differences. This method yielding partial information about the nonlinearity of isochrons is still computationally expensive, mostly due to the use of **PhCompBF**, but it is tractable for oscillators in arbitrary dimension, unlike the brute-force full isochron computation [1].

A.3 Isochron Approximation Errors

This section provides detailed comments on how the choice on a phase computation method can be made, considering isochron approximation errors committed by the methods (as touched upon briefly in Section 6.5.2). Local approximations for isochrons are a requisite if we would like to utilize the information that isochrons provide pertaining to the phase of an oscillator. Generating complete isochron portraits is out of the question, extracting partial information about isochrons as described above is still costly. Therefore, with the assistance of the phase gradient $\mathbf{v}_1(t)$ and phase Hessian $\mathbf{H}(t)$, we compute linear and quadratic approximations for isochrons. The fact that should be noted first and foremost is that these approximations provide only local information, i.e., in regions close to points on the limit cycle. Also, we note that in the close neighborhood of a point on γ, the quadratic approximation is more accurate than the linear one. However, even the quadratic approximation may fail to be an accurate fit, if the exact isochron exhibits rapidly varying nonlinear behavior around γ. Phase computation schemes, requiring perturbed solutions to be a priori computed, generate more or less accurate results at the expense of this information that must be provided beforehand. It seems that phase equations suffer the most from isochron approximation errors. In this sense, it may be right to declare that the LC oscillator has isochrons with highly nonlinear behavior, such that even **PhEqnQQ** depending on quadratic approximations generates phase shifts that are at variance with the exact results. **PhEqnQQ** and **PhEqnQL** are the best we have among the phase equations in terms of accuracy (i.e., superior over **PhEqnLL**), again not disregarding the fact that the superiority of quadratic approximations over linear ones is valid only locally, in the close neighborhood of γ. Therefore, it is now a question of how to decide in favor of **PhEqnQQ** or **PhEqnQL** to compute phase shift results, possibly not being content with the results of **PhEqnLL**. This is what we briefly discuss next.

A thorough study on the effects of isochron approximation errors on phase equation results should probably contain analyses that reveal the trend and rate, with which global errors in computed phase shifts accumulate. In this sense, the global error of a phase equation should be analyzed with respect to another phase equation of one higher order, e.g., it would be acceptable that we compare **PhEqnLL** to either **PhEqnQQ** or **PhEqnQL**. However, observing the level of intricacy entailed in the derivations of the proposed equations, we prefer not to delve into even more involved analyses as of the moment and adopt a local approach, instead of global, in assisting us with our selection of a phase equation to generate us accurate enough results, the choice being either **PhEqnLL** or one of the second order equations (**PhEqnQQ** or **PhEqnQL**).

The local approach in shaping our preferences of phase equations is grounded upon the actual phase $\hat{t} = \hat{t}(\mathbf{x}(t))$ and orbital deviation $\mathbf{y} = \mathbf{y}(t)$ computed at discrete points in time for a particular perturbation. Note that $\{\mathbf{v}_1(t), \mathbf{u}_2(t), \ldots, \mathbf{u}_N(t)\}$ is a basis set spanning $\Re^N$, since $\{\mathbf{u}_2(t), \ldots, \mathbf{u}_N(t)\}$ is a linearly independent set, and $\mathbf{v}_1(t)$ is orthogonal to all the vectors in this set. Therefore, it is wise to consider, through orthogonal projection, separately the components of $\mathbf{y}$ along $\mathbf{v}_1(t)$ and in the space spanned by $\{\mathbf{u}_2(t), \ldots, \mathbf{u}_N(t)\}$. If we compute the quantities $\hat{t}$ and $\mathbf{y}$ through any of the second-order equations, it is much likely that $\mathbf{y}(t)$ will contain a component along both the PPV $\mathbf{v}_1(\hat{t})$ and those Floquet modes called $\mathbf{u}_i(\hat{t})$ for $2 \leq i \leq N$. (Note again that ideally, $\mathbf{y}(t)$ does not contain a component along $\mathbf{v}_1(\hat{t})$ when a) if the computation is done with **PhEqnLL**, b) if the computation is done with any phase equation and the oscillator strictly has linear isochrons, i.e., hyperplanes, which is very unlikely.) Then, having $\mathbf{y}(t)$ and $\hat{t}$ available through a second-order equation, we may calculate the component of $\mathbf{y}(t)$ along $\mathbf{v}_1(\hat{t})$ through

$$\mathbf{y}_1(t) = \frac{\mathbf{v}_1(\hat{t})\mathbf{v}_1^{\intercal}(\hat{t})}{\mathbf{v}_1^{\intercal}(\hat{t})\mathbf{v}_1(\hat{t})}\mathbf{y}(t)$$

since $\mathbf{v}_1(\hat{t})$ is orthogonal to the other modes. Similarly, the component along these other modes is

$$\mathbf{y}_2(t) = \mathbf{y}(t) - \mathbf{y}_1(t)$$

We may now compare the norms of $\mathbf{y}_1(t)$ and $\mathbf{y}_2(t)$ and decide that if the norm of $\mathbf{y}_1(t)$ is considerable (by some pre-determined figure of merit) on average compared to that of $\mathbf{y}_2(t)$, then we may stick with second-order phase equations since the isochrons on average for the particular oscillator appear to be nonlinear, at least locally closer to quadratic. If the above is not the case, i.e., we are confident that on average the isochrons are close to being linear, we may well embrace the first-order and scalar phase equation **PhEqnLL**, reducing the computational cost of overall phase calculations. The described local method (suggested briefly in Section 6.5.2) of shaping our decisions on which type of equations to choose should be practically feasible in many cases, and thus we can avoid the intricate development of the theory needed to quantify the global errors introduced by phase equations. In fact, we have used a version of this method when computing the distance between linear and quadratic approximations along a certain direction confined in the linear approximation (see the results in Section 4.3.3. However, when the simple method for making our preferences is not adequate, more involved theory for error bound calculations has to be derived, such as the one we have already suggested. We finally note that such a derivation would make use of virtually any type of approximation employed in the derivations of our phase equations. In fact, we have listed every approximation and omitted terms throughout the thesis.

A.4 A Family of Scalar Phase Equations

In Section 6.3.5, it is suggested that formal model order reduction methods may be applied in order to reduce the simplified second-order phase equation into an even simpler equation of smaller size, solving for $\hat{t}$. However, at this point, we may describe a family of scalar phase equations (phase equations that consist of only one equation, i.e., the ultimate model reduction), that originate from the same methodology facilitating the derivation of **PhEqnQL**. As explained in the Section 6.3.5, (6.38) for $\hat{t}$ and (6.39) for $\mathbf{y}$ describe **PhEqnQL**. We observe that an approximate form of (6.38) is actually a special case of the following family of equations for $\hat{t}$.

$$\frac{\mathrm{d}\hat{t}}{\mathrm{d}t} = 1 + \left[\xi_n(\hat{t}, y)\right]^{\mathsf{T}} \mathbf{b}(\mathbf{x}_\mathrm{s}(t), t) \tag{A.10}$$

$\xi_n(\hat{t}, y)$ is the n th order approximation for $\mathbf{v}_1(\mathbf{x}_\mathrm{s}(\hat{t}) + \mathbf{y})$ obtained through n th order Taylor expansions. Observe that $n = 0$ in (A.10) leads to PhEqnLL directly and has no need for $\mathbf{y}$ calculations, for $n \geq 1$ we need $\mathbf{y}$. Fortunately, in our formulation for **PhEqnQL**, (6.39) is a differential equation for $\mathbf{y}$. We adopt this equation (6.39) to use in conjunction with (A.10) as well, and furthermore observe that the approximate solution $\mathbf{y}$ of (6.39) is explicitly given by (6.19). This means that we are free to substitute the explicit (6.19) into (A.10) for $\mathbf{y}$, obtaining a family of scalar phase equations for $n \geq 0$. We have already stated that $n = 0$ is equivalent to **PhEqnLL**. For $n \geq 2$, third and higher order partial derivatives of the phase $\hat{t}$ on γ are necessary, but these quantities are most probably intractable because of the computational cost they entail. For $n = 1$, we need $\mathbf{v}_1(t)$ and $\mathbf{H}(t)$, and another additional matrix quantity, for which numerical computation methods and preferably theoretical analysis schemes must be invented. Therefore, the case $n = 1$ leads to the derivation of an improved scalar model that may be tractable.

Appendix B

COMPUTATIONAL COMPLEXITIES

B.1 Computational Complexities in Chapter 6

In this section, we analyze the computational costs of phase computation schemes and phase equations, as used in Chapter 6. This section includes the derivation of the complexities given in Section 6.5.2. Below is a detailed account that leads us to justify and state the computational complexities in Section 6.5.2. Let us denote by N the number of states in an oscillator, K the number of timepoints along a single period, L the number of total timepoints along the interval where a phase computation method is run.

Preliminary statements on computational complexities are as follows. We assume as well-known complexities that $\mathbf{x}_{\mathrm{s}}(t)$, $\mathbf{G}(t)$ (assumed to be sparse), $\mathbf{u}_1(t)$ and $\mathbf{v}_1(t)$ are computable along a single period in $\mathcal{O}(N\,K)$ time. The computation of $\mathbf{H}(t)$ (which is usually not sparse) upon the stated quantities takes $\mathcal{O}(N^3\,K)$ time. Also, since phase computation schemes require it, we also state that computation of $\mathbf{x}_{\mathrm{p}}(t)$, a perturbed solution, along an interval of interest takes $\mathcal{O}(N\,L)$ time. We assume that if a matrix is sparse, then matrix vector multiplications and solving a linear system of equations involving this matrix can be done in linear time.

For **PhCompBF**, in order to compute the phase of a point $\mathbf{x}_{\mathrm{p}}(t_0)$, we have to integrate (3.1) with initial condition $\mathbf{x}_{\mathrm{p}}(t_0)$ for an ideally infinite number, namely n_{per}, of periods, so that the states vector can be assumed more or less to be tracing the limit cycle. This initial stage then takes $\mathcal{O}(n_{\mathrm{per}}\,K\,N)$ time. On top of this, assuming that we use uniform timesteps throughout the simulation, we must compute the FFT of the last period of the stated simulation result to compute the phase shift with respect to the unperturbed solution (the simulation that should simultaneously run along the stated one, but now with initial condition $\mathbf{x}_{\mathrm{s}}(t_0)$). The FFT computation takes $\mathcal{O}(K \log_2 K)$. Note that this last one is the complexity for the FFT of only one of the N states, since it is sufficient that we have the first order harmonics of only one state for the current computation of phase. Therefore, the phase computation of the single point $\mathbf{x}_\mathrm{p}(t_0)$ takes $\mathcal{O}(n_\mathrm{per}\, K\, N + K \log_2\, K)$. Since we have to carry out this computation for all of the L points along the interval, the overall complexity is $\mathcal{O}(n_\mathrm{per}\, K\, N\, L + L\, K \log_2\, K)$.

We next analyze **PhCompLin**. The phase computation of each point $\mathbf{x}_\mathrm{p}(t_0)$ requires several number of Newton iterations. In each Newton iteration of **PhCompLin**, one interpolated value for each of $\mathbf{x}_\mathrm{s}(t)$, $\mathbf{v}_1(t)$ and $\mathbf{G}(t)$ has to be computed. These computations (required by both function and Jacobian calculations) take each $\mathcal{O}(N)$ time, assuming that $\mathbf{G}(t)$ is sparse. Function and Jacobian evaluations then take $\mathcal{O}(N)$. The complexity of a single phase computation is $\mathcal{O}(m_\mathrm{lin}\, N)$, where m_lin is the maximum number of Newton iterations in obtaining the result. The overall complexity for all L points is then $\mathcal{O}(m_\mathrm{lin}\, N\, L)$.

Next is **PhCompQuad**. In line with the style of **PhCompLin**, we analyze first a single Newton iteration, which requires the computation of interpolated values for each of $\mathbf{x}_\mathrm{s}(t)$, $\mathbf{v}_1(t)$, $\mathbf{G}(t)$, $\mathbf{H}(t)$ and $\mathbf{M}(t)$ (a crucial matrix in phase Hessian theory, defined as in (4.23)). For the first three quantities, the complexity is $\mathcal{O}(N)$ as previously stated, but $\mathbf{H}(t)$ and $\mathbf{M}(t)$ require $\mathcal{O}(N^2)$ time as these are not usually sparse. Function and Jacobian computations are accordingly $\mathcal{O}(N^2)$. The overall complexity for all L points is $\mathcal{O}(m_\mathrm{quad}\, N^2\, L)$, where m_quad is the maximum number of Newton iterations in obtaining a single phase value.

The analysis of **PhEqnLL** can be done in two parts, since (6.12) for $\hat{t}$ is independent of (6.18) for $\mathbf{y}$, but not the other way around. Note that in the computation for (6.12) (which is a nonlinear equation, therefore requiring the use of Newton's method), calculation of interpolated values for $\mathbf{x}_\mathrm{s}(t)$ and $\mathbf{v}_1(t)$ is necessary. These both take $\mathcal{O}(N)$, as well as the scalar product $\mathbf{v}_1^\intercal(\hat{t})\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)$. Function and Jacobian (through finite differences) calculations take $\mathcal{O}(N)$, which makes the cost of a single Newton iteration $\mathcal{O}(N)$. Then, the computation of phase through (6.12) for all L points should take $\mathcal{O}(m_\mathrm{LL}\, N\, L)$, where again m_LL is the maximum number of Newton iterations in obtaining a single phase value. Before the integration of (6.18) for $\mathbf{y}$ (which is a linear differential equation in $\mathbf{y}$, therefore Newton's method is not required, i.e. a single Newton iteration is sufficient), several required values in the computation will already have been ready through the integration of (6.12), namely $\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)$, $\mathbf{v}_1^\intercal(\hat{t})\mathbf{b}(\mathbf{x}_\mathrm{s}(\hat{t}), t)$, $\mathbf{x}_\mathrm{s}(\hat{t})$ and $\mathbf{v}_1(\hat{t})$. However, on top of these, interpolated values for $\mathbf{G}(t)$ and $\mathbf{u}_1(t)$ are necessary for all L timepoints, which are readily computable in $\mathcal{O}(N\, L)$ time, noting the sparsity of $\mathbf{G}(t)$. Then, also making use of sparsity, a single $\mathbf{y}$

value can be computed in $\mathcal{O}(N)$ time, making the whole integration of (6.18) of complexity $\mathcal{O}(N\,L)$. We finally note that $\hat{t}$ and $\mathbf{y}$ computations through the two equations therefore incur $\mathcal{O}(m_{\mathrm{LL}}\,N\,L)$.

For **PhEqnQQ**, unlike the case of **PhEqnLL**, (6.32) for $\hat{t}$ and (6.36) for $\mathbf{y}$ are fully coupled nonlinear equations and they have to be solved simultaneously. We note that for each function evaluation necessary for Newton's method, interpolated values for $\mathbf{x}_{\mathrm{s}}(t)$ ($\mathcal{O}(N)$), $\mathbf{u}_1(t)$ ($\mathcal{O}(N)$), $\mathbf{v}_1(t)$ ($\mathcal{O}(N)$), $\mathbf{H}(t)$ ($\mathcal{O}(N^2)$), $\mathbf{G}(t)$ ($\mathcal{O}(N)$) and $\partial\mathbf{G}(t)/\partial\mathbf{x}_{\mathrm{s}}(t)$ ($\mathcal{O}(N^2)$) have to be computed, incurring the complexities given in parentheses after each item, noting again the sparsity of $\mathbf{G}(t)$ and non-sparsity of $\mathbf{H}(t)$. We then seek to find out the computational complexity for function evaluations, as we have done previously for the other methods. $\mathbf{v}_1(\hat{t})+\mathbf{H}(\hat{t})\mathbf{y}$ computation incurs $\mathcal{O}(N^2)$, and $b(\mathbf{x}_{\mathrm{s}}(\hat{t}),t)+\left[\partial b(\mathbf{x}_{\mathrm{s}}(\hat{t}),t)/\partial\mathbf{x}_{\mathrm{s}}(\hat{t})\right]\mathbf{y}$ is calculated in $\mathcal{O}(N)$ time since we assume that $\partial b(\mathbf{x}_{\mathrm{s}}(\hat{t}),t)/\partial\mathbf{x}_{\mathrm{s}}(\hat{t})$ is sparse. We assume that $\left[\partial\mathbf{G}(\hat{t})/\partial\mathbf{x}_{\mathrm{s}}(\hat{t})\right](\mathbf{y}\otimes\mathbf{y})$ incurs $\mathcal{O}(N^2)$, while $\mathbf{G}(\hat{t})\mathbf{y}$ and $\mathbf{u}_1(\hat{t})+\mathbf{G}(\hat{t})\mathbf{y}$ take $\mathcal{O}(N)$ time. Therefore, function evaluations in all incur $\mathcal{O}(N^2)$ and Jacobian calculations (through finite differences) cost $\mathcal{O}(N^3)$ time. Assuming sparsity again linear system of equations solutions should take $\mathcal{O}(N)$ time. In all, the whole integration of **PhEqnQQ** is to cost $\mathcal{O}(m_{\mathrm{QQ}}\,N^3\,L)$, where m_{QQ} is the maximum number of iterations in solving for a single phase value, as usual. The analysis for **PhEqnQL** is again similar and yields a complexity of the same form.

Note that in most of the above analyses, we have denoted the number of Newton iterations by appropriately subscripted m. If these m are bounded and do not scale with the number of states N (which in fact they do not most of the time), then all of the above m can be omitted in expressions for the complexities above.

The overview of the complexities analyzed tells us that **PhCompBF** is indeed the most expensive method as stated many times before. **PhCompLin** and **PhEqnLL** are linear in the number of states, N. **PhCompQuad** is of quadratic complexity in N, and the multi-dimensional **PhEqnQQ** and **PhEqnQL** are third order. More accurate methods should incur more computation time so that the problem in hand poses a trade-off between accuracy and speed, computational cost of the most accurate **PhCompBF** being very high.

B.2 Computational Complexities in Chapter 7

In this section, we analyze the computational costs of phase computation schemes and phase equations, as applied to SSA sample paths of discrete molecular oscillators (as in Chapter 7). Particularly, this section provides details on Section 7.5.4. We have found other implementations (compared to those of Chapter 6, therefore of Appendix Section B.1) for several of the methods to be more suitable for the problem in hand. This is why the derivations in this section are different. Let us denote by N the number of states in an oscillator, K the number of timepoints along a single period, L the number of total timepoints along the interval where a phase computation method is run, as in Appendix Section B.1. Note that although SSA simulates all N species, some of these species might be required to remain constant in molecule number throughout the simulation. In many of such cases, the continuous models are required to disregard these constant number species algorithmically. Therefore, in all of the expressions below we might be able to put instead of N a number little less than that, the effect reducing the overall complexities. Preliminary statements on computational complexities are as given in Appendix Section B.1.

We are using exactly the same algorithm for implementing **PhCompBF** as given in Appendix Section B.1. Therefore, the complexity does not change.

Analyzing the complexities of the phase equations requires first the evaluation of the modulated noise term that comes up in the update rule of (7.17), i.e. $\mathbf{S}\left[\mathbf{e}_j - \mathbf{a}\big(\mathbf{x}_s\big(\hat{t}(t)\big)\big)\,\tau\right]$. The propensity functions vector $\mathbf{a}$ in this expression can be obtained in $\mathcal{O}(M)$ time, where M is the number reactions. Then, the subtraction $\left[\mathbf{e}_j - \mathbf{a}\big(\mathbf{x}_s\big(\hat{t}(t)\big)\big)\,\tau\right]$ is also $\mathcal{O}(M)$. The rest of the complexity depends on the sparsity of the stoichiometry matrix $\mathbf{S}$. If $\mathbf{S}$ is sparse (i.e., if there is roughly one nonzero entry in each row), the last complexity figure in the evaluation becomes $\mathcal{O}(N)$, where N is the number of species. Else, if $\mathbf{S}$ is totally dense, the last operation, multiplication of $\mathbf{S}$ with the other term takes $\mathcal{O}(N\,M)$ time. So, one may deduce in this manner that the preparation of the modulated noise term that perturbs the system and accordingly the phase equations is in the worst case $\mathcal{O}(N\,M)$ and in the best case $\mathcal{O}(N+M)$.

As given by the update rule in (7.17), **PhEqnLL** requires an inner product with the possibly interpolated (for evaluation at an arbitrary phase $\hat{t}$) phase gradient $\mathbf{v}(t)$. Interpolation and the inner product are both $\mathcal{O}(N)$. The rest of the update rule is scalar, $\mathcal{O}(1)$. The update rule has to be run for all L timepoints along the interval of simulation. Therefore, **PhEqnLL** is in the worst case $\mathcal{O}(N\,M\,L)$ and in the best case $\mathcal{O}(M\,L + N\,L)$.

PhEqnQL update rule consists of forward Euler discretizations of the equations in (7.22) and (7.19). For (7.22), phase Hessian $\mathbf{H}(t)$ interpolations and the $\mathbf{H}(\hat{t})\,\mathbf{Y}$ product both take $\mathcal{O}(N^2)$, since $\mathbf{H}(\hat{t})$ is dense. These operations dominate $\mathbf{v}(t)$ interpolations ($\mathcal{O}(N)$) and the summation ($\mathcal{O}(N)$). The scalar product of $\mathbf{v} + \mathbf{H}\,\mathbf{Y}$ and the modulated noise term takes $\mathcal{O}(N)$. For (7.19), since $\mathbf{G}(t)$ is sparse, both $\mathbf{G}(t)$ and $\mathbf{u}(t)$ interpolations and the $\mathbf{G}\,\mathbf{Y}$ product takes $\mathcal{O}(N)$. Therefore, one may deduce that the update rule in (7.19) takes $\mathcal{O}(N)$. The whole update rule carried out at L timepoints then takes $\mathcal{O}(N^2\,L + N\,M\,L)$ in the worst case and $\mathcal{O}(N^2\,L + M\,L)$ in the best case.

The phase computation schemes consist of solving the algebraic equations in (7.24) or (7.25) (depending on whether the linear or quadratic scheme is preferred). We favor the bisection method to solve for the appropriate phase at each timepoint of an SSA sample path. The complexity of solution computation through bisections is analyzed as follows. In both types of phase computation schemes, the initial guess at a phase at the current timepoint is taken as the computed phase of the previous timepoint in the history of the sample path. The bisection method requires first an interval in which the solution is sure to exist. Therefore, an interval of length d_{min} is selected, in the middle of which the previously computed phase $\hat{t}_{\text{prev}}$ resides. It is tested whether the phase of the current timepoint lies in this interval. If not the interval length is doubled, keeping $\hat{t}_{\text{prev}}$ in the middle. A maximum value of the interval d_{max} is also predetermined. It is best to select d_{max} and d_{min} to be appropriate fractions of the period T. Then, the maximum number of iterations for interval finding in preparation for the bisection method is $\log_2 \left\lceil \frac{d_{\text{max}}}{d_{\text{min}}} \right\rceil$. After the interval is found, next comes the bisections. When the length of the interval is chopped down to some value d_{tol}, the current phase solution $\hat{t}_{\text{soln}}$ is assumed to be found. If relative tolerances are utilized, d_{tol} becomes also a function of $\hat{t}_{\text{soln}}$ as in $d_{\text{tol}} = \epsilon_{\text{rel}}\hat{t}_{\text{soln}} + \epsilon_{\text{abs}}$, where ϵ_{rel} and ϵ_{abs} are relative and absolute tolerances, respectively. Since d_{tol} indeed depends on the phase solution, let us choose for it the smallest d_{tol} value encountered in the history of the computation. Then, the maximum number of iterations in bisections is $\log_2 \left\lceil \frac{d_{\text{max}}}{d_{\text{tol}}} \right\rceil$. As the complexities of valid interval checking is the same in interval finding and bisections, the maximum total number of iterations in bisections with the preparation is $\log_2 \left\lceil \frac{d_{\text{max}}^2}{d_{\text{tol}}\,d_{\text{min}}} \right\rceil$. It can be deduced that a single valid interval check in the **PhCompLin** scheme is roughly the same in complexity and it is $\mathcal{O}(N)$. This complexity for **PhCompQuad** is roughly $\mathcal{O}(N^2)$. In all, the complexity of **PhCompLin** for L timepoints becomes

$$\mathcal{O}\Bigg(\; N \; L \; \log_2 \left\lceil \frac{d_{\max}^2}{d_{\mathrm{tol}}\, d_{\min}} \right\rceil \; \Bigg) \tag{B.1}$$

In similar fashion, the complexity of **PhCompQuad** can be derived to be

$$\mathcal{O}\Bigg(\; N^2 \; L \; \log_2 \left\lceil \frac{d_{\max}^2}{d_{\mathrm{tol}}\, d_{\min}} \right\rceil \; \Bigg) \tag{B.2}$$

Notice that in (B.2) only the term involving N has changed with respect to (B.1).

The essence of the above analyses is that accuracy demands a trade-off of complexity. For mildly noisy oscillators, the phase equations should remain somewhat close to the results of the golden reference **PhCompBF** and the other approximate phase computation schemes, which imitate **PhCompBF** very successfully with much less computation times. For more noisy oscillators, we should expect the phase computation schemes to do still well, although the phase equations will compute some inaccurate results very fast. **PhCompBF** is always very slow.

Appendix C

PROOFS OMITTED IN THE TEXT

C.1 The Differential Lyapunov Equation

Below is the proof of Lemma 4.2.1, which was omitted in the text and gives the generic solution of a differential Lyapunov equation.

Proof of Lemma 4.2.1: Recall once more the differential Lyapunov equation for the Jacobian of $\mathbf{v}_1$, which we call $\mathbf{H}$, the phase Hessian. All time arguments below have been omitted for convenience.

$$\frac{\mathrm{d}\Psi}{\mathrm{d}t} + \Psi\mathbf{G} + \mathbf{G}^{\intercal}\Psi = -\mathbf{M} \tag{C.1}$$

Let us multiply all terms with $\mathbf{u}_p$ to obtain

$$\frac{\mathrm{d}\Psi}{\mathrm{d}t}\mathbf{u}_p + \Psi\mathbf{G}\mathbf{u}_p + \mathbf{G}^{\intercal}\Psi\mathbf{u}_p = -\mathbf{M}\mathbf{u}_p \tag{C.2}$$

The following is again an identity.

$$\frac{\mathrm{d}\Psi}{\mathrm{d}t}\mathbf{u}_p = \frac{\mathrm{d}}{\mathrm{d}t}\Big[\Psi\mathbf{u}_p\Big] - \Psi\frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}t} \tag{C.3}$$

Plugging (C.3) into (C.2) and making use of (3.11),

$$\frac{\mathrm{d}}{\mathrm{d}t}\Big[\Psi\mathbf{u}_p\Big] - \Psi\Big(\mathbf{G}\mathbf{u}_p - \mu_p\mathbf{u}_p\Big) + \Psi\mathbf{G}\mathbf{u}_p + \mathbf{G}^{\intercal}\Psi\mathbf{u}_p = -\mathbf{M}\mathbf{u}_p \tag{C.4}$$

After a cancellation and some manipulations, (C.4) can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\Big[\Psi\mathbf{u}_p\Big] + \Big(\mathbf{G}^{\intercal} + \mu_p\mathbf{I}_M\Big)\Big[\Psi\mathbf{u}_p\Big] = -\mathbf{M}\mathbf{u}_p \tag{C.5}$$

Now let us multiply (C.5) by $\mathbf{u}_r^{\intercal}$ from the left to obtain

$$\mathbf{u}_r^{\intercal}\frac{\mathrm{d}}{\mathrm{d}t}\Big[\Psi\mathbf{u}_p\Big] + \mathbf{u}_r^{\intercal}\Big(\mathbf{G}^{\intercal} + \mu_p\mathbf{I}_M\Big)\Big[\Psi\mathbf{u}_p\Big] = -\mathbf{u}_r^{\intercal}\mathbf{M}\mathbf{u}_p \tag{C.6}$$

Again, making use of identities similar to (C.3) and (3.11), we get the simplified form of (C.6) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\Big\{\mathbf{u}_r^{\intercal}\Psi\mathbf{u}_p\Big\} + \Big(\mu_r + \mu_p\Big)\Big\{\mathbf{u}_r^{\intercal}\Psi\mathbf{u}_p\Big\} = -\mathbf{u}_r^{\intercal}\mathbf{M}\mathbf{u}_p \tag{C.7}$$

after some tedious manipulations.

Note that (C.7) is a scalar linear inhomogeneous equation with constant coefficients. Therefore, the form of its solution is readily available. We set as the initial condition $\mathbf{u}_r^{\mathsf{T}}(0)\Psi(0)\mathbf{u}_p(0)$. In the arguments of functions, we use the likes of t instead of $\mathbf{x}_{\mathrm{s}}(t)$, for convenience. The solution of (C.7) is given by

$$\begin{aligned}\mathbf{u}_r^{\mathsf{T}}(t)\Psi(t)\mathbf{u}_p(t) &= \mathbf{u}_r^{\mathsf{T}}(0)\Psi(0)\mathbf{u}_p(0)\exp\Big(-(\mu_r+\mu_p)t\Big)\\ &- \int_0^t \exp\Big(-(\mu_r+\mu_p)(t-\tau)\Big)\Big[\mathbf{u}_r^{\mathsf{T}}(\tau)\mathbf{M}(\tau)\mathbf{u}_p(\tau)\Big]d\tau \qquad \text{(C.8)}\end{aligned}$$

In handling (C.8) once more, we multiply this equation from the left with $\mathbf{v}_r$ and from th right with $\mathbf{v}_p^{\mathsf{T}}$ and double sum all over r and p. In view of the identities

$$\Bigg[\sum_{r=1}^N \mathbf{v}_r(t)\mathbf{u}_r^{\mathsf{T}}(t)\Bigg]\Psi(t)\Bigg[\sum_{p=1}^N \mathbf{u}_p(t)\mathbf{v}_p^{\mathsf{T}}(t)\Bigg] = \Big[\mathbf{I}_N\Big]\Psi(t)\Big[\mathbf{I}_N\Big] = \Psi(t)$$

and

$$\Bigg[\sum_{r=1}^N \exp(-\mu_r(t-0))\mathbf{v}_r(t)\mathbf{u}_r^{\mathsf{T}}(0)\Bigg]\Psi(0)\Bigg[\sum_{p=1}^N \exp(\mu_p(0-t))\mathbf{u}_p(0)\mathbf{v}_p^{\mathsf{T}}(t)\Bigg]$$
$$=\Big[\Upsilon^{\mathsf{T}}(t,0)\Big]\Psi(0)\Big[\Upsilon(0,t)\Big]$$

and

$$\Bigg[\sum_{r=1}^N \exp(-\mu_r(t-\tau))\mathbf{v}_r(t)\mathbf{u}_r^{\mathsf{T}}(\tau)\Bigg]\mathbf{M}(\tau)\Bigg[\sum_{p=1}^N \exp(\mu_p(\tau-t))\mathbf{u}_p(\tau)\mathbf{v}_p^{\mathsf{T}}(t)\Bigg]$$
$$=\Big[\Upsilon^{\mathsf{T}}(t,\tau)\Big]\mathbf{M}(\tau)\Big[\Upsilon(\tau,t)\Big]$$

we obtain from (C.8)

$$\begin{aligned}\Psi(t) &= \Upsilon^{\mathsf{T}}(t,0)\Psi(0)\Upsilon(0,t)\\ &- \int_0^t \Upsilon^{\mathsf{T}}(t,\tau)\mathbf{M}(\tau)\Upsilon(\tau,t)d\tau \qquad \text{(C.9)}\end{aligned}$$

which is the same as (4.32). ■

C.2 Orbital Deviation Characterizations in Phase Equations

This section provides the proofs, which were omitted in the text, for orbital deviation characterizations in phase equations. The first proof belongs to Theorem 6.2.1 of Section 6.2 on the first-order phase equation. The second proof is that of Theorem 6.3.3 in Section 6.3 for the second-order phase equation.

Proof of Theorem 6.2.1: Subtracting the corresponding sides of the identity in (6.13) from those of (6.16), also noting the decomposition in (6.17), we obtain exactly (6.18).

In order to find the solution in (6.19) for **y**, we have to contend that

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \left[\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}}\right]\left[\frac{\mathrm{d}\hat{t}}{\mathrm{d}t}\right] \approx \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}} \tag{C.10}$$

based on Assumption 6.2.1. Then, through (C.10), (6.18) can be approximated as

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}} = \mathbf{G}(\hat{t})\mathbf{y} + \sum_{i=2}^{N}\left[\mathbf{v}_i^{\mathsf{T}}(\hat{t})\mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}),t)\right]\mathbf{u}_i(\hat{t}) \tag{C.11}$$

Now, noting the identity

$$\frac{\mathrm{d}\mathbf{u}_i(\hat{t})}{\mathrm{d}\hat{t}} = \mathbf{G}(\hat{t})\mathbf{u}_i(\hat{t}) - \mu_i\mathbf{u}_i(\hat{t}) \tag{C.12}$$

which is another form of the eigenvalue problem statement in (3.11), and the proposed form of **y** in (6.9), (C.11) can be written alternatively as

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}} = \mathbf{G}(\hat{t})\mathbf{y} - \sum_{i=2}^{N} c_i\mu_i\mathbf{u}_i(\hat{t}) \tag{C.13}$$

Comparing (C.11) and (C.13), the identity

$$-\sum_{i=2}^{N} c_i\,\mu_i\,\mathbf{u}_i(\hat{t}) = \sum_{i=2}^{N}\left[\mathbf{v}_i^{\mathsf{T}}(\hat{t})\,\mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}),t)\right]\mathbf{u}_i(\hat{t}) \tag{C.14}$$

can be deduced. Defining the matrix **U** and the vector **d** as

$$\mathbf{U} = \left[\begin{array}{c}\mathbf{u}_2(\hat{t})\cdots\mathbf{u}_N(\hat{t})\end{array}\right] \quad \text{and} \quad \mathbf{d} = \left[\begin{array}{c} -\mu_2 c_2 \\ \vdots \\ -\mu_N c_N \end{array}\right] \tag{C.15}$$

respectively, it is possible to express the LHS of (C.14) in matrix-vector form as in

$$\mathbf{U}\,\mathbf{d} = \sum_{i=2}^{N}\left[\mathbf{v}_i^{\mathsf{T}}(\hat{t})\,\mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}),t)\right]\mathbf{u}_i(\hat{t}) \tag{C.16}$$

Note that since **U** is an $N\times(N-1)$-sized matrix with rank $N-1$, the solution of the system in (C.16) is unique if there exists one. The right-hand side of (C.16) is without doubt in the column space of $\mathbf{U}$. Therefore, (C.16) has a solution, and it is unique. Indeed, through (C.16), we can write

$$d_i = -\mu_i c_i = \mathbf{v}_i^{\mathsf{T}}(\hat{t})\mathbf{b}(\mathbf{x}_{\mathrm{s}}(\hat{t}), t) \tag{C.17}$$

which, after some manipulations and noting the form in (6.9), justifies (6.19). ■

Proof of Theorem 6.3.3: As will be shortly shown, much of the proof relies on the theory of the phase Hessian $\mathbf{H}(t)$ that is covered in Section 4.2.3. To conduct the proof, we have to recall that $\mathbf{H}(t)$ satisfies the differential equation in (4.22) where $\mathbf{M}(t)$ is defined by (4.23). The proof proceeds as follows.

Let us multiply (6.36) from the left by the transpose of $\mathbf{v}_1(\mathbf{x}_{\mathrm{s}}(\hat{t}) + \mathbf{y}) \approx \mathbf{v}_1(\hat{t}) + \mathbf{H}(\hat{t})\mathbf{y}$.

First, as in the proof of Theorem 6.2.1 (which we also had to omit because of space limitations), let us approximate $\mathrm{d}\mathbf{y}/\mathrm{d}t$ by $\mathrm{d}\mathbf{y}/\mathrm{d}\hat{t}$, assuming $\mathrm{d}\hat{t}/\mathrm{d}t \approx 1$.

Second, note that the last two terms on the right-hand side of (6.36) constitute the forcing term. We know through Theorem 6.3.2 that this forcing term has no component along $\mathbf{u}_1(\hat{t}) + \mathbf{G}(\hat{t})\mathbf{y}$. Therefore, applying the approximation scheme of Theorem 6.3.1, this forcing term multiplied by $\mathbf{v}_1(\hat{t}) + \mathbf{H}(\hat{t})\mathbf{y}$ from the left should vanish.

Third, we omit the scalar product of vectors $\mathbf{H}(\hat{t})\mathbf{y}$ and $1/2\left[\partial\mathbf{G}(\hat{t})/\partial\mathbf{x}_{\mathrm{s}}(\hat{t})\right](\mathbf{y}\otimes\mathbf{y})$ for it is higher-order.

Fourth, the scalar product that has not yet been examined, can be simplified through (6.35) and (4.23) as in

$$\begin{aligned}
&\mathbf{v}_1^{\mathsf{T}}(\hat{t})\left[\frac{1}{2}\frac{\partial\mathbf{G}(\hat{t})}{\partial\mathbf{x}_{\mathrm{s}}(\hat{t})}(\mathbf{y}\otimes\mathbf{y})\right] \\
= \;& \frac{1}{2}\sum_{i=1}^{N}\sum_{l=1}^{N} y_i y_l \left[\sum_{j=1}^{N}\frac{\partial G_{ji}(\hat{t})}{\partial x_{s,l}(\hat{t})} v_{1,j}(\hat{t})\right] \\
= \;& \frac{1}{2}\mathbf{y}^{\mathsf{T}}\mathbf{M}(\hat{t})\mathbf{y}.
\end{aligned} \tag{C.18}$$

In all, the computation of the current product yields, after the stated manipulations,

$$\left[\mathbf{v}_1(\hat{t}) + \mathbf{H}(\hat{t})\mathbf{y}\right]^{\mathsf{T}}\left[\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}} - \mathbf{G}(\hat{t})\mathbf{y}\right] - \frac{1}{2}\mathbf{y}^{\mathsf{T}}\mathbf{M}(\hat{t})\mathbf{y} = 0. \tag{C.19}$$

Let us now examine (C.19) in order to prove the claim. Simplifying, we obtain

$$\begin{aligned}
&\left[\mathbf{v}_1^{\mathsf{T}}(\hat{t})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}} - \mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{G}(\hat{t})\mathbf{y}\right] + \\
&\left[\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}} - \mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\mathbf{G}(\hat{t})\mathbf{y} - \frac{1}{2}\mathbf{y}^{\mathsf{T}}\mathbf{M}(\hat{t})\mathbf{y}\right] = 0.
\end{aligned} \tag{C.20}$$

Observe right away that

$$\left[\mathbf{v}_1^{\mathsf{T}}(\hat{t})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}} - \mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{G}(\hat{t})\mathbf{y}\right] = \frac{\mathrm{d}}{\mathrm{d}\hat{t}}\left[\mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{y}\right]. \tag{C.21}$$

Also, we have

$$\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}} = \frac{\mathrm{d}}{\mathrm{d}\hat{t}}\left[\frac{1}{2}\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\mathbf{y}\right] - \frac{1}{2}\mathbf{y}^{\mathsf{T}}\frac{\mathrm{d}\mathbf{H}(\hat{t})}{\mathrm{d}\hat{t}}\mathbf{y}. \tag{C.22}$$

Note also the identity

$$\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\mathbf{G}(\hat{t})\mathbf{y} = \frac{1}{2}\left[\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\mathbf{G}(\hat{t})\mathbf{y} + \mathbf{y}^{\mathsf{T}}\mathbf{G}^{\mathsf{T}}(\hat{t})\mathbf{H}(\hat{t})\mathbf{y}\right]. \tag{C.23}$$

Based on (4.22),

$$\frac{\mathrm{d}\mathbf{H}(\hat{t})}{\mathrm{d}\hat{t}} = -\mathbf{H}(\hat{t})\mathbf{G}(\hat{t}) - \mathbf{G}^{\mathsf{T}}(\hat{t})\mathbf{H}(\hat{t}) - \mathbf{M}(\hat{t}) \tag{C.24}$$

(C.22), (C.23), and (C.24) convey that

$$\begin{aligned}&\left[\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\hat{t}} - \mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\mathbf{G}(\hat{t})\mathbf{y} - \frac{1}{2}\mathbf{y}^{\mathsf{T}}\mathbf{M}(\hat{t})\mathbf{y}\right]\\ &= \frac{\mathrm{d}}{\mathrm{d}\hat{t}}\left[\frac{1}{2}\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\mathbf{y}\right]\end{aligned} \tag{C.25}$$

(C.21) and (C.25) imply that (C.20) simplifies to

$$\frac{\mathrm{d}}{\mathrm{d}\hat{t}}\left[\mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{y} + \frac{1}{2}\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\mathbf{y}\right] = 0. \tag{C.26}$$

(C.26) means that the value in the square brackets is a constant, but that constant is exactly equal to zero since at a particular point, namely $\mathbf{x}_{\mathrm{s}}(\hat{t})$, this value is zero (note again that $\mathbf{y} = \mathbf{x}(t) - \mathbf{x}_{\mathrm{s}}(\hat{t})$ and equating $\mathbf{x}(t) = \mathbf{x}_{\mathrm{s}}(\hat{t})$ we observe that the value in the brackets is zero). Therefore, $\mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{y} + 1/2\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\mathbf{y} = 0$ throughout for all $\mathbf{y}$ that could instantaneously satisfy the differential equation (6.36). Then observe also that the equality $\mathbf{v}_1^{\mathsf{T}}(\hat{t})\mathbf{y}+1/2\mathbf{y}^{\mathsf{T}}\mathbf{H}(\hat{t})\mathbf{y} = 0$ is nothing but (4.10) with $\hat{t}$ instead of t. Hence, the claim. ■

Appendix D

PRELIMINARIES FOR POISSON PROCESSES

This chapter provides simple preliminaries on Poisson processes (as would benefit comprehension of the material in Section 7.2.4), such as the probability distribution, the Master Equation for the process and the Gaussian approximation valid in some conditions. The material in this chapter is borrowed from [49, 48].

D.1 Poisson Processes

A Poisson process is a pure birth process that tells one how many occurences of a certain event has come to pass. Let us call N_t the number of times of occurences observed until time t. A crucial property that makes such a process Poisson is that the sojourn times between consecutive occurences are exponentially distributed with the same parameter λ. Therefore, if $\{S_1, S_2, S_3, \ldots\}$ are the instances in time when the event occurs, then $\{S_1, S_2 - S_1, S_3 - S_2, \ldots\}$ are all exponentially distributed with λ (i.e. $\mathbb{P}(S_1 < t) = 1 - \exp(-\lambda t)$), moreover we assume they are independent. Simple manipulations lead one to derive the probability law for the process N_t as in

$$\begin{aligned}\mathbb{P}(N_t = k) &= \underbrace{F_k(t)}_{\mathbb{P}(S_k<t)} \\ &- \underbrace{\int_0^t F_k(t-s)\,\lambda\exp(-\lambda s)ds}_{F_{k+1}(t)=\mathbb{P}(S_{k+1}<t)} \qquad \text{(D.1)} \\ &= \exp(-\lambda t)\frac{(\lambda t)^k}{k!} \qquad \text{(D.2)}\end{aligned}$$

The formula in (D.1) is a compact one relating the exponential distribution of sojourn times and that of the Poisson counting process N_t. Note that $F_k(t)$ is the k-fold convolution of the exponential distribution, computed in the manner given exactly in the integral expression of (D.1). Note also that the expression in (D.1) is true for $k \geq 1$. We know simply for $k = 0$ that $\mathbb{P}(N_t = 0) = \mathbb{P}(S_1 > t) = \exp(-\lambda t)$. Then, the formula in (D.2) is consistent in $k \geq 0$.

Also of some significance are the generating functions for discrete valued random processes. The generating function for the Poisson process can be shown to be

$$\mathbb{E}\left[\alpha^{N_t}\right] = \exp(\lambda t(\alpha - 1)) \tag{D.3}$$

Moments of the Poisson distribution can be calculated through the generating function as in

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\alpha}\mathbb{E}\left[\alpha^{N_t}\right]\Big|_{\alpha=1} &= \mathbb{E}\left[N_t\right] = \lambda t && \text{(D.4)} \\ \frac{\mathrm{d}^2}{\mathrm{d}\alpha^2}\mathbb{E}\left[\alpha^{N_t}\right]\Big|_{\alpha=1} &= \mathbb{E}\left[N_t(N_t - 1)\right] = (\lambda t)^2 && \text{(D.5)} \end{aligned}$$

From (D.4) and (D.5) it can be deduced that

$$\mathbb{E}\left[N_t\right] = var(N_t) = \lambda t \tag{D.6}$$

D.2 Master Equation for the Poisson Process

The Poisson process can indeed be shown to be a Markov process (continuous-time Markov chain) with a master equation derived briefly as follows. Let us assume the probability of a jump in a process N_t as (in fact the only type of jump that can occur for a Poisson process)

$$\mathbb{P}\left(N_t \,:\, n \to n + 1 \,\mathrm{in}\, \Delta t\right) = \lambda\,\Delta t \tag{D.7}$$

Then, through probabilistic manipulations and letting Δt go to zero, the following master equation is derived.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{P}\left(N_t = n\right) = \lambda\Big[\mathbb{P}\left(N_t = n - 1\right) - \mathbb{P}\left(N_t = n\right)\Big] \tag{D.8}$$

It is also possible to derive a differential equation for the generating function of N_t through (D.8), and then the generating function can be shown to be as in (D.3), which proves the master equation in (D.8) belongs to a Poisson process.

D.3 Gaussian Approximations

A Gaussian approximation can be derived if we observe the following facts about Poisson processes. Let L_t and M_t be independent Poisson processes with corresponding parameters λ_1 and λ_2. The generating function for the process $L_t + M_t$ is then

$$\mathbb{E}\left[\alpha^{L_t+M_t}\right] = \mathbb{E}\left[\alpha^{L_t}\right]\mathbb{E}\left[\alpha^{M_t}\right] = \exp((\lambda_1 + \lambda_2)t(\alpha - 1)) \tag{D.9}$$

Above, if we set $N_t = L_t + M_t$ and $\lambda = \lambda_1 + \lambda_2$, it can be observed that the sum of independent Poisson processes is another Poisson process, its parameter the sum of individual parameters. In this sense, it is also true that any Poisson process can be decomposed into a sum of independent individual Poisson processes. Utilizing the central limit theorem for i.i.d. processes, a Gaussian approximation for Poisson random variables with big values for parameters (decomposed into sums as such) can be derived. The mean and variance of the approximation will inherently be equal and the same as those of the Poisson process.

BIBLIOGRAPHY

[1] E.M. Izhikevich. *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. Chapter 10. MIT Press, 2007.

[2] Arthur T. Winfree. *The Geometry of Biological Time*. Springer, 2001.

[3] A. Goldbeter. *Biochemical Oscillations and Cellular Rythms*. Cambridge Univ Press, 1996.

[4] A. Demir and A. Sangiovanni-Vincentelli. *Analysis and Simulation of Noise in Nonlinear Electronic Circuits and Systems*. Kluwer Academic Publishers, 1998.

[5] A. Demir, A. Mehrotra, and J. Roychowdhury. Phase noise in oscillators: A unifying theory and numerical methods for characterisation. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 47(5):655–674, May 2000.

[6] J.M.G. Vilar, H.Y. Kueh, N. Barkai, and S. Leibler. Mechanisms of noise-resistance in genetic oscillators. *Proc Natl Acad Sci USA*, 99(9):5988–5992, Apr 2002.

[7] A. Goldbeter. Computational approaches to cellular rhythms. *Nature*, 420(6912):238–245, 2002.

[8] K. Josic, E. T. Shea-Brown, and J. Moehlis. Isochron. *Scholarpedia*, 1(8):1361, 2006.

[9] M. Farkas. *Periodic Motions*. Springer-Verlag, 1994.

[10] A. Demir. Fully nonlinear oscillator noise analysis: An oscillator with no asymptotic phase. *International Journal of Circuit Theory and Applications*, pages 175–203, March-April 2007.

[11] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer, 1983.

[12] A. Demir and J. Roychowdhury. A reliable and efficient procedure for oscillator PPV computation, with phase noise macromodelling applications. *IEEE Tran. on CAD of ICs and Systems*, 22(2):188–197, Feb 2003.

[13] G.B. Ermentrout. *Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students*. SIAM, 2002.

[14] Eric Brown, Jeff Moehlis, and Philip Holmes. On the phase reduction and response dynamics of neural oscillator populations. *Neural Comput*, 16(4):673–715, Apr 2004.

[15] W. Govaerts and B. Sautois. Computation of the phase response curve: A direct numerical approach. *Neural Comput*, 18(4):817–847, Apr 2006.

[16] I.G. Malkin. Methods of Poincare and Liapunov in theory of non-linear oscillations. *Gostexizdat*, Moscow, 1949.

[17] Y. Kuramato. *Chemical Oscillations, Waves, and Turbulence*. Springer-Verlag, New York, 1984.

[18] F. X. Kaertner. Analysis of white and $f^{-\alpha}$ noise in oscillators. *International Journal of Circuit Theory and Applications*, 18:485–519, 1990.

[19] F. C. Hoppensteadt and Eugene M. Izhikevich. *Weakly Connected Neural Networks*. Springer, 1997.

[20] O. Suvak and A. Demir. Quadratic approximations for the isochrons of oscillators: A general theory, advanced numerical methods and accurate phase computations. *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, 29(8):1215–1228, August 2010.

[21] O. Suvak and A. Demir. Computing quadratic approximations for the isochrons of oscillators: A general theory and advanced numerical methods. In *IEEE/ACM ICCAD*, November 2009.

[22] O. Suvak and A. Demir. On phase models for oscillators. *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, 30(7):972–985, July 2011.

[23] O. Suvak and A. Demir. Phase models and computations for molecular oscillators. In *Proc. 8th Internat. Workshop on Computational Systems Biology (WCSB 2011)*, June 2011.

[24] O. Suvak and A. Demir. Phase computations and phase models for discrete molecular oscillators. *EURASIP Journal on Bioinformatics and Systems Biology*, 2012:6, June 2012.

[25] T. Djurhuus, V. Krozer, J. Vidkjaer, and T. K. Johansen. Oscillator phase noise: A geometrical approach. *IEEE Tran. on CAS I: Regular Papers*, 56(7):1373–1382, July 2009.

[26] C. C. Canavier. Phase response curve. *Scholarpedia*, 1(12):1332, 2006.

[27] P. Gaspard. The correlation time of mesoscopic chemical clocks. *J. Chem. Phys.*, 117:8905–8916, 2002.

[28] W. Vance and J. Ross. Fluctuations near limit cycles in chemical reaction systems. *J. Chem. Phys.*, 105:479–487, 1996.

[29] A. Hajimiri and T.H. Lee. A general theory of phase noise in electrical oscillators. *IEEE Journal of Solid-State Circuits*, February 1998.

[30] K. Kundert, J. K. White, and A. Sangiovanni-Vincentelli. *Steady-State Methods for Simulating Analog and Microwave Circuits.* Norwell, MA: Kluwer, 1990.

[31] Z. Gajic and M.T. Javed Qureshi. *Lyapunov Matrix Equation in System Stability and Control.* Academic Press, 1995.

[32] Thilo Penzl. Numerical solution of generalized Lyapunov equations. *Advances in Computational Mathematics*, 8(1):33–48, 1998.

[33] P. Benner, J. R. Li, and T. Penzl. Numerical solution of large-scale lyapunov equations, riccati equations, and linear-quadratic optimal control problems. *Numerical Linear Algebra with Applications*, 15(9):755–777, 2008.

[34] The control and systems library SLICOT. *http://www.slicot.org/*.

[35] S. R. Taylor, R. Gunawan, L. R. Petzold, and F. J. Doyle III. Sensitivity measures for oscillating systems: Application to mammalian circadian gene network. *IEEE Trans. Automat. Contr.*, 153:177–188, 2008.

[36] P. Smolen, P.E. Hardin, B.S. Lo, D.A. Baxte, and J.H. Byrne. Simulation of Drosophila circadian oscillations, mutations, and light responses by a model with VRI, PDP-1, and CLK. *Biophysical Journal*, 86:2786–2802, May 2004.

[37] X. Lai and J. Roychowdhury. Automated oscillator macromodelling techniques for capturing amplitude variations and injection locking. In *IEEE/ACM ICCAD*, November 2004.

[38] L. Fu and C. C. Lee. The circadian clock: pacemaker and tumour suppressor. *Nat. Rev. Cancer*, 3:350–361, 2003.

[39] L. Fu, H. Pelicano, J. Liu, P. Huang, and C. Lee. The circadian gene Period2 plays an important role in tumor suppression and dna damage response in vivo. *Cell*, 111:41–50, 2002.

[40] S. Davis and D. K. Mirick. Circadian disruption, shift work and the risk of cancer: a summary of the evidence and studies in seattle. *Cancer Causes Contr.*, 17:539–545, 2006.

[41] E. S. Schernhammer, F. Laden, F. E. Speizer, W. C. Willett, D. J. Hunter, I. Kawachi, C. S. Fuchs, and G. A. Colditz. Night-shift work and risk of colorectal cancer in the nurses' health study. *J. Natl. Cancer Inst.*, 95:825–828, 2003.

[42] K. Straif, R. Baan, Y. Grosse, B. E. Secretan, F. E. Ghissassi, V. Bouvard, A. Altieri, L. Benbrahim-Tallaa, and V. Cogliano. Carcinogenicity of shift-work, painting, and fire-fighting. *Lancet Oncol.*, 12(8):1065–1066, 2007.

[43] M.B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, Jan 2000.

[44] D.T. Gillespie. Stochastic simulation of chemical kinetics. *Annu Rev Phys Chem*, 58:35–55, 2007.

[45] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[46] Daniel T. Gillespie. The chemical langevin equation. *Journal of Chemical Physics*, 113(1):297, 2000.

[47] Desmond J. Higham. Modeling and simulating chemical reactions. *SIAM Rev.*, 50(2):347–368, 2008.

[48] N.G. van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, 1992.

[49] C.W. Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*. Springer-Verlag, 1983.

[50] M. Amdaoud, M. Vallade, C. Weiss-Schaber, and I. Mihalcescu. Cyanobacterial clock, a stable phase oscillator with negligible intercellular coupling. *Proc. Natl. Acad. Sci. U.S.A.*, 104:7051–7056, 2007.

[51] L. G. Morelli and F. Julicher. Precision of genetic oscillators and clocks. *Phys Rev Lett.*, 98(22):228101, 2007.

[52] H. Koeppl, M. Hafner, A. Ganguly, and A. Mehrotra. Deterministic characterization of phase noise in biomolecular oscillators. *Phys. Biol.*, 8(5), 2011.

[53] T. Tomita, T. Ohta, and H. Tomita. Irreversible circulation and orbital revolution. *Prog. Theor. Phys.*, 52:1744–1765, 1974.

[54] D. J. Wilkinson. *Stochastic Modelling for System Biology, 1st Ed.* CRC Press, New York, 2006.

[55] R. Feistel and W. Ebeling. Deterministic and stochastic theory of sustained oscillations in autocatalytic reaction systems. *Physica A: Statistical and Theoretical Physics*, 93(1-2):114–137, 1978.

[56] M.A. Gibson and J. Bruck. Exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry A*, 104:1876–1889, 2000.

[57] A. Slepoy, A. P. Thompson, and S. J. Plimpton. A constant-time kinetic monte carlo algorithm for simulation of large biochemical reaction networks. *J. Chem. Phys.*, 128(205101), 2008.

[58] D. Gabor. Theory of communication. *J. IEE London*, 93:429–457, 1946.

[59] O. Suvak and A. Demir. Quadratic approximations for the isochrons of oscillators: A general theory, advanced numerical methods and accurate phase computations. *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, 29(8):1215–1228, August 2010.

[60] eXtensible multi-dimensional Simulator [Online]. Available: `http://www.xmds.org/`.

[61] W. H. Cropper. *Mathematica computer programs for physical chemistry.* Springer Verlag, 1998.

[62] Cellerator Model Repository [Online]. Available: `http://www.cellerator.info/nb.html`.

[63] Cellular Models [Online]. Available: `http://www.cds.caltech.edu/~hsauro/models.htm`.

[64] E-Cell Website [Online]. Available: `http://www.e-cell.org/ecell/`.

[65] B. J. Bornstein, S. M. Keating, A. Jouraku, and M. Hucka. LibSBML: An API library for SBML. *Bioinformatics*, 24(6):880–881, 2008.

[66] S. M. Keating, B. J. Bornstein, A. Finney, and M. Hucka. SBMLToolbox: an SBML toolbox for MATLAB users. *Bioinformatics*, 22(10):1275–1277, 2006.

[67] S.L. Kalpazidou. *Cycle representations of Markov processes.* Springer Verlag, 2006.

[68] T.L. Hill. *Free energy transduction and biochemical cycle kinetics.* Springer Verlag, 1989.