

Generalized Nonlinear Timing/Phase Macromodeling: Theory, Numerical Methods and Applications

Chenjie Gu and Jaijeet Roychowdhury
{gcj,jr}@eecs.berkeley.edu
EECS Department, University of California, Berkeley

Abstract—We extend the concept of timing/phase macromodels, previously established rigorously only for oscillators, to apply to general systems, both non-oscillatory and oscillatory. We do so by first establishing a solid foundation for the *timing/phase response* of any nonlinear dynamical system, then deriving a timing/phase macromodel via nonlinear perturbation analysis. The macromodel that emerges is a *scalar, nonlinear time-varying equation* that accurately characterizes the system’s phase/timing responses. We establish strong links of this technique with projection frameworks for model order reduction.

We then present numerical methods to compute the phase model. The computation involves a full Floquet decomposition – we discuss numerical issues that arise if direct computation of the monodromy matrix is used for Floquet analysis, and propose an alternative method that are numerically superior. The new method has elegant connections to the Jacobian matrix in harmonic balance method (readily available in most RF simulators).

We validate the technique on several *highly nonlinear* systems, including an inverter chain and a firing neuron. We demonstrate that the new scalar nonlinear phase model captures phase responses under various types of input perturbations, achieving accuracies considerably superior to those of reduced models obtained using LTI/LPTV MOR methods. Thus, we establish a powerful new way to extract timing models of combinatorial/sequential systems and memory (e.g., SRAMs/DRAMs), synchronization systems based on oscillator enslaving (e.g., PLLs, injection-locked oscillators, CDR systems, neural processing, energy grids), signal-processing blocks (e.g., ADCs/DACs, FIR/IIR filters), etc..

I. INTRODUCTION

Automatic macromodeling, also known as model order reduction (MOR), has been important in EDA for more than 20 years and is of increasing interest today to several other communities, including biology, aeronautics and energy. Given a large input-output system, these algorithmically-rooted techniques extract smaller models that match certain important behaviors of a given system, to within a fidelity acceptable for a given application.

The behaviors or fidelity metrics to be preserved determine, to a great extent, the characteristics of a given macromodeling method. For example, many linear time-invariant (LTI) [1], [2] and linear periodic time-varying (LPTV) [3] MOR techniques choose *moments* (of transfer functions) as fidelity metrics to preserve. Similarly, weakly nonlinear macromodelling methods [3]–[6] match moments of multivariate Volterra transfer functions [7], and many strongly nonlinear MOR techniques [8]–[10], being based on gluing together reduced models of linearizations, rely heavily on moment matching as well.

The value of moments as fidelity metrics stems chiefly from the fact that they are closely related to *timing* and *delay* properties of linear systems. For example, the first moment of an LTI system is the well-known Elmore delay, while higher moments capture finer details of timing. Timing is of fundamental interest in many disciplines. For example, interconnect and buffer delays, the longest (critical) path of a combinational circuit, setup/hold times of latches/registers, jitter and phase shifts in oscillators/PLLs, etc., are important in IC design. Similarly, the timing and synchronization properties of firing neurons are thought to be their most important functional characteristic in neuroscience [11]. Such is the importance of timing in applications that it can be plausibly argued that MOR/macromodelling has been driven chiefly by a single underlying goal, a desire to capture timing properties of complex systems well.

An interesting observation is that moments are an *indirect* means of getting to the underlying timing properties of a system, in the sense that they infer timing indirectly from waveform information. For example, consider the waveform $x(t) = \sin(\omega t + \tau(t))$, the essential timing feature of which is a time-varying delay $\tau(t)$. The same time-varying delay can be embedded in a differently-shaped waveform, e.g., $y(t) = \text{squarewave}(\omega t + \tau(t))$. [$x(t)$ could be, for instance, the output of a time-varying RC circuit (with slowly changing time constant); while $y(t)$ could be the output of the same circuit followed by a memoryless hard clipper.] Note that $x(t)$ and $y(t)$ have different Elmore delays, even though the underlying timing quantity of interest, $\tau(t)$, is the same for both waveforms. This example indicates that, especially for nonlinear systems, moments or other indirect means of inferring timing properties can have shortcomings; a more direct way of capturing $\tau(t)$, that does not rely on waveform shapes such as $\sin(\cdot)$ or $\text{squarewave}(\cdot)$, is therefore desirable.

In this paper, we develop a theory to identify and macromodel the underlying timing/phase properties of any system. Our approach is based on ideas originally developed for phase macromodelling of autonomous oscillators, but generalizes them considerably in order to arrive at techniques applicable to any kind of system, whether or not they are autonomous or oscillatory. The timing/phase macromodel we generate is a single *scalar, nonlinear* differential equation that captures any system’s input/output timing properties. We also establish pleasing connections of this timing macromodel with projection frameworks [4], [10] for model reduction – we show that it is a projection of the original system onto a time-varying subspace derived from trajectory linearizations. We prove that existing phase macromodelling techniques that apply only to autonomous oscillators [12] are simply a special case of our general timing macromodelling method.

We then develop numerical methods for implementing and applying the theory. A core step in extracting the timing macromodel is a full Floquet decomposition [13]. We show that straightforward techniques based on computing monodromy matrices can face significant numerical issues, to address which we develop an alternative technique that has far superior numerical properties. The new technique is based on exploiting elegant eigen-properties of frequency-domain Jacobian matrices that arise naturally in the standard numerical methods of harmonic balance, widely available in RF simulators.

Finally, we validate and explore the uses of the new timing modeling technique by applying it to representative examples drawn from circuits and biology. To obtain concrete insights into the technique’s properties, we first study a simple nonlinear system in some detail. We then apply the technique to a firing neuron and an inverter chain circuit. We show that the new method provides large speedups in simulating timing properties, while at the same time providing results considerably more accurate than those from existing LTI/LPTV model reduction techniques.

The remainder of the paper is organized as follows. In Section II, we define the generalized concept of timing/phase response and discuss applications where timing/phase is important. We also review the projection framework of MOR, and explain limitations of LTI/LPTV reduced models in systems that are highly nonlinear. We then develop the theory of the new timing/phase macromodel in Section III and present numerical methods for computing it in Section IV. In Section V, we validate this new model on a set

of benchmarks and compare results against full simulations of the original systems.

II. BACKGROUND AND MOTIVATION

A. Problem Definition and Applications

Consider a nonlinear dynamical system described by a set of differential algebraic equations

$$\frac{d}{dt}\vec{q}(\vec{x}(t)) + \vec{f}(\vec{x}(t)) + \vec{b}(t) = 0, \quad (1)$$

where $\vec{x} \in \mathbb{R}^n$ are state variables (for example, node voltages in circuit equations) and $\vec{b} \in \mathbb{R}^n$ are inputs. We aim to derive a differential equation in terms of the *phase variable* that captures the *phase response* of the system. To make sense of this goal, we define the phase response as follows:

Definition 2.1 (Phase response): Suppose in (1), the (unperturbed) response to (unperturbed) input $\vec{b}_S(t)$ is $\vec{x}_S(t)$, and the (perturbed) response to (perturbed) input $\vec{b}_p(t)$ is $\vec{x}_p(t)$. There exists $z(t) : \mathbb{R}^+ \rightarrow \mathbb{R}$, such that $\vec{x}_S(t+z(t))$ best approximates $\vec{x}_p(t)$ in some sense¹. We call z the phase variable, and $z(t)$ the phase response.

This concept can be best understood geometrically. As shown in Fig. 1, the solid (blue) orbit is the trajectory of $\vec{x}_S(t)$, and the dotted (red) orbit is the trajectory of $\vec{x}_p(t)$ (we show two closed orbits in Fig. 1 for simplicity, but there is no restriction of the orbit being closed in our definition – the orbit can be arbitrary). At time t , on the orbit of $\vec{x}_S(t)$, there exists a point given by $\vec{x}_S(t+z(t))$ which best approximates $\vec{x}_p(t)$ on the perturbed orbit. The remaining difference $\vec{y}(t) = \vec{x}_p(t) - \vec{x}_S(t+z(t))$ is called the *amplitude deviation* [12]. Therefore, we can view the response to input $\vec{b}_p(t)$ as combined effects of the phase response and the amplitude deviation, *i.e.*,

$$\vec{x}_p(t) = \vec{x}_S(t+z(t)) + \vec{y}(t). \quad (2)$$

According to (2), we see that when $\vec{y}(t)$ is small, *i.e.*, $\vec{x}_p(t)$ stays close to the orbit of $\vec{x}_S(t)$ in the state space, capturing phase response alone gives a reasonably good approximation to the perturbed solution.

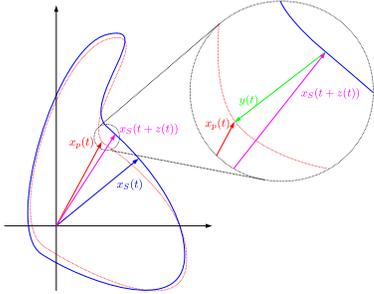


Fig. 1. Illustration of phase response.

With this definition of phase response, it is worth noting that although the phase response is an important characteristic of oscillators as pioneered by many researchers [11], [12], it is not a special characteristic only for oscillators, but generally for all kinds of dynamical systems. It has broad applications in many fields.

Phase response is of great importance in circuits in particular. For examples, in oscillators, phase noise or timing jitter is the phase response in a stochastic setting where noise/uncertainties are present; in digital gates and RC interconnects, timing information such as delay and slope are essentially derived from the phase response; in a pulse generator, the pulse width can also be derived from the phase response.

Scientists in biology and neuroscience also place a great emphasis on quantitative understanding of the phase response. For examples, normal human circadian rhythm disorders, which can be caused by insomnia, fatigue and jet lag, are phase responses of underlying physiological processes under different perturbations; synchronization of

a neural network is a phenomenon where the phase responses of all neurons settle down to the same value; some neuron firing/bursting behaviors, which are exhibited when currents are injected, can also be understood as the phase response of the neuron.

It must be noted that the assumption (that the trajectories of the system under various perturbations all cluster around a fixed orbit) is important for the phase response to be a good characteristic. In many systems, the validity of this assumption depends on both the system properties as well as input perturbations.

Fortunately, we see a lot of applications where this assumption is satisfied. For examples: in oscillators that are asymptotically orbital stable, the periodic orbit is an attractor which by definition attracts all trajectories nearby; in digital applications, although circuits never reside in a periodic steady state, their trajectories follow almost the same path since inputs are basically signals switching between “0” and “1” with variations of the slope, to the first-order approximation.

Among all possible perturbations, there is also a special one that is very useful and normally makes the above assumption satisfied. This is the phase perturbation, *i.e.*, the perturbed input $\vec{b}_p(t) = \vec{b}_S(t+p(t))$ is only a phase-modulated version of $\vec{b}_S(t)$ with some signal $p(t)$. For example, it can be used to represent an FM/PM signal in RF applications, or a delayed ramp signal with any slope and distortion in digital applications. In particular, when $p(t)$ is a constant, the perturbed solution $\vec{x}_p(t)$ lies exactly on the same orbit of $\vec{x}_S(t)$ if initial condition is set properly.

We conclude that with valid assumptions, phase response alone characterizes the system response, and therefore a good phase macromodel is of great importance in practice.

B. Projection Framework

Most previous MOR techniques follow a linear or nonlinear projection framework [4], [10], *i.e.*, they try to identify two low order ($q \ll n$) linear subspaces or nonlinear manifolds in the state space, and project the state variables and differential equations (residual) onto them respectively. This boils down to computing two projection functions $\vec{v} : \mathbb{R}^q \rightarrow \mathbb{R}^n$ and $\vec{w} : \mathbb{R}^n \rightarrow \mathbb{R}^q$, and deriving the reduced model of (1) as

$$\vec{w} \left(\frac{d}{dt} \vec{q}(\vec{v}(\vec{z}(t))) + \vec{f}(\vec{v}(\vec{z}(t))) + \vec{b}(t) \right) = 0, \quad (3)$$

where $\vec{z} \in \mathbb{R}^q$ are state variables of the reduced system.

We will show in Section III that our phase macromodeling technique indeed fits into this projection framework by carefully defining appropriate nonlinear projection functions.

C. Why Previous Reduced Models Can Fail

Most systems, including non-oscillatory ones, mentioned in Section II-A are highly nonlinear. Although LTI/LPTV reduced models have successes in modeling weakly nonlinear systems [4], they are almost determined to fail in these highly nonlinear applications.

LTI MOR techniques for nonlinear systems basically generate an LTI reduced model for the LTI system obtained by linearizing the nonlinear system around its DC operating point. Therefore, they inherit the assumption that inputs (perturbations) to the system must be small enough so that the system operates in its linear region. This assumption is valid in circuits like amplifiers, but fails in other systems. For example, digital circuits (such as an inverter) never operate in the linear region.

Similarly, LPTV MOR techniques for nonlinear systems generate an LPTV reduced model for the LPTV system obtained by linearizing the nonlinear system around its periodic steady state. It captures the nonlinearity of the system assuming that at time t , the small signal response to the nonlinear system can be approximated by

$$\frac{d}{dt} [C(\vec{x}_S(t))\vec{y}(t)] + G(\vec{x}_S(t))\vec{y}(t) + \Delta\vec{b}(t) = 0. \quad (4)$$

¹The notion of *best approximation* is defined in Section III.

It can be viewed as the combination of a series of linear models around every point on the periodic steady state, and it assumes that at any time t , the perturbed response $\vec{x}_p(t)$ is close to $\vec{x}_S(t)$.² However, this is not true in our applications. For example, if the input $\vec{b}(t) = \vec{b}_S(t + \tau)$ is a time-shifted version of $\vec{b}_S(t)$, then asymptotically the perturbed solution is $\vec{x}_p(t) = \vec{x}_S(t + \tau)$, which can be very far away from $\vec{x}_S(t)$. This simple fact makes the LPTV approximation fail, let alone the reduced model derived from it.

General nonlinear reduced models will capture the phase response correctly if the subspace or manifold is chosen correctly. Since the reduction criterion is normally to match transfer functions of many linearized systems and to cover training trajectories, the reduced model works in a general setting. However, if only the delay/phase property is of interest, these general-purpose reduced models are too redundant to be used.

As a brief summary, LTI/LPTV models fail since their basic assumptions are unsatisfied; nonlinear reduced models are good but still redundant and need to be tuned to capture the phase response.

III. GENERALIZED NONLINEAR TIMING/PHASE MACROMODEL

In this section, we derive the generalized phase macromodel. We show that a scalar nonlinear time-varying equation encodes the dynamics of the phase response. We further interpret this phase macromodel both in the traditional projection framework and via a nonlinear perturbation analysis from which we see clearly what system behaviors are characterized in the model. Note that our derivations make the assumption that the unperturbed system is in a periodic steady state. This assumption is crucial for both analysis and numerical methods. We then generalize the idea to the case where the unperturbed system is not in a periodic steady state in Section III-F.

A. Preliminaries and Notations

To understand derivations in following sections, we need to introduce a few notations and lemmas. For simplicity, we consider the system defined by a set of ordinary differential equations

$$\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t)) + \vec{b}(t). \quad (5)$$

Following [14], the results can be extended to differential algebraic equations. We omit derivations for this extension due to page limits.

We assume that the input $\vec{b}(t)$ is a periodic signal $\vec{b}_S(t)$ with period T , and that under this input, the asymptotic response of the system is $\vec{x}_S(t)$ which is also periodic with period T .

A traditional perturbation analysis using linearization can then be carried out: assuming that the response to the perturbed input $\vec{b}_p(t) = \vec{b}_S(t) + \vec{b}_w(t)$ is $\vec{x}_p(t) = \vec{x}_S(t) + \vec{w}(t)$, and substituting $\vec{x}_p(t)$ and $\vec{b}_p(t)$ in (5), we obtain

$$\frac{d}{dt}(\vec{x}_S(t) + \vec{w}(t)) = \vec{f}(\vec{x}_S(t) + \vec{w}(t)) + (\vec{b}_S(t) + \vec{b}_w(t)). \quad (6)$$

To the first order approximation, we have

$$\frac{d}{dt}\vec{w}(t) = G(t)\vec{w}(t) + \vec{b}_w(t), \quad (7)$$

where $G(t) = \frac{\partial \vec{f}}{\partial \vec{x}} \Big|_{\vec{x}_S(t)}$ is a time-varying matrix with period T .

(7) is an LPTV system, whose solution, according to Floquet theory

[13], is

$$\begin{aligned} \vec{w}(t) &= \Phi(t, 0)\vec{w}_0 + \int_0^t \Phi(t, s)\vec{b}_w(s)ds \\ &= U(t)D(t)V^T(0)\vec{w}_0 + U(t) \int_0^t D(t-s)V^T(s)\vec{b}_w(s)ds \\ &= \sum_{i=1}^n \vec{u}_i(t)e^{\mu_i t} \vec{v}_i^T(0)\vec{w}_0 + \sum_{i=1}^n \vec{u}_i(t) \int_0^t e^{\mu_i(t-s)} \vec{v}_i^T(s)\vec{b}_w(s)ds \\ &= \sum_{i=1}^n \vec{u}_i(t) \left(e^{\mu_i t} \vec{v}_i^T(0)\vec{w}_0 + \int_0^t e^{\mu_i(t-s)} \vec{v}_i^T(s)\vec{b}_w(s)ds \right), \end{aligned} \quad (8)$$

where \vec{w}_0 is the initial condition, $\Phi(t, s) = U(t)D(t-s)V^T(s)$ is the *state transition matrix* of (7), μ_i s are the *Floquet exponents*, $D(t) = \text{diag}(e^{\mu_1 t}, \dots, e^{\mu_n t})$, $U(t) = [\vec{u}_1(t), \dots, \vec{u}_n(t)]$, $V(t) = [\vec{v}_1(t), \dots, \vec{v}_n(t)]$, and $V^T(t)U(t) = I_n$. More theories and proofs about LPTV systems can be found in [13], and are omitted here due to page constraints.

We now introduce a lemma showing that $\frac{d\vec{x}_S(t)}{dt}$, the time derivative of the periodic solution $\vec{x}_S(t)$ of (5), satisfies an LPTV system.

Lemma 3.1: The time derivative of the periodic solution $\vec{x}_S(t)$ of (5), i.e., $\frac{d}{dt}(\vec{x}_S(t))$ satisfies

$$\frac{d}{dt}\vec{w}(t) = G(t)\vec{w}(t) + \frac{d\vec{b}_S(t)}{dt}, \quad (9)$$

and can be written as

$$\frac{d\vec{x}_S(t)}{dt} = U(t)\vec{c}(t) = \sum_{i=1}^n \vec{u}_i(t)c_i(t), \quad (10)$$

where

$$c_i(t) = \lim_{t \rightarrow \infty} \left(e^{\mu_i t} \vec{v}_i^T(0) \frac{d\vec{x}_S(t)}{dt} + \int_0^t e^{\mu_i(t-s)} \vec{v}_i^T(s) \left(\frac{d}{ds} \vec{b}_S(s) \right) ds \right). \quad (11)$$

Proof: Since $\vec{x}_S(t)$ satisfies (5), we have

$$\frac{d}{dt}\vec{x}_S(t) = \vec{f}(\vec{x}_S(t)) + \vec{b}_S(t). \quad (12)$$

Take the time derivative of (12) on both sides, we obtain

$$\frac{d}{dt} \left(\frac{d\vec{x}_S(t)}{dt} \right) = \frac{d}{dt} (\vec{f}(\vec{x}_S(t)) + \vec{b}_S(t)) = \frac{\partial \vec{f}}{\partial \vec{x}} \Big|_{\vec{x}_S(t)} \frac{d\vec{x}_S(t)}{dt} + \frac{d\vec{b}_S(t)}{dt}. \quad (13)$$

Therefore, $\frac{d}{dt}(\vec{x}_S(t))$ satisfies (9).

Since $\frac{d}{dt}\vec{x}_S(t)$ is the asymptotic periodic solution to (9), according to (8), we further have (10) with $\vec{c}(t)$ defined by (11). ■

B. Main Results via Nonlinear Perturbation Analysis

With the important assumption that the trajectory of the perturbed system stays close to the trajectory of $\vec{x}_S(t)$, the key idea is to show that under the perturbed input $\vec{b}_p(t)$, the perturbed response $\vec{x}_p(t)$ can be decomposed into the phase response $z(t)$ and the amplitude deviation $\vec{y}(t)$ in a reasonable way, i.e.,

$$\vec{x}_p(t) = \vec{x}_S(t + z(t)) + \vec{y}(t), \quad (14)$$

and that by defining the right differential equation for $z(t)$, $\vec{y}(t)$ is minimized in some sense.

To show this, we start by defining the *phase equation*, i.e., the differential equation for the phase response $z(t)$. We then show that the input $\vec{b}_p(t)$ can be decomposed into $\vec{b}_z(t)$ and $\vec{b}_y(t)$ such that when only $\vec{b}_z(t)$ is applied to (5), the perturbed response is exactly $\vec{x}_S(t + z(t))$. We then derive the first-order approximation of $\vec{y}(t)$ by linearizing original differential equations around the phase-shifted solution $\vec{x}_S(t + z(t))$, and show that $\vec{y}(t)$ is minimized in some sense.

Definition 3.1: We define the phase equation to be

$$\vec{c}^T(t+z)\vec{c}(t+z) \frac{dz}{dt} = \vec{c}^T(t+z)V^T(t+z) [\vec{b}_p(t) - \vec{b}_S(t+z)], \quad (15)$$

where $\vec{c}(t)$ is defined in (11) and $V(t)$ is defined in (8).

²Note that there is a crucial difference between this assumption and the assumption that the trajectories of $\vec{x}_p(t)$ and $\vec{x}_S(t)$ stay close to each other.

With the definition of $z(t)$, we present a theorem showing that part of the input $\vec{b}_p(t)$ contributes only to the phase response.

Theorem 3.2: Given any perturbed input $\vec{b}_p(t)$, define

$$\begin{aligned} \vec{b}_z(t) = & \vec{b}_S(t+z) + \frac{\vec{c}^T(t+z)}{\vec{c}^T(t+z)\vec{c}(t+z)} V^T(t+z) \\ & \cdot [\vec{b}_p(t) - \vec{b}_S(t+z)] U(t+z)\vec{c}(t+z), \end{aligned} \quad (16)$$

then $\vec{x}_S(t+z(t))$ is the solution to

$$\frac{d}{dt}(\vec{x}(t)) = \vec{f}(\vec{x}(t)) + \vec{b}_z(t). \quad (17)$$

With this input decomposition, it remains to show that by decomposing the perturbed response $\vec{x}_p(t)$ into phase response and amplitude deviation according to (15), the amplitude deviation is minimized in some sense. This is proven by the following theorem. The proof also gives another derivation of the phase equation (15).

Theorem 3.3: Suppose the perturbed response is $\vec{x}_p(t) = \vec{x}_S(t+z(t)) + \vec{y}(t)$, then to the first-order approximation, $\vec{y}(t)$ is

$$\vec{y}(t) = U(\tau) \int_0^\tau e^{\Lambda(\tau-s)} \vec{r}(s) ds, \quad (18)$$

where $\tau = t+z$, $\Lambda = \text{diag}(\mu_1, \dots, \mu_n)$ are the Floquet exponents of (7), and $\vec{y}(t)$ is minimized in the sense that $\|\vec{r}(s)\|_2$ is minimized.

Sketch of proof: Using the input decomposition in Theorem 3.2, we can perform a perturbation analysis of the original nonlinear system around its periodic solution. Assuming $\vec{x}_p(t) = \vec{x}_S(t+z) + \vec{y}(t)$, we can derive an LPTV system in terms of $\vec{y}(t)$. By applying Floquet theory (*i.e.*, (8)), we obtain (18), where

$$\vec{r}(t) = V^T(t+z) (\vec{b}_p(t) - \vec{b}_S(t+z)) - \frac{dz}{dt} \vec{c}(t+z). \quad (19)$$

The minimization of $\|\vec{r}(t)\|_2$ boils down to the problem of minimizing $\|A\vec{x} - \vec{b}\|_2$ where $A = \vec{c}(t+z)$, $x = a(t)$ and $\vec{b} = V^T(t+z) [\vec{b}_p(t) - \vec{b}_S(t+z)]$. The solution to this problem is simply

$$\frac{dz}{dt} = \frac{\vec{c}^T(t+z)}{\vec{c}^T(t+z)\vec{c}(t+z)} V^T(t+z) (\vec{b}_p(t) - \vec{b}_S(t+z)), \quad (20)$$

which again leads to (15). \blacksquare

C. Algorithm

Based on the above analysis, the algorithm to generate a general-purpose phase macromodel is shown in Algorithm 1.

Algorithm 1 Generalized Phase Macromodeling

- 1: Given input $\vec{b}_S(t)$, compute the periodic steady state $\vec{x}_S(t)$.
 - 2: Compute $\frac{d\vec{x}_S(t)}{dt}$, the time derivative of $\vec{x}_S(t)$.
 - 3: Perform Floquet decomposition of the LPTV system $\frac{d\vec{w}}{dt} = G(\vec{x}_S(t))\vec{w}(t)$ (compute $U(t)$, $V(t)$, and $D(t)$).
 - 4: Compute $\vec{c}(t) = V^T(t) \frac{d\vec{x}_S(t)}{dt}$.
 - 5: Precompute $\vec{c}^T(t)\vec{c}(t)$ and $\vec{c}^T(t)V^T(t)$ and store them in the model.
-

D. Interpretation in the Projection Framework

The model shown in Section III-B can be interpreted as a special reduced order model in the projection framework mentioned in Section II-B, where state variables are forced to lie in a q -dimensional manifold defined by $\vec{x} = \vec{v}(\vec{z})$, and the residual is forced to lie in a q -dimensional manifold defined by $\vec{r}_z = \vec{w}(\vec{r}_x)$.

To derive the same phase macromodel under the projection framework, we first enforce that the perturbed response $\vec{x}_p(t)$ only induces a phase shift, *i.e.*, the state variable lie on the same orbit as that of the unperturbed solution. This is equivalent to say that we project the state space onto the one-dimensional nonlinear manifold defined by

the unperturbed solution $\vec{x}_S(t)$ which has a natural parameterization by time t . Therefore, the projection function is defined by

$$\vec{x}(t) = \vec{v}(z(t)) = \vec{x}_S(t+z). \quad (21)$$

Substituting (21) in (5), we have

$$\frac{d}{dt} \vec{x}_S(t+z) = \vec{f}(\vec{x}_S(t+z)) + \vec{b}_p(t). \quad (22)$$

Therefore,

$$\begin{aligned} \frac{d}{dt} [\vec{x}_S(t+z)] &= \frac{d\vec{x}_S}{d\tau}(\tau) \frac{d\tau}{dt} = \frac{d\vec{x}_S}{d\tau}(\tau) \left(1 + \frac{dz}{dt}\right) \\ &= \frac{d\vec{x}_S}{d\tau}(\tau) + \frac{d\vec{x}_S}{d\tau}(\tau) \frac{dz}{dt}. \end{aligned} \quad (23)$$

On the other hand, we have

$$\frac{d}{d\tau} \vec{x}_S(\tau) = \vec{f}(\vec{x}_S(\tau)) + \vec{b}_S(\tau), \quad (24)$$

and therefore, (23) can be written as

$$\frac{d}{dt} \vec{x}_S(t+z) = \vec{f}(\vec{x}_S(\tau)) + \vec{b}_S(\tau) + \frac{d}{d\tau} \vec{x}_S(\tau) \frac{dz}{dt}. \quad (25)$$

Combining (22) and (25), we obtain

$$\frac{d}{d\tau} \vec{x}_S(\tau) \frac{dz}{dt} = \vec{b}_p(t) - \vec{b}_S(\tau), \quad (26)$$

or equivalently,

$$U(\tau)\vec{c}(\tau) \frac{dz}{dt} = \vec{b}_p(t) - \vec{b}_S(\tau). \quad (27)$$

Therefore, after the projection onto the periodic orbit, we obtain n equations (27) and 1 unknown variable z . To derive a reduced order model, we further project the residual, *i.e.*, (27), to a one-dimensional manifold. In our model, we have chosen the projection to be

$$\vec{w}(\vec{r}_x) = \vec{c}^T(\tau) V^T(\tau) \vec{r}_x, \quad (28)$$

which is shown to minimize the amplitude deviation in the sense of Theorem 3.3.

E. Connections to PPV Macromodel

The PPV phase macromodel [12] is specifically devised to characterize the phase response of an oscillatory system. Its limitation is that it is only applicable to oscillators. This excludes it from wide applications such as traditional timing analysis and excitory firing neuron network simulations.

Following previous derivations, the PPV macromodel is indeed a special case of our generalized phase macromodel. We can obtain the PPV model for oscillators by choosing $\vec{c}(t)$ in (15) to be $[1, 0, \dots, 0]^T$.

This choice of $\vec{c}(t)$ in fact is the definition in (11) for autonomous oscillators. To see that, we note that for autonomous oscillators, the following two conditions are typically satisfied: (1) one Floquet exponent of (9) to be 0 and others negative (*e.g.*, $\mu_1 = 0$, $\mu_i < 0$, $2 \leq i \leq n$), and (2) $\vec{b}_S(t)$ is constant. Therefore, (11) becomes

$$\begin{aligned} c_1(t) &= \lim_{t \rightarrow \infty} \vec{v}_i^T(0) \frac{d\vec{x}_S}{dt}(0) = 1, \\ c_i(t) &= \lim_{t \rightarrow \infty} e^{\mu_i t} \vec{v}_i^T(0) \frac{d\vec{x}_S}{dt}(0) = 0, \quad 2 \leq i \leq n, \end{aligned} \quad (29)$$

which leads to $\vec{c}(t) = [1, 0, \dots, 0]^T$, and the PPV model.

F. Generalization for Non-Periodic Trajectories

Above derivations are based on the assumption that $\vec{x}_S(t)$ is the asymptotic periodic orbit in the state space. This is not generally the case for some systems such as digital circuits. However, if we force the input to be a periodic signal (for example, a square wave), then the system will converge to a periodic steady state, and transient behaviors (for example, the rising and falling edge of waveforms)

are well-approximated by its periodic steady state. Hence, we can force the input to be periodic as long as the periodic trajectory well-approximates the non-periodic ones. As a result, the derived phase model will also characterize the transient phase responses.

IV. NUMERICAL METHODS FOR COMPUTING THE PHASE MODEL

As shown in algorithm 1, to compute the phase macromodel, we need to compute the periodic steady state $\bar{x}_S(t)$, its time derivative $\frac{d\bar{x}_S(t)}{dt}$, and perform a full Floquet decomposition of an LPTV system.

PSS analysis has been widely adopted in commercial tools and usually uses shooting method [15], FDTD method [15] or harmonic balance [15]. The time derivative of the PSS solution can be computed either in the time domain directly by finite difference approximation or in the frequency domain and be transformed back to time domain.

The main numerical challenge lies in the Floquet decomposition, which is numerically much more ill-conditioned. Implementations according to textbook definitions using Monodromy matrix fail without question. In this section, we first demonstrate that methods based on explicitly computing the Monodromy matrix are numerically extremely bad. We then present a much better-behaved method to solve this challenge, based on harmonic balance method.

A. Floquet Decomposition using Monodromy Matrix

We consider the Floquet decomposition of the LPTV system

$$\frac{d}{dt}\bar{x}(t) + G(t)\bar{x}(t) = 0. \quad (30)$$

Floquet decomposition [13] refers to computing matrices $U(t)$, $V(t)$ and $D = e^{\Lambda t}$ (where Λ is a diagonal matrix of all Floquet exponents) such that the state transition matrix of (30) is

$$\Phi(t, s) = U(t)D(t-s)V^T(s). \quad (31)$$

The definition for Floquet exponents are derived from the eigenvalues of the *monodromy matrix* defined by

$$B = \Phi(T, 0) = Pe^{\Lambda T}P^{-1}, \quad (32)$$

and $U(t)$ is then defined as $\Phi(t, 0)Pe^{-\Lambda t}$, and $V(t) = U^{-1}(t)$.

This definition naturally gives a straightforward implementation of Floquet decomposition:

Algorithm 2 Floquet Decomposition using Monodromy Matrix

- 1: Starting with initial condition $X(0) = I$, integrate (30) to compute $\Phi(t, 0)$ for $t \in [0, T]$.
 - 2: Compute the monodromy matrix $B = \Phi(T, 0)$.
 - 3: Eigen-decompose the monodromy matrix $B = Pe^{\Lambda T}P^{-1}$.
 - 4: Compute $U(t) = \Phi(t, 0)Pe^{-\Lambda t}$ and $V(t) = U^{-1}(t)$.
-

This algorithm will work in perfect precision. However, it almost never works in practice. For real systems, most Floquet exponents are normally negative. This means that using any initial condition P , the solution to (30) (which is $\Phi(t, 0)P$) tends to 0 exponentially. This makes the computed monodromy matrix B extremely numerically singular, and the eigen-decomposition of B provides useless results. Also, computing $V(t)$ using the inverse of $U(t)$ is not a good choice since the smoothness of $V(t)$ can easily be destroyed.

B. Equations in terms of $U(t)$ and $V(t)$

As the first and the most important step to tackle the problems in the Monodromy matrix method, we derive equations whose unknowns are $U(t)$ and $V(t)$, respectively. Because $U(t)$ and $V(t)$ are periodic, the resulting differential equations have solutions that are periodic, instead of decaying to 0. Therefore, this avoids numerical errors in the Monodromy matrix computation. Moreover, having equations for $U(t)$ and $V(t)$ separately avoids computing $V(t) = U^{-1}(t)$.

To derive these equations, we start with a fundamental matrix

$$X(t) = \Phi(t, 0) = U(t)D(t)V^T(0), \quad (33)$$

which solves (30). Substituting (33) in (30), we have

$$\begin{aligned} & \frac{d}{dt} [U(t)D(t)V^T(0)] + G(t)U(t)D(t)V^T(0) = 0 \\ \Leftrightarrow & \frac{d}{dt} [U(t)D(t)] + G(t)U(t)D(t) = 0 \\ \Leftrightarrow & \left[\frac{d}{dt} U(t) \right] e^{\Lambda t} + U(t) \left[\frac{d}{dt} e^{\Lambda t} \right] + G(t)U(t)e^{\Lambda t} = 0. \end{aligned} \quad (34)$$

Therefore,

$$\frac{d}{dt} U(t) + U(t)\Lambda + G(t)U(t) = 0. \quad (35)$$

Similarly, to derive the equations for $V(t)$, we apply the same technique to the adjoint system of (30)

$$\frac{dy(t)}{dt} - G^T(t)y(t) = 0, \quad (36)$$

which has a fundamental matrix

$$Y(t) = V(t)D(-t)U^T(0). \quad (37)$$

Substituting (37) in (36), we have

$$\begin{aligned} & \frac{d}{dt} (V(t)D(-t)) - G^T(t)V(t)D(-t) = 0 \\ \Leftrightarrow & \left(\frac{d}{dt} V(t)e^{-\Lambda t} + V(t)(-\Lambda)e^{-\Lambda t} \right) - G^T(t)V(t)e^{-\Lambda t} = 0. \end{aligned} \quad (38)$$

Therefore,

$$\frac{d}{dt} V(t) - V(t)\Lambda - G^T(t)V(t) = 0. \quad (39)$$

Therefore, we have shown that $U(t)$ and $V(t)$ satisfy differential equations (35) and (39), respectively. To solve for $U(t)$ and $V(t)$, we note that they are by definition periodic with period T , and therefore we need to *force* their periodicity during the computation. A well-known numerical method that forces the periodicity is HB, and we now show how to apply HB to solve for $U(t)$ and $V(t)$.

C. Floquet Decomposition via Harmonic Balance

1) *Notations:* We follow the notations in [16]: For any periodic matrix or vector $A(t)$ which has a truncated Fourier series $A(t) = \sum_{i=-M}^M A_i e^{j\omega t}$ ($\omega = \frac{2\pi}{T}$), we define the block vector $V_{A(t)}$ of A_i s and the block-Toeplitz matrix $T_{A(t)}$ of A_i s to be

$$V_{A(t)} = \begin{bmatrix} \vdots \\ \vdots \\ A_2 \\ A_1 \\ A_0 \\ A_{-1} \\ A_{-2} \\ \vdots \\ \vdots \end{bmatrix}, \quad T_{A(t)} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & A_0 & A_1 & A_2 & A_3 & A_4 & \cdots \\ \cdots & A_{-1} & A_0 & A_1 & A_2 & A_3 & \cdots \\ \cdots & A_{-2} & A_{-1} & A_0 & A_1 & A_2 & \cdots \\ \cdots & A_{-3} & A_{-2} & A_{-1} & A_0 & A_1 & \cdots \\ \cdots & A_{-4} & A_{-3} & A_{-2} & A_{-1} & A_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \quad (40)$$

Using this notation, two lemmas follow [16]:

Lemma 4.1: If $Z(t) = X(t)Y(t)$, and $X(t)$, $Y(t)$ are T -periodic, then,

$$\begin{aligned} V_{Z(t)} &= T_{X(t)}V_{Y(t)}, & V_{Z^T(t)} &= T_{Y^T(t)}V_{X^T(t)}, \\ T_{Z(t)} &= T_{X(t)}T_{Y(t)}, & T_{Z^T(t)} &= T_{Y^T(t)}T_{X^T(t)}. \end{aligned} \quad (41)$$

Lemma 4.2: If $X(t)$ is T -periodic, then

$$V_{\dot{X}(t)} = \Omega V_{X(t)}, \quad T_{\dot{X}(t)} = \Omega T_{X(t)} - T_{X(t)}\Omega, \quad (42)$$

where $\Omega = \Omega_0 \otimes I_n$, $\Omega_0 = j\omega \text{diag}(\cdots, 2, 1, 0, -1, -2, \cdots)$. (43)

2) *HB for Floquet decomposition:*

Theorem 4.3: The eigenvalues and eigenvectors of the HB Jacobian matrix for (30) ($J_{HB} = \Omega + T_{G(t)}$) are given by the diagonal of

$\Omega_0 \otimes I_n - I_{2M+1} \otimes \Lambda$ and $T_{U(t)}$, respectively.

The eigenvalues and eigenvectors of the HB Jacobian matrix for (36) ($\hat{J}_{HB} = \Omega - T_{G^T(t)}$) are given by the diagonal of $\Omega_0 \otimes I_n + I_{2M+1} \otimes \Lambda$ and $T_{V(t)}$, respectively.

Proof: Applying lemmas 4.1 and 4.2 to (35), we have

$$\begin{aligned} \Omega T_{U(t)} - T_{U(t)} \Omega + T_{U(t)} T_\Lambda + T_{G(t)} T_{U(t)} &= 0 \\ \Leftrightarrow (\Omega + T_{G(t)}) T_{U(t)} + T_{U(t)} (T_\Lambda - \Omega) &= 0, \end{aligned} \quad (44)$$

which is already in the eigen-decomposed form of $J_{HB} = \Omega + T_{G(t)}$.

Similarly, applying lemmas 4.1 and 4.2 to (39), we have

$$\begin{aligned} (\Omega T_{V(t)} - T_{V(t)} \Omega) - T_{V(t)} T_\Lambda - T_{G^T(t)} T_{V(t)} &= 0 \\ \Leftrightarrow (\Omega - T_{G^T(t)}) T_{V(t)} - T_{V(t)} (\Omega + T_\Lambda) &= 0, \end{aligned} \quad (45)$$

which is already in the eigen-decomposed form of $\hat{J}_{HB} = \Omega - T_{G^T(t)}$. ■

Therefore, solving for $U(t)$ and $V(t)$ boils down to the matrix eigenvalue problem of J_{HB} and \hat{J}_{HB} . To pick up the right eigenvectors for $U(t)$ and $V(t)$, we note that eigenvalues of two matrices are $\lambda_{i,k} = j\omega \mp \mu_k$, $-M \leq i \leq M$, $1 \leq k \leq n$. and the block vector $V_{U(t)}$ and $V_{V(t)}$ correspond to eigenvalues $\lambda_{0,k} = \mu_k$. Therefore, assuming μ_k s are real, we can pick up the eigenvectors corresponding to real eigenvalues to obtain $V_{U(t)}$ and $V_{V(t)}$. This leads to the following algorithm:

Algorithm 3 Floquet Decomposition via Harmonic Balance

- 1: Construct Jacobian matrices J_{HB} and \hat{J}_{HB} in Thm. 4.3.
 - 2: Eigen-decompose J_{HB} and \hat{J}_{HB} .
 - 3: Select real eigenvalues to be the Floquet exponents μ_k s, and eigenvectors corresponding to μ_k s to be $V_{U(t)}$ and $V_{V(t)}$.
 - 4: Perform inverse FFT to compute $U(t)$ and $V(t)$ in time-domain.
-

V. VALIDATION AND APPLICATIONS

In this section, we illustrate applications of the previously developed phase macromodeling technique to several nonlinear systems.

We first look at a simple two-dimensional nonlinear system in detail – we present results on phase macromodel construction, and show that our numerical methods to compute the phase macromodel (which mainly involves the full Floquet decomposition) have superior accuracy over the straightforward Monodromy matrix method. We then compare the simulation results of the phase model to those of the full model, and analyze the results for different types of perturbations, including perturbations on amplitude, frequency, phase and initial condition (impulse perturbation).

Following this case study, we present results on a firing neuron model and an inverter chain circuit, where special types of perturbations that are of interest in practice are discussed.

A. A Simple Nonlinear System

The first nonlinear system we consider is described by

$$\begin{aligned} \frac{d}{dt} x_1 + x_1 + 1000(x_1 - x_2)^3 - b(t) &= 0 \\ \frac{d}{dt} x_2 + x_2 - 1000(x_1 - x_2)^3 &= 0. \end{aligned} \quad (46)$$

The (nominal) periodic input $b_S(t)$ is set to $b_S(t) = \cos(2\pi t)$, and the periodic steady state $\bar{x}_S(t)$ is solved and shown in Fig. 2.

To derive the phase model described in Section III, we perform a full Floquet decomposition of the LPTV system obtained by linearizing (46) around its periodic steady state $\bar{x}_S(t)$. We apply the method described in Section IV which boil down to solve an eigenvalue problem of the HB Jacobian matrix. The computed eigenvalues of the HB Jacobian matrix (using 45 harmonics) for the LPTV system and its adjoint system are depicted in Fig. 3. From Fig. 3, we see that eigenvalues are approximately $\pm 1 + j2\pi$ and $\pm 20.68 + j2\pi$ ($i = -45, \dots, 45$). These eigenvalues match the theoretical results,

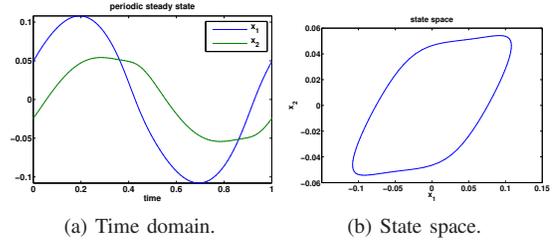


Fig. 2. Periodic steady state of (46).

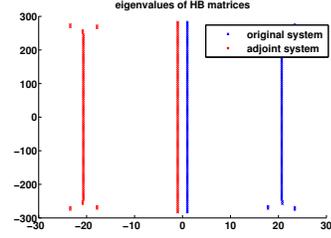


Fig. 3. Eigenvalues of HB Jacobians.

although the eigenvalues at the end of the line $x = \pm 20.68$ are off a little bit.

The eigenvectors of these matrices corresponding to the Floquet exponents -1 and -20.68 are then extracted to construct $U(t)$ and $V(t)$, which are plotted in Fig. 4(a) and Fig. 4(b), respectively. To examine the bi-orthonormality of the two matrices, we also plot the matrix $V^T(t)U(t)$ in Fig. 4(c). From Fig. 4(c), we see that $v_1^T(t)u_1(t)$ and $v_2^T(t)u_2(t)$ equal to constant 1, and $v_1^T(t)u_2(t)$ and $v_2^T(t)u_1(t)$ are much less than 10^{-12} . Therefore, the bi-orthonormality is preserved.

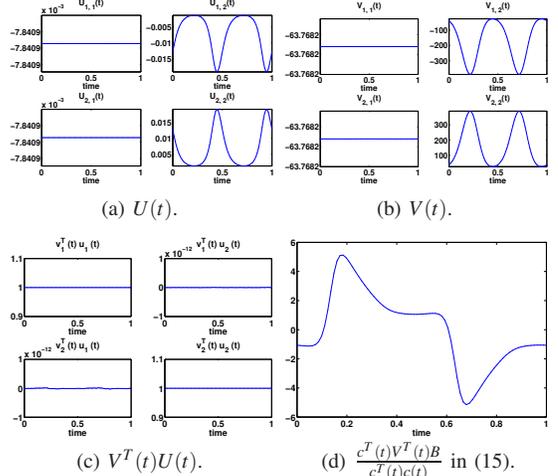


Fig. 4. Floquet decomposition and phase macromodel construction.

We have not shown results using the Monodromy matrix method, since these results do not make sense. For the fact that this system (as well as almost all practical systems) is stable and Floquet exponents are all negative, the Monodromy matrix is numerically extremely singular, and the eigen-decomposition produces meaningless results.

Using results from Floquet decomposition, we construct the phase macromodel, *i.e.*, the phase equation (15). The time-varying function $\frac{c^T(t)V^T(t)B}{c^T(t)c(t)}$ (assuming input $\vec{b}(t) = Bb(t)$) is plotted in Fig. 4(d). This key function models the nonlinearities that contribute to the phase response, and can be viewed as the sensitivity of inputs to the phase response. From Fig. 4(d), we can infer that inputs applied at time $t \simeq 0.2$ and $t \simeq 0.7$ have the most significant impact on the phase response.

We now examine results of our phase model under different types of input perturbations. Although any perturbation is applicable to the model, we specifically consider four types of perturbations here: for a periodic input $\vec{b}_S(t)$, we consider phase perturbation $\vec{b}_S(t) +$

p), frequency perturbation $\vec{b}_S(f_t)$, amplitude perturbation $A\vec{b}_S(t)$, and impulse perturbation $\vec{b}_S(t) + D\delta(t)$.

The simulation results under a constant phase perturbation $\vec{b}_p(t) = \vec{b}_S(t + 0.4)$ with initial condition $z = 0$ are plotted in Fig. 5. Fig. 5(a) depicts the waveform of $z(t)$, which shows that the phase is converging to $z = -0.6$. This corresponds to the time-domain solution $\vec{x}_S(t + 0.4)$. Fig. 5(b) plots the time-domain waveforms of x_1 and x_2 . It is observed that the initial transient behaviors are not matched exactly, but the waveforms stay quite close. To better understand the dynamics, we examine the time-evolution of state variables in the state space, which is depicted in the $x-t$ space in Fig. 5(c). It is observed that state variables of the phase model are confined on the orbit of $\vec{x}_S(t)$, but they try to follow the correct trajectory by choosing almost the closest point on the orbit of $\vec{x}_S(t)$. The simulation results for another initial condition $z = 0.2$ is also plotted in the $x-t$ space in Fig. 5(d), and this is equivalent to applying an impulse function at time $t = 0$. Similar behaviors of the phase model are observed, and it finally converges to the correct asymptotic response. Indeed, if a constant phase-shifted input $\vec{b}_S(t + p)$ is applied, no matter what the initial condition is, our phase model will reproduce the correct phase-shifted asymptotic output. This can be seen by the fact that $z(t) = p$ is a fixed point of (15).

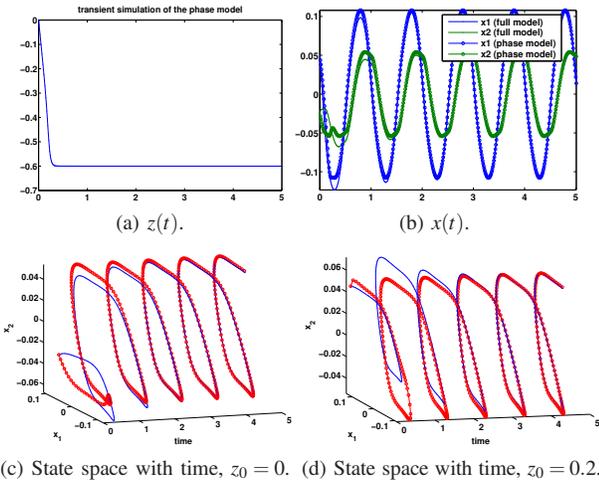


Fig. 5. Transient simulation of the phase model when $b(t) = \cos(2\pi(t + 0.4))$. In Fig. 5(c) and Fig. 5(d), red(circled): phase model, Blue(solid): full model.

We then make the phase perturbation time-varying – we apply an PM signal $b_p(t) = \cos(2\pi(t + 0.1 \sin(0.2\pi t)))$, and the simulation results for 100 cycles are shown in Fig. 6. It is seen in Fig. 6(b) that the response of the full model almost lies on the periodic orbit of $\vec{x}_S(t)$, and therefore, the phase model works perfectly.

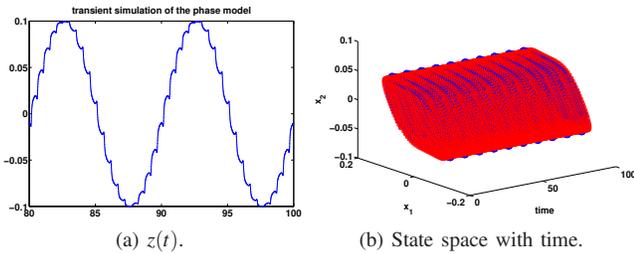


Fig. 6. Transient simulation when $b_p(t) = \cos(2\pi(t + 0.1 \sin(0.2\pi t)))$. In Fig. 6(b), red(circled): phase model, Blue(solid): full model.

Now we apply a frequency perturbed input $\vec{b}_p(t) = \vec{b}_S(1.2t)$, and the simulation results are shown in Fig. 7. It is seen that the periodic orbit has a large deviation from that of $\vec{x}_S(t)$, and the phase model is doing its best to approximate the right trajectory using points on $\vec{x}_S(t)$. Most importantly, although the resulting time-domain waveforms do not match exactly, the timing information is captured – the frequency of the output waveform is 1.2 which is the same as that of $\vec{b}_p(t)$. Also note that the frequency perturbation can be interpreted as a

phase perturbation $p(t) = 0.2t$ which can grow unboundedly as time evolves. This shows that the perturbation can be arbitrary large as long as the underlying assumption (that the trajectory does not change much) is satisfied.

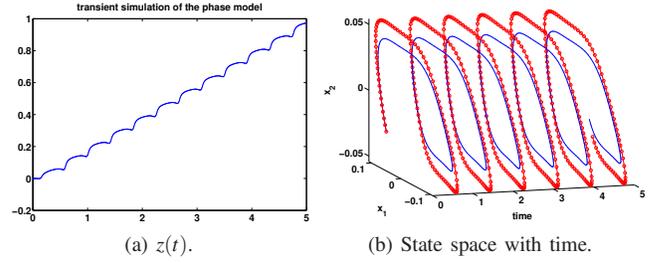


Fig. 7. Transient simulation of the phase model when $b_p(t) = \cos(2\pi(1.2t))$. In Fig. 7(b), red(circled): phase model, Blue(solid): full model.

Then an amplitude perturbed signal $\vec{b}_p(t) = 1.2\vec{b}_S(t)$ is applied, and the simulation results are shown in Fig. 8. Similar to previous results, the periodic orbit deviates from that of $\vec{x}_S(t)$, and the phase model produces a reasonably well-approximated waveforms. Note that in many applications such as digital circuits and neuron models, voltage/potential waveforms reach a saturated value in the nominal periodic solution $\vec{x}_S(t)$. This fact makes the periodic orbit insensitive to the amplitude perturbation, and therefore the assumption that the trajectory stays close to $\vec{x}_S(t)$ is satisfied. Therefore, the phase model generates good results in these cases, as we will show in next two examples.

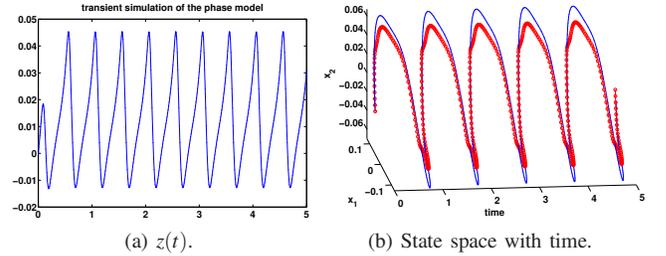


Fig. 8. Transient simulation of the phase model when $b_p(t) = 1.2 \cos(2\pi t)$. In Fig. 8(b), red(circled): phase model, Blue(solid): full model.

We have not provided results of LTI and LPTV reduced models since they simply produce meaningless results. The assumption that the additive perturbation input must be *small* is generally not satisfied. Specifically, in the case of the phase and frequency perturbations, suppose the magnitude of $b_S(T)$ is A , then the corresponding additive perturbation $\Delta b(t) = b_S(t + p(t)) - b_S(t)$ can have a magnitude of $2A$, and this usually breaks the small signal assumption.

The speedups are not easy to measure considering various factors including memory allocation and MATLAB optimization. For the examples we show in this paper, the measured transient runtime speedup is generally about $10 \times$ to $20 \times$. However, since the size of the phase model is 1, the speedups are expected to be much larger for larger systems, similar to PPV models [12].

B. A Firing Neuron Model

The firing neuron model we consider is known as Morris-Lecar model [17]. The differential equations for a single neuron are

$$\begin{aligned} \frac{d}{dt}V &= \frac{1}{C_M} (-g_L(V - V_L) - g_{Ca}M_\infty(V - V_{Ca}) - g_KN(V - V_K) + I) \\ \frac{d}{dt}N &= (N_\infty - N) \frac{1}{\tau_N}, \end{aligned} \quad (47)$$

where $M_\infty = 0.5(1 + \tanh((V - V_1)/V_2))$, $N_\infty = 0.5(1 + \tanh((V - V_3)/V_4))$, $\tau_N = 1/(\Phi \cosh((V - V_3)/(2V_4)))$. The input is the injection current I , and other parameters are chosen as $C_M = 20$, $g_K = 8$, $g_L = 2$, $V_{Ca} = 120$, $V_K = -80$, $V_L = -60$, $V_1 = -1.2$, $V_2 = 18$, $V_4 = 17.4$, $g_{Ca} = 4$, $\Phi = 1/15$, $V_3 = 12$, which are adapted from [17].

Using this set of parameters, the neuron is not self-oscillatory, and therefore the PPV model for oscillators is not applicable. However,

the neuron fires when adequate currents are injected. We set the input current to be a pulse signal shown in Fig. 9(a). The phase macromodel of this neuron is computed, and the time-varying function $\frac{c^T(t)V^T(t)B}{c^T(t)c(t)}$ is plotted in Fig. 10(a).

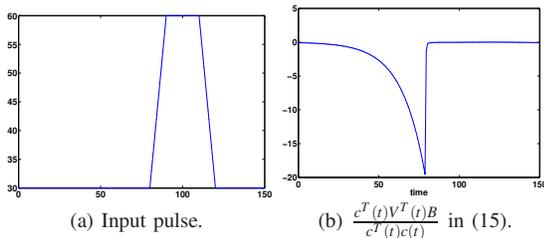


Fig. 9. Input signal and the phase model.

We then apply a pulse input whose magnitude is twice of the original one, and the simulation results are plotted in Fig. 10. It is seen that although the input amplitude is doubled, the periodic orbit only deviates a little bit, and the phase model results almost match those of the full model.

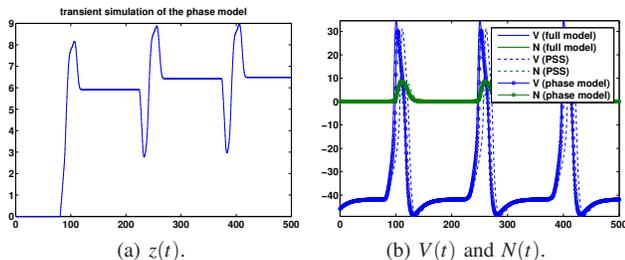


Fig. 10. Transient simulation under amplitude perturbation.

C. An Inverter Chain

We now apply the phase macromodeling technique to a nine-stage inverter chain circuit. The nominal periodic input is set to a square wave shown in Fig. 11(a), and the time-varying function $\frac{c^T(t)V^T(t)B}{c^T(t)c(t)}$ is plotted in Fig. 11(b). This circuit is a typical “digital circuit” in the sense that each node voltage settles to a saturated value in a short time, and this makes the trajectory in the state space less sensitive to input perturbations. However, the timing properties are sensitive to input perturbations.

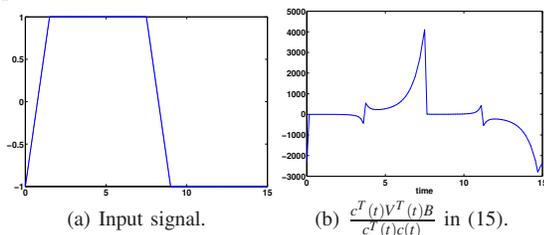


Fig. 11. Input signal and the phase model.

We have tried the several kinds of perturbations and similar results to those of previous examples are observed. Here, we show a special perturbation that is of interest in timing analysis – the perturbation is on the slope of the input ramp. In timing analysis and library generation, people normally compute a look-up table storing a map from input slope to output delay and slope. We show that our phase model reproduces accurate timing properties.

In Fig. 12, we plot the transient simulation results of the phase model and the full model when the slope of the rising edge is 0.6 (instead of 0.5 for the nominal input). It is observed that delays of waveforms change due to perturbation, and the phase model matches that of the full model.

VI. CONCLUSION

We have developed a generalized phase macromodeling technique both via nonlinear perturbation analysis and in projection framework.

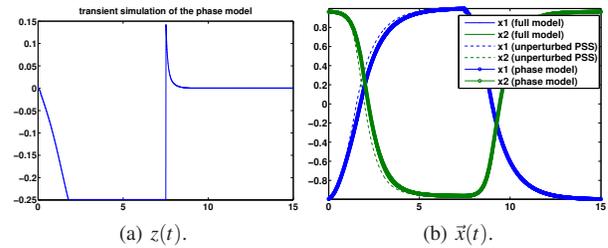


Fig. 12. Transient simulation under slope perturbation.

We have also devised a numerical method for computing the model, and it can be readily implemented in current RF simulators. We have shown several applications of this general phase model, and demonstrated that phase/timing responses are well-characterized by the macromodel.

ACKNOWLEDGMENTS

The authors would like to thank Alper Demir for fruitful discussions. The authors would also like to thank reviewers for constructive comments and suggestions.

REFERENCES

- [1] E.J. Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, University of Illinois, EE Dept, Urbana-Champaign, 1997.
- [2] A. Odabasioglu, M. Celik, and L.T. Pileggi. PRIMA: passive reduced-order interconnect macromodelling algorithm. In *Proceedings of the IEEE International Conference on Computer Aided Design*, pages 58–65, November 1997.
- [3] J. Roychowdhury. Reduced-order modelling of time-varying systems. *IEEE Trans. Ckts. Syst. – II: Sig. Proc.*, 46(10), November 1999.
- [4] J. R. Phillips. Projection-Based Approaches for Model Reduction of Weakly Nonlinear Time-Varying Systems. *IEEE Transactions on Computer-Aided Design*, 22(2):171–187, 2003.
- [5] P. Li and L. T. Pileggi. NORM: Compact Model Order Reduction of Weakly Nonlinear Systems. *Proceedings of the IEEE Design Automation Conference*, 2003.
- [6] Chenjie Gu. Qlmor: a new projection-based approach for nonlinear model order reduction. In *ICCAD '09: Proceedings of the 2009 International Conference on Computer-Aided Design*, pages 389–396, New York, NY, USA, 2009. ACM.
- [7] W. Rugh. *Nonlinear System Theory - The Volterra-Wiener Approach*. Johns Hopkins Univ Press, 1981.
- [8] M. Rewienski and J. White. A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices. *IEEE Transactions on Computer-Aided Design*, 22(2), February 2003.
- [9] S.K. Tiwary and R.A. Rutenbar. Faster, parametric trajectory-based macromodels via localized linear reductions. In *Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on*, pages 876–883, Nov. 2006.
- [10] C. Gu and J. Roychowdhury. Model reduction via projection onto nonlinear manifolds, with applications to analog circuits and biochemical systems. In *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, pages 85–92, 10-13 Nov. 2008.
- [11] Eugene M. Izhikevich. *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting (Computational Neuroscience)*. The MIT Press, 1 edition, November 2006.
- [12] Alper Demir, Amit Mehrotra, and Jaijeet Roychowdhury. Phase noise in oscillators: a unifying theory and numerical methods for characterization. *IEEE Trans. Circuits Syst. I*, 47:655–674, 2000.
- [13] Earl A. Coddington and Norman Levinson. *Theory of ordinary differential equations [by] Earl A. Coddington [and] Norman Levinson*. McGraw-Hill, New York, 1955.
- [14] Alper Demir. Floquet theory and non-linear perturbation analysis for oscillators with differential-algebraic equations. *International Journal of Circuit Theory and Applications*, 28(2):163–185, 2000.
- [15] K.S. Kundert, J.K. White, and A. Sangiovanni-Vincentelli. *Steady-state methods for simulating analog and microwave circuits*. Kluwer Academic Publishers, 1990.
- [16] A. Demir and J. Roychowdhury. A reliable and efficient procedure for oscillator ppv computation, with phase noise macromodeling applications. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 22(2):188 – 197, feb. 2003.
- [17] Kunichika Tsumoto, Hiroyuki Kitajima, Tetsuya Yoshinaga, Kazuyuki Aihara, and Hiroshi Kawakami. Bifurcations in morris-lecar neuron model. *Neurocomput.*, 69(4-6):293–316, 2006.