

TP-PPV: Piecewise Nonlinear, Time-Shifted Oscillator Macromodel Extraction For Fast, Accurate PLL Simulation

Xiaolue Lai and Jaijeet Roychowdhury

Department of Electrical and Computer Engineering, University of Minnesota

Email: {laixl, jr}@umn.edu

Abstract— We present a novel method for generating small, accurate PLL macromodels that capture transient response and jitter performance with unprecedented accuracy, while offering large speedups. The method extracts and uses a highly accurate oscillator phase macromodel termed the TP-PPV macromodel. The core idea behind the novel extraction procedure is to combine concepts from strongly nonlinear trajectory piecewise macromodeling techniques together with PPV-based time-shifted nonlinear phase macromodels. As a result, TP-PPV generated macromodels offer excellent global as well as local fidelity. These properties are necessary for handling large excursions in PLL control voltages during capture/lock in, e.g., hopping frequency synthesizers. We validate TP-PPV on a 5-stage interpolative ring VCO based PLL and compare results against full simulation, as well as against prior macromodels. We show that, unlike prior macromodels that only work well when the control voltage of the VCO has small excursions, the TP-PPV macromodel provides near-perfect matches against full SPICE-level simulation over a wide range of design scenarios, while achieving speedups of about three orders of magnitude.

I. INTRODUCTION

PLLs [1] are widely used in all mixed-signal and digital systems. Their applications include frequency synthesis, clock and data recovery (CDR), signal modulation and demodulation, low-phase-noise clock generation, etc.. Even though PLLs are familiar to designers, their design still presents significant challenges. Modern PLL design involves trade-offs between various conflicting design metrics such as phase noise/jitter, lock and capture range, acquisition time, etc., for different kinds of applications. Simulation tools are extensively used in design processes for finding a good balance between design metrics to meet the performance requirements. Unfortunately, existing full circuit simulation tools (such as SPICE [2]), are very inefficient for the simulation of PLLs at the transistor level [3]; and this problem worsens when dealing with frequency synthesizers with large divide ratios. It is not uncommon for many months to be required to finalize the design of today's advanced PLLs. As a result, a fast and accurate method for PLL simulation and jitter analysis is of great practical importance for the semiconductor design industry.

To improve simulation efficiency, macromodel based simulation is widely used in PLL designs — full circuit simulation is replaced by the use of small, simple behavioral models to approximate PLL responses [4], resulting in great speedups. In a typical PLL macromodel, the voltage controlled oscillator (VCO) is replaced by a phase-domain oscillator macromodel. Linear VCO macromodels [4]–[6], in which the VCO is represented as a simple linear integrator that converts input voltages to output phases, are widely used due to their conceptual and implementation simplicity. However, the linear approximation of the VCO in phase domain is an oversimplification [7]; as a result, they are not well suited for predicting many important aspects of PLL performance.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD'06, November 5-9, 2006, San Jose, CA

Copyright 2006 ACM 1-59593-389-1/06/0011...\$5.00

The shortcomings of linear VCO macromodels are largely overcome by the use of dynamically *time-shifted nonlinear phase macromodels* [7], [8], which have been successfully used for predicting injection locking [9] and loop non-idealities in PLLs [7]. In a nonlinear VCO macromodel, the VCO is represented by a simple one-dimensional time-shifted nonlinear phase equation [10]. A quantity called the perturbation projection vector (PPV) [10], [11] is the basis of the time-shifted nonlinear phase equation; it can be extracted automatically using numerical methods [10], [12], [13]. The far greater accuracy of the time-shifted nonlinear phase equation stems from its provable correctness for small perturbations [10].

However, despite being nonlinear, time-shifted oscillator macromodels still have the drawback that they are small signal models extracted from linearizing oscillators about their steady-states; they only work well when perturbations to the oscillator are small. This is not a big problem in the many applications where external perturbations signals do remain relatively small. However, in certain PLLs (especially frequency synthesizers), the control voltage of the VCO can undergo large swings, varying from 0 to V_{dd} . Large swings of the control voltage tend to invalidate even the time-shifted nonlinear phase macromodel; indeed, all previous PLL macromodels [7], [8] become unsuitable.

In this paper, we present an algorithm for generating strongly nonlinear phase domain macromodels of VCOs that take into account both time shift and amplitude nonlinearities. Our technique is based on generalizing the time-shifted PPV using piecewise techniques from TPWL and PWP [14], [15]. TPWL/PWP offer good global fidelity [14], especially useful for large control voltage swings, while nonlinear time-shifted phase macromodels ensure provably accurate local fidelity [10] for small perturbations such as power supply noise. By combining both techniques, we obtain a more globally accurate VCO macromodel that predicts oscillator responses well at any control voltage.

In our method, several nonlinear VCO phase macromodels are extracted about different control voltages, then combined together to form a uniform trajectory piecewise PPV-based (TP-PPV) nonlinear phase macromodel. We propose a new modified form for the phase equation of the combination, since the original nonlinear phase equation of [10] is no longer directly applicable when the free-running frequency of the oscillator is not a constant. The net result is a more generally applicable and accurate PLL macromodel than previously possible. This macromodel captures the dynamics of complex phenomena in PLLs, such as step input response, jitter induced by reference noise and low-pass filter noise, etc., and replicates qualitative features from full SPICE-level simulations more accurately than previous techniques, especially when the control voltage has large swings.

We validate the proposed technique using a PLL with an XOR phase detector, a $\div 2$ frequency divider and a 5-stage interpolative ring VCO. We simulate transient responses and jitter of the PLL using our method, and compare to other macromodeling techniques and full SPICE-level circuit simulation. The numerical results obtained confirm that our approach captures the transient responses of the PLL with much better fidelity than previous techniques, achieving, indeed, near-perfect matches with SPICE-level circuit simulation for PLL jitter prediction.

The remainder of the paper is organized as follows. In Section II, we briefly review previous PLL macromodeling techniques and piecewise techniques. In Section III, we present our trajectory piecewise nonlinear oscillator macromodeling approach. In Section IV, we illustrate the proposed technique using a 5-stage interpolative ring VCO based PLL. In Section V, we present simulation results of our macromodel, and compare with previous macromodeling techniques and full SPICE-level simulation.

II. RELEVANT PREVIOUS WORK

In this section, we summarize relevant macromodeling techniques that relate closely to our work, including the PPV-based oscillator and PLL macromodels and trajectory piecewise macromodeling techniques.

A. PPV Based PLL Macromodel

In PPV based PLL macromodels [7], [8], the VCO is modeled using a nonlinear time-shifted oscillator macromodel [10], [16]. [16] postulated that an oscillator's output under perturbation can be expressed as

$$x_p(t) = x_s(t + \alpha(t)) + y(t), \quad (1)$$

where $x_s(t)$ is the steady-state solution of the unperturbed oscillator, while $x_p(t)$ is the solution of the oscillator under perturbation. The effect of the perturbations is decomposed into two parts in the above: a phase shift $\alpha(t)$ and an amplitude deviation $y(t)$.

The amplitude deviation $y(t)$ can be ignored to first order for PLL analysis, since it is small and it is the phase that is of major concern. It was shown in [10] that $\alpha(t)$ is governed by the nonlinear differential equation

$$\dot{\alpha}(t) = \vec{v}_1^T(t + \alpha(t)) \cdot \vec{b}(t), \quad (2)$$

where $\vec{b}(t)$ is a vector of external perturbations to the oscillator and $\vec{v}_1(t)$ is a quantity called the Perturbation Projection Vector (PPV). The PPV can be extracted from the oscillator's steady-state solution using numerical methods [10], [12], [13]. It is a vector of periodic waveforms with the same period as the free-running oscillator. Each waveform represents the “nonlinear phase sensitivity” of the oscillator to perturbations applied at the corresponding circuit node.

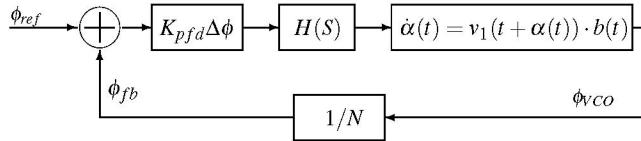


Fig. 1. PPV based PLL phase-domain macromodel.

Using the nonlinear phase equation (2) to replace the VCO circuit in the PLL, we obtain a PPV based PLL macromodel, as shown in Figure 1. It has been shown [7] that the using the PPV macromodel not only predicts the PLL's frequency variations due to control voltage changes, but also captures timing jitter due to other perturbations (e.g., power/ground supply noise) applied to the VCO. This can be seen more clearly if we rewrite (2) in the form

$$\dot{\alpha}(t) = v_{vc}(t + \alpha(t)) \cdot b_{vc}(t) + \vec{v}_n^T(t + \alpha(t)) \cdot \vec{b}_n(t), \quad (3)$$

where $b_{vc}(t)$ is the perturbation at the control voltage, $v_{vc}(t)$ is the PPV of the control node, $\vec{b}_n(t)$ is a vector of noise currents/voltages applied to the VCO, and $\vec{v}_n^T(t)$ is corresponding PPVs. The first term on the right hand side of (3) models the transient response of the PLL and jitter induced by the LPF's output; the second term on the right hand side models the PLL jitter due to other perturbations applied to the VCO.

When the phase deviation $\alpha(t)$ is obtained by solving (3), the total phase of the VCO in radians can be calculated using

$$\phi_{vco}(t) = \omega_0(t + \alpha(t)), \quad (4)$$

where ω_0 is the VCO's free-running frequency.

A drawback of this method is that the macromodel is a small-signal model, extracted for one specific VCO control voltage. If any “input” to the VCO (such as its control voltage) varies over large ranges, the macromodel becomes increasingly inaccurate.

B. Trajectory Piecewise Techniques

The key idea of trajectory piecewise techniques [14], [15] is to put together a nonlinear system using a number of different of small signal models valid in adjoining polytopes of a system's state space, each centered around an “expansion point”. The chief advantage of piecewise linear models is that, under the right circumstances, they feature good global fidelity for strongly nonlinear systems. This property is complementary to that of small-signal models, which are typically very accurate in small regions but suffer from poor global accuracy for strongly nonlinear systems.

Consider a nonlinear system in differential algebraic equation (DAE) form

$$E \frac{dx}{dt} = f(x) + Bu, \quad y = Cx, \quad (5)$$

where x is the unknown state vector and $f(x)$ is a nonlinear function vector.

A TPWL model [14] for (5) can be built as follows:

- 1) Linearize (5) at various expansion points $\{x_i\}$

$$E \frac{dx}{dt} = f(x_i) + A_i(x - x_i) + Bu, \quad y = Cx. \quad (6)$$

- 2) Each linearized model is reduced to

$$\hat{E} \frac{dz}{dt} = f(z_i) + \hat{A}_i(z - z_i) + \hat{B}u, \quad y = \hat{C}z. \quad (7)$$

using linear time-invariant model order reduction (MOR) techniques.

- 3) The reduced models are combined together using a scalar weight function to form a single reduced nonlinear model:

$$\hat{E} \frac{dz}{dt} = \sum_{i=1}^n w_i(z)(f(z_i) + \hat{A}_i(z - z_i)) + \hat{B}u, \quad y = \hat{C}z, \quad (8)$$

where $w_i(z)$ are weight functions, which satisfy $0 \leq w_i(z) \leq 1$ and $\sum w_i(z) = 1$.

III. TRAJECTORY PIECEWISE NONLINEAR VCO MACROMODEL

In this section, we combine the PPV phase macromodel with ideas from trajectory piecewise techniques, deriving a trajectory piecewise PPV-based (TP-PPV) macromodel for the VCO. Note that, unlike prior trajectory piecewise techniques [14], [15], each of the time-shifted PPV models being put together in our method is fundamentally and strongly nonlinear; this is crucially responsible for the accuracy of TP-PPV.

A. Nonlinear Phase Macromodel And Its Shortcoming For The VCO

The phase deviation due to small signal perturbations can be represented by the nonlinear time-shifted equation [10]

$$\dot{\alpha}(t) = \vec{v}_1^T(t + \alpha(t)) \cdot \vec{b}(t). \quad (9)$$

In this equation, $\alpha(t)$ is the phase deviation caused by a perturbation $\vec{b}(t)$; $\vec{v}_1(t)$ is the PPV [10].

As noted in the previous section, (9) works well when the perturbation $\vec{b}(t)$ is small, but its accuracy deteriorates under large fluctuations of, e.g., the control voltage. The reason for this is that the PPV $\vec{v}_1(t)$ can change significantly as a function of the input. For example, Figure 2 depicts PPV waveforms of an interpolative ring VCO with DC control voltages of 1.2V and 2.4V; it can be seen that they are quite different. Hence, if we use the PPV at $v_{ctrl} = 1.2V$ to solve the phase equation (9), the macromodel becomes very inaccurate as the control voltage changes to 2.4V.

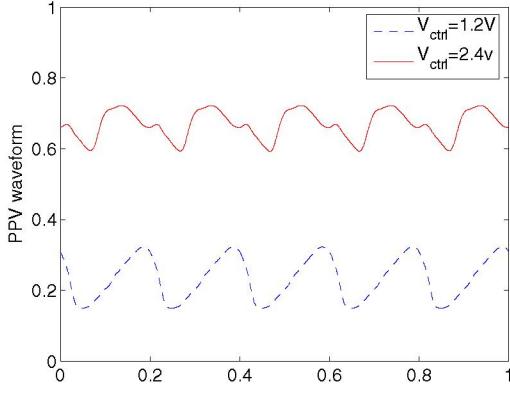


Fig. 2. PPV waveforms of a VCO with different control voltages.

One possible means of overcoming this drawback is that to calculate the PPVs of the VCO for different control voltages and then to (somehow) switch the PPV waveforms suitably during the simulation. However, this approach is ad-hoc; we prefer a more structured approach that provides a concrete nonlinear differential equation as the macromodel. The trajectory piecewise approach presented below concretizes the intuition of switching from one PPV to another appropriately, within the framework of differential equations.

B. Piecewise Nonlinear VCO Macromodel

We cannot simply apply the trajectory piecewise approach of TPWLPWP (Section II-B) to (9) directly. Firstly, the VCO has different free-running frequencies at different control voltages. As a result, the PPV in (9) also has different periods when the control voltages are different; it is not straightforward to simply combine equations valid at different frequencies directly. Secondly, the state variable $\alpha(t)$ in (9) is the time shift due to perturbation, which is used to calculate the VCO phase using (4). $\alpha(t)$ is only meaningful in the context of a particular free-running frequency, *i.e.*, $\alpha(t)$ at different free-running frequencies have different meanings. Hence, we cannot use $\alpha(t)$ as the state variable when we apply the piecewise technique, as the state variable in a piecewise system should be consistent for different expansion points. In the following, we derive a new phase equation that addresses these issues.

Assume that we solve for the steady-state solution of the VCO at several DC control voltages $\{v_{ci}\}$, $i = 1 \dots m$, and obtain corresponding steady-state frequencies $\{f_i\}$ and the PPVs $\{\vec{v}_{1i}(t)\}$. The nominal phase equation for each v_{ci} can be written as

$$\dot{\alpha}_i(t) = \vec{v}_{1i}^T(t + \alpha_i(t)) \cdot \vec{b}(t) \quad i = 1 \dots m. \quad (10)$$

Since the PPVs $\vec{v}_{1i}(t)$ have different frequencies, we cannot combine (10) directly. To make the combination possible, we first normalize the period of each $\vec{v}_{1i}(t)$ to 1; then, (10) can be rewritten as

$$\dot{\alpha}_i(t) = \vec{v}_{1i}^T(f_i \cdot (t + \alpha_i(t))) \cdot \vec{b}(t) \quad i = 1 \dots m, \quad (11)$$

where $\vec{v}_{1i}(t)$ is the normalized PPV with the period of 1, and f_i is the free-running frequency at the i^{th} value of the large-signal DC control voltage. A generic form of (11) can be written as

$$\dot{\alpha}(t) = \vec{v}_1^T(f \cdot (t + \alpha(t))) \cdot \vec{b}(t), \quad (12)$$

where $\vec{v}_1 \in \{\vec{v}_{1i}\}$ and $f \in \{f_i\}$. Within (12), define $\phi(t) = f \cdot (t + \alpha(t))$ to represent the phase of the VCO, which is only true if the VCO free-running frequency is a constant. However, if the VCO control voltage switches within $\{v_{ci}\}$, and the free-running frequency varies within $\{f_i\}$, the phase of the VCO can be generalized to be

$$\phi(t) = \int_0^t f(\tau) \cdot (1 + \dot{\alpha}(\tau)) d\tau, \quad (13)$$

where $f(\tau)$ is the VCO free-running frequency at time τ , determined dynamically by the changing control voltage $v_c(\tau)$. (12) can then be rewritten as

$$\dot{\alpha}(t) = \vec{v}_1^T \left(\int_0^t f(\tau) \cdot (1 + \dot{\alpha}(\tau)) d\tau \right) \cdot \vec{b}(t) \quad (14)$$

$$= \vec{v}_1^T \left(\int_0^t f(\tau) d\tau + \int_0^t f(\tau) \dot{\alpha}(\tau) d\tau \right) \cdot \vec{b}(t). \quad (15)$$

Multiplying both sides with the current free-running frequency $f(t) \in \{f_i\}$, the phase equation can be written as

$$f(t) \dot{\alpha}(t) = f(t) \cdot \vec{v}_1^T \left(\int_0^t f(\tau) d\tau + \int_0^t f(\tau) \dot{\alpha}(\tau) d\tau \right) \cdot \vec{b}(t), \quad (16)$$

Now we introduce the new variables

$$\phi_0(t) = \int_0^t f(\tau) d\tau, \quad \phi(t) = \int_0^t f(\tau) \dot{\alpha}(\tau) d\tau, \quad (17)$$

where $\phi_0(t)$ is the phase of the VCO without perturbations and $\phi(t)$ is the phase deviation of the VCO due to perturbation. Using these new variables, (16) can be written as

$$\dot{\phi}(t) = f(t) \cdot \vec{v}_1^T(\phi_0(t) + \phi(t)) \cdot \vec{b}(t). \quad (18)$$

In (18), $\phi_0(t)$ can be calculated incrementally at runtime. $\phi(t)$ is our new state variable which already incorporates frequency information implicitly and is hence a single, consistent phase variable even as when we switch between different free-running frequencies. Hence, we can apply (18) to generate a piecewise VCO phase equation.

Using an appropriate weight function [14], [15], the piecewise VCO phase equation can be expressed as

$$\dot{\phi}(t) = \sum_{i=1}^m w_i(t) \cdot f_i \cdot \vec{v}_{1i}^T(\phi_0(t) + \phi(t)) \cdot \vec{b}(t), \quad (19)$$

where $w_i(t)$ are weight functions. The phase of the VCO can be calculated using

$$\phi_{vco}(t) = \phi_0(t) + \phi(t). \quad (20)$$

Using (19) as the VCO phase macromodel, the resulting PLL macromodel has considerably better global fidelity than the approach in Section II-A, as shown in the next section.

IV. A PLL APPLICATION OF TP-PPV

In this section, we describe a realistic PLL application, shown in block form in Figure 3, of TP-PPV. We demonstrate how to generate a high-fidelity PLL macromodel using our technique, step by step. The PLL has an XOR phase detector, followed by a simple charge pump and a second-order low-pass filter. The VCO is a 5-stage interpolative ring oscillator with a tuning range of 160MHz – 390MHz when the control voltage is between 0 – 3V. A D flip-flop is used for $\div 2$ division. We first macromodel each constituent block separately, then put the macromodels together to form a high-fidelity PLL macromodel.

A. The VCO Macromodel

We sweep the control voltage from 0V to 3V in steps of 0.3V, simulate the VCO using harmonic balance simulation, and extract PPVs for different control voltages [11]. The nonlinear relationship between the control voltage and the oscillation frequency is plotted in Figure 4(a).

We normalize the frequency of the PPVs to 1, and plot the PPV waveform of the control node in Figure 4(b). In Figure 4(b), one horizontal axis is time, from 0 to 1s, since the frequency of the PPV is normalized to 1; the other horizontal axis is the control voltage, from 0V to 3V; the vertical axis is the PPV (or the phase sensitivity)

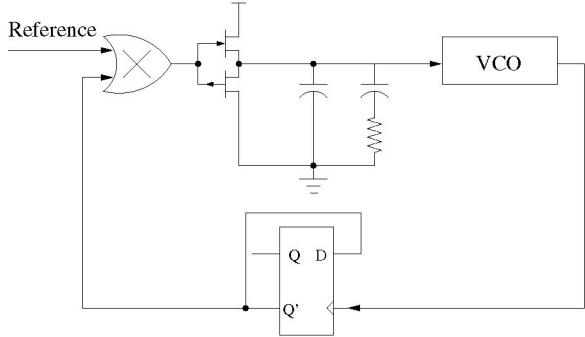


Fig. 3. PLL block diagram.

waveform) of the control node. We can see from the figure that the waveforms of the PPV depend strongly on the control voltage. Hence, TP-PPV is useful and necessary for this VCO.

Once $\{f_i\}$ and $\{\hat{v}_{1i}(t)\}$ are available, we can apply (19) to model the VCO using the trajectory piecewise technique outlined above. The trajectory piecewise VCO phase equation can be expressed as

$$\dot{\phi}(t) = \sum_{i=1}^m w_i(t) f_i(\hat{v}_{vc_i}(\phi_{vco}(t))(v_{ctrl}(t) - v_{c_i}) + \hat{v}_{ni}^T(\phi_{vco}(t)) \cdot \vec{n}(t)), \quad (21)$$

where $\phi_{vco}(t) = \phi_0(t) + \phi(t)$ is the VCO phase, f_i the VCO free-running frequency on control voltage v_{c_i} , and $w_i(t)$ is a weight function. In this equation, we split the dot product $\hat{v}_{1i}^T(\phi_{vco}(t)) \cdot \vec{b}(t)$ in (19) into two terms: the first term $\hat{v}_{vc_i}(\phi_{vco}(t))(v_{ctrl}(t) - v_{c_i})$ models the VCO's phase response due to fluctuation on the control voltage, in which $\hat{v}_{vc_i}(t)$ is the PPV of v_{ctrl} node on control voltage v_{c_i} , $v_{ctrl}(t)$ is the current control voltage; the second term $\hat{v}_{ni}^T(\phi_{vco}(t)) \cdot \vec{n}(t)$ captures the phase deviation of the VCO due to noise applied to other nodes in the VCO.

B. Phase Detector, Charge Pump and Low-pass Filter

Figure 5 depicts the phase/frequency detector, the charge pump and the low-pass filter of the PLL. In the figure, v_{ref} is the reference signal, v_{fd} is the voltage from the frequency divider, v_{pfd} is the output voltage of the phase/frequency detector, I_{pump} is the current of the charge pump, and v_{ctrl} the control voltage for the VCO.

From full simulations, we find that the response of the XOR phase detector to inputs v_{ref} and v_{fd} is very ideal. Thus, we model the phase detector using an ideal XOR function with a gate delay.

$$v_{pfd}(t) = \text{XOR}(v_{ref}(t - t_{pfd}), v_{fd}(t - t_{pfd})), \quad (22)$$

where t_{pfd} is the XOR gate delay.

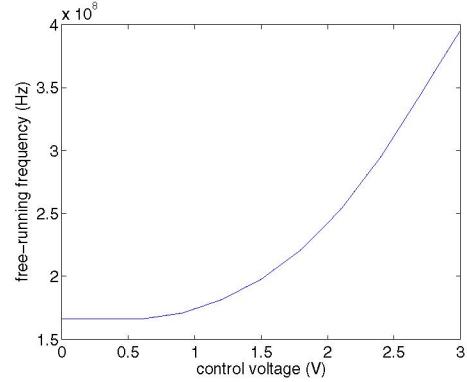
From Figure 5, we know the charge pump current I_{pump} depends on two input voltages: v_{pfd} and v_{ctrl} , or

$$I_{pump}(t) = F_{pump}(v_{pfd}(t), v_{ctrl}(t)), \quad (23)$$

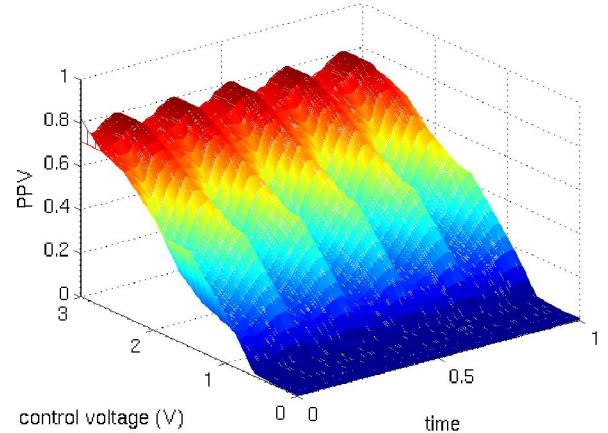
where $F_{pump}()$ is a nonlinear function, which can be identified numerically using DC sweep analysis. We perform a 2-dimensional DC sweep on v_{pfd} and v_{ctrl} , and plot I_{pump} as a function of v_{pfd} and v_{ctrl} in Figure 6.

Since $I_{pump}(t)$ can be evaluated by performing a 2D interpolation on Figure 6, the DAE equation of the low-pass filter can be written as

$$\begin{aligned} C_1 \frac{d}{dt} v_{ctrl}(t) + C_2 \frac{d}{dt} (v_{ctrl}(t) - v_1(t)) &= I_{pump}(t) \\ C_2 \frac{d}{dt} (v_1(t) - v_{ctrl}(t)) + \frac{v_1(t)}{R} &= 0. \end{aligned} \quad (24)$$



(a) Frequency vs control voltage.



(b) PPVs of the VCO on different control voltages.

Fig. 4. Frequency and the PPV waveforms of the VCO on different control voltages.

C. Frequency Divider

Since the output of the frequency divider goes to an ideal XOR phase detector, small deviations in the output's amplitude can be ignored; we model the frequency divider as a square wave generator with delay

$$v_{fd}(t) = S(\phi_{vco}(t - t_{fd})), \quad (25)$$

where $S()$ is a square wave function with the period of 2 (for a division ratio of 2), ϕ_{vco} is the VCO phase, and t_{fd} is the frequency divider delay.

D. TP-PPV PLL Macromodel

Putting together the macromodels of each component, we can write DAE equations to macromodel the behaviors of the PLL:

$$\begin{aligned} \dot{\phi}(t) &= \sum_{i=1}^m w_i(t) f_i(\hat{v}_{vc_i}(\phi_{vco}(t))(v_{ctrl}(t) - v_{c_i}) + \hat{v}_{ni}^T(\phi_{vco}(t)) \cdot \vec{n}(t)) \\ C_1 \frac{d}{dt} v_{ctrl}(t) + C_2 \frac{d}{dt} (v_{ctrl}(t) - v_1(t)) &= I_{pump}(t) \\ C_2 \frac{d}{dt} (v_1(t) - v_{ctrl}(t)) + \frac{v_1(t)}{R} &= 0 \end{aligned} \quad (26)$$

In (26), $\phi_{vco}(t)$ is the phase of the VCO, which can be calculated using (20). $I_{pump}(t)$ can be evaluated using

$$I_{pump}(t) = F_{pump}(\text{XOR}(v_{ref}(t - t_{pfd}), S(\phi_{vco}(t - t_d))), v_{ctrl}(t)), \quad (27)$$

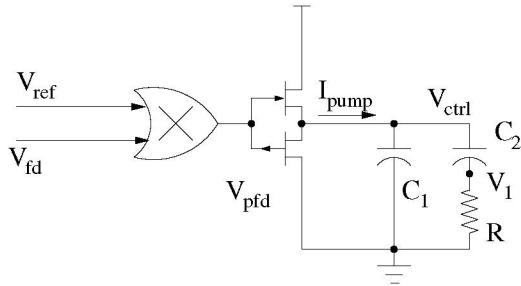


Fig. 5. Phase detector, charge pump and low-pass filter in the PLL.

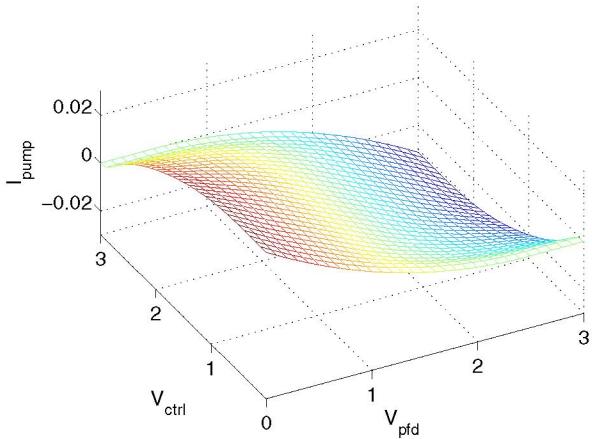


Fig. 6. Pump current I_{pump} vs phase detector output V_{pfd} and VCO control voltage V_{ctrl} .

where $t_d = t_{pfd} + f_{fd}$ is the total delay in the PLL loop. (26) can be solved using any DAE/ODE solver, with reference voltage $v_{ref}(t)$ as input.

V. RESULTS AND VALIDATION

In this section, we simulate the TP-PPV PLL macromodel of Section IV, and compare the results against full SPICE-level simulation and the PPV based PLL macromodel proposed in [7]. All simulations are performed using a MATLAB-based circuit/system simulation environment. Since local fidelity properties of the PPV based PLL macromodel have been extensively studied earlier [7], we focus on the global fidelity aspects in this work.

A. Step Response Simulation

In the simulation, we first apply a reference frequency of 100MHz to the PLL, and run the simulation until the PLL is in lock. We then increase the reference frequency and observe the PLL's transient response due to this frequency step. Since it is not easy to extract the PLL's changing frequency from the time-domain waveforms of full SPICE-level simulation, we plot the control voltages of the PLL as a validation metric.

Figure 7 depicts the PLL's response when the reference frequency changes from 100MHz to 104MHz. The control voltage of the PLL is plotted in this figure. There are small high frequency fluctuations in the control voltage, due to the non-ideality of the LPF. In this case, both macromodels (PPV and TP-PPV) work well: the patterns from the macromodels are very close to the result from full SPICE-level simulation. Since the step is very small and no switching between

different v_{ci} happens, the TP-PPV PLL macromodel provides almost identical results as the PPV based PLL macromodel.

Using the phase domain PLL macromodel leads to significant speedups in simulation time. For this PLL circuit, the runtime in MATLAB for full transient simulation with the SPICE level 1 device model is about 5000 seconds, for a simulation duration of 100 VCO cycles. However, it takes only 20 seconds to simulate the same number of cycles using the macromodel – an approximately 250 times speedup. If we use the BSIM3 device model for full circuit simulation, the simulation time increases to 19000 seconds, translating to a 950 times speedup.

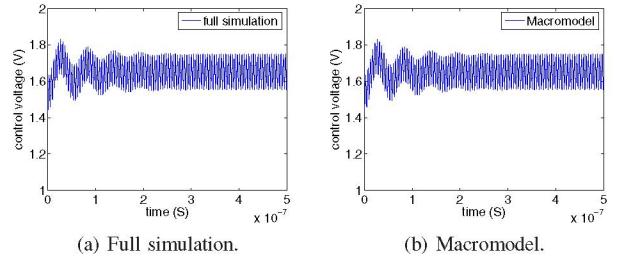
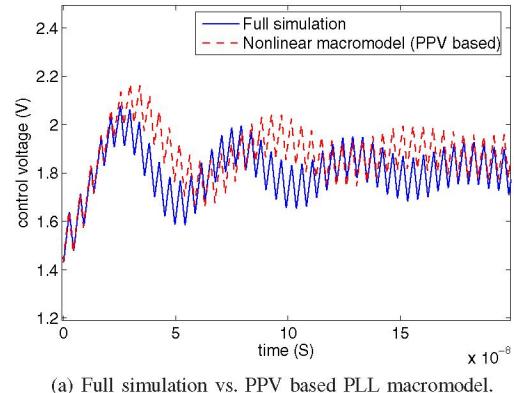
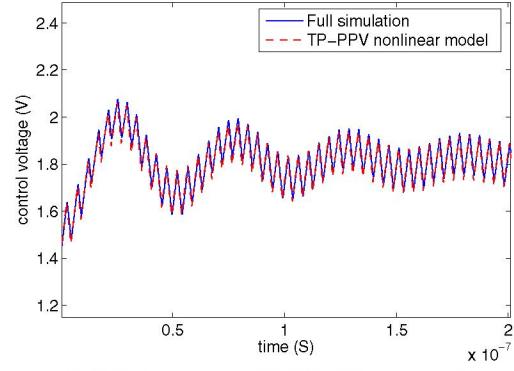


Fig. 7. Step response of the PLL ($f_{ref} = 104MHz$).

Now we apply a larger step input: we change the reference frequency from 100MHz to 111MHz. The simulation results are shown in Figure 8. From Figure 8(a), it is clear that the PPV based PLL macromodel is not able to match the full simulation well. In contrast, the TP-PPV PLL macromodel exhibits a close-to-perfect match, as shown in Figure 8(b).



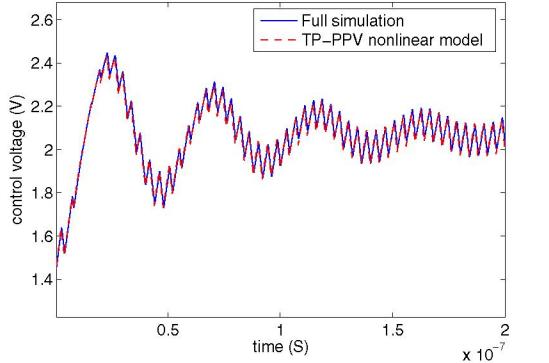
(a) Full simulation vs. PPV based PLL macromodel.



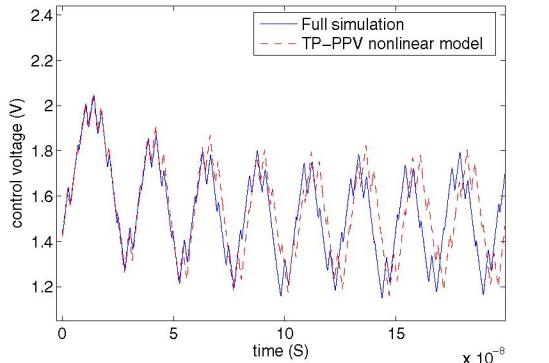
(b) Full simulation vs. TP-PPV PLL macromodel.

Fig. 8. Step response of the PLL ($f_{ref} = 111MHz$).

We keep increasing the step size of the reference, simulating the PLL with reference frequencies 125MHz and 143MHz . The results are plotted in Figure 9. With such large step sizes, the control voltage exhibits very large swings during the capture/locking process; the PPV based PLL macromodel makes totally wrong predictions in these cases. However, the TP-PPV PLL macromodel of this work still provides excellent predictions, with near-perfect transient response matches evident in Figure 9(a). When the reference frequency is 143MHz , the PLL is unable to lock, which is predicted correctly by the TP-PPV macromodel, as shown in Figure 9(b).



(a) Full simulation vs. TP-PPV PLL macromodel ($f_{ref} = 125\text{MHz}$).



(b) Full simulation vs. TP-PPV PLL macromodel ($f_{ref} = 143\text{MHz}$).

Fig. 9. Step response of the PLL.

B. Phase Deviation Simulation

Since PLL jitter simulation using the locally-accurate PPV macromodel has already been extensively studied in [7], here we provide a simple example demonstrating the jitter prediction ability of the TP-PPV macromodel. We first make the PLL lock to a reference frequency of 100MHz , then apply a small, periodic phase noise to the reference signal, and simulate the PLL phase deviation due to reference jitter. The simulation results are plotted in Figure 10. In this figure, the vertical-axis is the phase deviation of the PLL due to reference jitter. Again, our TP-PPV macromodel offers good matches against full SPICE-level simulation.

VI. CONCLUSIONS

We have presented a novel VCO macromodeling technique by combining strongly nonlinear trajectory piecewise macromodeling techniques with PPV-based time-shifted nonlinear phase macromodels. The resulting TP-PPV macromodeling technique offers not only local fidelity, but also excellent global fidelity by virtue of its

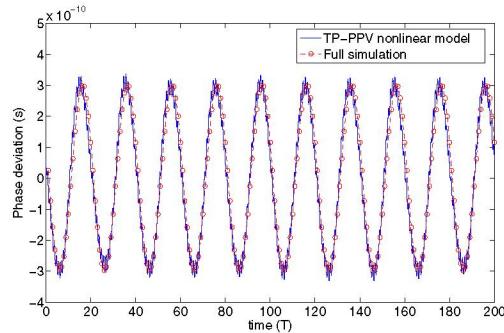


Fig. 10. Phase deviation due to reference jitter.

piecewise nature. This leads to more broadly applicable and accurate PLL macromodels than previously available. We have demonstrated the proposed technique on a 5-stage interpolative ring oscillator based PLL, illustrating the process of TP-PPV based PLL macromodeling step by step. The simulation results show the TP-PPV PLL macromodel is able to simulate a variety of PLL transient responses and capture PLL jitters at SPICE-level accuracy, with speedups of about three orders of magnitude.

REFERENCES

- [1] J.L. Stensby. *Phase-locked loops: Theory and applications*. CRC Press, New York, 1997.
- [2] L. Nagel. SPICE2: A Computer Program to Simulate Semiconductor Circuits. Electron. Res. Lab., Univ. Calif., Berkeley, 1975.
- [3] M. Takahashi, K. Ogawa, and K.S. Kundert. VCO jitter simulation and its comparison with measurement. In *Proceedings of Design Automation Conference 1999*, pages 85–88, June 1999.
- [4] K. Kundert. *Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers*. www.designers-guide.com, 2002.
- [5] A. Demir, E. Liu, A.L. Sangiovanni-Vincentelli, and I. Vassiliou. Behavioral simulation techniques for phase/delay-locked systems. In *Proceedings of the Custom Integrated Circuits Conference 1994*, pages 453–456, May 1994.
- [6] F. Herzl and B. Razavi. A study of oscillator jitter due to supply and substrate noise. *IEEE Trans. Circuits and Systems II*, 46:56–62, Jan 1999.
- [7] X. Lai, Y. Wan, and J. Roychowdhury. Fast pll simulation using nonlinear vco macromodels for accurate prediction of jitter and cycle-slipping due to loop non-idealities and supply noise. In *Proc. IEEE Asia South-Pacific Design Automation Conference*, January 2005.
- [8] A. Mehrotra. Noise analysis of phase-locked loops. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 49(9):1309–1316, September 2002.
- [9] X. Lai and J. Roychowdhury. Capturing Oscillator Injection Locking via Nonlinear Phase-Domain Macromodels. *IEEE Trans. Microwave Theory Tech.*, 52(9):2251–2261, September 2004.
- [10] A. Demir, A. Mehrotra, and J. Roychowdhury. Phase noise in oscillators: a unifying theory and numerical methods for characterization. *IEEE Trans. on Circuits and Systems-I:Fundamental Theory and Applications*, 47(5):655–674, May 2000.
- [11] A. Demir and J. Roychowdhury. A reliable and efficient procedure for oscillator ppv computation, with phase noise macromodelling applications. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(2):188–197, February 2003.
- [12] A. Demir. Phase noise in oscillators: Dases and colored noise sources. In *IEEE/ACM International Conference on Computer-Aided Design*, November 1998.
- [13] A. Demir, D. Long, and J. Roychowdhury. Computing phase noise eigenfunctions directly from steady-state jacobian matrices. In *IEEE/ACM International Conference on Computer Aided Design*, pages 283–288, November 2000.
- [14] 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 of Integrated Circuits and Systems*, 11:155–170, Feb 2003.
- [15] N. Dong and J. Roychowdhury. Piecewise polynomial nonlinear model reduction. In *Proc. IEEE Design Automation Conference*, June 2003.
- [16] FX Kaertner. Analysis of white and $f^{-\alpha}$ noise in oscillators. *International Journal of Circuit Theory and Applications*, 18:485–519, 1990.