A Multilevel Technique for Robust and Efficient Extraction of Phase Macromodels of Digitally Controlled Oscillators

Xiaolue Lai, Jaijeet Roychowdhury
Department of Electrical and Computer Engineering, University of Minnesota
Minneapolis, MN
laixl@ece.umn.edu, jr@umn.edu

ABSTRACT

PPV phase macromodels are important for speeding up simulation of oscillator related circuits, such as PLLs, without sacrificing accuracy. Prior numerical methods for extracting PPVs face very significant robustness and accuracy problems when confronted with digitally controlled oscillators (DCOs, core building blocks in digital phase-locked loops), due to large RC time-constants from gated capacitors. In this paper, we present a hierarchical harmonic balance based technique for numerically extracting the PPV of DCOs from their SPICE-level circuit descriptions. The proposed method applies hierarchical circuit partitioning and multi-level Newton methods to achieve dramatically superior convergence and PPV accuracy in the presence of large RC time-constants. We validate the method on a large DCO with many gated capacitors and demonstrate that it can extract the PPV efficiently and robustly, succeeding when prior methods fail. The method also provides speedups of an order of magnitude for large circuits, in addition to having significantly smaller memory requirements.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids – simulation

General Terms

Algorithms

Keywords

Simulation, macromodel, VCO, DCO, PLL, DPLL, PPV

1. INTRODUCTION

Phase-locked loops (PLLs) [1,2] are widely used building blocks in analog and mixed-signal systems. Traditionally, PLLs are analog circuits that employ a voltage-controlled oscillator (VCO) for converting voltage levels to frequencies. Although analog PLLs can feature high output frequencies and excellent spectral purity [3], they exhibit shortcomings such as narrow operating frequency ranges [4], highly nonlinear frequency versus voltage characteristics [5], *etc.*. These shortcomings come to the fore when PLL designs migrate to deep-submicron (DSM) CMOS processes, which offer very limited voltage headroom [6]. To overcome such DSM integration issues, digital phase-locked loop (DPLL) design techniques [6–8] have rapidly gained in popularity, especially for highly-integrated SoC applications. DPLLs provide characteristics such as fast switching, full digitization, and good frequency stability [9]. The core component of a DPLL is the switch-tuned digitally controlled oscillator (DCO) [10, 11], the digital counterpart of analog VCOs. A typical DCO consists of many gated capacitors or resistors which are turned on or off by digital control signals, as shown in Figure 1.

However, the advent of DCOs and DPLLs poses unique challenges for simulation and verification, already a difficult enough task for PLLs. SPICE-level "transient" simulation [12] is poorly suited for simulating PLLs in general, due to efficiency issues [17]. To alleviate this inefficiency, a popular approach is to perform approximate PLL simulation using *phase domain macromodelling* of sections of the PLL's feedback loop, including the VCO. Various phase domain macromodels for oscillators (*e.g.*, [13–16]) are available; among these, the perturbation projection vector (PPV) [16] model, a nonlinear time-shifted macromodel that can be extracted from SPICE netlists using numerical methods, has been shown to be best for predicting a variety of oscillator and PLL phenomena (such as injection pulling, lock/capture transients, phase noise and jitter, *etc.*) well [17, 18].

Currently available numerical methods for extracting PPV macromodels face significant challenges for DCOs, however, that severely compromise the accuracy and utility of the extracted phase macromodel. There are two techniques available for PPV extraction: the time domain monodromy matrix method [16, 19] and the augmented Jacobian method [20] in the frequency domain. Neither of these is robust when applied to DCOs featuring banks of digitally switched capacitors. The steady-state computations that precede PPV extraction are numerical very sensitive, often failing outright in Newton-Raphson convergence; the succeeding PPV computation is even more sensitive, resulting in large inaccuracies, especially for the components at the crucial "off"-capacitor nodes. The root of the problem stems from that "off" capacitors contribute large RC-like time constants to the oscillator system, degrading the conditioning of the Jacobian matrix. Breakdown of these PPV computation techniques poses a serious problem for the design of the largest and most advanced PLLs in the industry today, hence remedies are being urgently sought.

In this paper, we present a novel and effective approach for alleviating this problem. Our technique, which can extract the PPV of any oscillator, is particularly advantageous for large oscillator systems such as DCOs with many banks of switched capacitors. The key advance of our method is breaking the computation, of both the nonlinear steady-state and the PPV, down hierarchically into smaller parts, each of which not only has superior numerical characteristics but is easier to solve. The smaller parts are tied together using extensions of multilevel Newton concepts, originally proposed in [21], to harmonic balance (HB), PPV computation and steady-state computations in general that we also develop and employ in this paper. For the computation of a particular block, the effect of the other blocks is captured via the "nonlinear Fourier

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

transfer functions", and the small Jacobian matrices, of the connections to the block under consideration. Although each block consists of a set of nonlinear differential equations, this is possible because all computations deal simply with *algebraic* relationships between vectors of Fourier components.

We prove that our method is exactly equivalent to the augmented Jacobian method in the sense that, with perfect numerics and infinite precision, identical results would be obtained. Our hierarchical method is, however, far more robust and accurate for non-ideal non-ideal numerical computations with double-precision arithmetic, truncated Fourier series, *etc.*. The new hierarchical approach also has other important advantages. The size of each block reduces significantly when the circuit is partitioned, hence memory requirements for steady-state and PPV solution become much lower. This feature is of great importance for large DCOs in integrated RF applications, simulations of which are often memory limited. Another crucial feature of our method is that the large RC poles which corrupt the Floquet eigenmode are completely eliminated, since they are moved to a different hierarchical block which is solved separately. The remainder of the DCO becomes much smaller and has few, if any, remaining interfering eigenmodes. Because each block is smaller and has better numerical conditioning, steady-state computations enjoy more robust convergence and are also faster to run. In addition, since DCO circuits have many identically repeated cells (such as gated capacitor blocks), isomorphic simulation can apply in the HB solver to improve the simulation efficiency significantly. Finally, hierarchical solution using multilevel-Newton concepts also makes the algorithm well suited for threaded and parallel implementation.

We implement the direct and hierarchical PPV solvers in a Matlab/Python based analog/RF simulation platform, and evaluate them in detail using a cross-coupled LC DCO. Both methods work equally well when the circuit is small and there are no slow time constants in the circuit. If some gated capacitors are turned off, thereby introducing large RC time constants, the direct PPV method starts developing difficulties: the HB solver converges very slowly due to the poor conditioning of the Jacobian matrix, and the PPV extracted is not accurate. In contrast, the hierarchical solver converges robustly and goes on to find the PPV correctly. When the oscillator circuit is very large, with many gated capacitors, direct methods run out of memory on our machines, which have 1GB of RAM. However, the hierarchical method of this work continues to works well, without significant increases in memory requirements.

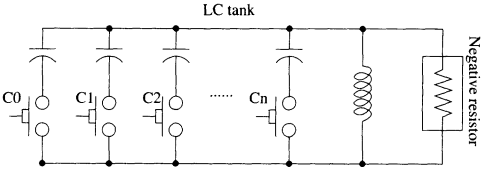

Figure 1: Diagram of an LC DCO circuit.

2. RELEVANT PREVIOUS WORKS

In this section, we provide a brief overview of methods for extracting the PPV waveforms, and explain the difficulties of those extraction methods when they apply to the DCOs.

2.1 Monodromy Method

Since the PPV is one of the Floquet eigenvectors of the linear periodic time-varying (LPTV) system linearized from the oscillator equation, we can apply the Floquet decomposition [22] to the LPTV system and extract the PPV [16, 19].

We use any simulation technique to simulate the free-running oscillator to steady state, and linearize it as an LPTV system

$$\dot{w}(t) = A(t)w(t) + b(t), \tag{1}$$

where $w(t)$ represents deviations due to perturbations $b(t)$, and $A(t)$ is a T-periodic matrix. According to Floquet theory [22], the state transition matrix of system (1) can be decomposed as

$$\Phi(t,\tau) = U(t)\exp(D(t-\tau))V(\tau), \tag{2}$$

where $U(t)$ and $V(t)$ are T-periodic nonsingular matrices, satisfying biorthogonality conditions $v_i^T(t)u_j(t) = \delta_{ij}$, and $D = diag[\mu_1, ..., \mu_n]$, where μ_i are the Floquet exponents. For an oscillator circuit, one of the Floquet exponents must be 0. Without loss of generality, we choose $\mu_1 = 0$ and $u_1(t)$ and $v_1(t)$ are corresponding left and right Floquet eigenvectors. It has been shown [16] that $u_1(t) = \dot{x}_s(t)$, the derivative of the steady state of the oscillator, and $v_1(t)$ is the PPV, satisfying $u_1(t)^T \cdot v_1(t) = 1$ at any time point.

To calculate $v_1(t)$, we need to calculate the monodromy matrix $\Phi(T,0)$ by numerically integrating

$$\dot{w}(t) = A(t)w(t) \tag{3}$$

from 0 to T, using initial value $w(0) = I_n$, where I_n is an identity matrix of size n. $v_1(0)$ is an eigenvector of $\Phi^T(T,0)$ corresponding to eigenvalue 1. $v_1(0)$ need to be scaled using $u_1(0)^T \cdot v_1(0) = 1$, and then $v_1(t)$ can be calculated by integrating the adjoint system of (3)

$$\dot{w}(t) = -A(t)^T w(t) \tag{4}$$

for one period, using $v_1(0)$ as initial value.

According to above procedure for calculating the PPV, we need to find an eigenvector of $\Phi^T(T,0)$ corresponding to the oscillatory eigenvalue 1. If the circuit has many large RC poles, which introduce eigenvalues very close to 1, we will have trouble to find the right eigenvector for the PPV extraction, as numerical integration error can accumulate in computing the monodromy matrix, causing the oscillatory eigenvalue become numerically indistinguishable.

2.2 Augmented Harmonic Balance Jacobian Method

To overcome the difficulty of the time domain method, a more robust frequency domain method is proposed [20]. We use the harmonic balance method to solve the free-running oscillator equation. Since oscillators have infinite steady-states, we add an extra unknown for frequency and a phase equation to make the solution unique. When harmonic balance converges, we obtain the Jacobian matrix, which we call augmented Jacobian matrix, as we add one extra row and column for phase equation. The augmented Jacobian matrix has the form of

$$J_{aug} = \begin{pmatrix} J & q \\ p^T & r \end{pmatrix}, \tag{5}$$

where J is the harmonic balance Jacobian of the oscillator equation, p is the vector of phase condition, $q = j\Omega X_s$ is the Fourier coefficient of $\dot{x}_s(t)$, and r is a scalar.

If the augmented Jacobian matrix is not singular, the PPV can be calculated by solving a linear equation

$$J_{aug}^T x = e, \tag{6}$$

where $e = [0, 0, ..., 0, 1]^T$, and $x = [V_1, d]^T$ in whichV_1 is the Fourier coefficients of the PPV, and d is a scalar.

This method enforces $u_1^T(t) \cdot v_1(t) = 1$ when solving (6). Hence, the correct PPV can always be obtained, even though there are other Floquet eigenvalues that are close to 1. However, when we apply this method to DCO circuits, we still experience difficult: firstly, there are many extremely large RC poles in the circuit, so that the augmented Jacobian matrix has very bad conditioning. As a result, the PPV obtained from solving (6) is not accurate due to numerical error. Secondly, since the augmented matrix can be close to singular in the worst case, the harmonic balance simulation cannot converge and this method cannot apply without an augmented Jacobian matrix.

3. HIERARCHICAL HARMONIC BALANCE SIMULATION

In this work, we adopt a multilevel Newton method [21], which was previously used for time domain simulation, in our harmonic balance solver. The method can eliminate the effect of large RC poles induced by gated capacitors/resistors, without affecting the PPV extraction. In this method, the circuit is partitioned into a tree hierarchy, as shown in Figure 2. In each Newton iteration of the top level, bottom level subcircuits are simulated and macromodeled as simple devices without internal nodes. The subcircuit macromodels are applied to higher level to improve the convergence of harmonic balance. The circuit is simulated and macromodeled from bottom level to top level until the top level converges.

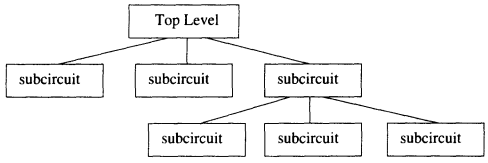

Figure 2: Tree structure of hierarchical circuit.

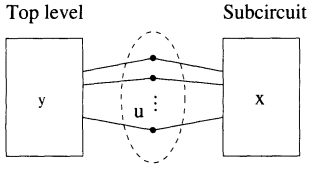

Figure 3: A simple circuit with a top level and a subcircuit.

For the purpose of simplicity, we use a simple circuit that has only one subcircuit, as shown in Figure 3, to explain our method. In Figure 3, y is set of all internal nodes in the top level circuit, x is set of all internal nodes in the subcircuit, and u is the set of interface nodes between the top level circuit and the subcircuit. u has size of p. The harmonic balance equation of the circuit can be written as

$$H(X,U,Y) = \begin{pmatrix} H_x(X,U) \\ H_u(X,U,Y) \\ H_y(U,Y) \end{pmatrix} = 0, \tag{7}$$

where X, U and Y are Fourier coefficients of unknowns in x, u, and in y, respectively. H_x is harmonic balance equations respect to nodes in x, H_u is harmonic balance equations respect to nodes in u, and H_y is harmonic balance equations respect to nodes in y. Corresponding harmonic balance Jacobian can be written as

$$J = \begin{pmatrix} J_{xX} & J_{xU} & 0 \\ J_{uX} & J_{uU} & J_{uY} \\ 0 & J_{yU} & J_{yY} \end{pmatrix}, \tag{8}$$

where $J_{xX} = \frac{\partial H_x}{\partial X}$, $J_{xU} = \frac{\partial H_x}{\partial U}$, *etc.*

Now we cut the circuit on the interface nodes u, and partition the circuit into two independent parts, as shown in Figure 4. For simplicity, we assume the top level circuit takes currents $I_u = [i_1, i_2, ..., i_p]^T$ as input, and outputs voltages on interface nodes $V_u = [v_1, v_2, ..., v_p]^T$; The subcircuit takes voltages $V_u = [v_1, v_2, ..., v_p]^T$ as input, and outputs currents $I_u = [i_1, i_2, ..., i_p]^T$ that is applied to the top level as input. Using this input/output relation, we can write the harmonic balance equation for the top level circuit as

$$H_{top}(I_u,U,Y) = \begin{pmatrix} H_{u_{top}}(I_u,U,Y) \\ H_y(U,Y) \end{pmatrix} = 0, \tag{9}$$

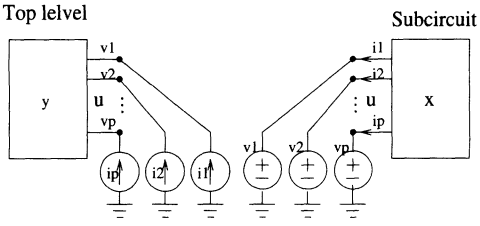

Figure 4: Partition of the simple circuit.

where U are unknowns corresponding to interface nodes u in the top level, which will be applied to the subcircuit as its input V_u, and I_u are input currents from the subcircuit. $H_{u_{top}}$ is different from H_u in (7): X is replaced by input current I_u, since u is split. The Jacobian of the top level is

$$J_{top} = \begin{pmatrix} J_{uU_{top}} & J_{uY} \\ J_{yU} & J_{yY} \end{pmatrix}, \tag{10}$$

where $J_{uU_{top}}$ is different from J_{uU} in (8), since u is split.

The harmonic balance equation for the subcircuit is

$$H_{sub}(X,U,I_u,V_u) = \begin{pmatrix} H_x(X,U) \\ H_{u_{sub}}(X,U,I_u) \\ KVL(U,V_u) \end{pmatrix} = 0, \tag{11}$$

where V_u are inputs from the top level, I_u are outputs corresponding to input V_u, and $KVL(U,V_u)$ is KVL equations added for inputs V_u. We need these extra equations for macromodelling the subcircuit. The Jacobian of the subcircuit is

$$J_{sub} = \begin{pmatrix} J_{xX} & J_{xU} & 0 \\ J_{uX} & J_{uU_{sub}} & I_p \\ 0 & I_p & 0 \end{pmatrix}, \tag{12}$$

where I_p is an identity matrix of size p, which is from the extra KVL equations for input V_u. Due to the partitioning, the Jacobian of interface nodes u is divided into two parts, which satisfy

$$J_{uU} = J_{uU_{sub}} + J_{uU_{top}}. \tag{13}$$

In our hierarchical harmonic balance simulation, we simulate the bottom level subcircuit first, and macromodel it as a simple device without internal nodes. The macromodeled Jacobian of the subcircuit is use to formulate the equation for the top level circuit. Since the top level has full information of the subcircuit, harmonic balance has good convergence, and the PPV extraction is possible. We know a device model needs to provide two informations: one is its output respect to given inputs; the other is the Jacobian matrix of the device. For the subcircuit in Figure 4, its output can be obtained by solving (11). To obtain the Jacobian $J = \frac{\partial I_u}{\partial V_U}$, we adopt a macromodelling technique proposed in [21].

Since we can always solve (11) to obtain the solution of X,U and I_u for any V_u applied to the subcircuit, X, U and I_u can be considered as functions of V_u, and we can rewrite (11) as

$$H_{sub}(X(V_u),U(V_u),I_u(V_u),V_u) = 0. \tag{14}$$

Since for any V_u, we can find $X(V_u)$, $U(V_u)$ and $I_u(V_u)$ to satisfy (14), in this solution space, we have

$$\begin{aligned} 0 &= \frac{\partial H_{sub}(X(V_u),U(V_u),I_u(V_u),V_u)}{\partial V_u} \\ &= \frac{\partial H_{sub}}{\partial V_u} + \left(\frac{\partial H_{sub}}{\partial X}, \frac{\partial H_{sub}}{\partial U}, \frac{\partial H_{sub}}{\partial I_u}\right) \cdot \begin{pmatrix} \frac{\partial X}{\partial V_u} \\ \frac{\partial U}{\partial V_u} \\ \frac{\partial I_u}{\partial V_u} \end{pmatrix} \end{aligned} \tag{15}$$

$$\Downarrow$$

$$\begin{pmatrix} \frac{\partial X}{\partial V_u} \\ \frac{\partial U}{\partial V_u} \\ \frac{\partial I_u}{\partial V_u} \end{pmatrix} = -\left(\frac{\partial H_{sub}}{\partial X}, \frac{\partial H_{sub}}{\partial U}, \frac{\partial H_{sub}}{\partial I_u}\right)^{-1} \cdot \frac{\partial H_{sub}}{\partial V_u} = -J_{sub}^{-1} \cdot \frac{\partial H_{sub}}{\partial V_u}, \tag{16}$$

where J_{sub} is the Jacobian matrix of the subcircuit, $\frac{\partial H_{sub}}{\partial V_u}$ is the input vector that can be obtained from subcircuit equation, and $\frac{\partial I_u}{\partial V_u}$ is the Jacobian of the macromodeled subcircuit we need for formulating top level circuit equations. Adding $\frac{\partial I_u}{\partial V_u}$ to $J_{uU_{top}}$ in (10), we make the Jacobian of the top level Jacobian complete.

This macromodel is an exact model that exactly represents the behavior of the subcircuit [21]. Once $\frac{\partial I_u}{\partial V_u}$ is calculated, we can apply this Jacobian to formulate the circuit equation for the top level simulation. Quadratic convergence can be achieved by using this method [21], and the PPV extracted from the top level Jacobian is valid, as shown in Section 4.

4. HIERARCHICAL PPV EXTRACTION

In this section, we show that the PPV extracted from the top level Jacobian of the hierarchical harmonic balance simulation is valid, and develop a method for extracting the PPV for all subcircuits hierarchically from their multilevel harmonic balance Jacobians.

We still use the simple circuit in Figure 3 for proving the validity of the PPV extracted from the hierarchical harmonic balance simulation. The proof can be extended to more complex circuits with multiple level hierarchy easily.

Since the PPV is a vector in which each entry represents the oscillator's phase sensitivity on corresponding node, we can express the PPV of the circuit as

$$V = \begin{pmatrix} V_x \\ V_u \\ V_y \end{pmatrix}, \tag{17}$$

where V is the PPV of the whole circuit, V_y is the PPV of nodes in the top level circuit, V_x is the PPV of the subcircuit, and V_u is the PPV of the interface nodes. The Jacobian matrix of the circuit in Figure 3 has the form of

$$J = \begin{pmatrix} J_{xX} & J_{xU} & 0 \\ J_{uX} & J_{uU} & J_{uY} \\ 0 & J_{yU} & J_{yY} \end{pmatrix}, \tag{18}$$

In [16], the PPV has been shown to be the only vector in the null space of J^T, Hence, we have the relationship $J^T V = 0$, or

$$\begin{pmatrix} J_{xX}^T & J_{uX}^T & 0 \\ J_{xU}^T & J_{uU}^T & J_{yU}^T \\ 0 & J_{uY}^T & J_{yY}^T \end{pmatrix} \cdot \begin{pmatrix} V_x \\ V_u \\ V_y \end{pmatrix} = 0. \tag{19}$$

If we partition the circuit as shown in Figure 4, the Jacobian of the top level circuit can be expressed as

$$J_{top} = \begin{pmatrix} J_{uU_{top}} & J_{uY} \\ J_{yU} & J_{yY} \end{pmatrix}. \tag{20}$$

And the the Jacobian of the subcircuit can be expressed as

$$J_{sub} = \begin{pmatrix} J_{xX} & J_{xU} & 0 \\ J_{uX} & J_{uU_{sub}} & I_p \\ 0 & I_p & 0 \end{pmatrix}, \tag{21}$$

Now we macromodel the subcircuit, the Jacobian of the macromodeled subcircuit is

$$\begin{pmatrix} \frac{\partial X}{\partial V_u} \\ \frac{\partial U}{\partial V_u} \\ \frac{\partial I_u}{\partial V_u} \end{pmatrix} = -J_{sub}^{-1} \cdot \frac{\partial H_{sub}}{\partial V_u} = -\begin{pmatrix} J_{xX} & J_{xU} & 0 \\ J_{uX} & J_{uU_{sub}} & I_p \\ 0 & I_p & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 \\ 0 \\ I_p \end{pmatrix} = -\begin{pmatrix} \cdot & \cdot & -J_{xX}^{-1} J_{xU} \\ \cdot & \cdot & I_p \\ \cdot & \cdot & -J_{uX} J_{xX}^{-1} J_{xU} + J_{uU_{sub}} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ I_p \end{pmatrix} \tag{22}$$

$$\Rightarrow \frac{\partial I_u}{\partial V_u} = J_{uX} J_{xX}^{-1} J_{xU} - J_{uU_{sub}}. \tag{23}$$

Applying the Jacobian of the macromodeled subcircuit to the top level Jacobian, we obtain the top level Jacobian matrix of the hierarchical HB solver

$$J_{top_h} = \begin{pmatrix} J_{uU_{top}} + J_{uU_{sub}} - J_{uX} J_{xX}^{-1} J_{xU} & J_{uY} \\ J_{yU} & J_{yY} \end{pmatrix} = \begin{pmatrix} J_{uU} - J_{uX} J_{xX}^{-1} J_{xU} & J_{uY} \\ J_{yU} & J_{yY} \end{pmatrix}. \tag{24}$$

By comparing (19) and (24), it is not difficult to show that $[V_u; V_y]$ in (19) satisfies

$$J_{top_h}^T \cdot \begin{pmatrix} V_u \\ V_y \end{pmatrix} = 0. \tag{25}$$

Hence, the top level Jacobian is oscillatory, since it has an eigenvalue of 0. We can apply the augmented Jacobian method to calculate the PPV for the top level circuit. If we apply the augmented Jacobian method to the top level Jacobian, and obtain the top level PPV from hierarchical simulation

$$V_{top_h} = \begin{pmatrix} V_{u_h} \\ V_{y_h} \end{pmatrix}, \tag{26}$$

which satisfy $J_{top_h}^T V_{top_h} = 0$, or

$$\begin{pmatrix} J_{uU}^T - J_{xU}^T J_{xX}^{-T} J_{uX}^T & J_{yU}^T \\ J_{uY}^T & J_{yY}^T \end{pmatrix} \cdot \begin{pmatrix} V_{u_h} \\ V_{y_h} \end{pmatrix} = 0. \tag{27}$$

Comparing (19) and (27), we find if we calculate the PPV for subcircuit x using

$$V_{x_h} = -J_{xX}^{-T} J_{uX}^T V_{u_h}, \tag{28}$$

the resulting PPV from hierarchical simulation

$$V_h = \begin{pmatrix} V_{x_h} \\ V_{u_h} \\ V_{y_h} \end{pmatrix} \tag{29}$$

satisfy (19). Hence, the PPV extracted using our macromodel based hierarchical simulation is valid: it is exactly the same PPV calculated using direct harmonic balance simulation. The PPV of the subcircuit can be calculated using (28), when the top level PPV is available.

5. SIMULATION RESULTS

In this section, we apply and evaluate the hierarchical PPV extraction technique. We implement the proposed method in our Matlab/Python based analog/RF simulation platform, and compare to direct augmented Jacobian method. The oscillator we use in our simulation is a cross-coupling LC oscillator, as shown in Figure 5. The oscillator has a bank of gated capacitors controlled by digital signals, for generating different frequencies. All simulations are performed in **MATLAB**, using BSIM3 device model.

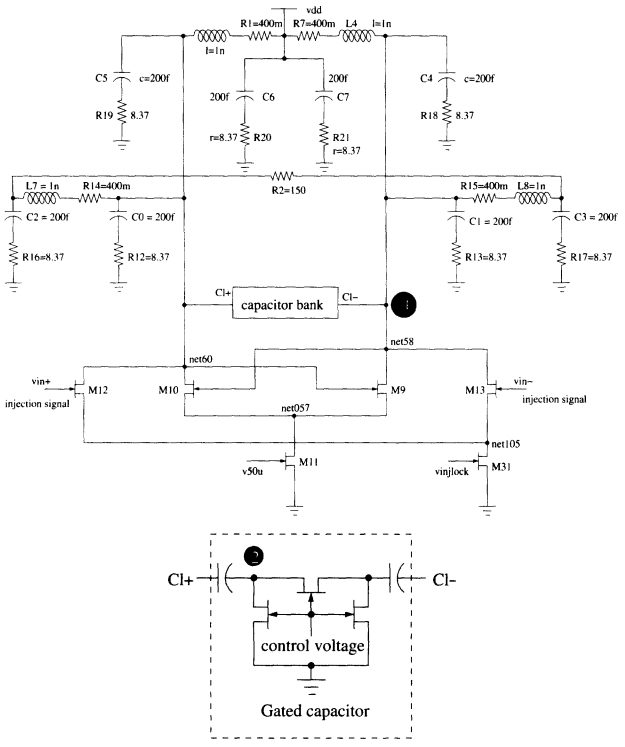

Figure 5: A cross coupling LC DCO with capacitor bank.

We first simulate the circuit with four gated capacitors that are all turned on, and extract the PPV, using both direct and hierarchical methods. The system size is 78 if the direct harmonic balance is applied. Using the hierarchical method, we simulate a top level circuit of size 44, and a subcircuit of size 14. The direct method takes about 4 minutes to finish the simulation and PPV extraction; the hierarchical method takes about 7 minute to do the same job. The direct method is faster in this case because the circuit is small and the conditioning of the circuit is good. The oscillation waveform and the PPV of the node (1) in Figure 5 are shown in Figure 6 and Figure 7. Two methods have exactly the same results. The direct method works well in this case, as all gated capacitors are on, so that there has no large RC pole in the circuit.

We then add one more gated capacitor to the circuit, which is turned off. This new gated capacitor should not affect the waveform

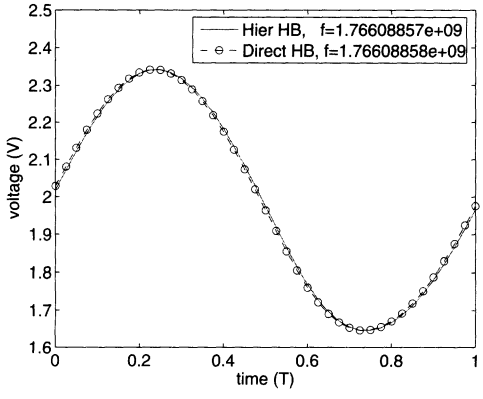

Figure 6: Oscillation waveform of node (1) (4 gated capacitors).

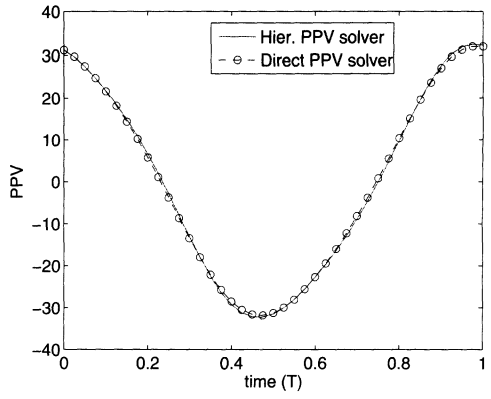

Figure 7: PPV waveform of node (1) (4 gated capacitors).

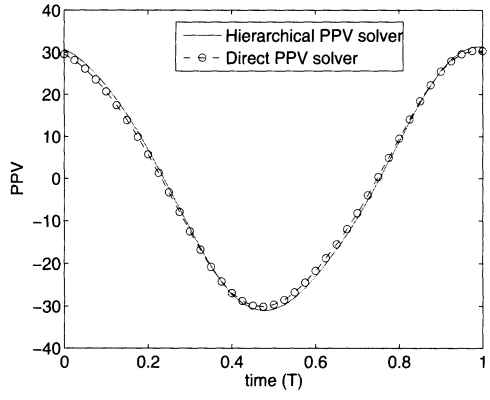

Figure 8: PPV waveform of node (1) (one capacitor is off).

and the PPV of the circuit, since it is off. The system size is 88 for direct method. Using hierarchical method, we reduce the circuit to a top level of size 44, and two subcircuits of size 14. The new gated capacitor introduces large RC poles to the circuit, so that the direct harmonic balance has trouble to converge: it converges in 29 Newton iterations, total computational time is about 14 minutes. For the hierarchical method, its iteration number does not increase, computational time is about 9 minutes. The 2 minutes increase in simulation time is due to we need to simulate one more subcircuit.

In Figure 8 and Figure 9 , we plot PPV waveforms of the node (1) and the node (2) from both methods. There is slight difference between direct method and hierarchical method in this case. By comparing Figure 8 with Figure 7 in the first case carefully, we observe the amplitude of the PPV waveform from direct method reduces slightly in Figure 8, which should not happen since the circuit does not changed. This small difference can be due to numerical errors when solving (6) using an augmented Jacobian matrix with bad conditioning. The hierarchical method does not has this problem, since it has the nature to isolate the bad conditioning subcircuit from the top level. If circuits are very large and have many gated capacitors that are off, we expect the direct method will have larger numerical error, and the hierarchical method will give the PPV more accurately.

Finally, we simulate and extract the PPV for the full circuit with 64 gated capacitors. The circuit has about 200 transistors, and the system size is more than 500. We have trouble to apply direct harmonic balance in this case due to memory issue. Using hierarchical method, we simulate a top level of size 44 and several subcircuits of

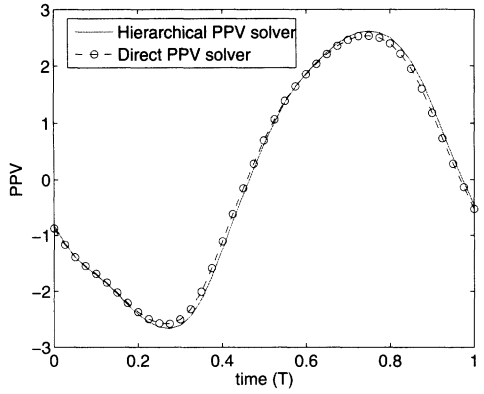

Figure 9: PPV waveform of node (2) (one capacitor is off).

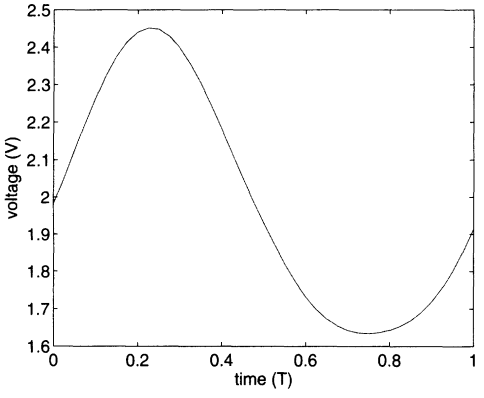

Figure 10: Oscillation waveform of node (1) (64 gated capacitors).

size 14. Hence, the memory requirement and computational time do not increase significantly, when the size of the circuit grows. The oscillation waveform and the PPV of node (1) are shown in Figure 10 and Figure 11.

6. CONCLUSION

We have presented an efficient and robust technique to extract phase macromodels from circuit descriptions of oscillators, by exploiting circuit hierarchy. The proposed technique is generally applicable for any oscillator circuit, especially for large DCOs with many large RC poles, to which traditional direct methods fail to apply. Numerical results show that our hierarchical method is able to extract the PPV of large DCO circuits more accurately and efficiently than direct methods. We are currently developing hierarchical method for extracting amplitude macromodels of large oscillator systems.

7. REFERENCES

[1] J.L. Stensby. Phase-locked loops: Theory and applications. CRC Press, New York, 1997.

[2] M. Gardner. Phase-Lock Techniques. Wiley, New York, 1966.

[3] A. Edwin. Direct digital synthesis applications. Microwave Journal, 33(1):149–151, 1990.

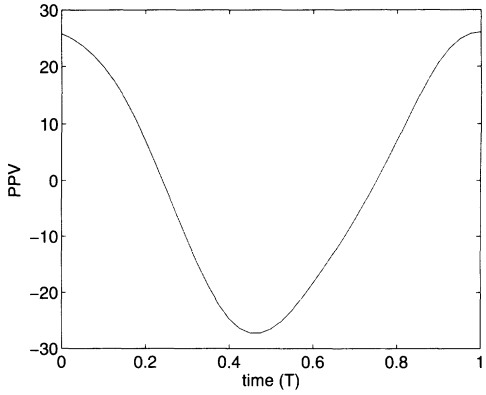

Figure 11: PPV waveform of node (1) (64 gated capacitors).

[4] P. Saul and M. Mudd. A direct digital synthesizer with 100-mhz output capability. IEEE J. Solid-State Circuits, 23(3):819–821, June 1988.

[5] R. B. Staszewski, D. Leipold, C. M. Hung, and P. T. Balsara. A first digitally controlled oscillator in a deep-submicron cmos process for multi-ghz wireless applications. In Proc. 2003 IEEE RFIC Symp, June 2003.

[6] R. B. Staszewski, D. Leipold, K. Muhammad, and P. T. Balsara. Digitally controlled oscillator (dco)-based architecture for rf frequency synthesis in a deep-submicrometer cmos process. IEEE T. on Circuits and Systems II, 50(11):815–828, November 1990.

[7] W. C. Lindsey and C. M. Chie. A survey of digital phase-locked loops. IEEE, Proceedings, 69:410–431, April 1981.

[8] I. Galton. Analog-input digital phase-locked loops for precise frequency and phase demodulation. IEEE T. on Circuits and Systems II, 42(10):621–629, October 1995.

[9] Jen-Shiun, Chiang, and Kuang-Yuan Chen. The design of an all-digital phase-locked loop with small dco hardware and fast phase lock. IEEE T. on Circuits and Systems II, 46(7):945–950, July 1999.

[10] B. Giebel, J. Lutz, and P. L. O'Leary. Digitally controlled oscillator. IEEE J. Solid-State Circuits, 24(6), June 1989.

[11] G. Donzellini, D. Caviglia, G. Parodi, D. Ponta, and P. Repettoe. A digital controlled oscillator based on controlled phase shifting. IEEE T. on Circuits and Systems, 36(8):1101–1105, August 1989.

[12] L. Nagel. SPICE2: A Computer Program to Simulate Semiconductor Circuits. Electron. Res. Lab., Univ. Calif., Berkeley, 1975.

[13] A. Hajimiri and T.H. Lee. A general theory of phase noise in electrical oscillators. IEEE Journal of Solid-State Circuits, 33(2), February 1998.

[14] 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.

[15] A. Costantini, C. Florian, and G. Vannini. Vco behavioral modeling based on the nonlinear integral approach. IEEE International Symposium on Circuits and Systems, 2:137–140, May 2002.

[16] 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.

[17] 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 Asian South Pacific Design Automation Conference, January 2005.

[18] 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.

[19] A. Demir. Phase noise in oscillators: Daes and colored noise sources. In IEEE/ACM International Conference on Computer-Aided Design, November 1998.

[20] 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.

[21] N. B. Rabbat, A. L. Sangiovanni-Vincentelli, and H. Y. Hsieh. A multilevel newton algorithm with macromodeling and latency for the analysis of large-scale nonlinear circuits in the time domain. IEEE T. on Circuits and Systems, 26(9):733–741, September 1979.

[22] R. Grimshaw. Nonlinear Ordinary Differential Equations. Blackwell Scientific, New York, 1990.