A Robust Envelope Following Method Applicable to Both Non-autonomous and Oscillatory Circuits

Ting Mei, Jaijeet Roychowdhury
University of Minnesota
Twin Cities, USA
{meiting,jr}@umn.edu

ABSTRACT

In this paper, we propose a novel envelope-following method which is uniformly applicable to both non-autonomous and oscillatory circuits. A key feature of our technique is the use of an efficient minimum least squares solution technique to solve an underdetermined envelope system directly. This leads to a general purpose approach which is much easier to solve than previous phase condition based envelope-following method, improving numerically robustness dramatically. We validate our method on a variety of autonomous and non-autonomous circuits, including a PLL in transition to lock. The new method provides speedups of 1-2 orders of magnitude over transient simulation, while obtaining results that are equally or more accurate.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

envelope following, least squares, phase condition

1. INTRODUCTION

Signals with widely separated time scales arise in many eclectical systems. For example, in communication circuits, fast carriers are often modulated by information signals that vary orders of magnitude slower. Circuits that feature this characteristic include voltage controlled oscillators (VCOs), phase locked loops (PLLs), mixers, *etc.*. For such circuits, SPICE-like transient simulation (*e.g.*, [1, 2]) is often extremely ineffective. Simulation timesteps are constrained by fast signal variations to be very small; but at the same time, the total simulation time needs to be long enough to capture the slow components, also known as envelopes. The situation is even worse for oscillators. Due to their fundamental property of marginal stability, small numerical errors in oscillator phase accumulate without limit. Therefore, extremely small timesteps are required during transient simulation to obtain acceptable results.

To solve such systems more efficiently, researchers have proposed various techniques. One category of techniques, proposed by Petzold [3] and later used for circuit simulation (*e.g.*, [4, 5, 6, 7, 8]), are *time-domain envelope-following methods*, in which only a few selected fast cycles are simulated using transient, while the rest are interpolated. Another popular envelope approach is the frequency-domain Fourier-envelope method [9, 10, 11], which uses a combination of Harmonic balance and time integration methods. Recently, a new class of envelope methods, based on Multi-time Partial Differential Equation (MPDE) circuit formulations [12, 13, 14], provide both time-domain and frequency-domain technique in a unified manner with the use of artificial time scales.

The above techniques are applicable to non-autonomous circuits; to extend them to oscillators, additional refinements are required. Unlike forced circuits, which operate at fixed frequencies determined by the inputs, frequencies of oscillators are internally ("autonomously") determined and may change dynamically during operation. Therefore, a major issue in oscillator envelope following is how to find the dynamically changing frequency accurately, such that envelopes vary smoothly and can be tracked efficiently. The majority of solutions to this problem have focused on the estimation of the changing periods of oscillators. For example, Petzold [3] proposed a minimization based approach for estimating the periods of oscillators in her envelope following method. Later, Gear developed a simpler method to estimate the period by identifying certain points that appear repeatedly in a waveform [15]. Another common approach towards estimating the changing period is to add an extra variable T to the circuit equation system, and to solve it with the help of a "phase condition" equation, such as those used in Fourier [12, 13, 14] and MPDE based [16] envelope methods. Recently published work [17] also accounts for the cumulative effect of the changing periods of cycles skipped during envelope simulation. By adding an extra variable to represent the envelope step taken, the robustness of oscillator envelope is significantly improved.

However, the envelope methods available today require that oscillators be treated differently, using a structurally different envelope algorithm, from non-autonomous circuits. There appears to be no general-purpose envelope technique available that is applicable to both autonomous and non-autonomous systems. This constitutes a significant limitation of envelope methods, especially when compared with transient simulation, which is broadly applicable to any kind of circuit. Further, it is not always possible to tell whether the circuit being simulated is an oscillator or not, especially since a circuit can change its autonomous nature during operation. Such situations are not the exception but are commonly encountered in practice. For example, when an oscillator is injection locked, it loses its natural frequency to match the frequency of the injection signal. During this locking process, the oscillator changes from an autonomous system to a forced system. Another very important practical application where this happens is phase-locked loops (PLLs), which contain an autonomous VCO which becomes locked to an external reference over time.

In this paper, we present a novel envelope-following technique that applies uniformly to both autonomous and autonomous circuits, and can cope naturally with transitions from one to the other during operation. The method (which we call LSENV) modifies the recently-published MPCENV method [17] by introducing the important notion of minimum least-squares solution. We retain the concept of using two extra variables to represent the potentially changing period and the cumulative effect of changing periods, respectively. The key advance of our technique, however, is the use of a minimum least-squares method to solve an *underdetermined envelope* system directly. By exploiting a Moore-Penrose pseudo-inverse based Newton solution method [18], our method is not only more general, but also makes numerical envelope solution much easier and more robust, compared to previous approaches based on square systems via the introduction of phase conditions. The computational efficiency of LSENV is virtually identical to that of non-autonomous envelope-following methods, since we perform least-squares solution efficiently by deflating away a 2-dimensional null space. Replacing phase conditions by minimum least square solution also results in the algorithm's naturally finding smooth envelopes that would be very difficult to identify even with manual knowledge and effort; this has further positive implications on efficiency and robustness. We present numerical strategies and a phase corrector method to further check and ensure the smoothness of the envelopes solved.

We validate our method extensively on a variety of circuits, both oscillatory and non-autonomous. In addition to applying LSENV to oscillators, mixers, *etc.*, separately, we also consider circuits that change nature from autonomous to non-autonomous during operation. Using LSENV, we are able to predict the onset of injection locking in forced oscillators effectively. Even more importantly from an application perspective, we demonstrate that LSENV is suitable for simulating the capture/lock process in PLLs. We obtain excellent matches of results from LSENV against transient simulation, but with speedups of 1-2 orders of magnitude. In view of the effectiveness and robustness of the technique and also its relative simplicity of implementation, we feel that LSENV, though new, can already be profitably deployed in industrial simulators.

The remainder of the paper is organized as follows. In Section 2, we briefly review existing non-autonomous and autonomous envelope-following methods. In Section 3, we describe LSENV, discuss numerical solution details and describe refinements to ensure good solutions. In Section 4, we present results from applying LSENV to various kinds of circuits to demonstrate its generality and effectiveness.

2. ENVELOPE FOLLOWING METHODS

In this section, we first review envelope following methods for forced circuits. We then discuss envelope following for autonomous circuits, *i.e.*, oscillators [3, 15, 17].

2.1 Envelope following for non-autonomous circuits

Generally, a circuit can be described by a system of differential equations:

$$\dot{q}(x) + f(x) = b(t), \tag{1}$$

where x is a vector of state variables (such as node voltages and branch currents), q contains capacitor charge or inductor flux terms, and f represents resistive terms [13].

We assume that waveforms of a circuit feature both fast and slow characteristics. For example, the waveform shown in Figure 1 contains fast oscillations whose amplitude changes much more slowly. When such waveforms are sampled at every fast oscillation period, the resulting samples can be interpolated as a slowly varying curve. This curve is defined as the envelope (depicted by the dashed line in Figure 1). Envelope methods [3] gain efficiency by solving for the envelope via transient simulation on only a few selected fast cycles, while skipping the simulation of many intermediate fast cycles. More specifically, a fast cycle is first simulated accurately using transient analysis, as shown in Figure 1 (from point A to B). A secant line between A and B can then be extrapolated over a large "envelope timestep", which can have many fast cycles, to obtain the solution at point C. This process is repeated until the end of the simulation interval. The process described above is analogous to solving the envelope by conventional forward-Euler integration. Similar algorithms, which apply explicit and implicit Adams formulae, can also be derived [3]. For instance, the backward-Euler based envelope-following method, popular for circuit simulation applications, can be described by the difference equation:

$$x(t+mT) = x(t) + mT\frac{x(t+mT) - x(t+(m-1)T)}{T}, \tag{2}$$

where T is the period of the fast cycles and m the number of cycles in one envelope step (between point A and C in Figure 1).

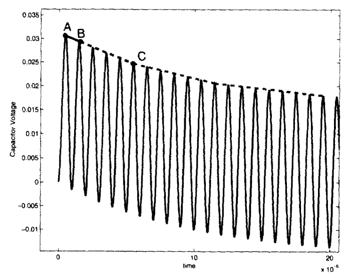


Figure 1: Illustration of envelope and Petzold's method.

2.2 Envelope following for oscillators

The main difficulty in applying the above envelope following techniques to oscillatory circuits arises from the fact that the fast oscillation period T is not known *a priori* and can, indeed, change dynamically during circuit operation. It is extremely important that T be accurately estimated in order to sample the fast waveforms in a synchronized manner. Otherwise, the resulting envelope will not be smooth and cannot be followed using large envelope steps.

One technique for estimating the oscillation period, proposed by Petzold [3], is to minimize $||y(t+T) - y(t)||_2$, where y(t) is the vector of circuit unknowns to be solved. A simpler approach is to find certain points that appear repeatedly in the waveform, such as zero-crossing points [15]. Another method to determine the changing T uses a "phase condition" equation to augment the steady state equations used to solve each of the fast cycles encountered during envelope simulation. In spite of their estimating T dynamically, these envelope approaches lack robustness because they are unable to account for the cumulative de-synchronizing effect of the changing periods of the cycles skipped.

A recent technique, MPCENV [17], introduced the notion of implicitly calculating the envelope step taken, by adding one more phase condition to ensure synchronization of the envelope samples. In MPCENV, two extra system unknowns are added to the circuit equations. One unknown represents the instantaneous changing period (T) of oscillators; the other unknown (envelope step T_{env}) takes into account cumulative effect of the changing periods of the skipped cycles. Extra phase condition equations are then added in order to ensure synchronized envelope sampling and to form a "square" system that can be solved uniquely. For example, the augmented backward-Euler based oscillator envelope following equation system is

$$\begin{aligned} \frac{x(t+T_{env}+T) - x(t+T_{env})}{T} &= \frac{x(t+T_{env}) - x(t)}{T_{env}}, \\ \frac{dx_l}{dt}\Big|_{t+T_{env}} &= 0, \\ \frac{dx_l}{dt}\Big|_{t+T_{env}+T} &= 0, \end{aligned} \tag{3}$$

where x_l is a single state variable on which the phase condition constraints are applied.

2.3 Lack of uniformly applicable envelope following methods

The reader will have noted above that practically effective envelope formulations for non-autonomous and autonomous circuits differ structurally, *i.e.*, it is necessary to choose the appropriate algorithm based on the nature of the circuit. Especially when contrasted with the general applicability of, *e.g.*, plain transient simulation, this constitutes a significant limitation for envelope following methods in general, for it may not be known a-priori if a circuit is oscillatory or not. Even more significantly, circuits can dynamically change from oscillatory to non-oscillatory and vice-versa. For example, during the process of injection locking, an oscillator gradually loses its autonomous frequency and eventually becomes a forced circuit that oscillates at exactly the same frequency as an external input. Similarly, during PLL startup and frequency switches, the VCO typically starts as an autonomous system with dynamically changing frequency but end up as a forced system when the PLL is locked.

In principle, it is possible to apply MPCENV to non-oscillatory circuits and to circuits that transition between oscillatory and non-oscillatory. However, the phase conditions of MPCENV can be difficult to satisfy numerically and can cause problems for flat waveforms (*e.g.*, square-like waveforms in ring oscillators). In the next section, we present a generally-applicable envelope method that dispenses with the need for specifying phase conditions.

3. LEAST SQUARES BASED ENVELOPE FOLLOWING

In this section, we propose an envelope following method suitable for both autonomous and non-autonomous systems. The method eliminates the requirement for extra phase conditions by solving an underdetermined system of nonlinear equations using least squares techniques. Dubbed LSENV, the method is also numerically more robust than phase condition based methods for oscillator envelopes.

3.1 LSENV formulation and Newton solution

In LSENV, we retain the two extra unknowns of MPCENV: T and T_{env}; *i.e.*, the first envelope-following equation in (3) is used unchanged. Without the phase conditions, however, this is an underdetermined system since it has two more unknowns than equations. (3) can be rewritten using the state transition function ϕ, as

$$\frac{\phi(x_0,t_0,t_0+T)-x_0}{T}-\frac{x_0-x_s}{T_{env}}=0. \tag{4}$$

At each envelope step, x_0 are the unknown state variables (at t_0) to be solved, while x_s are the known starting state from a previous envelope point. $\phi(x_0,t_0,t_0+T)$ are the circuit state at t_0+T (starting from x_0 at t_0).

Instead of adding two extra phase conditions to help solve (4) as in MPCENV, we directly solve the underdetermined nonlinear system using a Moore-Penrose pseudo-inverse based Newton-Raphson solver [18]. At each Newton iteration for solving (4),

$$J\Delta x=-F(x) \tag{5}$$

is solved. Here $F(x)$ is the function evaluation of the left hand side of (4) and J is the Jacobian matrix. J is a rectangular matrix of size $n\times(n+2)$, where n is the number of circuit unknowns. Therefore, (5) has an infinite number of solutions. We choose the unique solution with minimum norm that satisfies (5), *i.e.*, the Minimum Least Squares solution (MLS). It can be shown that Newton's method (which retains its convergence properties when the MLS solution is used) converges to a point on the continuous nonlinear manifold of solutions of (4) that is "near" the initial guess it was provided [18].

3.2 Numerical solution of LSENV and the phase corrector

During the numerical solution of LSENV (4), $\phi(x_0,t_0,t_0+T)$ is evaluated by integrating over one cycle T, starting from x_0. The calculation of the Jacobian matrix involves the evaluation of the derivative of the state transition function (known as the sensitivity matrix). This can be computed during transient simulation with additional work, just as for the shooting method [20].

For example, if the trapezoidal method (TRAP) is used during the transient (from t_0 to t_0+T, while evaluating $\phi(x_0,t_0,t_0+T)$), $\frac{d\phi}{dx_0}$ can be calculated by repeatedly applying [21]

$$\frac{dx_n}{dx_0}=(\frac{C_n}{h_n}+\frac{G_n}{2})^{-1}(\frac{C_{n-1}}{h_n}-\frac{G_{n-1}}{2})\frac{dx_{n-1}}{dx_0}, \tag{6}$$

until $\frac{dx(t0+T)}{dx_0}$ is obtained. Here, $C_i=\frac{dq(x_i)}{dx_i}, G_i=\frac{df(x_i)}{dx_i}, h_i=t_i-t_{i-1}$ and $\frac{dx_0}{dx_0}=I$. x_n is the circuit state at t_n $(t_0\le t_{n-1}\le t_n<t_0+T)$. Note that $\frac{C_n}{h_n}+\frac{G_n}{2}$ is the Jacobian matrix of (1) during transient simulation.

Unlike the phase condition based envelope-following method, LSENV does not reply on extra phase conditions to help solve the system, which improves the numerical robustness of envelope following quite dramatically. If finite differences are used to approximate the derivative, then the phase condition in [17]

$$\frac{dx}{dt}=0 \tag{7}$$

is often difficult to satisfy and results in convergence failure of Newton's method. In addition, the phase condition based approach can cause problem for flat waveforms, such as the square waveforms generated by ring oscillators. In this case, there are many continuous points that satisfy the phase condition. By eliminating the phase condition requirement, our method introduces extra freedom in the solution space. Starting from different initial guesses for Newton's method, we obtain different solutions with the use of MLS to update the initial guesses. Therefore, the chances of converging to one of the solutions are much higher, leading to improved robustness in numerical solution.

On the other hand, however, if this extra freedom is improperly exploited, total breakdown of the envelope method can result. For example, if the solutions calculated at subsequent envelope step have very different phases, then sampling synchronization is completely lost and the "envelope" is not smooth anymore, hence cannot be followed efficiently using large envelope steps. Therefore, proper initial guesses for each envelope step need to be provided such that the resulting solutions form a smooth envelope. In other words, the solutions should have either the same phase or phases that vary only slowly between envelope steps.

In our implementation, we first start the simulation with a random initial condition that satisfies (1), obtained by performing a short transient simulation and picking any solution. Since the envelope changes orders of magnitude slower than the fast oscillations, its variation after one envelope is small. At each envelope step, we use the previous starting state (x_s) as the initial guess of x_0 when solving (4) (more sophisticated predictors could also be used). The start state x_s is either the initial condition (given at the first envelope step) or a known state from previous step. In our experience, choosing the initial guess for each envelope step in this manner provides solutions with slowly varying phase, resulting in a smooth envelope in most cases.

In a few numerically sensitive cases, even slightly different phases in the envelope sampling points can cause large changes in the envelope amplitude. To ensure smooth envelopes for such cases, we perform a minor postprocessing or "corrector" operation after least squares solution in each step to correct the phase, treating the solution from MLS as a predictor. An obvious option for the corrector is to identify "the same phase" of a particular waveform by finding the zero-crossing point for the derivative of the solution. Since the solution obtained by least square usually has a small phase shift, the correction only needs a little additional work, at most one fast cycle of transient simulation.

A considerably superior method for the corrector, however, is to find the point in the new cycle being sampled that is *closest to* x_s

in the n-dimensional solution space. Not only does this minimum-distance corrector eliminate dependence of the algorithm on zero-crossings, maxima, *etc.*, of a particular waveform, it enhances and refines what we feel is a key property of LSENV, responsible for its efficacy and robustness: that the Moore-Penrose pseudo-inverse based Newton automatically "finds" the smoothest and most natural envelope in the abstract n-dimensional space of solutions. Indeed, for most circuits, postprocessing to correct the phase is not even necessary, since LSENV automatically converges to smooth envelopes.

4. APPLICATIONS AND VALIDATION

In this section, we apply and validate LSENV on various circuits. We first test our method in oscillators: for predicting envelopes in LC and ring VCOs and startup transient envelopes in high-Q oscillators. We then apply our method to mixers, which are non-autonomous circuits that frequently feature slow envelopes. Finally, we apply LSENV to circuits that change their nature from non-autonomous to autonomous during operation, applying it to predict injection locking/pulling of oscillators and PLL capture/lock envelopes. Results from LSENV show excellent matches against SPICE-like transient simulation, with speedups of 1-2 orders of magnitude. All simulations were performed using a MATLAB-based circuit simulation platform, on a 2.4GHz AthlonXP PC running Linux.

4.1 Oscillator envelope following

4.1.1 LC VCO

Figure 2 shows a simple LC VCO from [16]. The capacitance of the LC tank is varied by a controlling voltage, resulting in changes to the VCO's frequency. The oscillator has a nominal frequency of 10MHz; the controlling voltage is sinusoidal with a frequency of 1kHz. The nonlinear negative resistor characteristic is given by [16]:

$$i = f(v) = (G_0 - G_\infty)V_k tanh(\frac{v}{V_k}) + G_\infty v, \qquad (8)$$

where $G_0 = -0.1, G_\infty = 0.25$, and $V_k = 1$.

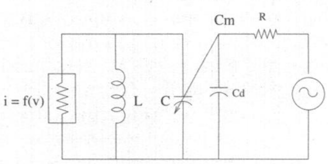
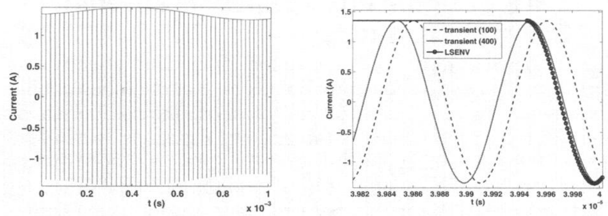


Figure 2: Circuit schematic of a LC VCO. Here $R = 1k\Omega$, $Cd = 0.3uF$, $C = \frac{1}{2\pi}10^{-7}F$, $L = \frac{1}{2\pi}10^{-7}H$, $Cm = \frac{1}{4\pi}10^{-7}F$.

The simulation is started from the nominal steady state of the VCO. We choose the initial condition with the inductor current at the peak of the fast oscillation waveform. The inductor current from LSENV is shown in Figure 3(a). By skipping a large number of fast cycles in each envelope step (and hence avoiding the phase errors that would have accumulated during their simulation), LSENV is actually able to achieve better accuracy than detailed transient simulation, while also being more efficient. Figure 3(b) shows detailed comparison of results from our method and transient simulation. As can be seen, results from LSENV with 100 timesteps per fast cycle match transient simulation result with finer timesteps (400 timesteps per fast cycle).

Furthermore, the envelope solved by LSENV is smooth and the sampling points in envelope are roughly at the same phase (peak of the fast cycles). Hence, for this example, the varying period solved for by LSENV captures the frequency change due to the variation of the capacitance in LC tank, as shown in Figure 4. In this simulation, we use an envelope step of about 200 fast cycles, resulting in a speed up of around 200 over transient simulation with similar accuracy.



(a) LSENV solution (b) Detailed comparison with transient solution

Figure 3: LC VCO: solution of the inductor current from LSENV and comparison with transient simulation. The result in blue dashed line is from transient simulation with 100 timesteps per fast cycle. The result in red solid line is from transient simulation with 400 timesteps per fast cycle. The result marked in circles is from LSENV with 100 timesteps per fast cycle.

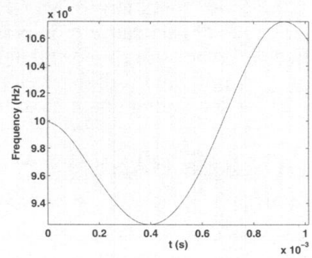


Figure 4: LC VCO: frequency modulation.

4.1.2 3 stage ring VCO

A ring VCO with 3 identical inverting stages is shown in Figure 5. The oscillator has a free-running frequency of 100MHz. Its frequency is varied by changing the stage-to-stage delay, using a MOS resistor controlled by a voltage source. The controlling voltage is $10 + 2sin(2\pi 10^4 t)$.

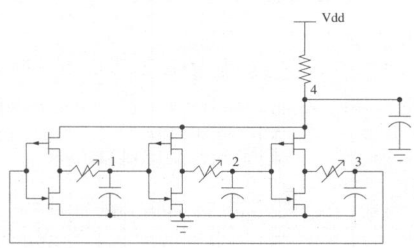
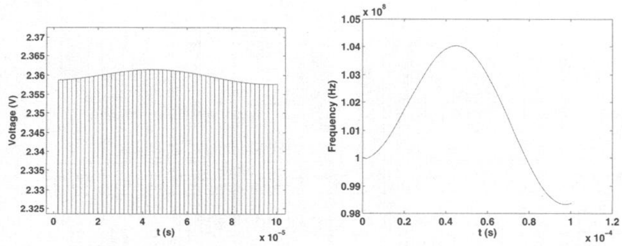


Figure 5: Circuit schematic of a ring VCO.

We start the simulation from a steady state obtained after not changing the VCO control for a long time. Figure 6(a) shows results from LSENV at the output of the 1st stage. Note that it is not necessary for the envelope solution to be at specific phase points, such as the peaks of fast cycles, for the envelope simulation to be useful. However, if necessary, it is easy to obtain the envelope at the peak/bottom of the fast cycles, by applying the postprocessing phase corrector in Section 3.2 to LSENV's results. We observe similar accuracy properties for LSENV compared with transient in this example as in the previous one (transient simulation results are omitted here due to lack of space and the density of the fast oscillations). The slowly varying oscillation frequency is shown in Figure 6(b). A nominal envelope timestep of 200 cycles is used in this example, resulting in a speedup of about $150\times$.



(a) Output of the 1st stage

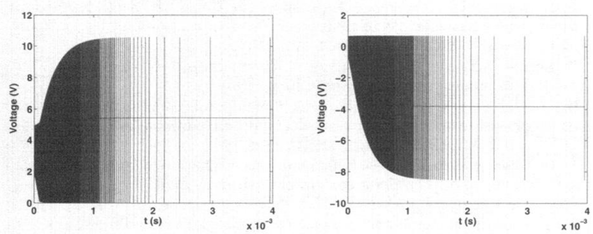
(b) frequency modulation.

Figure 6: Ring VCO: solutions from LSENV.

4.1.3 *Startup transient of a high-Q oscillator*

We use a Pierce crystal oscillator from [22, 23] to demonstrate startup transient of a high-Q oscillator. It has a high quality factor Q about 2.5×10^4, resulting in a very slow startup transient before reaching its steady state, making it ideally suited for investigating envelope algorithms.

Variable envelope stepsizes are used in this case, resulting in further speedups for LSENV. Results from LSENV at the base and the collector of the BJT are shown in Figure 7. As can be seen, the envelope step is small at the beginning but grows as the oscillator approaches steady state and the envelope settles. In this example, the phase at which the selected fast cycles are sampled by the envelope shifts slightly with each envelope step — as mentioned earlier, this is a feature of least-squares solution. The changes in phase are slow and the envelope remains smooth. An average envelope step of about 90 cycles is taken in the simulation, leading to a speedups of 40 over the transient for this simulation.



(a) At the collector

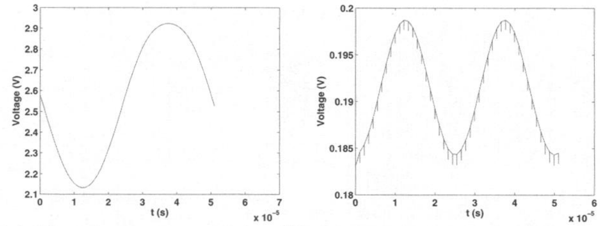
(b) At the base

Figure 7: LSENV solutions at the collector and the base of the BJT in the pierce crystal oscillator.

4.2 LSENV on non-autonomous circuits

We use a balanced CMOS down-conversion mixer from [24] as an example. In this circuit, the current generated by the lower pair of MOSFETs doubles the frequency of the driving LO. The upper pair of MOSFETs is a differential pair that effectively multiplies the RF signal by a distorted LO signal with large components at twice the LO frequency. High frequency components at the output are filtered out by the simple RC filter, thus retaining only low-frequency down-converted components. The LO signal is a sinusoid at 450 MHz, while the RF signal is a 900 MHz carrier modulated by a sinusoid at 10kHz.

In this example, we start the simulation from the DC operating point. Due to the double frequency effect, one fast cycle of LO signal results in two cycles of fast oscillations at the drains of the lower MOSFETs. Hence the amplitude envelope change at this node caused by the slight phase shift may not be small. Therefore, we use the postprocessing phase corrector to correct the phase at each step for this example. Figure 8 shows the results from our method. We take a nominal envelope step of 500 cycles due to the wide separation of frequencies in the signals. A speedup of about 100 is obtained for this example.



(a) Waveform at one of the differential outputs

(b) Waveform at drains of the lower MOSFETs

Figure 8: Waveforms from LSENV.

4.3 Injection locking/pulling in oscillators

Injection locking/pulling [25, 26] is an interesting phenomenon that occurs when a periodic signal is injected into an oscillator. Under the influence of perturbations, an oscillator can change its natural frequency to match the frequency of the injection signal, *i.e.*, it "locks" to the injection signal. Even if the injection signal does not succeed in locking the oscillator, it still pulls the frequency of the oscillator, often resulting in periodic changes in amplitude and frequency. The locking process can be slow and long, especially in high Q oscillators. During the injection pulling process, oscillators start off as autonomous systems; but when locked, essential properties (such as small-signal stability properties) change to mimic those of non-autonomous systems like mixers.

For illustration, we use the simple nonlinear LC oscillator similar to Figure 2. The Q factor of the tank is about 2000, leading to relatively slow injection locking/pulling. The injection signal is a sinusoidal current source in parallel with the LC tank. Figure 9 shows the inductor current from LSENV when the oscillator is finally locked to the injection signal. Figure 10 shows waveforms when the oscillator is not in lock; periodic changes in the amplitude envelope can be seen. These results are compared against careful transient simulation in both cases; they are in good agreement. In this example, we obtain a speedup of 1-2 orders of magnitude. More speedup can be obtained with the higher Q oscillators.

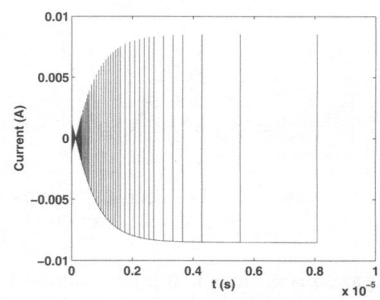
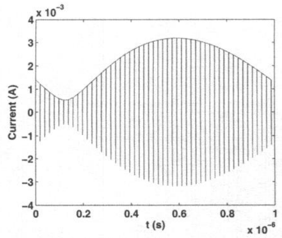


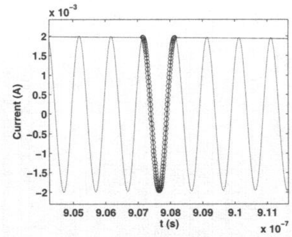
Figure 9: LSENV solution of the inductor current from our method. Oscillator is in lock. Injection signal is: $3 \times 10^{-6} sin(2\pi 1.013 \times 10^9 t)$.

4.4 PLL simulation with LSENV

The PLL used in our simulation is shown in Figure 11. Here, a mixer is used as the phase detector. The center frequency of the LC VCO is about 100 MHz. Figure 12 shows the startup transient at the controlling voltage of the PLL, obtained from LSENV and compared against transient simulation for validation. Similarly, the capacitor voltage within the VCO is shown in Figure 13. As can been, the solution from our method matches transient simulation well. In this case, we obtain relatively modest speedups of $2\times$ and $8\times$, over coarse (100 steps per fast cycle) and finer (400 steps per cycle) transient simulation, respectively. The small speedup is due



(a) LSENV solution



(b) Detailed comparisons of the inductor current from our method and transient.

Figure 10: Results when the oscillator is not in lock. Injection signal is: $3 \times 10^{-6} sin(2\pi 1.014 \times 10^{9} t)$.

to the high bandwidth of the low pass filter in the design, leading to a fairly small separation between the fast and slow time scales of the signals in the circuit. If the bandwidth of the low pass filter is decreased (this is typical in practice, to reduce the noise and non-idealities), larger speedups can be obtained.

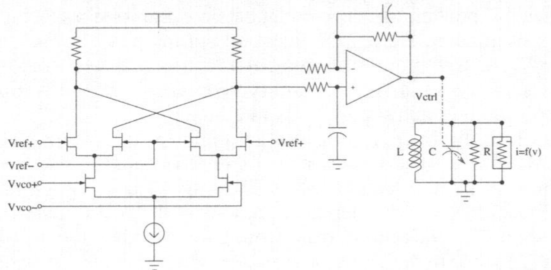
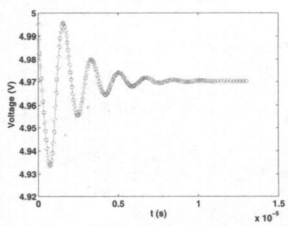
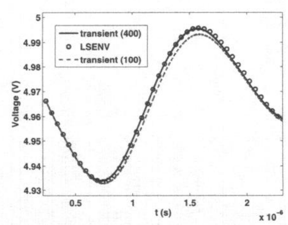


Figure 11: Schematic of the PLL used in our simulation.



(a) LSENV solution



(b) Detailed comparison with transient solution

Figure 12: Comparisons of the startup transient of the controlling voltage from our method and transient. The result in red dashed line is from transient simulation with 100 timesteps per fast cycle. The result in black solid line is from transient simulation with 400 timesteps per fast cycle. The result marked in circles is from LSENV with 100 timesteps per fast cycle.

5. CONCLUSIONS

We have presented a robust and general technique that is capable of tracking envelopes of both non-autonomous and autonomous systems. We have demonstrated that our method works robustly and uniformly for these two types of systems, and is suitable for predicting injection locking process in single oscillators, as well as PLL circuits, in which circuits change their nature from autonomous to non-autonomous. Our simulation results show perfect match with SPICE-like transient simulation, validating efficiency and accuracy of our method.

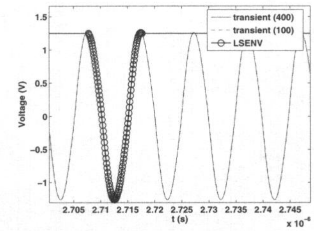


Figure 13: Detailed comparison of waveform of the capacitor voltage in the VCO from our method and transient solution.

6. REFERENCES

[1] L.W. Nagel. SPICE2: a computer program to simulate semiconductor circuits. PhD thesis, EECS department, University of California, Berkeley, Electronics Research Laboratory, 1975. Memorandum no. ERL-M520.

[2] Thomas L. Quarles. SPICE 3C.1 User's Guide. University of California, Berkeley, EECS Industrial Liaison Program, University of California, Berkeley California, 94720, April 1989.

[3] L. Petzold. An efficient numerical method for highly oscillatory ordinary differential equations. SIAM J. Numer. Anal., 18(3):455 – 479, 1981.

[4] K. Kundert, J. White, and A. Sangiovanni. An envelope-following method for the efficient transient simulation of switching power and filter circuits. In Proc. ICCAD, pages 446 – 449, 1988.

[5] J. White and S.B. Leeb. An envelope-following approach to switching power converter simulation. IEEE Transactions on Power Electronics, 6:303 – 307, Apr 1991.

[6] C. Deml and P. Turkes. Fast simulation technique for power electronic circuits with widely different time constants. In Thirty-Second IAS Annual Meeting, IAS '97, volume 2, pages 1204 – 1210, Oct 1997.

[7] L.M Silveira, J. White, and S. Leeb. A modified envelope-following approach to clocked analog circuit simulation. In Proc. ICCAD, pages 20 – 23, Nov 1991.

[8] A. Brambilla and P. Maffezzoni. Envelope-following method to compute steady-state solutions of electrical circuits. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 50:407 – 417, Mar 2003.

[9] E. Ngoya and R. Larchevéque. Envelop transient analysis: a new method for the transient and steady state analysis of microwave communication circuits and systems. In Proc. IEEE MTT Symp, 1996.

[10] D. Sharrit. New Method of Analysis of Communication Systems. In MTTS'96 WMFA: Nonlinear CAD Workshop, June 1996.

[11] P. Feldmann and J. Roychowdhury. Computation of circuit waveform envelopes using an efficient, matrix-decomposed harmonic balance algorithm. In Proc. ICCAD, 1996.

[12] H. G. Brachtendorf, G. Welsch, R. Laur, and A. Bunse-Gerstner. Numerical steady state analysis of electronic circuits driven by multi-tone signals. Electrical Engineering, 79:103–112, 1996.

[13] J. Roychowdhury. Analyzing circuits with widely separated time scales using numerical PDE methods. IEEE Transactions on Circuits and Systems — I: Fundamental Theory and Applications, 48(5):578–594, 2001.

[14] J. Roychowdhury. Efficient methods for simulating highly nonlinear multi-rate circuits. In Proc. IEEE DAC, 1997.

[15] C. W. Gear and K. Gallivan. Automatic methods for highly oscillatory ordinary differential equations. In Lecture Notes in Mathematics No. 912: Numerical Analysis, pages 115–124, Berlin, 1981. Springer-Verlag.

[16] O. Narayan and J. Roychowdhury. Analysing Oscillators using Multitime PDEs. IEEE Transactions on Circuits and Systems — I: Fundamental Theory and Applications, 50(7):894–903, 2003.

[17] T. Mei and J. Roychowdhury. An efficient and robust technique for tracking amplitude and frequency envelopes in oscillators. ICCAD, 2005.

[18] E.L. Allgower and K. Georg. Numerical Continuation Methods. Springer-Verlag, New York, 1990.

[19] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Numerical Recipes – The Art of Scientific Computing. Cambridge University Press, 1989.

[20] K.S. Kundert, J.K. White, and A. Sangiovanni-Vincentelli. Steady-state methods for simulating analog and microwave circuits. Kluwer Academic Publishers, 1990.

[21] T.J. Aprille and T.N. Trick. Steady-state analysis of nonlinear circuits with period inputs. Proc. IEEE, 60:108–114, 1972.

[22] M. Gourary, S. Ulyanov, M. Zharov, S. Rusakov, K.K.Gullapalli, and B.J. Mulvaney. Simulation of high-q oscillators. In Proc. ICCAD, pages 162 – 169, 1998.

[23] M. Gourary, S. Ulyanov, M. Zharov, S. Rusakov, K.K.Gullapalli, and B.J. Mulvaney. A robust and efficient oscillator analysis technique using harmonic balance. Computer Methods in Applied Mechanics and Engineering, 181:451–466, Jan 2000.

[24] Z. Chen Z. Zhang and J. Lau. A 900mhz cmos balanced harmonic mixer for direct conversion receiver. In Proc. IEEE Radio and Wireless Conference, pages 219–222, 2000.

[25] R. Adler. A study of locking phenomena in oscillators. Proc. IEEE, 61:1380–1385, 1973.

[26] K. Kurokawa. Injection locking of microwave solid-state oscillators. Proc. IEEE, 61:1336–1410, 1973.