Implementing Nonlinear Oscillator Macromodels using Verilog-AMS for Accurate Prediction of Injection Locking Behaviors of Oscillators

Ben Gu, Kiran K. Gullapalli, Steven Hamm, Brian Mulvaney,
MICA Circuit Simulation Group
Freescale Semiconductors
Xiaolue Lai, and Jaijeet Roychowdhury
Department of Electrical and Computer Engineering
University of Minnesota

***Abstract*— Oscillators are widely used in electronic systems but are inefficient to simulate using traditional methods. Due to this, the macromodeling of oscillators to speed up circuit simulations has been studied intensively, but various drawbacks have limited the applicability of the methods. Recently, a more physically correct approach to macro-model oscillators has been proposed. The new macromodel utilizes the results of Floquet analysis [1] and successfully captures the nonlinear dynamics of oscillators and correctly predicts their phase response in the presence of interference. The new method requires far less computational effort compared to SPICE–level simulations.**

In this paper, we present an implementation this nonlinear oscillator macromodel using the Verilog-AMS language. Several features in Verilog-AMS allow for a concise implementation. The Verilog-AMS macromodel is used to study the injection locking behavior of an LC oscillator, and the simulation results match full SPICE transient simulations accurately. An advantage of this implementation is that it can be used in many SPICE–like circuit simulators due to the wide availability of Verilog-AMS compilers and the standardization of the language.

I. Introduction

Oscillators are critical components of electronic systems. They are especailly important in the front end of communication circuits [2]. Design of oscillating systems is an important part of the overall system design. However, simulating oscillators presents unique challenges because of their fundamental property of neutral phase stability.

Traditional simulation methods in SPICE–like simulators often consume significant CPU time to simulate the transient behavior of oscillating systems, such as phase-locked loops (PLLs). This is especially true for oscillators under interference, since small time steps and many simulation cycles are required. As a result, macromodels of oscillators have been used for decades to speed up circuit simulations and facilitate system–level simulations. This has been an active research topic and extensive results have been published, but most of them have been based on the linear perturbation analysis and have failed to correctly predict important nonlinear transient behaviors, such as injection locking and cycle slipping.

Recently a fundamentally more correct modeling approach was proposed [3], [4]. The new macromodel utilizes the results of nonlinear Floquet analysis, which was used in the rigorous analysis of free–running oscillators' phase noise [1]. The new method consists of a novel algorithm to compute the oscillators' phase and amplitude deviations by perturbations using the Floquet vectors at the nodes where the perturbations are applied. The macromodel produced is a combination of a scalar nonlinear differential equation and a reduced linear time–varying system which is computationally much simpler and of smaller dimension than the original oscillator equations, resulting in significant speedups in simulations. It has been demonstrated that the new nonlinear oscillator macromodel can accurately predict those important nonlinear behaviors which linear macromodels cannot. This nonlinear macromodeling concept was used in [3] to model and simulate the phase behavior of VCOs in PLLs and was shown to be more accurate than linear phase–domain approaches [5].

In this work, as an extension to the original results in [4], we present implentation details of this nonlinear oscillator macromodel using the Verilog–AMS language [6]. Because of its wide availibility in various commerical/in–house SPICE–like simulators and its standardization across the industry, the Verilog-AMS language is becoming popular in the behavioral modeling and simulation communities. Implementing the nonlinear oscillator model using Verilog-AMS language makes it possible for circuit and system designers to simulate the oscillators together with other blocks modeled at the SPICE or Verilog-AMS level. We show in this work that the nonlinear macromodel can be implemented very concisely in Verilog–AMS language and be readily plugged into a SPICE-like circuit simulator.

We use the Verilog–AMS oscillator macromodel in Mica, Freescale's in-house analog/RF/mixed-signal circuit simulator, to simulate the injection locking behavior of an LC oscillator, which the linear macromodels fail to predict accurately. We provide comparisons of simulations using the Verilog–AMS oscillator macromodel generated for an LC oscillator versus full SPICE–level simulations, under different perturbations. Our simulation results show the validity of the Verilog–AMS implementation and illustrate the advantages of the nonlinear macromodel over linear macromodels.

The remainder of the paper is organized as follows: In Section II, we briefly review the theory of the nonlinear oscillator macromodel. In Section III, we present the details of implementing the nonlinear macromodel using Verilog–AMS. In Section IV, we present simulation results of applying the Verilog–AMS macromodel to study the injection locking behaviors of an LC oscillator and compare the results with full SPICE–level simulation.

II. The Nonlinear Oscillator Macromodel

The standard approach for analyzing perturbed nonlinear systems is to linearize around an unperturbed trajectory. However, this approach is not sufficient for analyzing oscillators. In [1], a novel phase macromodel based on nonlinear perturbation analysis was presented. In this section, we provide a brief review of the nonlinear phase model in [1], which was adapted in [4] in order to build the nonlinear oscillator phase domain macromodel.

A general oscillator under perturbation can be expressed as

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

where $b(t)$ is a perturbation applied to the free-running oscillator and $x(t)$ is a vector of the oscillator's state variables. For small perturbations, we can linearize (1) about its unperturbed orbit as

$$\begin{aligned} \dot{w}(t) \approx & -\frac{\partial f(x)}{\partial x}\Big|_{x_s(t)} w(t) + b(t) \\ = & A(t)w(t) + b(t), \end{aligned} \tag{2}$$

where $w(t)$ represents deviations due to perturbations and $x_s(t)$ is the unperturbed steady-state solution of the oscillator. The periodic time-varying linear system ((2)) can be solved using Floquet theory [7] which obtains an expression for its state transition matrix

$$\Phi(t,\tau) = U(t)\exp(D(t-\tau))V(\tau). \tag{3}$$

$U(t)$ and $V(t)$ are T-periodic nonsingular matrices, satisfying biorthogonality condition $v_i^T(t)u_j(t) = \delta_{ij}$, and $D = diag[\mu_1, ..., \mu_n]$, where μ_i are the Floquet exponents. As shown in [1], one of the Floquet exponents must be 0, and $\dot{x}_s(t)$ is one of the solutions of $w(t) = A(t)w(t)$, the homogenous part of (2).

Without loss of generality, we choose $\mu_1 = 0$ and $u_1(t) = \dot{x}_s(t)$. The *perturbation projection vector* (PPV) $v_1(t)$ satisfies $v_1^T(t)u_1(t) = 1$ [1], [8]. The PPV, which represents the oscillator's phase sensitivity to perturbations, is a periodic vector waveform with the same period as the unperturbed oscillator.

The particular solution of (2) is given by

$$w(t) = \sum_{i=1}^{n} u_i(t) \int_0^t \exp(\mu_i(t-\tau)) v_i^T(\tau) b(\tau) d\tau, \tag{4}$$

where $\mu_1 = 0$. A small perturbation $b(t)$ with the same frequency as $v_1(t)$ can always be chosen to satisfy that $v_i^T(t)b(t)$ has a nonzero average value; hence $w(t)$ can be made to grow unboundedly with t, despite $b(t)$ always remaining small. This contradicts the basic assumption for perturbation analysis, i.e. that $w(t)$ is always small.

A. Nonlinear Phase Macromodel

To resolve this contradiction, a key innovation of [1] was to rewrite (1) with the perturbation $b(t)$ split into two components

$$\dot{x} - f(x) = b_1(t) + \tilde{b}(t), \tag{5}$$

where

$$b_1(t) = v_1^T(t+\alpha(t))b(t)u_1(t+\alpha(t)) \tag{6}$$

induces *only phase deviations* to the unperturbed system, while

$$\tilde{b}(t) = \sum_{i=2}^{n} v_i^T(t+\alpha(t))b(t)u_i(t+\alpha(t)) \tag{7}$$

contributes to orbital deviations. The solution of $\dot{x} + f(x) = b_1(t)$ is in fact given by

$$x_p(t) = x_s(t+\alpha(t)), \tag{8}$$

where $\alpha(t)$ is the phase deviation due to the perturbation $b_1(t)$. Indeed, $\alpha(t)$ satisfies the nonlinear differential equation

$$\dot{\alpha}(t) = v_1^T(t+\alpha(t)) \cdot b(t). \tag{9}$$

where $v_1(t)$ is the PPV and $\alpha(t)$ in nonlinear has units of time. Phase deviation in radians can be obtained by multiplying $\alpha(t)$ by the free running oscillation frequency ω_0. With the PPV $v_1(t)$ available for a given oscillator, its phase deviations due to perturbations can be efficiently evaluated by solving the one–dimensional nonlinear differential equation (9). Effective numerical methods are available for computing the PPV from a SPICE–level description of the oscillator [1], [8] in either time or frequency domains.

III. Verilog–AMS implementation of the Nonlinear Oscillator Macromodel

In this section, we use the Verilog–AMS language to implement the nonlinear macromodel discussed in the last section. We use a simple example: the LC oscillator presented in [4]. Figure 1 depicts the block diagram of a simple LC oscillator, whose differential equations are

$$\begin{aligned} -C\frac{d}{dt}v(t) &= \frac{v(t)}{R} + i(t) + S\tanh\left(\frac{G_n}{S}v(t)\right) + b(t) \\ L\frac{d}{dt}i(t) &= v(t). \end{aligned} \tag{10}$$

$L = 4.869 \times 10^{-7}/(2\pi)\, H$, $C = 2 \times 10^{-12}/(2\pi)\, F$, $R = 100\,\Omega$, $S = 1/R$ and $G_n = -1.1/R$. With these parameters, the LC tank has a resonance frequency of 1 GHz, and the inductor current has amplitude $A_0 = 1.2$mA.

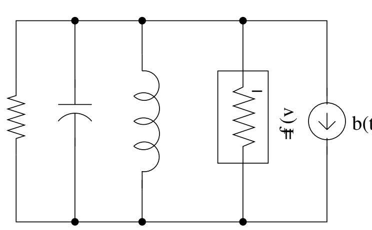


Fig. 1. A simple LC oscillator.

This simple circuit can be trivially simulated using Mica's oscillator analysis algorithm, and one period of the PPV waveform of node 1 is presented in Figure 2.

After knowing $v_1(t)$, the phase deviation $\alpha(t)$ of the LC oscillator under any perturbation, $b(t)$, at the output node, can be predicted by solving the nonlinear differential equation (9), and the output of the oscillator becomes $V(t+\alpha(t))$. The Verilog–AMS code of the nonlinear macromodel of the LC oscillator is presented in Figure III.

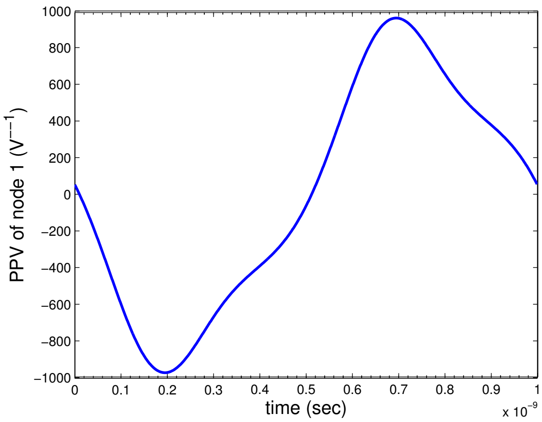


Fig. 2. PPV of the output node of the LC oscillator.

In Verilog–AMS, the LC oscillator can be implemented as a module with two nodes, an input node taking perturbation, and an output node generating the output oscillating waveform. (9) can be set up by creating an internal node, alpha, and two branches, alpha1 and alpha2, between alpha and ground. The voltage at node alpha, V(alpha), is the α in (9). The two branches are contributed by ddt(V(alpha)) and $-v_1$(V(alpha)+$abstime) $\cdot$ V(in). When this Verilog–AMS module is plugged into Mica for a transient simulation, as simulation proceeds, $abstime increases and the KCL equation at node alpha will force (9) to be solved at each time step. After (9) is solved, the perturbed time, $t+\alpha(t)$, is obtained, and the output waveform of the oscillator is just the unperturbed waveform shifted by $\alpha(t)$.

A feature of Verilog–AMS, $table_model is particularly useful in setting up (9). The $table_model function models the behavior of a system by interpolating between data points that are samples of that system's behavior. To build such a model, users need to provide a set of sample points, $(x_{i1}, x_{i2}, ..., x_{iN}, y_i)$ so that $f(x_{i1}, x_{i2}, ..., x_{iN}) = y_i$ [6]. For our purpose, we need to build a table describing the dependence of $v_1(t)$ on t. Taking into account of the periodicity of the PPV waveform, $v_1(t)$, we only need to store one period of it in a table, "osc_ppv.table", and index the table using the perturbed time's period modulus to get the corresponding PPV value, $v_1(t+\alpha(t))$. Similarly we store one period of the oscillator's output waveform in another table, "osc_output.table", and use it to generate the perturbed output.

An example of the PPV table with 17 time points is given in Table I. However, we used a much larger table (513 time points) to produce all the results presented in this work. The second column of Table I is the value of the PPV, $v_1(t)$, at t given in the first column, which is obtained from Mica's oscillator analysis using harmonic balance method.

The macromodel building process is very streamlined. The major step is simulating the full-size oscillator (the SPICE level description) using Mica's oscillator analysis algorithms either in time or frequency domain to obtain one period of the steady state and the PPV waveforms of the interesting outputs and inputs (where perturbations are applied).

```
`include "discipline.h"
`include "constants.h"

// nonlinear macromodel implemented in Verilog-AMS
module oscillator(in, out);
// define nodes
inout      in, out;
electrical in, out, alpha;

// define variables and parameter
real phase, ppv, period, omega,
     perturbed_time;

parameter real freq=1.0e9 from(0:inf);

// define branches
branch (alpha) alpha1;
branch (alpha) alpha2;

analog
begin
@(initial_step)
    begin
    //set up initial condition for eq(9)
    V(alpha) <+ 0;
    omega =  2.0*`M_PI*freq;
    period = 1.0/freq;
    end

    // real perturbed time is given by
    // $abstime + V(alpha)
    // but our ppv table has only one period
    // so take the modulus of period

    perturbed_time = ($abstime+V(alpha)) % period;

    // look up table to get ppv value
    // $table_model(arg1, arg2, arg3)
    // arg1: input (independent) variable
    // arg2: name of the file storing the table
    // arg3: interpolation method, L-linear

    ppv =  $table_model(perturbed_time,
           "osc_ppv.table", "L");

    // right hand side of eq(9)
    // ppv:  v1(t+alpha(t))
    // V(in):b(t)
    I(alpha1) <+ -ppv*V(in);

    // left hand side of eq(9)
    I(alpha2) <+  ddt(V(alpha));

    // The KCL at alpha forces
    //        I(alpha1) + I(alpha2) = 0
    // so that eq(9) is solved at each time point

    phase = V(alpha)*omega; // output only

    // look up table to generate output
    V(out) <+ $table_model(perturbed_time,
               "osc_output.table", "L");
end
endmodule
```

Fig. 3. Verilog–AMS code of the nonlinear macromodel of the LC oscillator.

TABLE I

PPV TABLE USED IN VERILOG–AMS NONLINEAR OSCILLATOR MACROMODEL

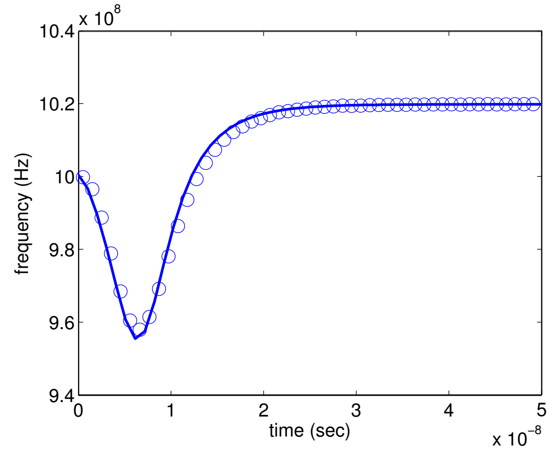
time (sec)	PPV (V^{-1})
0.000000e+00	5.842040e+01
5.881575e-11	-3.02776e+02
1.176315e-10	-7.10414e+02
1.764472e-10	-9.54355e+02
2.352630e-10	-9.06342e+02
2.940787e-10	-6.87832e+02
3.528945e-10	-4.95333e+02
4.117102e-10	-3.59686e+02
4.705260e-10	-1.87722e+02
5.293417e-10	1.069284e+02
5.881575e-10	5.126373e+02
6.469732e-10	8.659865e+02
7.057890e-10	9.649073e+02
7.646047e-10	8.038073e+02
8.234205e-10	5.817091e+02
8.822362e-10	4.252280e+02
9.410520e-10	2.848354e+02

As discussed in [4], the same approach can be used to build macromodels for oscillators' amplitude deviation as well. However, in this work, we limit our discussions to phase domain macromodel only because of its significance in oscillator behaviors.

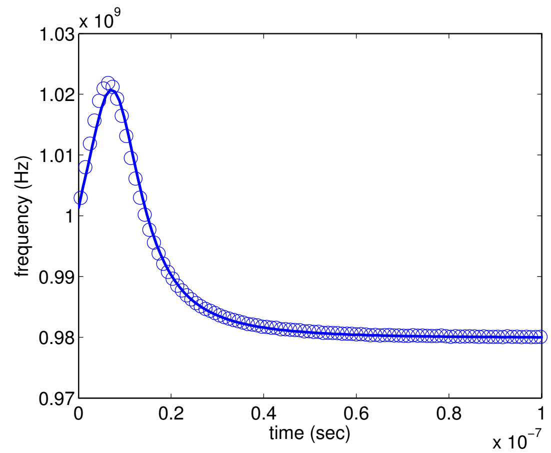
IV. Simulation Results

One of the most interesting applications of the nonlinear oscillator macromodel is to simulate the injection locking behavior of oscillators, a phenomenon which the linear macromodels cannot correctly predict. Injection locking is a nonlinear dynamical phenomenon occurring in all oscillators. When an oscillator is perturbed by a weak external signal close to its free–running frequency, the oscillator's frequency changes to become identical to that of the perturbing signal [9]. Capturing injection locking using traditional simulation presents challenges. SPICE–level simulation of oscillators is usually inefficient, since oscillators often require thousands of cycles to lock to an injecting signal, with each simulation cycle requiring large numbers of very small timesteps for acceptable phase accuracy. If the frequency of the injected signal is close to oscillator's free-running frequency, it also becomes very difficult to determine injection locking from observing time-domain waveforms.

In this section, we study the injection locking behavior using the Verilog–AMS macromodel from Section III. Sinusoidal perturbations at different frequencies and amplitudes are injected at the output node of the LC oscillator, and its transient behavior is simulated using the Verilog–AMS macromodel and the full SPICE transient simulation respectively. The frequency changes of the oscillator under different sinusoidal perturbations are presented in Figure 4 and Figure 5, where symbols are the results of the full SPICE simulations and solid lines are those using the Verilog–AMS macromodel. When the frequency of the injection perturbation is close to the oscillation frequency, the oscillator can lock into the injection signal, as shown in Figure 4. The simulation results using the Verilog–AMS match those by full SPICE transient simulations closely. As a result of the nonlinear dynamics of the oscillator being well captured by (9), the nonlinear macromodel predicts the transient locking process as well as the eventual locking behavior accurately. If the frequency of the perturbation is out of the oscillator's locking range [4], the oscillator cannot synchronize with the injection, and the system shows an unstable oscillation, as shown in Figure 5. Even in this case, the Verilog–AMS macromodel's prediction of the transient behaviors still matches the full SPICE simulations accurately.



(a) Perturbation frequency = 1020MHz; amplitude=100uA.

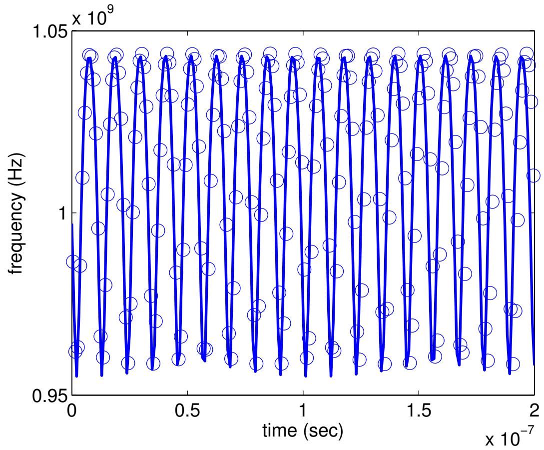


(b) Perturbation frequency = 980MHz; amplitude=50uA.

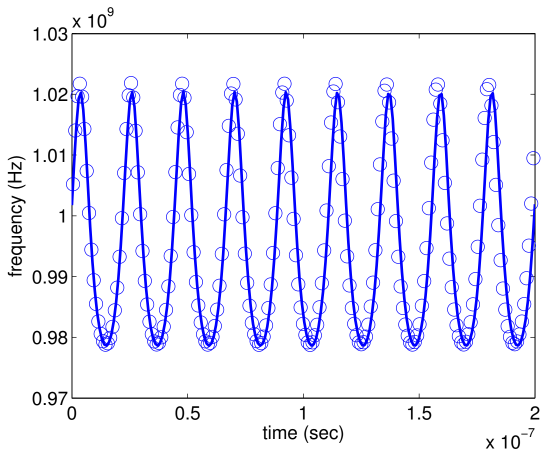
Fig. 4. Frequency Change of the LC oscillator under the perturbations close to its oscillation frequency.

V. Conclusions

We have presented a Verilog–AMS implementation of the nonlinear oscillator macromodel. The Verilog–AMS language allows for a very concise implementation of the nonlin-



(a) Perturbation frequency = 1100MHz; amplitude=100uA.



(b) Perturbation frequency = 950MHz; amplitude=50uA.

Fig. 5. Frequency Change of the LC oscillator under the perturbations out of its locking range.

ear macromodel. The generated macromodel can be readily plugged into any SPICE–like simulator which supports Verilog–AMS to study the behaviors of oscillators and perform system simulations. We have used the macromodel to investigate the injection locking behaviors of an LC oscillator. The simulation results using the Verilog–AMS match the full SPICE simulations faithfully.

VI. Acknowledgement

The authors in Freescale are grateful to all the members of the Freescale Mica team for being wonderful to work with, and to the Freescale Australian team for developing the Verilog–AMS compiler and evaluator integrated in Mica which were used to produce all the results presented in this work.

References

[1] 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.

[2] J. Laskar, B. Matinpour, and S. Chakraborty. *Modern Receiver Front–Ends*. John Wiley & Sons, 2004.

[3] X. Lai, Wan Y, 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 *IEEE Asia South-Pacific Design Automation Conference*, January 2005.

[4] X. Lai and J. Roychowdhury. Automated oscillator macromodelling techniques for capturing amplitude variations and injection locking. In *IEEE/ACM International Conference on Computer Aided Design*, pages 687–694, November 2004.

[5] K. Kundert. *Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers*. http //www.designers-guide.org, 2002.

[6] Accellera Verilog Analog Mixed-Signal Group. *Verilog-AMS Language Reference Manual*. http //www.eda.org/verilog-ams, 2005.

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

[8] 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.

[9] R. Adler. A study of locking phenomena in oscillators. *Proceedings of the I.R.E. and Waves and Electrons*, 34:351–357, June 1946.