Fast and accurate time-domain simulations of integer-N PLLs

Citation for published version (APA):

De Luca, G., Bolcato, P., Larcheveque, R., Rommes, J., & Schilders, W. H. A. (2017). Fast and accurate time-domain simulations of integer-N PLLs. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(4), 931-944. [7792570]. <https://doi.org/10.1109/TCSI.2016.2628323>

DOI:

[10.1109/TCSI.2016.2628323](https://doi.org/10.1109/TCSI.2016.2628323)

Document status and date:

Published: 01/04/2017

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Fast and Accurate Time-Domain Simulations of Integer-N PLLs

Giovanni De Luca, Pascal Bolcato, Remi Larcheveque, Joost Rommes, Wil H. A. Schilders

Abstract—We present a methodology to simulate industrial integer- N phase-locked loops (PLLs) at a verification level, as accurate as and faster than transistor-level simulation. The accuracy is measured on the PLL factors of interest, i.e., locking time, power consumption, phase noise and jitter (period and long-term). The speedup factor tends to the division ratio N for device-noise simulations.

We develop a unifying technique which is able to deal with both noise-free and device-noise analyses, taking into account nonlinear and second-order effects visible at transistor-level simulation only, whereas previous works focused on one of the two analyses, separately. The procedure is based on oscillator's sensitivity analysis and on the creation of a phase macromodel for the voltage-controlled oscillator (VCO) together with the loop divider (the phase model is called VCODIV), whilst the other PLL's blocks remain at transistor level. The macromodel's phase law is characterized by a piecewise linear curve, representing the sensitivity of the VCODIV output's phase deviation with respect to the voltage variation of the VCO's control pin, and by the effects of all the VCO's and divider's noise sources on the model's output. We show two experiments on industrial PLLs, and provide guidelines for designers which highlight the steps needed to implement the methodology by using well-known analyses in circuit simulation and Verilog-A for the creation of the macromodel.

Index Terms—Phase-locked loops, noise-free/device-noise transient analyses, phase noise, jitter, phase model, piecewise linear reduction, perturbation projection vector.

I. INTRODUCTION

CIRCUIT miniaturization (Moore's law) and diversification (More than Moore) are very common trends in the electronic world. These imply high-volume, power-efficiency and functional heterogeneity in integrated circuit (IC) and System-on-Chip (SoC) designs, all while increasing performance (e.g., extended battery life of portable devices, potentially faster computations, patient monitoring through wearable medical devices). However, these trends also result in an increased complexity of the design and verification process in the Electronic Design Automation (EDA) industry: design considerations, i.e., Process, Voltage and Temperature variation, power consumption, process constraints and sensitivity to noise have to be taken into account.

The verification process of ICs incorporating intellectual property blocks heavily relies on pre- and post-layout SPICE

simulation. There is a constant request to increase the simulation speed, while keeping a high accuracy in order to run more tests and other statistical analyses (e.g., Monte Carlo). Circuits for which even Faster-SPICE engines have very long simulation time are the phase-locked loops (PLLs). PLLs find wide applications in areas such as communications, wireless systems, digital circuits and medical devices. These are still medium size circuits ($10^3 - 10^4$ active devices and $10^3 - 10^5$ linear devices) but designers want to simulate them for a period of physical time that is orders of magnitudes higher than the characteristic time of the circuit meaningful phenomena [1] (e.g., at 45nm the rise/fall time is 10^{-12} seconds while the PLL calibration/lock could be in the range of 10^{-3} seconds). Besides, to validate the design in the presence of noise sources acting on the circuit, the analysis becomes more time-consuming to see modulation effects and spectral purity of the generated signal. All this translates into a long Time-to-Market.

PLLs are multi-rate circuits, i.e., described by differential equations with widely separated time constants (fast- and slow-varying dynamics). Consequently, time-domain numerical methods require a huge number of time points where to compute the solutions. When used as frequency synthesizers (divider block in the feedback loop), PLLs can be classified as integer- N and fractional- N . They generate a signal with high oscillating frequency f_{out} from a slower reference with f_{ref} . When switched on, a PLL has a transient behavior followed by a locking condition, in which $f_{out} = N \cdot f_{ref}$ and the phase difference between the two signals is constant.

The factors which industries are interested in are *power consumption*, *locking time*, *phase noise* and *jitter*. Estimation of average power consumption is very important for portable devices, as well as it is related to the life of the circuit's components [2]; locking time determines the instant in which the PLL starts to work properly (transient effects die out and a periodic steady-state (SST) is approached) and is a measure of how fast the PLL responds to a frequency sweep; jitter and phase noise [3], [6] are two key-factors identifying the quality of the produced signal (in the locked state), important to assess when the PLL is used to produce a clock driving other blocks or to quantify inter-channels interferences for communication systems. All these characteristics are usually extracted through time-domain simulations, which are currently the most reliable for this purpose.

Since all the factors of interest mentioned above are equally important, and to the best of our knowledge a single method to extract all of them is still missing in the state-of-the-art, we propose a technique, targeted to integer- N PLLs with

G. De Luca and W. H. A. Schilders are with the Department of Mathematics and Computer Science, Technical University of Eindhoven, NL, (see <http://www.win.tue.nl/~gdeluca/> and <http://www.win.tue.nl/~wschilde/>).

P. Bolcato (pascal_bolcato@mentor.com), R. Larcheveque (remi_larcheveque@mentor.com) and J. Rommes (joost_rommes@mentor.com) are with Mentor Graphics®, Grenoble, FR.

Manuscript received MM DD, YY; revised MM DD, YY.

high N , to accurately estimate them while accelerating noise-free and device-noise simulations (for any kind of noise), by replacing only the VCO and divider block with a phase macromodel (for fractional- N PLLs, some works have been done in, e.g., [4], [7] and [8]). We are motivated in creating a phase model just for the VCO and divider (seen as a single oscillator) since the perturbation theory on oscillators is robust [13], while doing so for the other PLL's blocks might require simplifications on the circuit's behavioral model, resulting in a loss of accuracy with respect to transistor-level simulations. Besides, our methodology is based on well-known analyses in the circuit simulation area, i.e., SST computation-based methods (Harmonic Balance (HB) [9] or the Shooting method [10]) and SST noise analyses (the conversion matrix method [11], [12] or the perturbation projection vector (PPV) [13], [14]) for the construction of the phase macromodel, so that designers could be able to validate it with a little effort. The creation of the phase model can be done, e.g., in Verilog-A language, enriching it with equations and input parameters we provide here, together with guidelines for our procedure, with some remarks on how to automate it.

The remainder of this paper is organized as follows. In Section II we present the problem of long simulation time for PLLs and the state-of-the-art. In Section III we describe our methodology, based on oscillators' perturbation and periodic circuit analyses, as well as we formalize the factors of interest and how they are estimated from both standard transistor-level simulation and with our macromodel. At the end of this section, we provide guidelines to test/implement the method. In Section IV we show results from simulating two industrial PLLs with division ratio $N = 25$ and $N = 50$, respectively. Section V is devoted to conclusions and some future perspectives.

II. PROBLEM FORMULATION

Usually, a circuit is described by a system of nonlinear differential algebraic equations (DAEs) of the form

$$\begin{cases} \frac{dq(x(t))}{dt} + i(x(t)) = b(t), \\ x(t_0) = x_0, \end{cases} \quad (1)$$

to solve for $t \in [t_0, t_{end}]$, which comes from applying the Kirchhoff's current/voltage laws and branch equations to the circuit. In (1), $x(t) \in \mathcal{R}^n$ is the vector-solution of dimension n , $q(\cdot), i(\cdot) : \mathcal{R}^n \rightarrow \mathcal{R}^n$ are the nonlinear functions of the dynamic and static elements, respectively. $b(t) \in \mathcal{R}^n$ is the vector of time-varying and constant sources; for a device-noise simulation, some of its components are enriched with equivalent, random noise sources, making (1) a set of stochastic differential equations. A time-domain analysis can be performed by numerical integration schemes on (1) within $[t_0, t_{end}]$ [15]. The simulation time is then affected, among other factors, by the time step taken during the integration scheme.

For PLLs, the time step is related to the signal with the highest frequency, thus a huge number of steps needs to be retained especially for PLLs working at GHz frequencies. Besides, when running a device-noise simulation, the time

step is even more limited because of the large noise frequency bandwidth to consider and to discriminate between numerical and (equivalent) physical noise. In fact, due to unavoidable numerical errors introduced by, e.g., integration schemes (truncation of high-order terms in approximating continuous-time with finite-difference derivatives), the time step has to be taken small enough to properly compute waveforms' zero-crossing values, useful to extract phase noise and jitter as post-processing procedures, which are not error-free routines. Furthermore, these last two metrics require a long simulation time when the PLL is locked, in order to see the effect of the noise on the PLL's output period (for jitter) and to compute the output signal's power which is spread around its frequency (for phase noise). All this results in an intractable simulation time.

In the literature, many works have been done to speed up PLL's analyses, both at design level (where accuracy can be sacrificed in favor of a faster simulation), and at the verification level (where accuracy is more important). Some are based on behavioral models (partial or full replacement of PLL's blocks by macromodels), while others on full transistor-level description of the circuit. However, none of them reports all the factors of interest we mentioned before. Hereafter, we list a few works.

In [16], the authors present an approach for phase noise behavioral simulation of charge pump-PLLs at steady state. They provide closed-form expressions for PLL blocks' noise transfer functions to extract the phase noise spectrum. However, their method assumes the PLL is in its SST, thus no power consumption nor locking time can be extracted during transient behaviors. Same assumption (PLL at SST) was made in [3] for integer- N PLLs and in [4] for fractional- N PLLs, where the circuit's blocks are recast as phase macromodels to predict phase noise and jitter. Neither locking time nor power estimation at SST are computed.

In [17], the authors use phase macromodels for each PLL's block. They compute perturbation-projection vectors [13] for different VCO's steady states, and interpolate the operating regions through a trajectory piecewise linear (PWL) approach. By doing so, they are able to simulate the PLL's transient and SST periods, reporting a huge speedup factor. Nevertheless, they build a phase model for the charge pump with an ideal response. Phase models are robust for oscillators, whereas for other blocks simplifications might generate over- or under-estimations of noise effects on the whole circuit. Besides, no power consumption nor jitter results are reported.

In [18], they present a method targeted to PLLs' noise performance with respect to variation of some parameters. They use three different Fourier expansions for the VCO, PLL's input signal and noise, respectively. The authors report phase noise quantities, but neither clear speedup nor the other factors of interest.

Lastly, [19] describes an algorithm for multi-rate circuit simulation based on spline wavelets. They decouple fast and slow dynamics by means of an adaptive time-grid, and apply different time steps to slow and fast dynamics. However, they do not deal with noise analyses, which imposes small time steps with a possible consequence of decreasing the benefit

from the multi-rate approach.

Instead, we develop a single methodology to extract all the PLL's factors of interest. This technique can be used at a verification stage, where transistor-level simulation is considered the most reliable analysis to assess the quality of the circuit. Besides, to model the effect of noise sources at the output of our macromodel, we provide different formulas representing VCO's phase noise shapes that we have identified, which makes our procedure a general one, independent of the oscillator technology.

III. METHODOLOGY

Here we describe our technique, as well as formalize the factors of interest, and finally show the steps to test/implement the methodology.

A. Noise-free Analysis

To speed up time-domain simulations we need to reduce the number of time steps, which are retained according to the fastest-varying dynamic, i.e., the VCO's output. By replacing the VCO and divider by a single phase macromodel (that we call *VCODIV*), we eliminate the VCO's output and keep all the other VCO's and divider's pins, interfaces with the rest of the PLL which is still at transistor level. By doing so, the macromodel is characterized by a phase equation, whereas the other PLL's components are modeled by voltage/current-domain equations (which common simulators deal with). Thus, we need to create a link between the phase equation and the modified nodal analysis form.

An integer- N PLL block model is shown in Fig. 1. The circuit's intrinsic behavior consists in adjusting the VCO's control voltage value $v_c(t)$ such that in the PLL's SST the oscillator produces a periodic output signal with frequency f_{out} . Before the locked state, the PLL has a transient period, in which we say that it is characterized by a "set of (perturbed) SST conditions", each of them with a specific frequency $f_{out,i}$ and $v_c(t) = V_{c,sst,i}$, for some $i = 1, 2, \dots, p$. Stated in another way, the PLL acts as perturbing the actual VCO's SST related to a specific control voltage value. Thus, we can consider the *VCODIV* pins' waveforms as perturbed versions of their corresponding SST solutions.

When we replace the VCO and divider in Fig. 1 with a phase macromodel, we need to specify the contribution of the *VCODIV* to the Kirchhoff's laws at each model's pin. We use equivalent sources at the pins, as depicted in Fig. 2. For instance, $i_{dd}(t + \frac{\Delta\theta(t)}{2\pi f_{out}})$ can be the current at one of the *VCODIV* power supply (related to either the VCO or the divider), and $v_{out}(t + \frac{\Delta\theta(t)}{2\pi f_{out}})$ is the divider's output voltage. In general, $i(t + \frac{\Delta\theta(t)}{2\pi f_{out}})$ (or $v(t + \frac{\Delta\theta(t)}{2\pi f_{out}})$) is the perturbed signal (with $\frac{\Delta\theta(t)}{2\pi f_{out}}$ a phase-shift, in seconds) of its SST solution $i(t) = i(t + T_{out})$, with $T_{out} = 1/f_{out}$ the period at SST. The SST solution is computed through HB or Shooting, extracting the Fourier coefficients and f_{out} . Thus, we reconstruct the equivalent sources through inverse truncated Fourier series

$$\begin{cases} i(t + \frac{\Delta\theta(t)}{2\pi f_{out}}) = \sum_{k=-N_h}^{N_h} I_k e^{\sqrt{-1}k \cdot \theta_{tot}(t)}, \\ \theta_{tot}(t) = \frac{2\pi f_{out}}{N} \cdot t + \Delta\theta(t), \end{cases} \quad (2)$$

Fig. 1. An integer- N PLL.

Fig. 2. *VCODIV* macromodel's equivalent sources, to close the gap between phase- and voltage-domain equations.

with $\Delta\theta(t)$ the phase deviation (in radians) and N_h the number of harmonics.

Alternatively, if Shooting is used to compute the SST, one has a number of solutions within one period of the oscillating waveform. Then, to extract the perturbed version of the SST solution at time $t' + \frac{\Delta\theta(t')}{2\pi f_{out}}$, one can interpolate the solutions $i(t')$ and $i(t'')$, where $(t' + \frac{\Delta\theta(t')}{2\pi f_{out}}) \in [t', t'']$. Next, we show a law for $\Delta\theta(t)$ governing the macromodel.

Our analysis is based on the VCO's nonlinear relation f_{out} vs v_c , i.e., how f_{out} varies with respect to variations of v_c . E.g., for a linear VCO model, it is

$$f_{out}(v_c(t)) = K_{vco} \cdot v_c(t), \quad (3)$$

with K_{vco} the constant slope of the f_{out} vs v_c curve. During a time-domain simulation, $v_c(t)$ might take values from a DC solution $V_{c,DC}$ to the supply V_{dd} . Besides, because the curve is generally nonlinear, we cannot use directly (3) to characterize the phase macromodel. Instead, the global phase law is obtained by combining a set of constant slopes $K_{macro,i}$, each of which is the sensitivity of the VCO's phase/frequency to variation of its control voltage, valid around the corresponding

Fig. 3. Approximating a nonlinear f_{out} vs v_c curve with a PWL combination of intersecting lines, where the i th one has slope $K_{macro,i}$.

$V_{c,sst,i}$, as

$$K_{macro,i} = \left. \frac{\partial f_{out}(v_c(t))}{\partial v_c(t)} \right|_{v_c(t)=V_{c,sst,i}}. \quad (4)$$

The phase-deviation equation, due to perturbation of the VCODIV's control pin, reads as

$$\frac{d}{dt} \Delta\theta_c(t) = \frac{2\pi K_{macro}(V_{c,sst,i})}{N} \cdot \Delta V_c(t), \quad (5)$$

with $\Delta\theta_c(0) = 0$ and

$$\begin{cases} K_{macro}(V_{c,sst,i}) = K_{macro,i} & \text{when } v_c(t) \in [V_{c,l}, V_{c,l+1}), \\ \Delta V_c(t) = v_c(t) - V_{c,sst,i} \end{cases}$$

where $V_{c,l}, V_{c,l+1}$ are the values for which a switch happens between two consecutive SSTs, with $l = 1, 2, \dots, p-1$.

Fig. 3 shows the original, nonlinear curve (*data*) approximated through the PWL curve $K_{macro}(V_{c,sst,i})$, whose i th line $K_{macro,i}$ is computed with (4) around $(f_{out,i}, V_{c,sst,i})$, for which the i th set of Fourier coefficients SST_i (diamond) is extracted and used within the switching points $[V_{c,l}, V_{c,l+1})$ (circles).

By definition, $K_{macro,i}$ in (4) is the oscillator's phase sensitivity with respect to small perturbations acting at the control pin node. Related to that node, $K_{macro,i}$ corresponds to the DC Fourier coefficient (or time average value) of the oscillator's PPV, under a quasi-static assumption. Thus, we have that $PPV_{DC,c,i} = K_{macro,i}$ for the i th point in the f_{out} vs v_c curve. Since the PPV is valid for small perturbations around a periodic solution, one can compute several PPVs in the f_{out} vs v_c curve and use the right DC coefficient between two switching points.

In case multiple SST conditions are needed, $\theta_{tot}(t)$ in (2) is of the form

$$\theta_{tot}(t) = \int_0^t \frac{2\pi f_{out}(\tau)}{N} d\tau + \Delta\theta_c(t),$$

This formulation is valid by ensuring a smooth passage between different SST parameters. This in turn is achieved by

ensuring that the phase in (2) is continuous in correspondence of the switching points in the f_{out} vs v_c curve. In fact, two consecutive SST waveforms are characterized by different oscillating frequencies, say $f_{out,1}, f_{out,2}$, and offset phases θ_1, θ_2 . At a switching point we have to avoid discontinuities due to abrupt changes of frequency and phase values. We smoothen the change of frequencies by integration and align the offset phase of the different SSTs (the phase offset is extracted from the SST analysis and is random, since autonomous oscillators have no phase reference). Formally, at the switching point we have to ensure that

$$\frac{f_{out,1}}{N} \cdot t + \Delta\theta_c(t) + \theta_1 = \frac{f_{out,2}}{N} \cdot t + \Delta\theta_c(t) + \theta_2.$$

It might be that $\theta_1 \neq \theta_2$. So, in correspondence of each SST we remove the actual phase offset θ_i . The final phase $\theta_{tot}(t)$ in (2), for noise-free analysis, is

$$\begin{cases} \theta_{tot}(t) = \frac{2\pi f_{mid}}{N} t + \int_0^t \frac{2\pi f_{diff,i}(\tau)}{N} d\tau + \Delta\theta_c(t) - \theta_i, \\ f_{diff,i}(t) = f_{out}(v_c(t)) - f_{mid}, \end{cases} \quad (6)$$

where f_{mid} can be chosen as the centered frequency in the f_{out} vs v_c curve and $\Delta\theta_c(t)$ obtained by solving (5).

Remark This work is similar to [5]. However, we provide a clearer relationship f_{out} vs v_c to approximate the curve as close as possible to the originally nonlinear one, through a method which chooses the points in the curve where to extract the Fourier coefficients. Clearly, the method is not unique, but it has to minimize the phase errors between transistor-level's and macromodel's simulations, which in turn determines the accuracy of locking time estimation for a large sweep of the reference signal's frequency, as well as power consumption at PLL's SST and during transient. Besides, the PWL approach was presented in [22] together with the use of a phase macromodel for VCO and divider, extracting locking time estimations, but neither power consumption nor noise figures are obtained. The PWL approximation was also used in [23], but with assumptions on the type of phase/frequency detector and loop filter. Our technique is instead independent of the PLL's blocks type under analysis.

Regarding the currents at the VCODIV supply pins from which we extract average power consumptions, we use first-order interpolation of the DC Fourier coefficients, computed at the different SSTs. The i th coefficient estimates the instantaneous average power consumption as $P_{avg} = I_{DC,i} \cdot V_{dd}$ (with $I_{DC,i}$ from the VCO or divider), which is supposed to match the value of the transistor-level's one during the simulation when $v_c(t) = V_{c,sst,i}$. Instead, when $v_c(t) \in (V_{c,sst,i}, V_{c,sst,i+1})$, we approximate the macromodel's supply current as

$$\begin{cases} I_{DC,M}(t) = I_{DC,i} + s_{macro} \cdot (v_c(t) - V_{c,sst,i}), \\ s_{macro} = \frac{I_{DC,i+1} - I_{DC,i}}{V_{c,sst,i+1} - V_{c,sst,i}}, \end{cases} \quad (7)$$

where s_{macro} is the i th line intersecting two consecutive SST points, with the values of the control voltage and DC currents available from HB or Shooting.

However, the interpolation on the supply currents produces an error in the approximation of the VCODIV power consumption

through (7) with respect to the transistor level, especially during the transient period of the PLL. The error is due to (i) the assumption that when $v_c(t) = V_{c,sst,i}$, average supply currents from the transistor level and macromodel simulations during the transient period are same, but this is not entirely true since we extract $I_{DC,i}$ from an SST-based analysis, thus neglecting the “transient nature” characterizing the evolution of the corresponding transistor-level waveforms, and (ii) discrepancies between $v_c(t)$ from the transistor-level and macromodel simulations. The first error source is hard to remove, but we can improve the approximation related to differences in control pin dynamics.

In the following section, we shall provide a way to minimize the error of the control pin evolutions. Since both the macromodel’s output phase (6) and the supply currents (7) are related to $v_c(t)$, the minimization of the error from the control pin dynamics implies reducing both phase and supply current errors.

1) Building the macromodel to minimize phase and power measurement errors: To build the macromodel, an automated procedure needs to select the control pin voltages $V_{c,sst,i}$, for some i , and to perform HB (or Shooting) at those points. We propose a greedy algorithm selecting the points with the goal of reducing the error between transistor-level’s and macromodel’s control pin dynamics. If $v_c(t)$ from the macromodel will be close as much as possible to that from the transistor level, then frequency (or phase) variations will match as well. This means that we need to accurately approximate the original f_{out} vs v_c curve with the PWL one. The PWL approach creates a frequency error with respect to the transistor-level curve between any two consecutive SST points. To minimize this error, we proceed as following: assume we have already computed two SST points, thus we know the sensitivity $K_{macro,i}$ and the pairs $(V_{c,sst,i}, f_{out,i})$ and $(V_{c,sst,i+1}, f_{out,i+1})$, where the error of the macromodel’s approximation is supposed to be minimum. Taking a look at Fig. 3 and drawing a line intersecting two consecutive SST points (diamonds), the point $f_{line} = f_{out,i} + \frac{f_{out,i+1} - f_{out,i}}{V_{c,sst,i+1} - V_{c,sst,i}} \cdot (v^* - V_{c,sst,i})$ can be easily computed. We check whether the condition

$$\begin{aligned} |\Delta f_{line}| &= |f_{line} - f_{macro}| \\ &= \left| \left(\frac{f_{out,i+1} - f_{out,i}}{V_{c,sst,i+1} - V_{c,sst,i}} - K_{macro,i} \right) \cdot (v^* - V_{c,sst,i}) \right| \\ &\leq tol_{\Delta f}, \end{aligned} \quad (8)$$

is satisfied, with $tol_{\Delta f} \in [0, 1]$ an a-priori fixed tolerance and f_{macro} is the macromodel’s frequency at the switching point $v_c(t) = v^*$ (circle). Note that (8) is a condition stricter than the one which checks on $\Delta f = f_{translev} - f_{macro}$, the actual difference between the transistor-level’s and the macromodel’s frequencies at the switching point, since $|\Delta f_{line}| \geq |\Delta f|$, but it just requires the knowledge of available information (the SST pairs and the i th sensitivity) without the need to perform an SST analysis at the switching point in case (8) is satisfied. Besides, we expect to minimize the error of the supply currents due to the interpolation (7), because it is related to the quality of the approximation of the f_{out} vs v_c curve.

If (8) is satisfied, then we consider that the macromodel

produces phase and average supply current evolutions close to the transistor-level’s ones when, during simulation, the VCODIV’s control pin takes values within $[V_{c,sst,i}, V_{c,sst,i+1})$; otherwise an SST analysis will be performed on v^* (becoming an SST point), and other two switching points will be created around it and checked with (8).

B. Device-noise Analysis

When a device-noise simulation is required, equivalent current/voltage noise sources are connected to each circuit’s component. Thus, the right-hand side of (1) is augmented with an additional term representing those sources, and the DAEs become stochastic. Same numerical integration schemes are used as for a noise-free analysis, but the time step has to be reduced, because values of the equivalent noise sources are small and their frequency content (better represented in terms of power spectral density (PSD)) has to be wide enough to include white noise effects [20].

Noise affects the behavior of circuits. It perturbs current and voltage waveforms, causing deviations of their parameters (e.g., amplitude, phase) from their nominal values (in the absence of noise). In PLLs, special attention is given to the effects of noise on the phase of the VCO’s output signal, from which jitter and phase noise are extracted. Amplitude noise is not as much important, especially when oscillators are orbitally stable [13] and there are amplitude-limiters placed after the VCO’s and divider’s output in the PLL. Thus, in building the macromodel we account for phase deviations only and in particular frequency-modulation effects due to noise [21], because the VCODIV model represents an autonomous system.

Next, we describe a novel method to incorporate noise effects into the VCODIV’s output, which are under the form of two time-dependent phase-deviation signals $\Delta\theta_{noise,vco}(t)$ and $\Delta\theta_{noise,div}(t)$, due to (white and flicker) noise generated from the VCO only and divider only, respectively. Then, during a device-noise simulation of the PLL with the VCODIV, we add the phase deviations to $\theta_{tot}(t)$ in (6), to reconstruct the VCODIV’s output which now accounts for variations of the control pin and noise sources, i.e.,

$$\begin{aligned} \theta_{tot}(t) &= \frac{2\pi f_{mid}}{N} t + \int_0^t \frac{2\pi f_{diff,i}(\tau)}{N} d\tau + \Delta\theta_c(t) - \theta_i \\ &\quad + \frac{\Delta\theta_{noise,vco}(t)}{N} + \Delta\theta_{noise,div}(t). \end{aligned} \quad (9)$$

The $\Delta\theta_{noise,vco}(t)$ and $\Delta\theta_{noise,div}(t)$ are obtained once the phase noise’s PSDs of the VCO’s and divider’s outputs are known [24]. The PSDs are extracted when building the macromodel through SST noise analyses.

A common oscillator’s phase noise shape is approximated by

$$S_{\theta,vco}(\Delta f) = \frac{1}{\Delta f^2} \cdot \left(\frac{c_{fl}}{\Delta f} + c_{wh} \right) + c_{floor}, \quad (10)$$

where Δf is the frequency offset from the VCO’s output frequency at PLL’s SST, c_{fl} is the noise power level related to flicker noise, c_{wh} is the level of the white noise and c_{floor}

Fig. 4. An oscillator's phase noise.

is due to remaining noise floor [3], [6]. From an SST noise analysis on the VCO, the simulator computes $S_{\theta,vco}(\Delta f)$ for some Δf in the spectrum of interest; then, by inspecting the shape of $S_{\theta,vco}(\Delta f)$, the noise levels c_{fl} , c_{wh} and c_{floor} are extracted. This can be done for each retained SST point and the power levels are interpolated (as for the supply currents) to provide the correct values corresponding to those characterizing the phase noise of the PLL's output when the circuit is locked. Then, during a device-noise simulation the simulator computes

$$\Delta\theta_{noise,vco}(t) = 2\pi \int_0^t [n_{fl}(c_{fl}, \tau) + n_{wh}(c_{wh}, \tau)] d\tau \quad (11)$$

$$+ n_{floor}(c_{floor}, t),$$

with $n_{cl}(c_{fl}, t)$, $n_{wh}(c_{wh}, t)$ and $n_{floor}(c_{floor}, t)$ functions generating time-domain, noise signals from the knowledge of their power content [24] (available in Verilog-A as well). Note that (11) is the time-domain formulation of (10).

Other phase noise shapes and related formulas can be identified, depending on the topology of the circuit. E.g., in Fig. 4 one can note that the slope of the region starting from $f_c = 80\text{MHz}$ is steeper than -20dB per decade (characterizing white noise region). This effect can be seen as an equivalent low-pass filtering of $S_{\theta,vco}(\Delta f)$ in (10) at its tail, i.e., a filter-like transfer function with 0dB flat spectrum before the corner frequency f_c , and -10dB after that frequency which determines a multiplicity equal to one for the filter's pole. The phase noise shape in Fig. 4 can be formulated (in power) as

$$S_{\theta,vco}(\Delta f) = \left[\frac{1}{\Delta f^2} \cdot (c_{fl} + c_{wh}) + c_{floor} \right] \cdot \frac{1}{1 + (\frac{\Delta f}{f_c})^2}, \quad (12)$$

that in time domain is equivalent to

$$\frac{1}{2\pi f_c} \cdot \frac{d\Delta\theta_{noise,vco}(t)}{dt} + \Delta\theta_{noise,vco}(t) - \Delta\theta_1(t) = 0. \quad (13)$$

In this case, during the simulation of the PLL with the VCODIV, the additional equation (13) is solved for $\Delta\theta_{noise,vco}(t)$ by the simulator, with $\Delta\theta_1(t)$ obtained as in (11).

In conclusion, to compute $\Delta\theta_{noise,vco}(t)$ from $S_{\theta,vco}(\Delta f)$, the simulator inspects the shape of $S_{\theta,vco}(\Delta f)$, fits the coefficients c_{fl} , c_{wh} , c_{floor} and use the functions $n_{fl}(c_{fl}, t)$,

$n_{wh}(c_{wh}, t)$ and $n_{floor}(c_{floor}, t)$. Same strategy is used to compute $\Delta\theta_{noise,div}(t)$ due to device noise in the divider, for which the noise functions used are $n_{fl}(c_{fl}, t)$ and $n_{wh}(c_{wh}, t)$.

In alternative to the presented method to account for device-noise's effects on the VCODIV's output phase, the PPV theory can be exploited.

Remark: A rule of thumb for a device-noise analysis is to set the maximum frequency f_{max} for the noise sources as 10 to 20 times the f_{out} , which translates into a tight constraint for the maximum allowed time step in the simulation. Instead, since we take into account for high-frequency noise folding into the VCO spectrum when we compute $S_{\theta,vco}(\Delta f)$ through the SST noise, we can set $f_{max} = f_{vco}/2$, then release the time step and further gain in simulation time.

C. Factors of interest

In this section we define and show how to extract the PLL's factors of interest after simulations of the transistor-level circuit and that with the VCODIV, as post-processing routines. A relative error to check the accuracy of our methodology is provided as well.

Locking time: Assume we run a noise-free simulation on a integer- N PLL with reference frequency f_{ref1} ; then, when the transient dies out and the PLL reaches its SST, the PLL's output frequency is $f_{out} = N \cdot f_{ref1}$. At time t_1 we sweep the reference frequency to f_{ref2} and the PLL has another transient period. At time t_2 , the PLL reaches again a locked state with $f_{out} = N \cdot f_{ref2}$.

We can define the locking time (in seconds) as

$$t_{lock} = t_2 - t_1. \quad (14)$$

This formulation is valid for both transistor level and macro-model.

Another way to estimate locking time is by changing the divider's ratio N during simulation, as done in [25], where the authors take into account the dynamical properties of the oscillator related to an abrupt change of N . Instead, here we are limited to locking time estimation for a reference frequency sweep, but the dynamic information could be incorporated in our model in case loop filter and other blocks' gain parameters are available.

Average power consumption: Usually, in a PLL simulation the power consumption is estimated at the PLL's SST and for one period T_{ref} of the reference signal.

The average power consumption from the transistor-level's VCO (or divider) is

$$P_{avg,VCO} = \frac{1}{T_{ref}} \int_0^{T_{ref}} i_{dd,VCO}(\tau) \cdot v_{dd,VCO}(\tau) d\tau. \quad (15)$$

When simulating the macromodel, in correspondence of values of $v_c(t)$ near to $V_{c,sst,i}$, measures of $i_{dd,VCO}(t)$ are close to the transistor-level's ones; when far from these regions, the macromodel provides accurate results as more points are retained in the f_{out} vs v_c curve, by using the interpolation formula (7).

Phase noise: From the transistor-level simulation, the phase noise is not readily available and must be extracted with post-processing techniques [26]. One of these is described next.

Assume that the VCO's output period T_{out} at SST is known, and define the i th cycle of the VCO's waveform as the time period $T(t_i) = t_{i+1} - t_i$ between two zero-crossings of the waveform at t_i and t_{i+1} , with either positive or negative slope. Then:

- 1) compute the uncertainty in the zero-crossing time as $j(t_i) = T(t_i) - T_{out}$, $\forall i \in \{1, \dots, n_c - 1\}$, with $n_c = T_{sim,noise}/T_{out}$ the number of VCO's output cycles within a device-noise simulation of time $T_{sim,noise}$, with, e.g., the method described in [27];
- 2) compute the time-domain phase noise as $\Delta\theta_{PLLout}(t_i) = 2\pi j(t_i)/T_{out}$;
- 3) compute the PSD (Welch's periodogram) of $\Delta\theta_{PLLout}(t)$, obtaining $S_{\theta,PLL}(\Delta f)$.

The aforementioned process might bring additional numerical inaccuracies to the result and become expensive with increasing $T_{sim,noise}$.

Instead, from simulating the macromodel, we directly compute the time-domain phase noise as

$$\Delta\theta_{PLLout}(t) = N \cdot \Delta\theta_c(t) + 2\pi(f_{out,i} - f_{out}) \cdot t \quad (16)$$

$$+ \Delta\theta_{noise,vco}(t),$$

where $2\pi(f_{out,i} - f_{out}) \cdot t$ removes the phase offset from $N \cdot \Delta\theta_c(t)$ due to a (possible) difference between the actual SST frequency $f_{out,i}$ retained during the building phase of the macromodel and the f_{out} corresponding to the locked PLL. The reason for this term is briefly explained in the following. From simulating the transistor level, we expect that at the PLL's steady state, with frequency f_{out} , the VCO's output phase noise $\Delta\theta_{c,TL}(t)$ due to perturbations at the control pin is slightly oscillating around V_{sst} with sensitivity K_{sst} . However, with our macromodel $\Delta\theta_c(t)$ at SST is related to the triplet $(V_{c,sst,i}, K_{macro,i}, f_{out,i})$. By assuming that $K_{macro,i} \approx K_{sst}$, it might be that $V_{c,sst,i} \neq V_{sst}$ and $f_{out,i} \neq f_{out}$, such that $\Delta\theta_c(t)$ is oscillating around a ramp with slope proportional to $(V_{c,sst,i} - V_{sst})$. Thus

$$\begin{cases} K_{macro,i} \approx K_{sst}, \\ \frac{d\Delta\theta_{c,TL}(t)}{dt} = 2\pi(f(t) - f_{out}) = K_{sst} \cdot (v_c(t) - V_{sst}), \\ \frac{d\Delta\theta_c(t)}{dt} = 2\pi(f(t) - f_{out,i}) = K_{sst} \cdot (v_c(t) - V_{c,sst,i}). \end{cases}$$

Since we want $d\Delta\theta_c(t)/dt \approx d\Delta\theta_{c,TL}(t)/dt$, this implies

$$V_{c,sst,i} = V_{sst} + \frac{2\pi(f_{out,i} - f_{out})}{K_{sst}}. \quad (17)$$

By plugging (17) into $d\Delta\theta_c(t)/dt$, we have that

$$\frac{d\Delta\theta_c(t)}{dt} = K_{sst} \cdot (v_c(t) - V_{sst}) - 2\pi(f_{out,i} - f_{out}),$$

which, by adding the term $2\pi(f_{out,i} - f_{out})$, approximates $\Delta\theta_{c,TL}(t) \forall t$, so that (16) holds.

It is worth mentioning that we avoid the first two steps used for the transistor-level phase noise computation and reduce possible numerical errors. The only drawback is that, since during simulation we use a larger time-step length, we cannot see the tail of $S_{\theta,PLL}(\Delta f)$ (the maximum readable frequency is related to the minimum time-step size [28]). However, we have a remedy for that, that we present in what follows.

Period jitter: It is the variation of the VCO's perturbed output period from the nominal T_{out} , characterized by the sequence $\{j(t_i) = T(t_i) - T_{out}\}_i$ [6], with $T(t_i)$ same as in the phase noise factor's definition. Jitter is a stochastic process, so it is best expressed in terms of RMS value as

$$j_{per,RMS} = \sqrt{\frac{1}{n_c - 1} \sum_{i=1}^{n_c-1} j(t_i)^2}. \quad (18)$$

When simulating the PLL with the VCODIV, we eliminate the VCO's output and compute solutions with a larger time-step size, speeding up the analysis. Nevertheless, we can easily reconstruct the VCO's waveform as a fast post-processing procedure and extract the period jitter.

In fact, when building the macromodel the SST analysis allows to extract the Fourier coefficients of the VCO's output, which, at the end of a device-noise analysis, can be reconstructed in the time domain with a finer time grid and by using linear interpolation to extract the values of $\Delta\theta_c(t)$ and $\Delta\theta_{noise,vco}(t)$ between the time points retained during the simulation:

$$v_{vco,out}(t) = \sum_{k=-N_h}^{N_h} I_k e^{\sqrt{-1}k(2\pi f_{out} \cdot t + N \cdot \Delta\theta_c(t) + \Delta\theta_{noise,vco}(t))}, \quad (19)$$

with f_{out} the PLL's locking frequency. Eq. (19) can be quickly evaluated as a post-processing routine, since it only requires interpolation of the phase-deviation solutions retained during the simulation, in the new time grid, and it is highly parallelizable. Besides, from (19) one can compute the VCO's phase noise spectrum, obtaining its tail.

Long-term jitter: It can be described as the variation in time of the cumulative period of adjacent n_c -cycle samples. In other words, it is the set of n_c -period jitter RMS values $\{j_{lt,RMS}(k)\}_k$, expressed by

$$\begin{cases} j_{lt,RMS}(k) = \sqrt{\frac{1}{n_c-1} \sum_{i=1}^{n_c-1} j_k(t_i)^2}, \\ j_k(t_i) = t_{i+k} - t_i - k \cdot T_{out}. \end{cases} \quad (20)$$

Notice that, when $k = 1$ then $j_{lt,RMS}(1) = j_{per,RMS}$ [20]. For transistor-level's simulation, long-term jitter is extracted from the VCO's output; same can be done from the analysis with VCODIV, since we interpolate the VCO's output with (19).

Lastly, to compare the accuracy of a factor of interest's set of values $\{x(t_i)\}_i$ between the transistor-level (TL) and the macromodel (M) simulations, we define the relative error as

$$\text{Error} = \frac{|x_{TL}(t_i) - x_M(t_i)|}{|x_{TL}(t_i)|}. \quad (21)$$

D. Guidelines

Here, we list the steps of our technique. The methodology is a prototype: we use the Eldo® circuit simulator (Mentor Graphics) and Matlab® to build the macromodel, then only Eldo for the time-domain simulations. However, the final target is to have a fully automated method, hence we highlight the critical points to accomplish this.

Assume that $N_{h,vco}$ and $N_{h,div}$ are the number of harmonics used by HB performed on VCO and divider, respectively, whose blocks are clearly identifiable from the netlist.

- 1) Choose $N_{h,vco}$ and $N_{h,div}$ based on the expected waveforms in the VCO and divider. E.g., if square wave-like or triangular signals are involved in the blocks, then one should choose a high number of harmonics, say $N_{h,vco} \geq 50$; accordingly, $N_{h,div} = N \cdot N_{h,vco}$;
- 2) let the range for $v_c(t)$ be the PLL's locking range $[V_{c,lockMin}, V_{c,lockMax}]$; alternatively, choose $V_{c,lockMin} = V_{c,DC}$. In this case, $V_{c,DC}$ can be estimated through a DC analysis on the transistor-level PLL;
- 3) if the VCO is provided with additional bias pins other than the control one for $v_c(t)$, then a DC analysis on the transistor-level PLL can be performed to extract the basing values at those pins and keep them when computing the SSTs. Same is done for the loop divider. Besides, compute the equivalent capacitance C_{eq} in Fig. 2 through an AC analysis on the VCO's control pin;
- 4) from some experiments on different VCOs, a good heuristic to start with in approximating the nonlinear f_{out} vs v_c curve (to satisfy $tol_{\Delta f}$) is to compute HB in correspondence of $[V_{c,lockMin}, V_{c,mid1}, V_{c,mid2}, V_{c,lockMax}]$, where $V_{c,mid1} = V_{c,lockMin} + 1/3(V_{c,lockMax} - V_{c,lockMin})$ and $V_{c,mid2} = V_{c,lockMin} + 2/3(V_{c,lockMax} - V_{c,lockMin})$. Thus, build the model as explained in Sec. III-A, specifying $tol_{\Delta f}$ to check (8) at each switching point, and save the Fourier coefficients of the waveforms to be synthesized through (2);
- 5) compute the sensitivity in (4) with the PPV theory or by finite difference, as explained in Sec. III-A;
- 6) inspect the phase noise spectra of the VCO and divider to extract the parameters to be used in the macromodel (to evaluate the additional phase-deviation terms due to device noise) as explained in Sec. III-B, or use the PPV theory;
- 7) the macromodel's equations to use during the time-domain simulation are: Eq. (5), the phase law; Eq. (2), and (6) for noise-free or (9) for device-noise analysis, to reconstruct the VCODIV's output; Eq. (7) to interpolate the DC coefficients of the supply currents; Eq. (16) as an additional VCODIV's pin, to extract the phase noise.

For the experiments presented in Section IV, we used Eldo in steps 1, 2, 3; the approximation of the original nonlinear f_{out} vs v_c curve with the PWL approach was implemented in Matlab, basically extracting the switching points where (8) has to be checked (step 4), while using the sensitivities (computed with Eldo through the HB method); instead, the Fourier coefficients were saved in text files, properly stored by Eldo and used during simulation. After computing the phase noise shapes of the VCO and divider with Eldo, we obtain the corresponding power noise levels by knowing the values of $S_{\theta,vco}(\Delta f)$ (step 6); finally, we run simulations of the PLL with the macromodel with Eldo.

In order to automate the methodology, some efforts are needed under a computer science viewpoint when parsing the netlist,

to identify the VCO and divider blocks and perform the SST analyses. Instead, the routine to approximate the f_{out} vs v_c curve with the PWL approach and the extraction of the noise levels can be easily implemented in any available circuit simulator.

IV. RESULTS

We tested our method with the Eldo simulator and compare results against transistor-level simulations. Experiments were ran with an Intel® Xeon® CPU, 8 processors at 3GHz machine, on two industrial PLL frequency synthesizers with division ratio $N = 25$ and $N = 50$, respectively.

To validate the device-noise analysis so that noise from the circuit's components can be clearly discriminated from the numerical one, the simulator adaptively adjusts simulation tolerances (Newton, maximum time step). Tests can be done by looking at, e.g., the VCO's/divider's output frequency evolutions: after noise is injected at a specific simulation time $tstart_noise$, the frequency's dynamic has a peak-to-peak intensity higher than the one characterizing the simulation before $tstart_noise$.

A. $N = 25$

The first PLL has number of equations and active devices equal to 1271 and 1009, respectively. The simulation ending time is set to 20μ seconds (s) and to $1ms$ for the noise-free and the device-noise simulations, respectively. The reference signal is originally $f_{ref1} = 100MHz$ and swept to $f_{ref2} = 115MHz$ during the noise-free simulation to estimate the locking time.

To test the accuracy of the power consumptions, we choose $tol_{\Delta f} = 0.03$ (3%) and compare the VCO's and divider's supply currents between the transistor level and macromodel within one period of the reference frequency, at some points. The algorithm retains 6 points in the f_{out} vs v_c curve, that we plot in Fig. 5. The most time consuming routine was performing HB on the divider at 6 control voltage values, each of which requires 17s, together with an SST noise which takes 20s.

In Table I we show the error values of the VCO's and divider's supply currents, estimated with (21) in correspondence of some steady-state (SST), switching (SW) and SST-SW middle (MID) control voltage values. The power is then extracted with (15) with constant v_{dd} . The best approximation is in correspondence of the SST point $v_c(t) = 0.323V$, and the discrepancy between the DC Fourier coefficients extracted from the SST analysis (for the macromodel) and the average value of the currents with a transient nature (for the transistor level) is remarkable at the SST point $v_c(t) = 0.3931V$; at the switching points, the linear interpolation routine works quite well; the worst case is at $v_c(t) = 0.3738V$ (MID). The interpolation process is more effective as more points are retained in the f_{out} vs v_c curve.

In Fig. 6 and Fig. 7, we show dynamics from a noise-free simulation of the VCO control voltage (V_c) and locking signals (Lk), respectively, for transistor level (TL) and macromodel (M). The waveforms are almost indistinguishable. Here, the PLL reaches the first locking condition at $4.6\mu s$; at around

Fig. 5. Integer-25 PLL: Approximating the original nonlinear f_{out} vs v_c (red) with a PWL curve (black).

TABLE I
INTEGER-25 PLL: ACCURACY OF SUPPLY CURRENTS ($tol_{\Delta f} = 3\%$).

$v_c(t)$ [V]	err_{VCO}	err_{DIV}
0.3230 (SST)	0.1%	1.0%
0.3931 (SST)	1.7%	5.6%
0.2836 (SW)	0.4%	1.8%
0.4284 (SW)	2.0%	3.5%
0.4108 (MID)	0.3%	1.0%
0.3783 (MID)	3.0%	8.0%

Fig. 6. Integer-25 PLL: control voltage (V_c) of transistor level (black) and macromodel (magenta).

$t_1 = 6.7\mu s$, we sweep to $f_{ref2} = 115\text{MHz}$, then the PLL loses the lock and reaches another locked state at around $t_2 = 8.6\mu s$. Locking time is extracted as in (14).

In Fig. 8 we show the PLL's phase noise from device-noise time-domain simulations on the transistor level, obtained through the steps listed in Sec. III-C, and on the macromodel

Fig. 7. Integer-25 PLL: locking signal (Lk) of transistor level (black) and macromodel (magenta).

by computing the PSD on (16). Simulation results match well from 200KHz to 200MHz (a maximum error of 4dB in this band). Because of the chosen simulation time interval, to obtain accurate near-carrier results one has to increase that interval. We cannot see the tail on the macromodel's $S_{\theta, PLL}(\Delta f)$ because we compute solutions with a larger time step.

To have another reference for comparison, we also provide the phase noise from an SST noise analysis on the PLL at transistor level ($SSTN$). Fig. 9 plots the phase noise from the SST noise and from the PSD on the macromodel's $\Delta\theta_{PLLout}(t)$ in (16), interpolated as a post-processing routine, to appreciate the phase noise's tail. We can see that results match very well up to 500MHz; instead, the phase noise's tail obtained from the time-domain simulation on the transistor level seems to be more accurate than the macromodel, compared with the SST noise (Fig. 8). In Fig. 9 one can see the near-carrier phase noise due to flicker noise sources in the PLL and phase-frequency detector/charge pump blocks, and the “bump” in the PLL's bandwidth due to noise in the loop filter's resistor. However, we did not simulate noise from the PLL's reference signal, but this can be easily done by the simulator, for any noise analysis. PLL's noise levels depend on the technology under analysis (e.g., see [29]).

However, it is hard to define which result represents the “truth”, and the only way is to have measurements from the silicon. In fact, designers use a bunch of methods, together with experience, at different level of the design to test PLLs before the manufacturing process. Despite of the small differences shown in Fig. 8 and Fig. 9, it's worth noticing that all the outcomes are very close to each other.

Fig. 10 depicts the long-term jitter measurements. Results are very accurate for many adjacent VCO's cycles: we have an error of 8.5% on the period jitter (607fs for the transistor level vs 665fs for the macromodel), an error peak of 11% at $k = 2$ for the long-term jitter in (20) and for all other ks it stays around 5%.

Fig. 8. Integer-25 PLL: phase noise (dB) from device-noise analysis on the transistor level (green), macromodel (magenta) and SST noise on transistor level (blue).

Fig. 9. Integer-25 PLL: phase noise (dB) from SST noise on transistor level (blue) and PSD on interpolated macromodel's $\Delta\theta_{PLLout}(t)$ (magenta).

In Table II, we report results of the macromodel's and the transistor-level's measurements for the locking time t_{lock} (s), total PLL's power consumption P_{tot} (Watt), $S_{\theta, PLL}(\Delta f)$ (dB) with measures taken for some frequency decades, (RMS) period jitter and (RMS) long-term jitter for some VCO's cycles, as well as their relative errors with respect to transistor-level's data. The total power consumption is evaluated at SST for $T_{ref} = 0.01\mu s$ and is mainly due to: the charge pump block ($147,09\mu W$ (TL) against $147,05\mu W$ (M)), VCO ($5.874mW$ (TL) against $5.946mW$ (M)) and divider ($2.349mW$ (TL) against $2.415mW$ (M)). VCO and divider errors are small even because the average value of the supply currents from the transistor-level simulation are measured when the PLL is

Fig. 10. Integer-25 PLL: long-term jitter results for VCO's output from transistor level (green) and macromodel (magenta).

TABLE II
INTEGER-25 PLL: ACCURACY OF THE FACTORS OF INTEREST.

	t_{lock}	P_{tot}	$S_{\theta, PLL}$	j_{per}	$j_{lt}(k)$
TL	1.9742μ	$8.370m$	¹	$607f$	³
M	1.9743μ	$8.508m$	²	$665f$	⁴
Error	$8 \cdot 10^{-6}\%$	1.6%	$\leq 4dB$	9.5%	$< 11\%$

¹ $[-89, -100.8, -114]$ at $[5, 30, 90]$ MHz from f_{out} .

² $[-85, -100.7, -112]$ at $[5, 30, 90]$ MHz from f_{out} .

³ $[0.607, 2.50, 13.36]p$ at VCO's cycle $k = [1, 5, 50]$.

⁴ $[0.665, 2.73, 13.06]p$ at VCO's cycle $k = [1, 5, 50]$.

TABLE III
INTEGER-25 PLL: SIMULATION TIME RESULTS.

TRAN	# retained steps	t_{build} (min)	t_{sim} (min)	speedup
TL	$3.7 \cdot 10^6$	-	131	-
M	$4.0 \cdot 10^5$	2	13	8.7x
NOISE	# retained steps	t_{build} (min)	t_{sim} (hour)	speedup
TL	$2.6 \cdot 10^8$	-	338.0	-
M	$2.1 \cdot 10^7$	-	20.6	16.4x

locked, so that the extracted DC Fourier coefficients used by the macromodel match them. Table III shows the retained time steps and the simulation times for the noise-free and device-noise simulations, with speedup factors. These include the time t_{build} spent to build the macromodel with 6 points (for the noise-free analysis only, since it is a one-time cost). We can say that, for device-noise simulations the speedup factor tends to the PLL's division ratio N and is proportional to the reduction of computed time steps.

B. $N = 50$

This PLL has number of equations and active devices equal to 3936 and 1599, respectively. The simulation ending time is set to $170\mu s$ and to $0.5ms$ for the noise-free and device-noise simulations, respectively. The reference signal is originally

Fig. 11. Integer-50 PLL: Approximating the original nonlinear f_{out} vs v_c (red) with a PWL curve (black).

$f_{ref1} = 10\text{MHz}$ and swept to $f_{ref2} = 14\text{MHz}$ during the noise-free simulation to estimate the locking time.

To build the macromodel, we set $tol_{\Delta f} = 0.05$ (5%), obtaining 10 SST points to approximate the f_{out} vs v_c curve, that we show in Fig. 11. In Table IV we show the error evaluated at some control voltage values. At $v_c(t) = 0.721\text{V}$, where the transistor-level's and macromodel's control voltage dynamics match, we have the best accuracy during the PLL's transient period; at $v_c(t) = 0.5267\text{V}$ the approximation of the divider's supply current is worst because the control pin evolutions are not overlapped, as well as the transient nature of the transistor-level's currents is not properly represented by the SST nature of the macromodel; instead, at $v_c(t) = 1.223\text{V}$ the PLLs are close to the second SST (after the reference frequency sweep) and control pin dynamics match, so that errors are small. Again, the accuracy can be improved by retaining a higher number of SST points in the macromodel, e.g., by tightening $tol_{\Delta f}$.

The time spent to build the macromodel is approximatively equal to 32 minutes (min), to perform 10 times HB on the divider with a higher number of harmonics with respect to the previous experiment, and the SST noise on it to extract the noise power level (8 min). In fact, when the divider has a high ratio N , the CPU time to build the model might have an impact on the time performance of the noise-free, macromodel's simulation. However, since usually the device-noise analysis is much more expensive, it dominates the time spent in forming the model.

Fig. 12 and Fig. 13 show the transistor-level's (TL) and macromodel's (M) control voltage and locking signal, respectively, which match with a good accuracy.

In Fig. 14 we plot the phase noise shapes obtained with the steps in Sec. III-C on the transistor level and on the interpolated $\Delta\theta_{PLLout}(t)$ by computing the PSD on (16). Because of possible inaccuracies due to the first two post-processing steps in obtaining the transistor-level's phase noise, we also show the phase noise from an SST noise on the transistor-level

TABLE IV
INTEGER-50 PLL: ACCURACY OF SUPPLY CURRENTS ($tol_{\Delta f} = 5\%$).

$v_c(t)[\text{V}]$	err_{VCO}	err_{DIV}
0.7210 (SST)	3.0%	4.0%
0.5267 (SST)	4.6%	16.0%
0.8611 (SW)	2.7%	10.0%
0.7911 (MID)	2.5%	5.4%
1.2230 (MID)	1.3%	4.2%

Fig. 12. Integer-50 PLL: control voltage (V_c) of transistor level (black) and macromodel (magenta).

Fig. 13. Integer-50 PLL: locking signal (L_k) of transistor level (black) and macromodel (magenta).

PLL (SSTN): we can note this last curve lies in between the first two, thus statistically reducing the discrepancy between the results obtained with the macromodel and transistor-level time-domain simulations, which stays below 7dB. The SST noise analysis performed to extract the phase noise spectrum is local truncation error-free, and it gives a reliable estimation

Fig. 14. Integer-50 PLL: phase noise (dB) from device-noise analysis on transistor level (green), PSD on macromodel's $\Delta\theta_{PLLout}(t)$ (magenta) and SST noise on transistor level (blue).

Fig. 15. Integer-50 PLL: long-term jitter results for VCO's output from transistor level (green) and macromodel (magenta).

of the PLL's phase noise.

Lastly, in Fig. 15 we depict the long-term jitter. The results show that the curves differ for some k in (20), with a peak error of 30%. However, because of the considerations done for the phase noise plots and since the PLL's long-term jitter is directly related to its phase noise [20], differences in jitter values should be smaller as well.

In Table V, we report results on the accuracy of the macromodel's and transistor-level's measurements. The total power consumption is evaluated at SST for $T_{ref} = 0.1\mu s$ and is mainly due to: the charge pump block ($119.67\mu W$ (TL) against $101.77\mu W$ (M)), bias block ($10.35\mu W$ (TL) against $10.35\mu W$ (M)), VCO ($2.997mW$ (TL) against $2.996mW$ (M)) and divider ($200.79\mu W$ (TL) against $209.75\mu W$ (M)). Table VI shows simulation times and speedup factors.

TABLE V
INTEGER-50 PLL: ACCURACY OF THE FACTORS OF INTEREST.

	t_{lock}	P_{tot}	$S_{\theta, PLL}$	j_{per}	$j_{lt}(k)$
TL	25.06713μ	$3.328m$	¹	$1.29p$	³
M	25.06690μ	$3.318m$	²	$1.10p$	⁴
Error	$9 \cdot 10^{-7}\%$	0.3%	$\leq 7dB$	15%	$< 30\%$

¹ $[-97, -121, -130.3]$ at $[2, 15, 50]$ MHz from f_{out} .

² $[-93, -116, -130.4]$ at $[2, 15, 50]$ MHz from f_{out} .

³ $[1.29, 4.20, 7.84]p$ at VCO's cycle $k = [1, 10, 25]$.

⁴ $[1.10, 4.95, 9.77]p$ at VCO's cycle $k = [1, 10, 25]$.

TABLE VI
INTEGER-50 PLL: SIMULATION TIME RESULTS.

TRAN	# retained steps	t_{build} (min)	t_{sim} (min)	speedup
TL	$9.0 \cdot 10^6$	-	561	-
M	$6.3 \cdot 10^5$	40	38	7.2x
NOISE	# retained steps	t_{build} (min)	t_{sim} (hour)	speedup
TL	$2.90 \cdot 10^7$	-	65.0	-
M	$1.78 \cdot 10^6$	-	2.5	26x

The speedups we obtain with our macromodel are smaller than those appearing in the literature. For instance, in [19] they obtain a speedup factor of almost 300x for noise-free analysis, at a verification level. In [30] the authors obtain the phase noise at the PLL's VCO output with an accurate macromodel and a speedup of 600x. This is suitable for verification level assuming the PLL is in SST. Same is done in [1], where a very good phase noise approximation is extracted from an approach using macromodels, resulting in a speedup proportional to the divider's ratio. However, we stress the fact that with our approach we can estimate metrics of a PLL from both its transient and SST behavior.

V. CONCLUSIONS

We have presented a method to speedup time-domain simulations of industrial integer- N PLLs, while accurately extracting locking time, average power consumptions, phase noise and jitter, whereas previous works in the literature are not able to extract all of them. The speedup factors are satisfying, especially when the divider's ratio is high and for PLLs working at GHz frequencies.

In the second experiment, we have experienced some inaccuracies in the estimation of the jitter obtained with our method, compared to results obtained from post-processing the transistor-level PLL's output waveform. This procedure is not error-free, so it is hard to define a clear reference for comparison and the only way is to have measurements from the silicon. However, we have shown phase noise estimations from different methods and that they are close to each other, thus reducing the discrepancies, like those in the long-term jitter.

The piecewise linear approach could be adopted for other multi-rate circuits, e.g., Switching-Mode Power Supplies. A macromodel for these systems could be created, with a law characterized by the sensitivity of the output model (e.g., duty cycle of the signal driving the switch in a buck converter),

with respect to variations at the input (e.g., circuit's output voltage), while removing the fast-varying dynamics.

ACKNOWLEDGMENT

The authors are very grateful for the reviewers' comments, which helped improve this work. The authors want to thank some members of the Mentor Graphics in Grenoble for valuable discussions about several aspects on circuit simulation, especially the RF world. The first author is also grateful for the financial support from the Marie Curie Action.

REFERENCES

- [1] B. Wang and E. Ngoya, *Integer-N PLLs verification methodology: large signal steady state and noise analysis*. in IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, n. 11, pp. 2738-2748, 2012.
- [2] L. Scheffer, L. Lavagno and G. Martin (editors), *EDA for IC Implementation, Circuit Design, and Process Technology*. Taylor & Francis, 2006.
- [3] K. Kundert, *Predicting the phase noise and jitter of PLL-based frequency synthesizers*. www.designers-guide.com. 2003.
- [4] X. Mao, H. Yang and H. Wang, *Behavioral modeling and simulation of jitter and phase noise in fractional-N PLL frequency synthesizer*. in Proc. IEEE on Behavioral Modeling And Simulation (BMAS) Conf., pp. 25-30, 2004.
- [5] B. Wang, *Modeling and simulation techniques for the accurate verification of Integer-N PLLs*. Ph.D. Thesis, University of Limoges, 2009.
- [6] D. C. Lee, *Analysis of jitter in phase-locked loops*. in IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 49, n. 11, pp. 704-711, 2002.
- [7] M. Biggio, F. Bizzarri, A. Brambilla and M. Storace, *Efficient transient noise analysis of non-periodic mixed analogue/digital circuits*. in IET Circuits, Devices and Syst., vol. 9, n. 2, pp. 73-80, 2015.
- [8] V. R. Gonzalez-Diaz, J. M. Munoz-Pacheco, G. Espinosa-Flores-Verdad, L. A. Sanchez-Gaspariano, *A Verilog-A Based Fractional Frequency Synthesizer Model for Fast and Accurate Noise Assessment*. in Radioengineering, vol. 25, n. 1, pp. 89-97, 2016.
- [9] K. S. Kundert and A. Sangiovanni-Vincentelli, *Finding the steady-state response of analog and microwave circuits*. in Proc. IEEE on Custom Integrated Circuits Conf. (CICC), pp. 6.1/1-6.1/7, 1988.
- [10] R. Telichevesky, K. S. Kundert and J. K. White, *Efficient Steady-State Analysis Based on Matrix-Free Krylov-Subspace Methods*. in 32nd Design Automation Conf. (DAC), pp. 480-484, 1995.
- [11] P. Bolcato, J. C. Nallatamby, R. Larcheveque, M. Prigent and J. Obregon, *A unified approach of PM noise calculation in large RF multitone autonomous circuits*. in IEEE MTT-S Int. Microw. Symp. Digest., vol. 1, pp. 417-420, 2000.
- [12] E. Ngoya, J. Roussel and D. Argollo, *Rigorous RF and microwave oscillator phase noise calculation by envelope transient technique*. in IEEE MTT-S Int. Microw. Symp. Digest., vol. 1, pp. 91-94, 2000.
- [13] A. Demir, A. Mehrotra and J. Roychowdhury, *Phase noise in oscillators: a unifying theory and numerical methods for characterization*. in IEEE Trans. Circuits Syst. I, Fundam. Theory App., vol. 47, pp. 655-674, 2000.
- [14] A. Demir and J. Roychowdhury, *A reliable and efficient procedure for oscillator PPV computation, with phase noise macromodeling applications*. in IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 22, pp. 188-197, 2003.
- [15] F. N. Najm, *Circuit Simulation*. Wiley-IEEE Press, 2010.
- [16] P. Maffezzoni and S. Levantino, *Analysis of VCO phase noise in charge-pump phase-locked loops*. in IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, n. 10, pp. 2165-2175, 2012.
- [17] X. Lai and J. Roychowdhury, *TP-PPV: piecewise nonlinear, time-shifted oscillator macromodel extraction for fast, accurate PLL simulation*. in Int. Conf. Comput.-Aided Design (ICCAD), Hassoun, Soha, 2006.
- [18] S. Sancho, A. Suarez and J. Chuan, *General envelope-transient formulation of phase-locked loops using three time scales*. in IEEE Trans. Microw. Theory Techn., vol. 52, n. 4, pp. 1310-1320, 2004.
- [19] K. Bittner and H. G. Brachtendorf, *Adaptive multi-rate wavelet method for circuit simulation*. in Radioengineering, vol. 23, pp. 300-307, 2014.
- [20] Mentor Graphics Corp., *Eldo® User Manual*. 2015.
- [21] K. S. Kundert, *Modeling and simulation of jitter in phase-locked loops*. in Analog Circuit Design, pp. 359-379, Springer US, Boston, 1997.
- [22] I. Harasymiv, M. Dietrich and U. Knochel, *Fast mixed-mode PLL simulation using behavioral baseband models of voltage-controlled oscillators and frequency dividers*. in 11th Int. Workshop on Symbolic and Numerical Methods, Modeling and Appl. to Circuit Design (SM2ACD), pp. 1-6, 2010.
- [23] C. Wiegand, C. Hangmann, C. Hedayat and U. Hilleringmann, *Modeling and simulation of arbitrary ordered nonlinear charge-pump phase-locked loops*. in Semiconductor Conf. Dresden (SCD), pp. 1-4, 2011.
- [24] P. Bolcato, M. S. Tawfik, R. Poujois and P. Jarron, *A new efficient transient noise analysis technique for simulation of CCD image sensors or particle detectors*. in Proc. IEEE Custom Integrated Circuits Conf. (CICC), pp. 14.8.1-14.8.4, 1993.
- [25] B. De Smedt and G. Gielen, *Models for systematic design and verification of frequency synthesizers*. in IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 46, n. 10, pp. 1301-1308, 1999.
- [26] A. L. Lance, W. D. Seal, and F. Labaar, *Phase Noise and AM Noise Measurements in the Frequency Domain*. in Infrared Millim. Waves, vol. 11, pp. 239289, 1984.
- [27] J. McNeill, *Jitter in ring oscillators*. in Proc. IEEE Int. Symp. on Circuits Syst. (ISCAS), vol. 6, pp. 201-204, 1994.
- [28] P. B. Osgood, *Lecture Notes for EE 261 the Fourier Transform and Its Applications*. CreateSpace Independent Publishing Platform, 2014.
- [29] A. Mehrotra, *Noise analysis of phase-locked loops*. in IEEE Trans. Circuits Syst. I, Fundam. Theory App., vol. 49, n. 9, pp. 1309-1316, 2002.
- [30] P. Maffezzoni, S. Levantino, C. Samori, A. L. Lacaita, D. D'Amore and M. Santomauro, *Behavioral phase-noise analysis of charge-pump phase-locked loops*. in Proc. 20th Eur. Conf. Circuit Theory and Design (ECCTD), pp. 357-360, 2011.

Giovanni De Luca received the MSc degree in Computer Science and Automatic Control Engineering (2008) in L'Aquila (IT), collaborating with the Dept. of Geniuses of Automated Production at Ecole de Technologie Supérieure in Montreal (CA). In 2012, he was a visiting researcher at the Max Planck Institute in Magdeburg (GE), with the goal of speeding up the analysis of transmission line models with Model Order Reduction and parallelization techniques. Since 2014, he is a PhD candidate at the TU Eindhoven, to speed up time-domain simulations of PLLs, Switching Power Supplies and Delta-Sigma Modulators (ASIVA14 project: www.itn-asiva14.eu) with partner the Mentor Graphics in Grenoble (FR). His fields of interest include circuit simulation, model reduction techniques and optimization of algorithms.

Pascal Bolcato received the Msc degree in Electrical Engineering from the Grenoble University (France) in 1990, and the PhD degree from the INP Grenoble (France) in 1994. He is Engineering Director at Mentor Graphics in charge of Analog Simulation (SPICE, RF and Fast-SPICE).

Remi Larcheveque holds a PhD degree (1997) in Electrical Engineering from University of Limoges (FR). Since 1996, he has been working with the Mentor Graphics in Grenoble. His main expertise is RF simulation and validation, with emphasis in steady-state analyses and noise characterization.

Joost Rommes has been with Mentor Graphics in Grenoble since May 2014, as technical leader for large-scale circuit reduction and simulation. He holds a PhD in Computational Science (2007) and worked for NXP Semiconductors (2007 – 2014) on linear and nonlinear circuit reduction, large linear system solvers, EDA tools for design and optimization of parameterized circuits, and program management for new process technology introduction. He supervised and supervises several MSc and PhD students.

Wil H. A. Schilders received the MSc degree in mathematics from Radboud University in Nijmegen (NL) in 1978, and the PhD degree from Trinity College Dublin (IE) in 1980. Since then, he has been working in the electronics industry, first with Philips Research Laboratories in Eindhoven, and since 2006 with NXP. Here, he developed algorithms for simulating semiconductor devices, electronic circuits, organic light emitting diodes, and electromagnetic problems (TV tubes, interconnects, magnetic resonance imaging). Since 1999, he has been a part-time

Professor of numerical mathematics for industry with TU Eindhoven. He is the Managing Director of the Platform for Mathematics in The Netherlands. He is currently the President of the European Consortium for Mathematics in Industry.