RF Library based on Block Diagram and Behavioral descriptions

Benjamin Nicolle
Mentor Graphics - LEAT
250, rue Albert Einstein
06560 Valbonne - France
+33(0)492942870
bnicolle@unice.fr

William Tatinian, Gilles Jacquemod
LEAT – CNRS UNSA UMR 6071
250, rue Albert Einstein
06560 Valbonne - France
+33(0)492942851
tatinian@unice.fr

Jean-José Mayol, Jean Oudinot
Mentor Graphics
13/15, rue Jeanne Braconnier
92360 Meudon-la-forêt, France
+33(0)140947455
jean_oudinot@mentor.com

ABSTRACT

This paper presents a RF behavioral library. The objective of this library is to validate the future RF transceiver system, for example for BER specifications. The library includes key RF blocks such as Low Noise Amplifier, Channel, Mixers, Power Amplifier, Filters, used in RF transceivers. The model is based on block diagram and behavioral descriptions. The models present two aspects, the first is based on the development of RTL code. The second uses the advantages of block diagram simulator as phase noise and noise figure generation to reduce drawbacks of behavioral limitations and the complete library is available in a unified simulation RF environment to simplify the simulation setup overhead, hence reducing simulation time while still obtaining acceptable accuracy versus Spice level simulation.

Keywords

RF, Modeling, Behavioral, Library, Mixed, Transceiver, Circuit simulation, Simulation software.

1. INTRODUCTION

Today shorter time-to-market, higher integration complexity and nanometer challenges increase the integrated circuit design difficulties. The different parts of the system are usually specified and designed separately by different engineers using different EDA tools.

To cope with the system level specifications, we need to mix different levels of abstraction in order to explore the implementation architectures and to validate the final design at the circuit level.

A generic RF library, taken into account the critical parameters (noise figure, phase noise, harmonics), allows accuracy fine-tuning and optimization at the system level. Combining with a unified simulation environment, it provides a new powerful approach for RF system design.

Now, with this approach, gap between specification block extraction and block design disappears. In this configuration, in our library, Simulink block diagram or feature generation can be exported and used as such with ADVance MS (ADMS). The link between Simulink and ADMS is ADVance System Model Extractor (ADVSME – MathWorks Integration) [1].

We present in this paper a RF library by unifying models from two different environments instead of using co-simulation. Co-simulation solution can be a time consuming task. We used, here, export between block diagram and behavioral simulators and we can mix different abstraction levels in the same environment, illustrated on figure 1.

Simulink provides in Communications Toolbox and RF library, different baseband modeling blocks. Indeed, we start this paper with a discuss between RF and baseband because RF is often the bottleneck in most of system level simulation which integrates baseband blocks. The block modeling presentation and the corresponding results are presented.

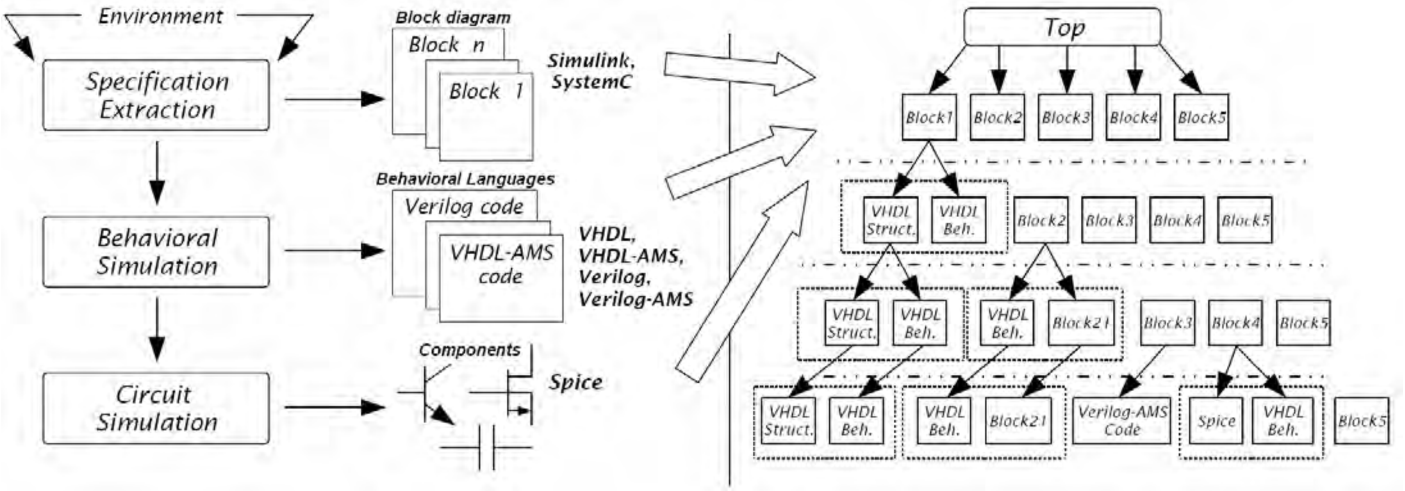

Figure 1 : Design and Simulation Support

978-1-4244-1567-0/07/$25.00 © 2007 IEEE 52

2. BASEBAND VERSUS RF SIMULATIONS

Simulink proposes a baseband modeling simulation performed for RF systems which significantly reduces the number of simulation steps.

For example, in a Bluetooth transceiver, a wireless system transmits a GFSK modulated signal at 1 MHz, split into 79 sub-carriers, at a carrier frequency around 2.4 GHz [2].

But, in this case, baseband modeling represents only spectral peaks within a bandwidth around the carrier frequency. So, one important drawback is the loss of nonlinear behavior outside of this bandwidth. Particularly, the carrier harmonics can have an impact on the receiver performances. So, for behavioral and functional approaches, we need to run different RF simulations to analyze a complete system. Due to theses restrictions, baseband modeling must not be neglected and we used it into Simulink with RF baseband library.

3. LANGUAGES PRESENTATION

The tools and languages for hardware description of VLSI circuits have changed over the years. We focused in this section on the languages selected for this work.

Mathwork's Simulink is a graphical What You See Is What You Get (WYSIWYG) editor based on the Matlab engine. This simulator covers the system level and we can obtain easily and quickly results due to the interactivity. It allows complete transmission systems simulation containing both RF, MAC, modems and channel by combining digital and analog parts. In fact, with this simulator type, the simulation accuracy is restricted for analog system parts and strong key is the whole system concept verification.

To describe analog, digital and particularly RF behavior, VHDL-AMS and Verilog-AMS are chosen as description languages which are an IEEE standard and a quasi industrial standard respectively.

4. LIBRARY PRESENTATION

A RF transceiver chain modeling contains key models as Low Noise Amplifiers, Mixers, Channel description, Power Amplifier, Phase Locked Loop with Voltage Controlled Oscillator, Filters. So, in this section, we presented these models.

4.1 Low Noise Amplifier (LNA)

The Low Noise Amplifier (LNA) is the first on-chip block. Then it is essential to have good signal amplification with low noise and a high linearity, according to Friis equation [3]. From a modeling point of view, these characteristics need to be modeled in the most accurate way. On the contrary, for Power Amplifier (PA), though we use the same modeling principles, noise is not modeled because it is not a critical parameter. The main focus of this kind of amplifier, unlike low-frequency amplifiers, is the large signal behavior and non-linear distortion due to harmonics and inter-modulations. All these effects, on top of power transmission (and therefore matching impedances) which link power and voltage gains, are modeled with equations (1) (2) and (3) [4] in VHDL-AMS code or Simulink.

$$A_v = \sqrt{\frac{R_{output}}{R_{input}} \cdot 10^{\frac{G_{pdB}}{10}}} \qquad (1)$$

$$IIP3_v = \sqrt{2 \cdot R_{input} \cdot 10^{\frac{IIP3_{dBm} - 30}{10}}} \qquad (2)$$

$$V_{outmax} = \frac{2}{3} \cdot A_v \cdot \sqrt{\frac{A_{v1}}{3 \cdot A_{v3}}} \qquad (3)$$

where A_v is the voltage gain, R_{out} and R_{in} are output and input resistances, G_{pdB} is the power gain in decibels, $IIP3_v$ is the input voltage Intercept Point of the 3rd order harmonics, $IIP3_{dBm}$ is the same described in power, V_{omax} is the maximal output voltage, A_{v1} and A_{v3} the voltage gains of the first order and the third order.

VHDL-AMS does not support noise generation, so we have chosen to use Matlab/Simulink to model the noise and the noise figure. To do so, we added a white Gaussian noise simulated in the time domain, which variance can be computed from the LNA specifications by equation (4) :

$$\overline{v^2}_{noise} = 4 \cdot k_B \cdot T_0 \cdot R_{input} \cdot (F - 1) \qquad (4)$$

where F is the linear noise factor, k_B the Boltzmann constant, T_0 the temperature of interest and R_i the LNA input impedance.

Usually, the VHDL-AMS LNA gain (G_{LNA}) is frequency dependent where it comes the following transfer function (5).

$$G_{LNA}(j\,\omega) = \frac{G(\omega_0) \cdot \frac{\omega_0}{Q} \cdot j\,\omega}{\omega^2 + j\,\omega \cdot \frac{\omega_0}{Q} + \omega_0^2} \qquad (5)$$

where $G(\omega_o)$ is the voltage gain and Q the quality factor. One can also find these characteristics expressed as S-parameters but here we focus on voltage and power modeling which is easier to handle with VHDL-AMS description.

The figure 2 summarizes the LNA model.

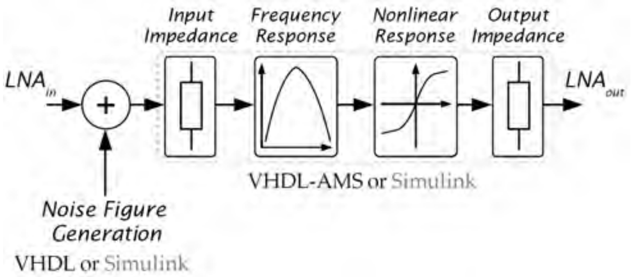

Figure 2 : LNA Block Diagram

4.2 Mixer

A mixer block uses nonlinear circuit characteristics for frequency conversion in the RF applications. In our work, we focused on a Gilbert cell but the results can be extended to other topologies. It either multiplies an RF input signal with a local oscillator (LO), then we obtain the output signal (down conversion) or it multiplies a LO and a IF signals, then we obtain an RF signal (up conversion).

In the modeling process, we used the same principles and equations as for the LNA, but another important characteristic for mixer modeling is the Phase Noise (PN). It is the result of random fluctuations in instantaneous phase or frequency due to intrinsic device noise or noise from the supply and control lines. Again, Simulink and Matlab are used to model this feature and mixer architecture is described with VHDL-AMS or Simulink. The principle of PN generation is presented in figure 3. We added the phase noise in the LO to match future mixer phase noise measurements. We generated a measurement-like PN with the white noise response of a transfer-function. The transfer-function output is the phase due to PN fluctuations, and is added to the ideal phase generated from an ideal LO.

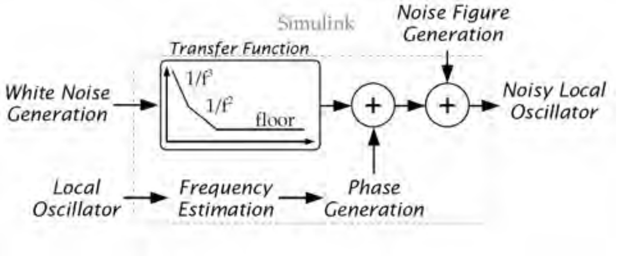

Figure 3 : Mixer Phase Noise Generation

The figure 4 summarizes the mixer model.

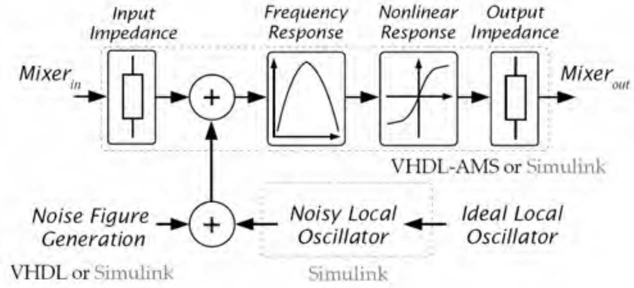

Figure 4 : Mixer Block Diagram

4.3 Channel

This channel model includes the main effects that affect a signal between the transmitter and the receiver antennas. We include in this model the Gaussian noise, delay, losses linked to frequency and distance, and also antenna impedances. Listing 1 presents the channel code.

Additive white Gaussian noise (AWGN) with power level *pn_dBm* requires the signal power level *ps_dBm* and the signal-to-noise ratio *snr* to generate distributed random numbers *var1* and *var2*. The core uses a digital process that generates a time discrete white Gaussian noise. The losses are computed in the air and linked to the frequency *freq_c*, distance *distance* between the two antennas and the signal is delayed with a *tdelay* after filtering. So, the input and output antennas impedances are modeled as ohmic resistances like LNA and Mixer models. We superimposed the input signal with a noisy signal and the difference between these two signals represents the AWGN. The next step for channel modeling is the implementation of the transmitting and receiving antennas directivity diagram.

```
LIBRARY DISCIPLINES;
LIBRARY IEEE;
USE DISCIPLINES.ELECTROMAGNETIC_SYSTEM.ALL;
USE IEEE.MATH_REAL.ALL;

entity CHANNEL is
  generic (
    distance : real := 100.0; -- distance in meters
    freq_c : real := 2.4e9; -- carrier frequency in Hz
    ps_dbm : real := -100.0; -- signal power in dBm
    snr : real := 100.0; -- signal to noise ratio
    fs_noise : real := 1.0e6; -- noise frequency
    tdelay : real := 0.0; -- channel delay
    rin : real := 50.0; -- input impedance
    rout : real := 50.0); -- output impedance
  port (
    terminal p_in : electrical; -- channel input
    terminal p_out : electrical); -- channel output
end entity CHANNEL;

architecture awgn of CHANNEL is

  -- constant declarations
  constant period : real := 1.0/fs_noise/2.0; -- time period
  constant pn_dbm : real := ps_dbm - snr; -- noise power
  constant pn : real := 10.0**((pn_dbm-30.0)/10.0); -- linear
  constant vn : real := sqrt(pn*rout); -- voltage.
  constant loss : real := 2.0*MATH_2_PI*distance*freq_c/
299792458.0;
  -- intern terminal & quantities declaration
  terminal n_int : electrical;
  quantity v_rin across i_rin through p_in;
  quantity v_int across i_int through n_int;
  quantity v_rout across i_rout through n_int to p_out;
  quantity v_noise, v_lpf : real;
  signal s_noise : real := 0.0;

begin
  -- time noise generation
  process is
    variable var1,var2,var : real := 0.0;
    variable SD1 : positive := 111;
    variable SD2 : positive := 333;
  begin
    uniform(SD1,SD2,var1); -- value between 0 and 1
    uniform(SD1,SD2,var2); -- same principle
    var := vn*cos(MATH_2_PI*var1)*SQRT(-2.0*log(var2));
    s_noise <= var;
    wait for period;
  end process;
  -- conversion, filter, impedance, delay, losses implementation
  v_noise == s_noise;
  break on s_noise;
  v_lpf == v_noise'ltf((0=>1.0),(1.0,1.0/MATH_2_PI/fs_noise/
2.0));
  v_rin == rin*i_rin;
  v_int == 2.0*(loss*v_rin'delayed(tdelay)+v_lpf);
  v_rout == rout * i_rout;
end architecture awgn;
```

Listing 1 : Channel modeling code

4.4 Voltage Controlled Oscillator

This model presented in [5] gives good results versus the Spice but the phase noise is not yet implemented. In a Phase-Locked-Loop (PLL), the VCO block is the main phase noise contributor. Then, to predict the phase noise influence on a PLL simulation, we need to develop and use a noisy VCO model. We used this VCO in the DCS standard configuration to validate our generic model.

Our model includes some important characteristics like timing jitter and phase noise, non linear tuning characteristic, power consumption and input and output impedances. Quiescent frequency, VCO gain, output amplitude, initial phase are made generics to future work.

Figure 5 presents the VCO model with phase noise. We have modeled the phase noise as explained in the previous section, but here, the VCO model includes the three phase noise contributions: flicker noise frequency, floor noise frequency and level and neighbor noise frequency as expressed in equation 6.

$$F_{nbn} = \frac{F_o}{(2 \cdot Q)} \qquad (6)$$

where F_o is the quiescent frequency, Q is the loaded resonator.

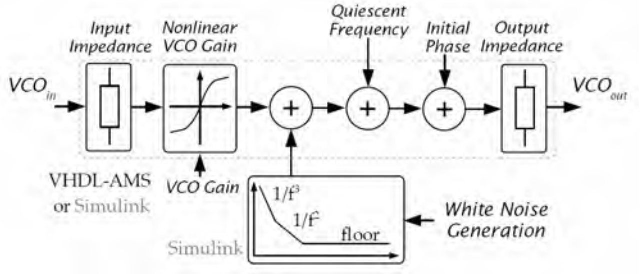

Figure 5 : VCO Block Diagram

5. RESULTS

5.1 Low Noise Amplifier Results

To improve the model accuracy, we start from looking at the voltage gain versus frequency simulated at transistor level (figure 6). For the first approach, we choose to consider the gain as constant for the behavioral and block-level modeling. A first improvement for this model is the addition of a band-pass filter to model the gain drop around the frequency of interest. The filter order is determined from the LNA topology. In case the frequency of interest does not correspond to the maximal gain, there is a need to extract the maximal gain frequency and translate the band-pass filter around this frequency.

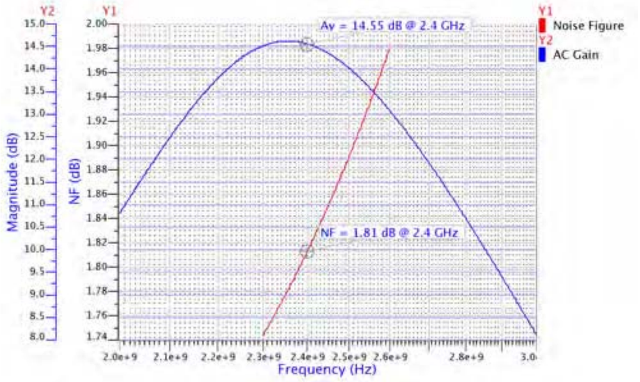

Figure 6 : LNA AC Gain & Noise Figure

The comparison of output power simulations is given in figure 7 ; the high level and low level model match pretty well in terms of power gain. The small discrepancies between the ideal model and the low level simulation can be explained by the clipping effect behavior versus the real simulation (Spice). But, we focused on the amplifying zone. However, we obtain a small difference between power gain (15.06 dB) and voltage gain (14.55 dB) due to input and output impedances values (closed to 50 ohms).

Even in the ideal case, the power gain is close enough to SPICE simulations and can be considered as constant for low input powers.

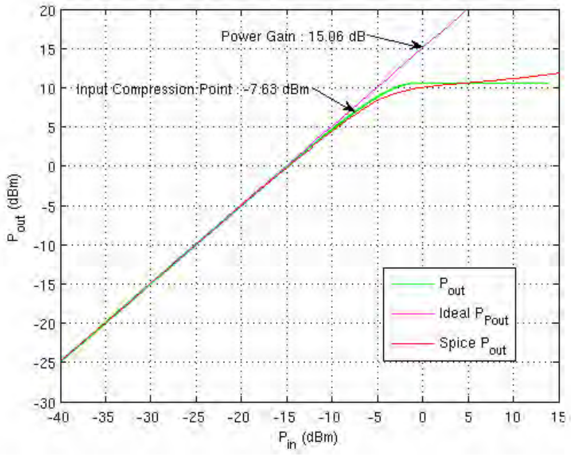

Figure 7 : LNA Simulation Comparison - Spice / Mixed

This means that it is possible to use high level models in global system simulations as long as the noise level can be neglected. For aggressive system design, one may want to add high level modeling noise contribution. For this purpose, the model we propose allows to take the noise figure into account.

The figure 8 shows the output power level for low input power level. The output power is the sum of the ideal output power and a white Gaussian noise, which in fact corresponds to the input noise amplification added to the internal LNA noise. Here, the Spice Noise Figure extraction results in a 1.81 dB NF (figure 6) and we use this parameter to generate the adequate noise. Although the noise is generated in the time domain, the model can also be used in the frequency domain via a FFT.

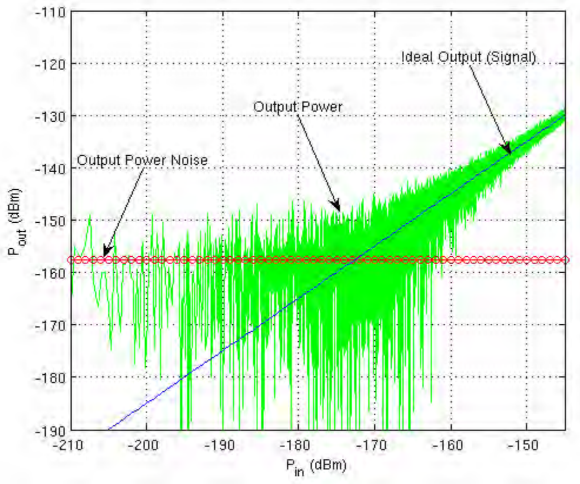

Figure 8 : Noise modeling for LNA and low power

5.2 Mixer Results

The mixer uses the same modeling principle as the LNA, but includes the phase noise. We use nonlinear circuit characteristics for frequency conversion. For demonstration purposes an RF simple tone (1 KHz, -70 dBm) and a local oscillator of 500 Hz and 0 dBm are used. The carrier is mixed with itself, which is known as a Direct Conversion receiver, since no intermediate frequency is used for demodulation. Figure 9 shows the simulation results after the mixer stage in the frequency domain. We found up and down conversions (500 Hz and 1.5 KHz) with an asymmetric gain (due to 3rd order nonlinear characteristics). Other spurious signals appear after conversion well below carrier and signal like IP3.

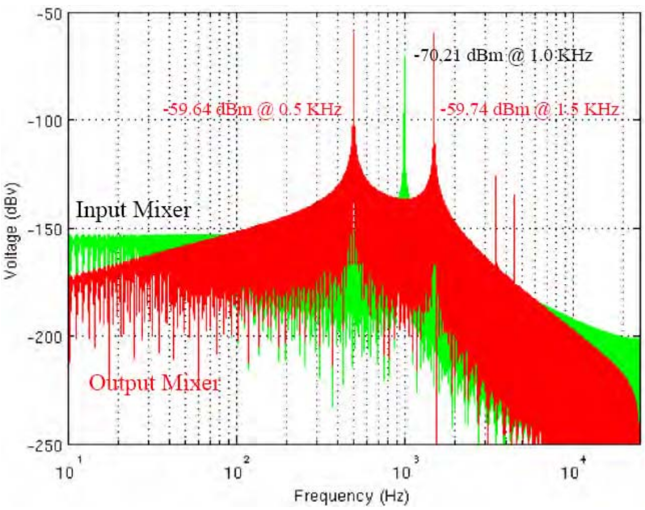

Figure 9 : Mixer modeling results

5.3 Channel Results

In the simulation example a single-tone of 1 GHz and 0 dBm was fed into the channel model. We used a SNR around 30 dB and a sampling frequency 1000 times greater than the signal frequency. Figure 10 shows the time simulation results. The input signal is superimposed with a noisy signal. Here, the distance between the two antennas is 0.1 meter. We obtain a SNR equal to 28.3 dB which matches 30 dB.

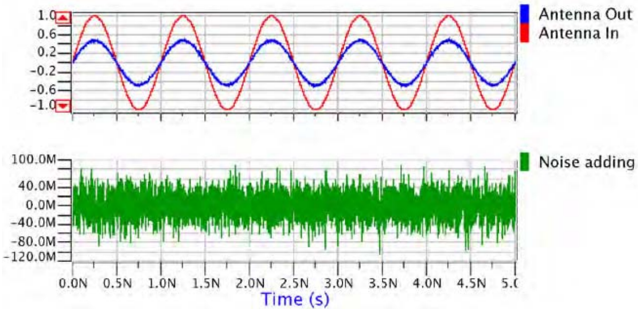

Figure 10 : Channel modeling - Time Simulation Results

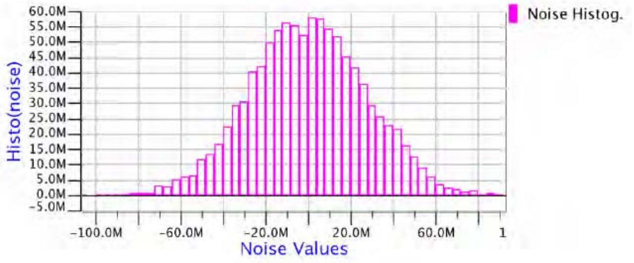

Figure 11 : Noise signal Time domain

Figure 11 presents a histogram computed from the difference between the noisy and pure signals. We observed the Gaussian nature of that noise.

5.4 Voltage Controlled Oscillator Results

In this part, we present only the phase noise modeling due to rich background for VCO model [5]. But, three constraints appear when we model PN.

The first is the need to increase the model sampling frequency with a 10 minimal ratio to keep convergence on the simulation.

Also, we needed to have a small DC Gain for small frequency offset to avoid integrator behavior. Without it, the block has a pure integrator behavior and diverge.

Thirdly, Mixer PN is only based on floor noise (zero slope) and neighbor noise ($1/f^2$ and -20dB/dec) whereas the VCO PN is based on three different PN : floor noise, neighbor noise and flicker conversion ($1/f^3$ and -30dB/dec). Simulink does not support decimal power and we needed to approach with multiple stages the transfer function to obtain -30dB/dec (7).

$$PN = G \cdot \frac{(p-\omega_f)(p-\omega_{nbn})(p-\omega_{z1})(p-\omega_{z2})(p-\omega_{z3})}{p(p-\omega_{p1})(p-\omega_{p2})(p-\omega_{p3})(p-\omega_{p4})} \qquad (7)$$

where PN represents the transfer function, G equals to the floor noise level, ω_f the flicker cut-off frequency, ω_{nbn} the neighbor cut-off frequency, ω_{pi} and ω_{zi} zeros and poles computed from ω_f.

Figure 12 shows the results between measurements and modeling with decimal and integer approach for transfer function.

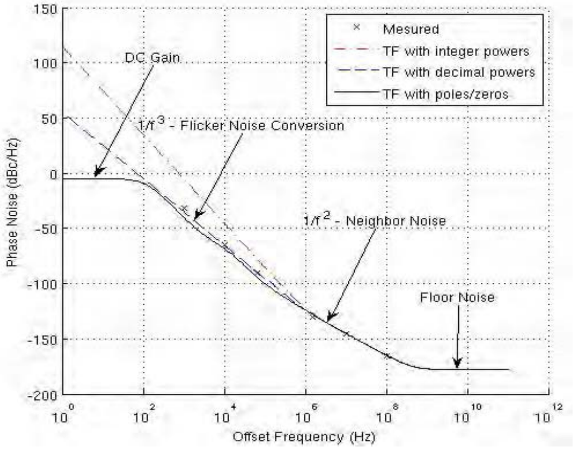

Figure 12 : VCO Phase Noise Modeling

We obtain a good result with decimal approach to match our VCO phase noise DCS measurements. Integer approach realizes only -20 and -40 dB/decade.

6. ACKNOWLEDGMENTS

The authors want to thank the CIM-PACA design platform for its support.

7. CONCLUSION

We've presented the development a RF library combining two representations: behavioral models and block diagrams. Model export from Simulink allows the combination of different model abstractions including features such as phase noise or noise figure generation, together with VHDL-AMS behavioral models used in a unified ADVance MS mixed-signal simulation environment. Such approach allows accuracy refinements of key parameters and architecture exploration at high level.

The next step is to use the library to validate the specifications of a complete transceiver including DSP (SystemC), MAC (block diagram) and RF blocks (VHDL-AMS, SPICE). Ultimately there is need to map such RF behavioral library to the circuit level to create a real RF Intellectual Properties (IP) to allow re-use in a "top-down" methodology. The results on a Bluetooth transceiver will be presented on the conference presentation. We have mixed different description languages (from System-C to SPICE) to decrease complexity and reduce simulation time with a good compromise to keep accuracy. With this study and a single EDA environment, we can extract RF blocks specifications, validate BER or EVM.

8. REFERENCES

[1] ADVance MS User's Manual, v4.5_1, AMS 2006.2

[2] http://www.bluetooth.com

[3] H.T. Friis, «Noise figures of radio receivers», Proc. IRE, vol. 32, pp.419-422, July 1944.

[4] «Modeling and Simulation for RF System Design» - Ronny Frevert et al. - Springer 2005

[5] B. NICOLLE et al., «VHDL-AMS modeling of a multi-standard PLL», IEEE ICECS 2005 Proceedings, pp. 33-36.