

MATLAB TOOLBOX FOR RF RECEIVER MODELING

BY

BOTOND S. KIREI*, **MARIUS GH. NEAG** and **MARINA D. ȚOPA**

Technical University of Cluj-Napoca

Received, May 31, 2011

Accepted for publication: July 19, 2011

Abstract. A novel RF receiver modeling approach is presented. The novelty consists in the usage of the object oriented programming instead of the usual imperative programming. Classes were defined for entities as *signal*, *non-ideality* and *RF/analog block*, in order to achieve a basic RF receiver model. The main circuit non-idealities were identified, non-ideality parameters and their analytical models were encapsulated into classes. The effectiveness of the proposed modeling approach was demonstrated by developing in MATLAB a fairly complete model of a direct conversion receiver.

Key words: direct conversion receiver; modeling of RF/analog blocks; thermal noise; nonlinearity; I/Q imbalance; phase noise; carrier frequency offset.

1. Introduction

The latest trend in wireless receiver development is the co-designing of the RF/analog front-end and the (usually digital) algorithms used to compensate the RF tuner non-idealities. Modern communication standards push ever higher both the carrier frequency and the channel bandwidth; as a result, the requirements imposed on the RF/analog blocks are more difficult to meet. In principle, this is (at least partially) mitigated by the usage of high-performance

* Corresponding author: *e-mail*: botond.kirei@bel.utcluj.ro

digital signal processing (DSP), including non-ideality compensation algorithms and error corrections. However, in order to assess the combined effect of RF/analog block non-idealities and determine whether or not a given DSP is able to handle it, one needs very detailed modeling of both the RF & analog front-end and the DSP.

When modeling a modern communication standard transmission chain the imperative programming may not be the most suitable approach as the code may become unstructured, unreadable and hard to reuse. Following on this observation we propose the usage of object oriented programming (OOP) in order to achieve not only the accurate modeling of block-level non-idealities but also to enable a framework for transmission chain modeling.

This paper is organized as follows: the next section briefly introduces the direct conversion receiver (DCR) architecture, highlighting the main blocks within its structure; the third section comprises a brief description of the proposed toolbox classes; the fourth section describes the DCR model developed with this toolbox; finally, conclusions are drawn and a few possible directions for further developments are discussed.

2. The Direct Conversion Receiver

Let us first review the main blocks of the DCR architecture (Gu, 2005), with the aim of identifying the main parameters/functions/limitations of the RF/analog blocks within the signal path of the receiver.

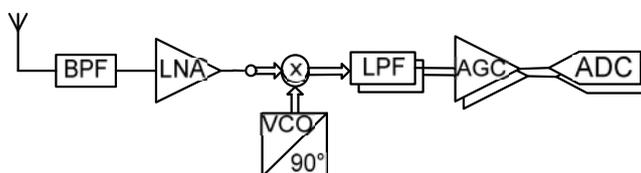


Fig. 1 – Basic DCR architecture.

The classic DCR receiver structure is depicted in Fig. 1. The antenna collects the RF signal, which is passed through a band pass filter (BPF) that selects the band used by the communication standard. The filtered RF signal is amplified by the low noise amplifier (LNA). The amplified signal is multiplied with a complex carrier generated by the local oscillator, based on a voltage controlled oscillator (VCO). This multiplication is performed by a quadrature mixer (MIX); thus, the RF signal is down-converted into a complex base-band signal, further processed by the in-phase and in-quadrature (I/Q) signal paths: the low pass filtering (LPF) is needed to remove unwanted spectral components – in particular the neighboring channels. The automatic gain control (AGC) block amplifies the filtered signal, now comprising mainly the wanted channel, to the full input range of the next stage, the analog to digital converter (ADC).

The main non-idealities affecting such a receiver are: the non-linearity

and noise of each block within the signal paths, the phase noise of the local oscillator, the I/Q imbalance due to phase and gain errors in the complex carrier and mismatches between the I and Q baseband paths, the aperture and clock jitter in the ADC (Fettweis, 2007).

3. Defined Classes

3.1. General Classes

Each class in the proposed toolbox was derived from the same parent class, called the BaseClass and having the diagram presented in Table 1. The aim of this class is to provide general functionalities that could be used in any derived class, for example the read and write of encapsulated properties.

The LNA, BPF, MIX, VCO and ADC are *RF/analog blocks* and these are communicating with the use of *signals*. Abstraction of the term *signal* leads to a class that includes properties (samples, sampling frequency, and center frequency) and methods (signal plot, arithmetic operations). The diagram of SignalClass is presented in Table 2.

Table 1
BaseClass Diagram

	Field name	Description
Methods	Write (<i>property,value</i>)	writes <i>value</i> in field named <i>property</i> ;
	<i>value</i> = Read (<i>property</i>)	returns the value from field <i>property</i>

Table 2
SignalClass Diagram.

SignalClass inherits all filed of BaseClass		
	Field name	Description
Properties	Samples	samples of the signal, can be a scalar, vector or matrix
	<i>F_s</i>	scalar, sampling frequency
	<i>F_c</i>	scalar, center frequency
Methods	object=Constructor (<i>Samples, F_s, F_c</i>)	object constructor; the first argument contains the samples of the signal; the second and third argument stands for the sampling frequency and center frequency
	object=Constructor(<i>signal</i>)	OOP allows polymorphism of functions; thus other constructor functions can be defined; this form takes as input argument an instantiation of SignalClass
	Plot	Plotting the contained values.

Abstraction of *RF/analog blocks* leads to the definition of BlockClass. This class is derived from BaseClass too. The common *RF/analog block* parameters are the input impedance (Z_{in}) and output impedance (Z_{out}).

BlockClass has a dedicated procedure, called *add*, to insert objects (class instances) into one of its field called *architecture*. Another important method is called *process*. This method can be used to implement the functionality of the blocks (amplification, filtering, mixing). The BlockClass is summarized in Table 3.

Table 3
BlockClass Diagram

BlockClass inherits all the fields of BaseClass		
	Field name	Description
Properties	architecture	Field architecture is a structure that contains objects of BlockClass. With use of the procedure <i>add</i> (de below) an abject can be added to the architecture field
	signal	the return value is saved into field signal
	Z_{in}	input impedance
	Z_{out}	output impedance
	Constructor(<i>input</i>)	<i>input</i> is an object of SignalClass, mainly used to initialize the fields inherited from SignalClass (<i>Samples</i> , <i>Fs</i> and <i>Fc</i>)
Methods	add(<i>block</i>)	<i>block</i> is an object of BlockClass and it will be added to the architecture field
	<i>result</i> =process(<i>signal</i>)	signal is an object of SignalClass. The return value <i>result</i> is computed in the next way: the process function of objects inserted in field architecture are invited in the order they were added in the architecture, passing the input signal from one to other.

3.2. Nonlinearity Class

Widely known parameters of the non-linearity found in the literature are: power gain, power at saturation P_{sat} , 1dB output compression point – OP1, third-order intercept point, IIP3 (Egan, 2003). The effect of non-linearity can be described with relation

$$y(t) = \sum_{k=1}^K a_k x^k(t), \quad (1)$$

where $x(t)$ is the input signal, $y(t)$ is the output signal and a_k – real coefficients. Usually the third order component is considered only, so relation (1) becomes

$$y(t) = x(t) \left[a_1 + \frac{3a_3}{4} x^2(t) \right]. \quad (2)$$

Eq. (2) is the starting point for the abstraction of nonlinearity, conducting to NonlinearityClass (Table 4). The coefficients a_1 and a_3 are properties of the class. Eq. (2) does not contain the saturation that can occur in circuits, thus further fields for the input and output amplitude limit, AI_{sat} , and

AO_{sat} , are computed using relation

$$AI_{sat} = \frac{2}{3} \sqrt{\frac{a_1}{3|a_3|}}, \text{ respectively } AO_{sat} = \frac{2}{3} AI_{sat}. \quad (3)$$

Due to the facilities provided by the OOP it is possible to implement the automatic update of a_1 and a_3 coefficients whenever the property values of AI_{sat} or AO_{sat} are accessed (modified). The processing described by eq. (2) is implemented in the method *process*.

Table 4

NonlinearityClass Diagram

NonlinearityClass inherits all the fields of BaseClass		
	Field name	Description
Properties	a_1	gain of the first order term
	a_3	gain of the third order term
	AI_{sat}	maximum input amplitude
	AO_{sat}	maximum output amplitude
Methods	$output=process(input)$	$input$ and $output$ are instances of SignalClass; this member executes the processing in eq. (2)
	$update(object)$	this member updates the values a_1 and a_3 of when $aisat$ or $aosat$ is accessed

Table 5

TermalNoiseClass Diagram

TermalNoiseClass inherits all the fields of BaseClass		
	Field name	Description
Properties	K	Boltzmann's constant 1.38×10^{-23} J/K
	R	Resistance
	B	Noise Band
	Temp.	Temperature
	NF	Noise Figure
Methods	$output=process(input)$	$input$ and $output$ are instances of SignalClass; process executes the mathematical model in eq. (5)
	$update(object)$	the function updates the value of Temp when is accessed

3.3. Thermal Noise Class

The electron flow in a resistance, R , operating at the temperature, T , generates a noise voltage; its value, integrated over the frequency range B , is given by

$$e^2 = 4RkTB, \quad (4)$$

where $k = 1.38 \times 10^{-23}$ J/K (Boltzmann's constant). Thermal noise is present in

each RF/analog block and its effect is described by

$$y(t) = x(t) + en(t), \quad (5)$$

where $n(t)$ is a random process with unitary power and normal distribution. This non-ideality is encapsulated in `TermalNoiseClass` (inspect Table 5 for the description of the class' fields). The noise figure (NF) of an RF/analog block expresses the degradation of the input signal to noise ratio (SNR) to output SNR. Thus the equivalent thermal noise, [K], of an RF/analog block can be computed from its noise figure, [dB]

$$T_e = 290(10^{NF/10} - 1). \quad (6)$$

3.4. Phase Noise

The signal generated by the VCOs is not purely sinusoidal in a silicon implementation. A frequency synthesizer comprises a reference oscillator – usually crystal-based – and a phase locked loop (PLL); the PLL consists of a phase detector, a loop filter, a voltage controlled oscillator and a divider with a given division factor. Even when the PLL is locked at the desired frequency, the inherent noise results in a slight variation of the phase of its output signal compared to the reference signal.

Table 6
PhaseNoiseClass Diagram

PhaseNoiseClass inherits all the fields of BaseClass		
Field name		Description
Properties	multiplier	this field points to a function that computes $e^{j\Phi(t)}$
Methods	<code>output=process(input)</code>	<code>input</code> and <code>output</code> are instances of <code>SignalClass</code> ; <code>process</code> executes the mathematical model in eq. (6)

Several models for phase noise can be found in literature; they all agree that the effect of phase noise can be modeled as a phase modulation of the baseband signal

$$y(t) = x(t)e^{j\Phi(t)}, \quad (7)$$

where $\Phi(t)$ is the time domain description of the phase noise. Many interpretation exists for $\Phi(t)$ (Horlin & Bourdoux, 2008).

A special case of the phase noise is the carrier frequency offset. In this case phase noise, $\Phi(t)$, is considered to be a linear function of time, $\Delta f t$, where Δf is the frequency offset. Thus eq. (7) becomes

$$y(t) = x(t)e^{j\Delta f t}. \quad (8)$$

PhaseNoiseClass (Table 6) provides a mean for the user to give its own function for $\Phi(t)$. This is implemented by property field dedicated for a function handler. Method process will invite the specified function to generate $e^{j\Phi(t)}$ and multiply the returned value and the input signal.

3.5. I/Q Imbalance (Mismatch)

The I/Q imbalance is a common problem for both direct conversion and low-IF receivers. The orthogonality between the I/Q paths is easily compromised by the amplitude and phase errors in the complex carrier generated by a quadrature oscillator. It was shown (Mailand *et al.*, 2006) that the I/Q imbalance (in the case of a direct conversion receiver) will cause the superposition of the desired signals conjugate over the desired signal

$$y(t) = k_1 x(t) + k_2 x^*(t), \quad (9)$$

where $x^*(t)$ is a complex valued, (representing the in-phase and quadrature signals), stands for conjugation, and coefficients are given by

$$k_1 = \frac{1 + g e^{-j\varphi}}{2}, \quad k_2 = \frac{1 - g e^{j\varphi}}{2}, \quad (10)$$

g being the amplitude and φ – the phase error. I/Q imbalance is measured using the image rejection ratio (IRR)

$$\text{IRR} = 20 \log \frac{|k_2|}{|k_1|}. \quad (11)$$

Considering the wideband nature of the communications standards used nowadays I/Q imbalance should be considered frequency selective. Frequency selectivity is the result of the many processing stages in the RF receiver. The main contributors to frequency selective (FD) I/Q imbalance are the lowpass filters placed on the I/Q paths for desired band selection. Due to the physical implementation the LPFs' places on the in phase and quadrature paths are different, they have a deviation from their nominal values. Thus FD I/Q imbalance is modeled by

$$y'(t) = h_i(t) \Re\{k_1 x(t) + k_2 x^*(t)\} + h_q(t) \Im\{k_1 x(t) + k_2 x^*(t)\}, \quad (12)$$

where h_i and h_q are the deviations from the LPFs nominal value. From (11) one can easily obtain

$$y'(t) = \left(\frac{h_i + g h_q e^{-j\varphi}}{2} \right) x(t) + \left(\frac{h_i - g h_q e^{j\varphi}}{2} \right) x^*(t). \quad (13)$$

The diagram of the created class for the I/Q imbalance is presented in Table 7.

Table 7
IQImbalClass Diagram

I/QImbalClass inherits all the properties of BaseClass		
	Field name	Description
Properties	g	amplitude error
	φ	phase error
	k_1	imbalance parameter, k_1
	k_2	imbalance parameter, k_2
	IRR	image rejection rate
Methods	$output=process(input)$	$input$ and $output$ are instances of SignalClass; process executes the mathematical model in eq. (9)
	update	the update function for k_1 , k_2 and IRR; they will be updated whenever the values of g or φ are accessed

4. Simulation Example

Let us consider the DCR architecture depicted in Fig. 2, with the parameters summarized in Table 8 (Oneț *et al.*, 2010). For simplicity we considered all RF/analog block input and output impedances equal to 50Ω .

Table 8
RF/Analog Block Parameters

Block	Gain, [dB]	Noise Figure, [dB]	Other
BPF	-3	0	
LNA	15	1.5	
MIX	15	7.8	IRR = 29 dB, CFO = 1 MHz
LPF	0	6	$F_{\text{pass}} = 0.4F_{\text{sample}}$
AGC	50	20	

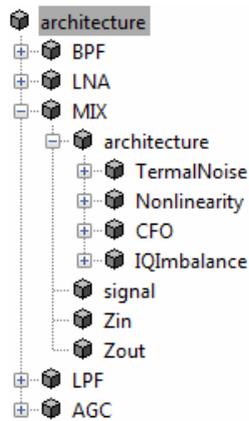


Fig. 2 – Tree view of the DCR container object properties.

One advantage of the OOP approach is a well-defined boundary between the core model and support/add-on features such as graphical user interface. For example, it was an easy task to generate the tree view of the RF receiver architecture depicted in Fig. 2. The top level architecture contains a BPF, LNA, MIX, LPF and AGC objects. Each object is a container of selected non-idealities. *E.g.* the object MIX contains instances of non-ideality classes (*TermalNoise*, *Nonlinearity*, *IQImbalance* and *CFO*). Beside these non-idealities the MIX contains properties as: Z_{in} , Z_{out} and *signal*.

The input signal, depicted in Fig. 3 *a*, consists of a desired and an adjacent channel, both an 8 MHz wide OFDM signal of $N = 1,075$ carriers. The desired and adjacent channel levels are -60 dB and, respectively, -70 dB.

The input signal undergoes a BPF filtering. This processing step does not modify substantially the spectrum of the desired signal, the gain of the BPF being -3 dB. Then the LNA amplifies the signal with 15 dB (Fig. 3 *b*). The thermal noise introduced by the LNA is about -140 dB.

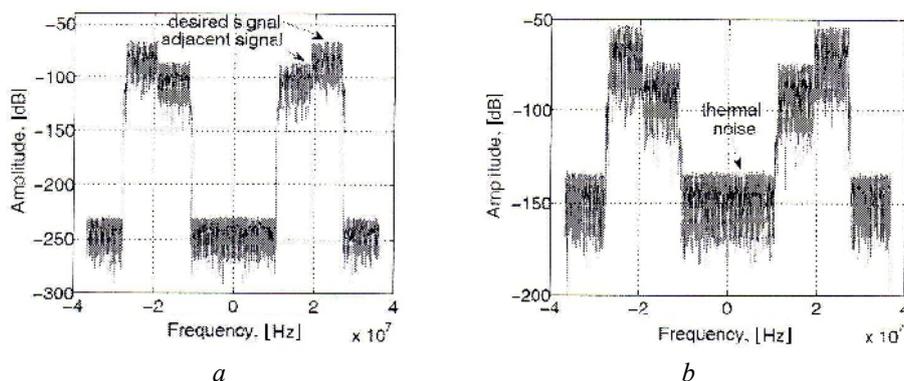


Fig. 3 – *a* – RF signal collected by the antenna; *b* – signal spectrum after LNA.

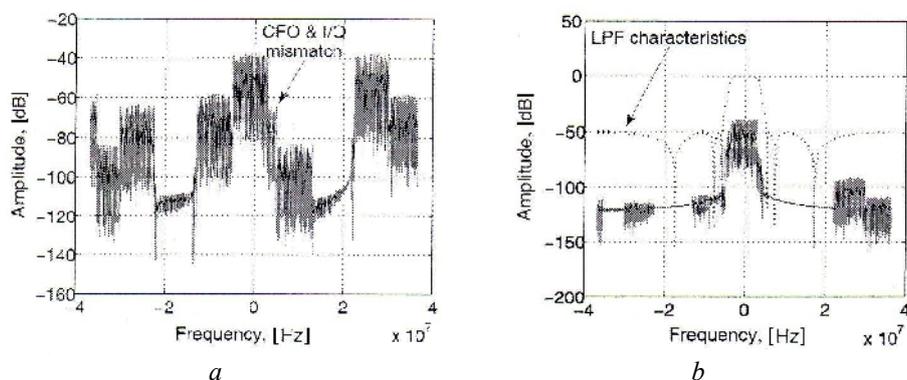


Fig. 4 – Signal spectrum after: *a* – MIX; *b* – LPF.

The amplified signal enters into the down-conversion stage carried out by MIX. CFO and I/Q imbalance are modeled in the mixer. The CFO is set for an exaggerated 1 MHz and the IRR for 29 dB, in order to make these two effects visible in Fig. 4 *a*. As one can see on the right side of the desired signal there is a side lobe, which is the combined result of CFO and I/Q imbalance. The MIX contributes with 15 dB gain in the overall receiver chain. Fig. 4 *b* is the desired signal spectrum after LPF. The LPF characteristics (6th order Chebyshev type II) are depicted too in this figure. Finally the AGC amplifies the received signal to the ADC input range.

5. Conclusions

A MATLAB toolbox for modeling the non-ideality of RF/analog blocks was developed. The novelty of this simulation toolbox is the OOP approach in the creation of the toolbox blocks. This approach makes it easy to encapsulate non-ideality parameters and their effects in one entity. The usage of the toolbox accelerates the creation of RF receiver modeling, thus potentially accelerating the verification of various algorithms employed in wireless communication, such as I/Q imbalance compensation, CFO mitigation, etc. Future work will comprise: completion of the toolbox with harmonic image distortion model and ADC specific non-ideality models (quantization noise, aperture and clock jitter, nonlinearity); the implementation of automated performance characteristics plotting (signal to noise and distortion ration (SINAD), difference with respect to reference (DwrtR, (Oneț *et al.*, 2010))); the development of a graphical user interface for the construction of RF receiver models.

Acknowledgment. This paper was supported by the project “Development and Support of Multidisciplinary Postdoctoral Programmes in Major Technical Areas of National Strategy of Research – Development – Innovation” 4D-POSTDOC, contract no. POSDRU/89/1.5/S/52603, project co-funded by the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013 and the Romanian National University Research Council through project “A New Approach in the Design of Multi-Standard Integrated Radio Receivers for Digital Radio and Mobile TV” under grant no. ID 2534/2008.

REFERENCES

- Egan W.F., *Practical RD System Design*. J. Wiley & Sons, Inc., Hoboken, New Jersey, 2003.
- Fettweis G., Lohning M., Petrovic D., Windish M., Zillman P., Rave W., *Dirty RF: A New Paradigm*. Internat. J. of Wireless Inform. Networks, **14**, 2 (2007).
- Gu Q., *RF System Design of Transceivers for Wireless Communications*. Springer Science + Business Media LLC, New York, 2005.
- Horlin F., Bourdoux A., *Digital Compensation for Analog Front-Ends*. J. Wiley & Sons Ltd, Chichester, UK, 2008.
- Mailand M., Richter R., Jentschel H.-J., *I/Q-Imbalance and its Compensation for Non-Ideal Analog Receivers Comprising Frequency-Selective Blocks*. Adv. in Radio Sci., **4**, 189–195 (2006) ([Http://Www.Adv-Radio-Sci.Net/4/189/2006/Ars-4-189-2006.Pdf](http://www.Adv-Radio-Sci.Net/4/189/2006/Ars-4-189-2006.Pdf), Last Viewed 20.10.2010).
- Oneț R., Popescu V., Neag M., Sărăcuț I., Țopa M., McDonagh S., *Matlab Modeling and Analysis of the Signal Path in Zero-IF DVB-T/H Radio Receivers*. Proc. of the IEEE Internat. Symp. on Electron. and Telecomm. ISETC '10, Timișoara, Romania, November 11-12, 2010.

UTILITAR MATLAB PENTRU MODELAREA RECEPTOARELOR DE RADIOFRECVENȚĂ

(Rezumat)

Este prezentată o nouă abordare pentru modelarea receptoarelor de radiofrecvență. Noutatea utilitarului constă în utilizarea principiilor de programare orientate pe obiecte în loc de programare imperativă. Pentru a realiza descrierea unui receptor s-au definit clase pentru entități cum sunt *semnalele*, *neidealitățile* și *blocurile* de radiofrecvență. S-au identificat principalele neidealități de circuit, parametrii și modelele analitice ale neidealităților au fost încapsulate în clase. Eficiența utilitarului propus a fost demonstrată prin dezvoltarea în MATLAB al unui model complet al receptoarelor cu conversie directă.