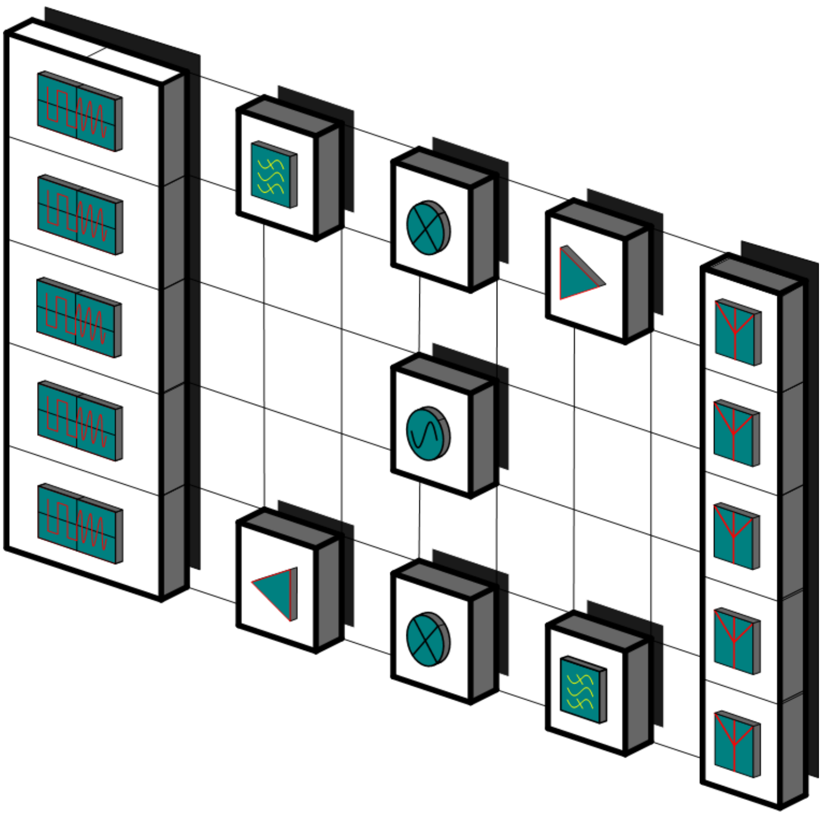
CHALMERS



Behavioral Modeling of RF front end devices in Simulink®

EYAD ARABI & SADIQ ALI

Signal processing group
Department of Signals and Systems
Chalmers University of Technology

Göteborg, Sweden, 2008 EX021/2008

MASTER’S THESIS EX021/2008

Behavioral Modeling of RF frond end devices in Simulink®

Master's Thesis in the Department of Signals and Systems

EYAD ARABI & SADIQ ALI

Signal processing group
Department of signals and systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2008

Behavioral Modeling of RF front end devices in Simulink®
Master's Thesis in the Department of Signals and Systems
EYAD ARABI & SADIQ ALI

Master's Thesis in the department of signals and systems
Department of Signals & Systems
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Printed by Chalmers Reproduction Service
Göteborg, Sweden 2008

Abstract

Simulation of wireless systems has become a key issue in analyzing, optimizing and designing wireless systems. In this thesis, modeling RF front end devices in Simulink® is investigated. The capabilities of Simulink® and RF Blockset are tested. Their different behavioral models for nonlinearity, noise, phase noise and mismatch are analyzed. A model for the power amplifier that takes into account memory effects is implemented in Simulink to extend the RF Blockset to model wideband applications such as WCDMA. This model for the power amplifier implements a memory-polynomial model. Memory-polynomials prove to be both accurate and easy to implement.

Keywords: RF front end modeling, behavioral modeling, power amplifier modeling, memory-polynomial, Simulink, RF Blockset.

Acknowledgment

The authors wish to thank Professor Mats Viberg for his smooth and effective supervision, Professor Thomas Ericsson for his invaluable directions and comments and Ali Soltani who has provided the measurement data for the power amplifier used in this work.

Contents

1 Introduction

The main goal of modern wireless communication systems is to provide a variety of communication services to anybody, anywhere, any time. These services for the next generation of communications include high speed data, video and multimedia traffic as well as voice communication. To provide these services communication systems have to overcome different types of impairments in the transmitter, the receiver and the channel. Including these impairments in a system level simulator will facilitate both design and analysis of wireless systems. This thesis work is done within the Charmant research center at the Signals and Systems department of Chalmers University of technology. It is a part of a project on simulating an end to end MIMO system for WCDMA applications. This work is an extension to a previous master thesis on MIMO channel modeling. In this thesis a model for the RF front end is implemented.

1.1 RF front end systems

An RF front end system refers to the analog front end of the wireless communication system. The digital base-band signals cannot be transmitted directly through wireless channels due to properties of the electromagnetic waves. Therefore, these signals must be converted to analog, up converted to higher frequencies, and transmitted through the channel. The received signals are down converted to the base-band frequency then converted to digital again. An overall wireless communication system is shown in Fig. 1.1. Processes done to the analog signal in the RF front end includes filtering, amplification, and mixing. These processes are imperfect and add various impairments to the received signal, in this thesis the modeling of major impairments added by each RF component is investigated, and an overall model for the receiver and the transmitter are implemented in Simulink® .

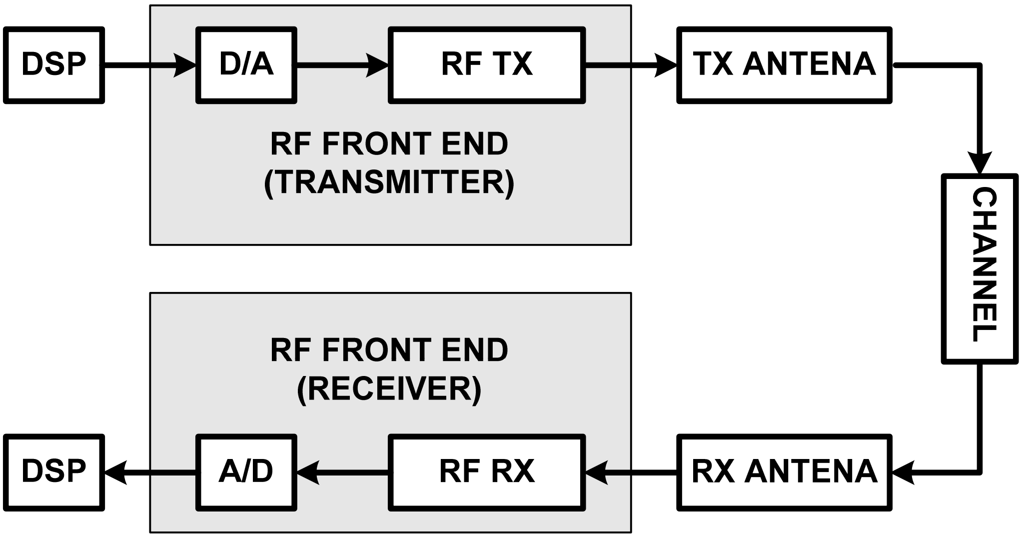


Fig. 1.1 A schematic of an overall wireless system structure. Antennas are sometimes included with RF frond end.

1.1.1 RF Transmitters

Fig.1.2. Shows a block diagram for a typical RF transmitter, only the power amplifier and the mixer are included in the modeling, because these components add the most serious impairments to the transmitter. Other components that do not add serious impairments, like filters, are ignored in this thesis. The mixer introduces phase noise, spurious frequencies and nonlinearity. The power amplifier introduces nonlinearity. The models for these blocks are illustrated in subsequent chapters.

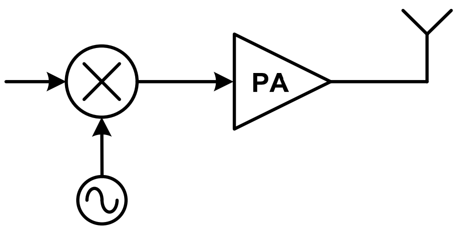


Fig. 1.2 A Schematic of an RF transmitter. Only components that introduce effective impairments are included.

1.1.2 RF Receivers

Fig.1.3. shows a block diagram for an RF receiver. Only the low noise amplifier (LNA) and the mixer are included in the model. The LNA introduces noise and nonlinearity. The mixer introduces phase noise and nonlinearity.

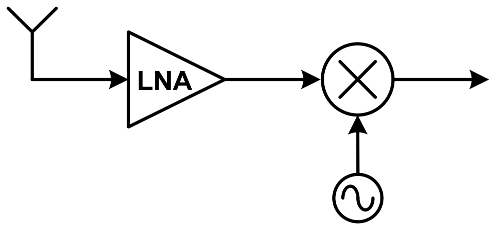


Fig. 1.3 A schematic of an RF receiver. Only blocks that introduce effective impairments are included.

1.2 Types of modeling

Modeling is the process of representing real-world objects or phenomena as sets of mathematical equations. Modeling can be divided into two types, Physical modeling and behavioral or mathematical modeling.

1.2.1 Physical modeling

A physical model requires knowledge of the elements that comprise the real system, their constitutive relations, and the theoretical rules describing their interactions. These types of modeling are appropriate for circuit level simulation and can be very accurate [17].

1.2.2 Behavioral modeling

A behavioral model, also called black box model, does not require prior knowledge of the physical systems. Its internal structure relies only on input and out measurements. The parameters of behavioral models are indentified from input and output measurement data. Therefore measurement techniques and the quality of data affect the accuracy of these models [17]. All the models presented in this thesis are behavioral.

1.3 Base-band and Pass-band modeling

A pass-band signal can be represented by the following equation [4]:

$$x_{PP}(n) = r(n)\cos\big(\omega_c n + \varphi(n)\big) \qquad (1.1)$$

where

ω_c is the center carrier frequency

$r(n)$ is the amplitude of the signal

$\varphi(n)$ is the modulated phase

This pass-band signal can be re-written as

$$x_{PB}(n) = I(n)\cos(w_c n) - Q(n)\sin(w_c n) \qquad (1.2)$$

where $I(n) = r(n)\cos\,[\varphi(n)]$ and $Q(n) = r(n)\sin[\varphi(n)]$ are the in phase and quadrature components respectively. Equation (1.2) can be re-written as

$$x_{PB}(n) = r(n)\cos\,[\varphi(n)]\cos(w_c n) - r(n)\sin[\varphi(n)]\sin\,(w_c n) \qquad (1.3a)$$

$$= Re\big[r(n)e^{j\varphi(n)}e^{jw_c\cdot n}\big] \qquad (1.3b)$$

$$= Re\big[x(n)e^{jw_c\cdot n}\big] \qquad (1.3c)$$

where

$x(n)$ is the complex base-band signal, which can be represented by

$$x(n) = r(n)e^{j\varphi(n)} \qquad (1.4a)$$

$$= I(n) + jQ(n) \qquad (1.4b)$$

Simulations can be done either in pass-band or complex base-band. Pass-band simulations are simpler and more accurate. However, they consume more resources and simulation time. To show this, assume a signal of center frequency f_c and an operational bandwidth B. To simulate one second of real time for this signal in Pass-band, $f_c + B/2$ simulation cycles are needed. In contrast, base-band simulations require B cycles only. Therefore, the reduced accuracy by using base-band simulations is usually a reasonable trade for the improved simulation time.

The main effect, that complex base-band simulations do not consider while pass-band simulations do, is the fluctuations in the carrier frequency, these fluctuation can also be referred to as phase noise. Therefore, if these fluctuations are to be modeled, either use an extra base-band model for these fluctuations or Pass-band simulations should be used.

1.4 Harmonics and intermodulation products in nonlinear devices

All electronic devices are inherently non-linear. Nonlinearity is desirable in some devices for example mixers, and not desirable for other devices like power amplifiers. The effect of nonlinearity of electronic devices can be characterized by applying single tone and multi tone signals to their inputs and observe the outputs at each case.

If a single tone signal with frequency f is applied to a nonlinear device the output will include signals of frequencies of the form nf, where n is an integer greater than one. These frequencies are called *harmonics*.

If a two tone signal with frequencies f_{RF} and f_{LO} is applied to a nonlinear device the output will include frequencies of the form $mf_{RF} \pm nf_{LO}$where m and n are two integers greater than zero. Theses frequencies are called intermodulation products. The order of a given intermodulation product is defined as $|m| + |n|$. The most serious intermodulation product in electronic devices is the 3rd order intermodulation product, because it is usually close to the desired frequency and cannot be completely filtered out. This product is usually characterized by the third order intercept point refereeing either to the input or the output (IIP_3 or OIP_3). This is best defined by looking at Fig. 3.2. It can be shown that the slope of the linear gain for input and output powers in dBs is unity, likewise the slope of the third gain of the third order intermodulation component is 3 [7], the point where the third order line intersects with the linear gain line is the third order intercept point.

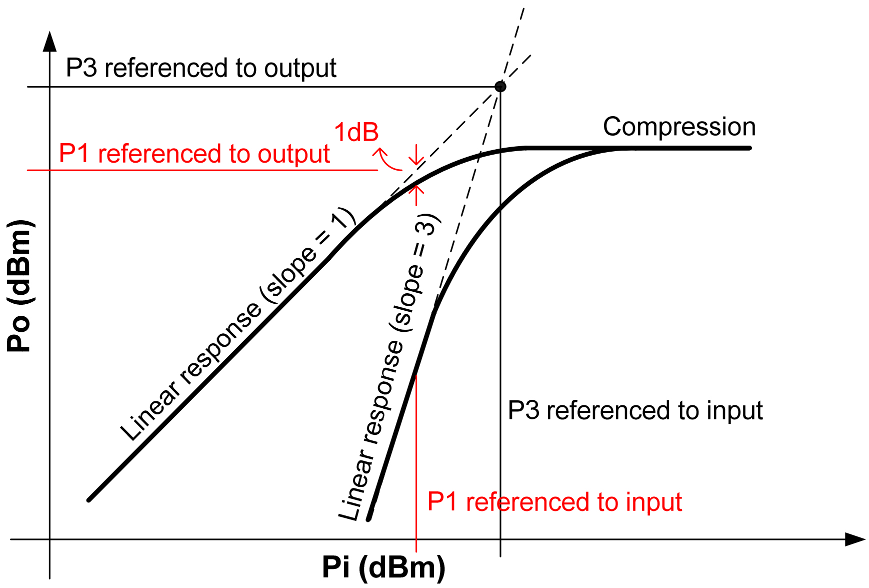


Fig. 3.2. An illustration of the first and third order intercepts points.

Another figure of merit to characterize nonlinearity is the 1dB compression point. For a nonlinear device, the 1dB compression point is defined as the point where the difference between the device's output and the linear output is exactly 1 dB. The 1 dB compression point is typically 12 to 15 dB less than the 3rd order intercept point assuming they are referenced at the same point.

2 Power Amplifiers (PA)

2.1 Introduction

An amplifier is a device designed to increase signal power levels. There are mainly two types of amplifiers in RF front end circuits; these are power amplifiers (PA), and low noise amplifiers (LNA). Power amplifiers are mainly present in the transmitters, and are designed to raise the power level of the signal before passing it to the antenna. This power boost is crucial to achieve the desired signal to noise ratio at the receiver, and without which received signals would not be detectable.

For the power amplifier it is necessary to have as high gain as possible, while adding as little distortion to the signal as possible i.e. be as linear as possible. For small and mobile transmitters there is usually another factor not less important; that is power efficiency since these devices are usually battery driven. Unfortunately, from a circuit design point of view, increasing the power efficiency would mean driving the device more and more into nonlinearity region which means that the amount of distortion will increase. The problem of nonlinearity is not so serious for applications where information is put in the carrier signal's phase. However, with the recent jam in the frequency bandwidth, new regulations have made strict bands that modulation schemes can work in, while at the same time higher data rates need to be achieved, therefore, it is logical to move to modulation schemes that carry information in both amplitude and phase. If the amplitude carries information then irreversible envelope distortions will affect the received signal quality and in turn increase the bit error rate of the received signal. Therefore, in modeling a power amplifier for wide band applications the most important aspect to model is the device nonlinearity that the amplifier introduces to the system.

The target application for this work is WCDMA. An important characteristic of this scheme is the large band width. Large bandwidth generally leads to higher frequency of the signal envelop, this feature will bring another important phenomena to the surface that is memory effect. Almost every device has a memory temporal dynamics or response delay, these effects are due to the biasing of the circuit and the capacitance or inductance built in the device. For wide band applications, memory effects cannot be ignored. Therefore, a model that takes into account memory effects must be used to accurately model the power amplifier's distortion.

In Section 2.2 an overview of the most commonly considered models for power amplifiers are presented, a comparison between these models is also discussed to select the most suitable model. In Section 2.3 the implementation of the selected model in Simulink® is explained in detail.

2.2 Power amplifier modeling

The power amplifier's nonlinearity broadens the input signal's bandwidth. This is known as Spectral re-growth and is undesired. Spectral re-growth causes interference with adjacent channels and increases the probability of violations of the out-of-band emission requirements mandated by regulatory bodies. It also causes distortions within the signal bandwidth, which affects the bit error rate at the receiver. Most recent transmission schemes, such as Wideband Code Division multiple access (WCDMA) or Orthogonal Frequency Division Multiplexing (OFDM), are especially vulnerable to the nonlinear distortions due to high fluctuations in their power levels.

To analyze a power amplifier (PA) system for future communication system it is important to model the behavior of PA nonlinearity and memory effect accurately. Nonlinear amplifier behavioral models can be divided into three types; memory-less (static), quasi memory-less and models with memory. In the following subsections some popular behavioral models are presented and compared.

2.2.1 AM-AM and AM-PM Modeling

The AM-AM conversion for a nonlinear system is the relation between the amplitude of the system's output and the amplitude of the system's input. The AM-PM conversion for a nonlinear system is the relation between the phase change of the system's input and output, and the amplitude of the input signal. This is shown in Fig. 2.1.

Assuming the pass-band input signal (1.1), the output of the AM-AM and AM-PM model $y_{PB}(n)$ can be written as

$$y_{PB}(n) = g\big(r(n)\big)\cos\left(\omega_0 n + \varphi(n) + f\big(r(n)\big)\right) \qquad (2.1)$$

where

$g(n)$ is the amplitude nonlinearity or AM-AM conversions

$f(n)$ is the phase nonlinearity or AM-PM conversions

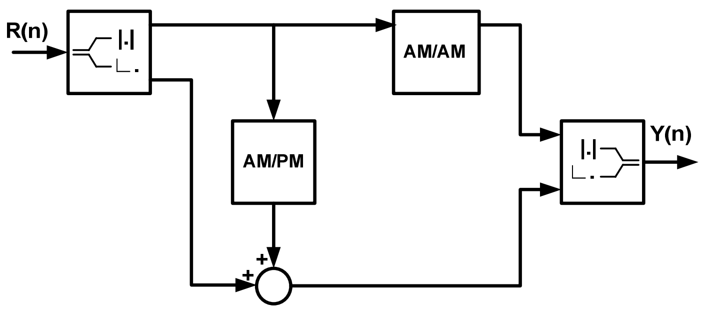


Fig.2.1. Amplitude-phase nonlinear model structure for a complex base band input and output signals.

In the following Sections different models for power amplifiers are presented.

2.2.2 Memory-less and quasi memory-less nonlinear models

In memory-less (static) power amplifier models the output signal is a nonlinear function of the current input signal only and previous values of the signal have no effect on the output of the model. Memory-less models only consider AM-AM conversions, and assume no phase change. On the other hand Quasi-memory-less models take into account both amplitude and phase distortions. Therefore, they are represented by the amplifier AM/AM as well as AM/PM transfer functions.

Static models give reasonable accuracy for applications with a narrow-band frequency spectrum or when memory effects are not important. Quasi-memoryless models have better accuracy for narrowband applications. In the following subsections some memory-less and quasi-memoryless models are discussed. Most of these are already implemented in Matlab Simulink® .

2.2.2.1 Polynomial model

The polynomial memory-less power amplifier model can be represented in base-band by the following equation

$$y(n) = \sum_{k=1}^{K} b_k \, x(n)|x(n)|^{k-1} \tag{2.2}$$

where

$x(n)$ is the input complex base-band signal.

$y(n)$ is output complex base-band signal.

b_k are real-valued coefficients.

Making the coefficients b_k complex in equation (2.2) will result in a quasi memory-less polynomial model for the power amplifier [2].

2.2.2.2 The Rapp Model

The Rapp model uses three parameters, and models amplitude distortion but no phase distortion. The general expression of the AM-AM conversions is as follows

$$g\big(r(n)\big) = \frac{r(n)}{\left[1 + \left(\frac{r(n)}{O_{sat}}\right)^{2s}\right]^{1/2s}} \tag{2.3}$$

where

O_{sat} is a parameter that sets the output saturation level.

s is a parameter that sets the smoothness of the transition from linear to saturation states, the smaller S the smoother the transition.

The technique of this model is quite simple. It assumes linear performance until the saturation point is approached. When the saturation point is approached, a transition towards a constant saturated output is applied [18].

2.2.2.3 The Saleh model

The Saleh model is a quasi-memoryless model. It uses four parameters to fit the model to measurement data. Its AM-AM and AM-PM conversion functions are described by the following equations

$$g\big(r(n)\big) = \frac{\alpha_a r(n)}{1 + \beta_a r(n)^2} \tag{2.4a}$$

$$f\big(r(n)\big) = \frac{\alpha_\varphi r(n)^2}{1 + \beta_\varphi r(n)^2} \tag{2.4b}$$

where

$[\alpha_a\ \alpha_\phi\ \beta_a\ \beta_\phi]$ are the model's parameters, [13].

2.2.2.4 The Ghorbani model

The Ghorbani model uses eight parameters to fit the model to measurement data, this model is quasi- memoryless, and its AM-AM and AM-PM conversions functions are described by the following equations

$$g(r(n)) = \frac{x_1 r(n)^{x2}}{1 + x_3 r(n)^{x2}} + x_4 r(n) \tag{2.5a}$$

$$f(r(n)) = \frac{y_1 r(n)^{y2}}{1 + y_3 r(n)^{y2}} + y_4 r(n) \tag{2.5b}$$

where

$x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$ are the model's parameters, which are calculated from measurement data by means of curve fitting [12].

2.2.2.5 The Hyperbolic tangent model

The Hyperbolic tangent model is quasi-memoryless. It has five parameters these are, IIP3, linear gain G, upper power limit P_U, lower power limit P_L and the linear phase gain G_{PM}. One characteristic of this model is that its parameters are related to physical attributes like IIP3. Its AM-AM conversion function is described by the following equation

$$g(r(n)) = \tanh\left[\frac{3}{IIP3}\, r(n)\right] \mathrm{G} \tag{2.6a}$$

where

$IIP3$ is the third order intercept point referred to the input.

G is the linear gain.

The AM-PM conversion is linear and is specified by the slope of the AM-PM conversion G_{PM} in *degree/dB*. This linearity is bounded by two parameters which are P_U and P_L. If the power magnitude of the input is less than P_L then no phase distortion is added, and if the power magnitude of the input is greater than P_U then a constant phase shift of $G_{PM}(P_U - P_L)$ is applied. This can be represented by the following equation

$$f\big(r(n)\big) = \begin{cases} G_{PM}(P_U - P_L) & |r(n)| > P_U \\ |r(n)|G_{PM} & P_L \le |r(n)| \le P_U \\ 0 & |r(n)| < P_L \end{cases} \tag{2.6b}$$

where

G_{PM} is the slope of the AM-PM linearity.

P_U and P_L are the upper and lower power limit parameters respectively.

2.2.3 Nonlinear models with memory

In reality the output of the power amplifier depends on previous inputs as well as the current input of the amplifier. This phenomenon is called memory effect, or simply temporal dynamics. These memory effects are due to thermal effects, and long time constants in DC bias circuits. It can be observed as asymmetries in lower and upper sidebands, and bandwidth dependent variations in the magnitude of intermodulation products. For higher bandwidth applications, e.g. WCDMA, the memory effects becomes severe, and cannot be ignored. Hence, memory-less and quasi-memoryless models are not accurate enough. Therefore, a model which considers memory effects should be used for such applications.

In the following Sections some of the most common models with memory are presented.

2.2.3.1 Volterra series

The Volterra model can be used to describe any nonlinear stable system with fading memory, with an arbitrary small error. However, its main disadvantages are the dramatic increase in the number of parameters with respect to nonlinear order and memory length, which causes drastic increase of complexity in the identification of parameters. This is the reason why it is highly unpractical to use volterra series for systems with high nonlinear orders and memory lengths. This model can be expressed mathematically as follows [2]

$$y(n) = \sum_{k} \sum_{l1} \dots \dots \sum_{l_{2k+1}} h_{2k+1}\,(l_1, l_2 \dots \dots .., l_{2k+1}) \prod_{i=1}^{k+1} x(n - l_i \prod_{i=k+2}^{2k+1} x^*\,(n - l_i). \quad (2.7)$$

From the equation above it is clear that the number of coefficients of the Volterra series increases exponentially as the memory length and the nonlinear order increase.

2.2.3.2 Wiener and Hammerstein models

As mentioned above Volterra series is unpractical for modeling power amplifiers in real time applications. This reason motivated researchers to investigate special cases of Volterra series. The Wiener model, the Hammerstein model, and the Wiener-Hammerstein model are included in the category of special cases of Volterra series for modeling nonlinear power amplifiers.

The Wiener model consists of a linear time invariant (LTI) system followed by a memory-less nonlinearity as illustrated in Fig. 2.2.

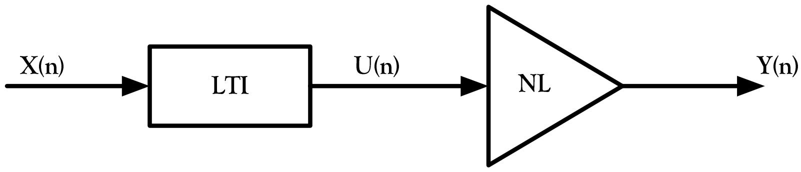


Fig. 2.2: Wiener Model

The model is given by the following mathematical formulas. The output of the LTI system is given as

$$u(n) = \sum_{q=0}^{Q} h_q \, x(n-q) \tag{2.8a}$$

The output of the static nonlinear (NL) block is given as

$$y(n) = \sum_{k=1}^{K} b_{2k-1} \, u(n)|u(n)|^{2(k-1)} \tag{2.8b}$$

Inserting (2.8a) in to (2.8b) gives

$$y(n) = \sum_{q=1}^{Q} h_q \sum_{k=1}^{K} b_{2k-1}|x(n-q)|^{2(k-1)} x(n-q) \tag{2.8c}$$

The Hammerstein model is a memory-less nonlinearity followed by an LTI, which is clear from the Fig. 2.3

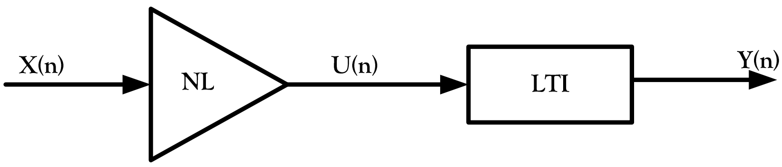


Fig. 2.3. Hammerstein model

The model is represented by the following equations.

$$u(n) = \sum_{k=1}^{K} b_k\, x(n)|x(n)|^{k-1} \tag{2.9a}$$

$$y(n) = u(n) * h(q) \tag{2.9b}$$

$$y(n) = \sum_{k=1}^{K}\sum_{q=0}^{Q} b_k\; h(q)x(n-q)|x(n-q)|^{k-1} \tag{2.9c}$$

The Wiener-Hammerstein (W-H) model consists of an LTI system followed by a memory-less nonlinearity, which is in turn followed by another LTI system. The Wiener-Hammerstein model is shown in Fig. 2.4

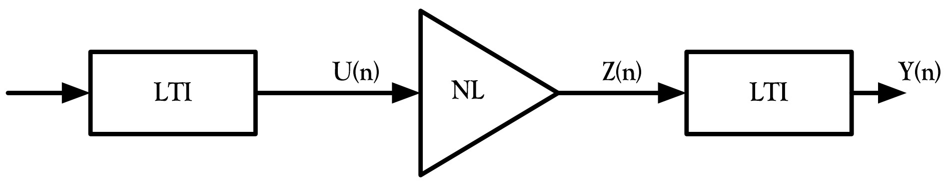


Fig. 2.4: Wiener-Hammerstein Model

The output is modeled by [2]

$$u(n) = \sum_{q1=0}^{Q1} a_{q1}\, x(n-q1), \tag{2.10a}$$

$$z(n) = \sum_{k=1}^{K} b_{2k-1}\, u(n)|u(n)|^{2(k-1)} \tag{2.10b}$$

$$y(n) = \sum_{q1=0}^{Q1} h_{q1} \sum_{k=0}^{K} b_{2k-1} \left[\sum_{q2=0}^{Q2} a_{q2} x(n - q_1 - q_2) \right] \left| \sum_{q2=0}^{Q2} a_{q2} x(n - q_1 - q_2) \right| \quad (2.10c)$$

where

Q_1 and Q_2 are parameters to specify memory depths of LTI.

2.2.3.3 The Parallel-Hammerstein model

The Parallel-Hammerstein is an extension of the standard Hammerstein model. The model is illustrated in Fig. 2.5. The system in this case is modeled by [2][3]

$$y(n) = \sum_{k=1}^{K} H_{2k-1}(q) |x(n)|^{2(k-1)} x(n) \quad (2.11a)$$

$$= \sum_{q=0}^{Q} \sum_{k=1}^{K} a_{2k-1,q} \, |x(n-q)|^{2(k-1)} x(n-q) \quad (2.11b)$$

where

$x(n)$ is the input complex base-band signal.

$y(n)$ is the output complex base-band signal.

$H_k(q)$ is the transfer function of the filter for the kth polynomial contribution.

$a_{k,q}$ are complex valued parameters.

Q is the memory depth.

K is the order of the polynomial.

The main difference between the Parallel-Hammerstein and the standard Hammerstein models is that in the Parallel-Hammerstein model, different static nonlinear orders are filtered with different LTI systems. For example, the first term of the polynomial is filtered with $H_1(q)$, and the 2nd odd power term i.e. $x|x|^2$ is filtered with $H_3(q)$ and so on.

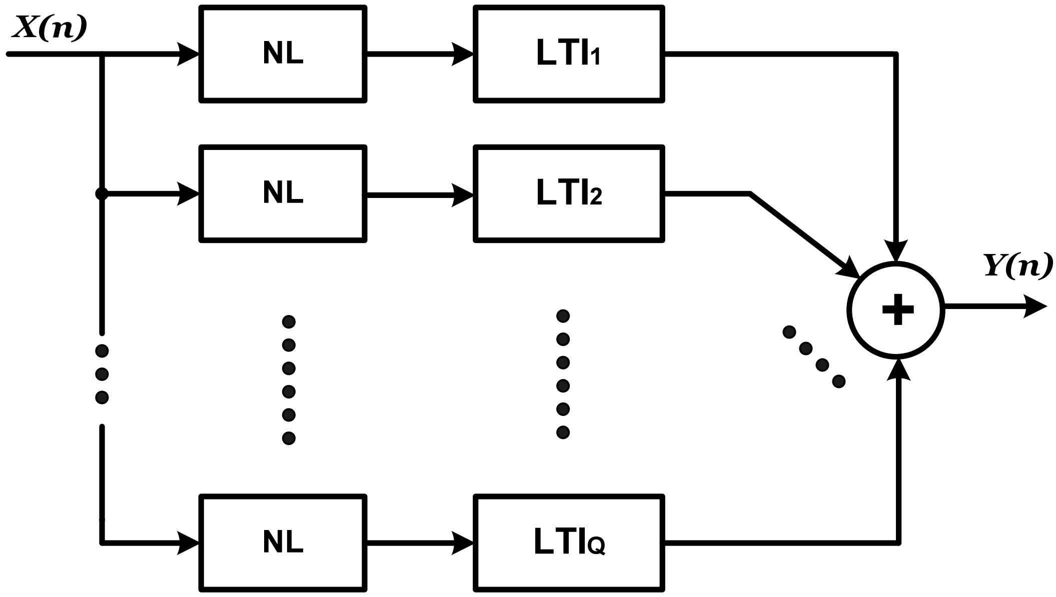


Fig.2.5 Block diagram for the Parallel- Hammerstein

2.2.3.4 The Memory- Polynomial Model

The memory-polynomial model, [1] consists of several delay taps and nonlinear static functions. This model is a truncation of the general Volterra series, which consists of only the diagonal terms in the Volterra kernels. Thus, the number of parameters is significantly reduced compared to general Volterra series. The model is shown in Fig. 2.6

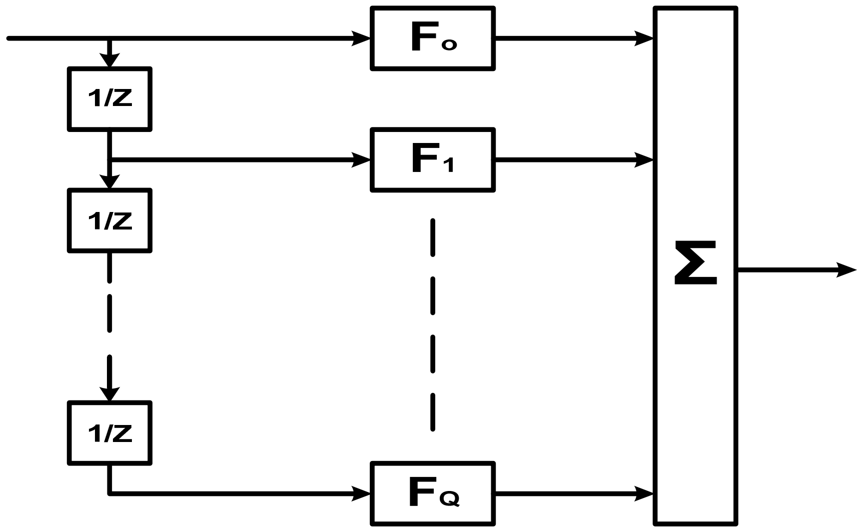


Fig. 2.6. The memory-polynomial model.

A Memory-polynomial model considering memory effects and nonlinearity is given by the following equation

$$y(n) = \sum_{q=0}^{Q} \sum_{k=1}^{K} a_{2k-1,q} \, |x(n-q)|^{2(k-1)} x(n-q) \tag{2.12}$$

where

$x(n)$ is the input complex base-band signal.

$y(n)$ is the output complex base-band signal.

$a_{k,q}$ are complex valued parameters.

Q is the memory depth.

K is the order of the polynomial.

This model considers only odd-order nonlinear terms, because the even-order terms are usually outside of the operational bandwidth of the signal and can be easily filtered out. This model considers polynomials with orders up to $2k-1$, where K is a design parameter.

2.2.4 Comparisons between nonlinear models with memory

The Volterra series is the most general model and is the most accurate one. However, for the Volterra series to be accurate enough, the number of parameters needed increases dramatically. This motivates the use of subsets of the Volterra series. The most popular subsets are the Memory-polynomial, Hammerstein, Wiener, Parallel-Hammerstein and Parallel-Wiener models.

Comparing the memory-polynomial model (2.12) with the Hammerstein model (2.9), it can be observed that the Hammerstein model is a special case of the Memory-polynomial model when only the odd polynomial terms are considered for the nonlinearity of the Hammerstein model and

$$a_{2k-1,q} = c_k h(q) \tag{2.13}$$

When comparing the memory-polynomial model (2.12) with the Parallel-Hammerstein (2.11**),** it is observed that the Memory-polynomial is equivalent to the Parallel Hammerstein model. It can also be shown that the Memory-polynomial model is a special case of the Parallel-Wiener model [2].

In summary, when considering polynomial type of nonlinearities, both the Parallel -Wiener and the Parallel-Hammerstein models are special cases of the Volterra series. In fact, it can be shown that the memory-polynomial model is equivalent to the Parallel-Hammerstein model. It can be shown [2] that a memory- polynomial model is special a case of the Parallel-Wiener model. Obviously, the Parallel-Hammerstein model includes the Hammerstein model as a special case, and the

Parallel- Wiener model includes the Wiener model as a special case. Hammerstein and Wiener models are the most specialized with the least number of coefficients, but are by no means the easiest to identify. The memory- polynomial model, however, offers a good compromise between generality and ease of parameter estimation and implementation [1] [2].

2.3 Memory-Polynomial Model

The memory-polynomial model discussed in the previous Section is implemented in Simulink®. Simulink® is a platform for multi-domain simulation and model-based design for dynamic systems. It provides an interactive graphical environment and customizable set of block libraries, and can be extended for specialized applications [6]. Simulink® was chosen because it is easy for implementing system level models compared to Matlab. Systems implemented in Simulink® can be easily modified and upgraded with minimum coding. It provides a user friendly environment and interface.

2.3.1 Model's implementation

As shown in Section 2.2.4 memory-polynomial provides a good tradeoff between accuracy and complexity. A memory-polynomial system can be expressed as follows

$$y(n) = \sum_{q=0}^{Q} \sum_{k=1}^{K} a_{2k-1,q} \, |x(n-q)|^{2(k-1)} x(n-q) \tag{2.14}$$

This equation can be rewritten as follows

$$y(n) = \sum_{q=0}^{Q} F_q(n-q) \tag{2.15}$$

$$\begin{aligned} &= F_0(n) + F_1(n-1) + F_2(n-2) + \ldots + F_q(n-q) \\ &\quad + \ldots . + F_Q(n-Q) \end{aligned} \tag{2.16}$$

where

$$F_q(n) = \sum_{k=1}^{K} a_{2k-1,q} |x(n)|^{2(k-1)}\, x(n) \tag{2.17}$$

which means that

$$\begin{aligned} F_q(n) = \; & a_{1,q}x(n) + a_{3,q}|x(n)|^2 x(n) + a_{5,q}|x(n)|^4 x(n) + \ldots \\ & + a_{2k-1,q}|x(n)|^{2(k-1)} x(n) + \ldots \\ & + a_{2K-1,q}|x(n)|^{2(K-1)} \end{aligned} \tag{2.18}$$

A block diagram is shown in Fig. 2.7 and Fig. 2.8

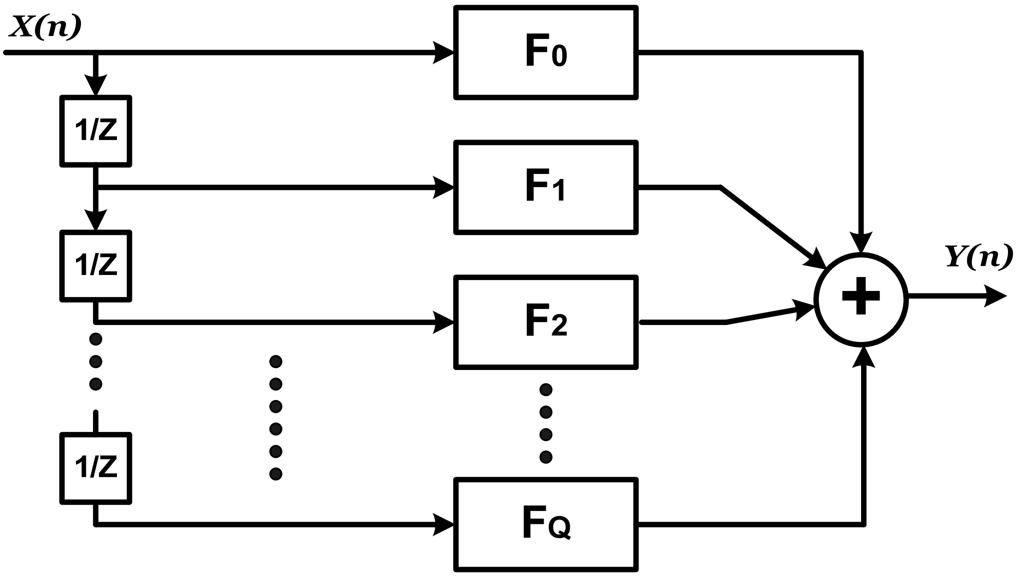


Fig.2.7. Implementation of equation 2.19

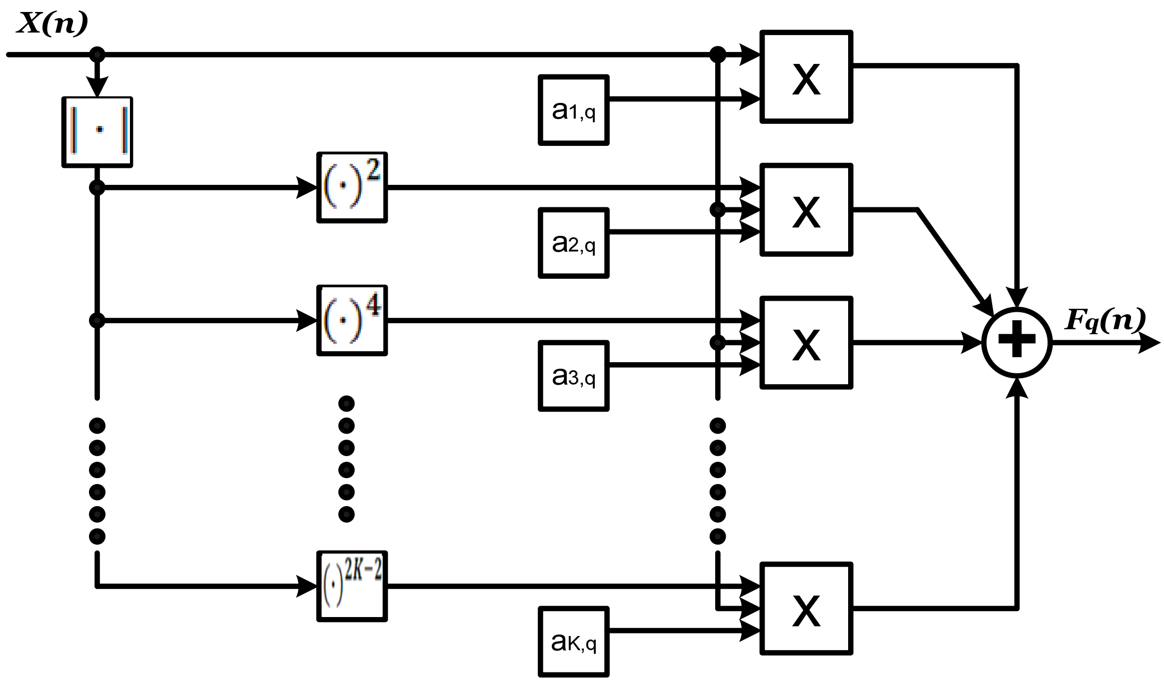


Fig. 2.8 Implementation of (2.17)

All of these bocks are available in Simulink® and the implementation is straightforward.

2.3.2 Model's Identification

Identification of the parameters of the memory-polynomial is very easy compared to other models. The least square error (LSE) technique is used to find the coefficients from measurement data. The process can be explained as follows [1].

To identify the coefficients, **Y** and **H** matrices are first defined as follow

$$\boldsymbol{Y} = [y(n) \quad y(n+1) \quad \cdots \quad y(n+N-1)]^T \tag{2.19}$$

and

$$\boldsymbol{H} = [\boldsymbol{H_0} \quad \cdots \quad \boldsymbol{H_q} \quad \cdots \quad \boldsymbol{H_Q}] \tag{2.20}$$

where

$y(n)$ is the output measured data elements. Baseband data is used.

N is the size of measured data set.

and

$$\boldsymbol{H_q} = \begin{bmatrix} h_{1,q}(n) & h_{3,q}(n) & \cdots & h_{2k-1,q}(n) \\ h_{1,q}(n+1) & h_{3,q}(n+1) & \cdots & h_{2k-1,q}(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ h_{1,q}(n+N-1) & h_{3,q}(n+N-1) & \cdots & h_{2K-1,q}(n+N-1) \end{bmatrix} \tag{2.21}$$

so

$$h_{2k-1,q}(n) = |x(n-q)|^{2(k-1)} x(n-q) \tag{2.22}$$

Let the complex coefficients be represented as follows:

$$\boldsymbol{a} = [\boldsymbol{a_0} \quad \cdots \quad \boldsymbol{a_q} \quad \cdots \quad \boldsymbol{a_Q}]^T \tag{2.23}$$

where

$$\boldsymbol{a_q} = [a_{1,q} \quad a_{3,q} \quad \cdots \quad a_{2k-1,q}] \tag{2.24}$$

Then the following matrix equation holds

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{a} \tag{2.25}$$

If $\hat{\boldsymbol{a}}$ is the estimated parameter matrix then to have minimum RMS error between the measured and simulated output $\hat{\boldsymbol{a}}$ can be calculated from the following equations

Since

$$(\mathbf{H}\,\mathbf{H}^*)(\mathbf{H}^*\,\mathbf{H})^{-1} = \mathbf{I} \tag{2.26}$$

where

$\mathbf{H}^*$ is the conjugate transpose of $\mathbf{H}$, also known as the *Hermitian* transpose or the *adjoint* matrix.

Then (2.25) can be rewritten as

$$(\mathbf{H}\,\mathbf{H}^*)(\mathbf{H}^*\,\mathbf{H})^{-1}\mathbf{Y} = \mathbf{H}\,\mathbf{a} \tag{2.27}$$

Then $\boldsymbol{a}$ can be approximated according to the least square criterion by $\hat{\boldsymbol{a}}$ as follows

$$\hat{\boldsymbol{a}} = \mathbf{H}^*(\mathbf{H}^*\,\mathbf{H})^{-1}\mathbf{Y} \tag{2.28}$$

where

$$\hat{\boldsymbol{a}} = [\hat{\boldsymbol{a}}_{\mathbf{0}} \quad \cdots \quad \hat{\boldsymbol{a}}_{\boldsymbol{q}} \quad \cdots \quad \hat{\boldsymbol{a}}_{\boldsymbol{Q}}]^T \tag{2.29}$$

and

$$\mathbf{H}^+ = \mathbf{H}^*(\mathbf{H}^*\,\mathbf{H})^{-1} \tag{2.30}$$

where

$\mathbf{H}^+$ is the pseudo inverse matrix of $\mathbf{H}$ defined above.

then

$$\hat{\boldsymbol{a}} = \mathbf{H}^{+}\mathbf{Y} \tag{2.31}$$

The simulated output can be calculated from the input and estimated parameters as follows:

$$\widehat{\boldsymbol{Y}} = [\hat{y}(n) \quad \hat{y}(n+1) \quad \cdots \quad \hat{y}(n+N-1)]^{T} \tag{2.32}$$

$$= \boldsymbol{H}\hat{\boldsymbol{a}} = \sum_{q=0}^{Q} \boldsymbol{H_q}\,\hat{\boldsymbol{a}}_{\boldsymbol{q}} \tag{2.33}$$

Then the error between measured and simulated data can be defined as

$$\boldsymbol{E} = \boldsymbol{Y} - \widehat{\boldsymbol{Y}} = [e(n) \quad e(n+1) \quad \cdots \quad e(n+N-1)]^{T} \tag{2.34}$$

The least square estimate is constructed in such a way that $\|\boldsymbol{E}\|^2$ is minimized.

3 Low Noise Amplifier

3.1 Introduction

Low Noise Amplifier (LNA) is the first amplifier in the RF receiver frontend; typically it is the first or second component after the antenna. It is designed to increase the power of the received signal which is usually very weak (could be as weak as -200dBm). LNAs are designed to add as little noise as possible, such that the signal to noise ratio (SNR) stays above the minimum required SNR of the receiver. The SNR is defined as the ratio between the wanted signal and the noise and is usually specified in dBs. Every receiver has a minimum SNR at its input, if the SNR drops below this value, the error in the received signal will be high. Another important performance figure for RF receivers is the noise factor F which is defined as the SNR_{in}/SNR_{out}. It is a measure of how much noise the receiver will add. If F is measured in dBs instead of linear scale, it will be known as the noise factor. The notation F is usually used for both noise figure and noise factor. For more explanation of F see section 3.2.4.

Ideally, an LNA should introduce a linear gain and no noise to the received signal. However, in reality electronics devices are inherently nonlinear and rather noisy.

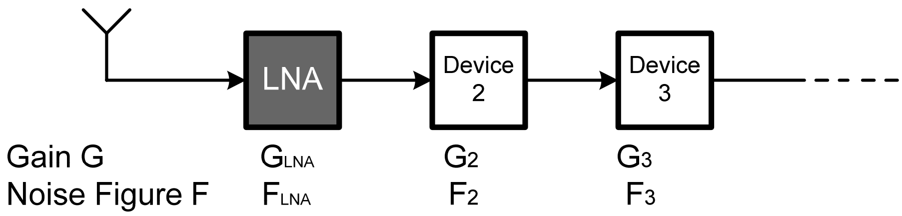


Fig. 3.1: Noise figure calculation for receiver front end

$$F_N = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \cdots \quad (3.1)$$

From (3.1) and Fig. 3.1, it is clear that the noise figure of the overall network depends mainly on the noise figure of the first component with high gain. This is typically the LNA. This implies that an accurate modeling of the noise added by the LNA is crucial for a good receiver model. Another important impairment that should be included for the modeling of LNA is the nonlinearity introduced by the solid state transistors. Mismatching effect is also an important impairment which causes an increase in the bit error rate and needs to be considered for accurate LNA modeling [7].

In the following Sections modeling of the above mentioned impairments in Matlab RF block set is presented.

3.2 LNA's Model in RF Blockset®

RF Blockset® extends Simulink® with a library of blocks for modeling RF systems that include RF filters, transmission lines, amplifiers, and mixers. During the simulation, all blocks are modeled using a time-domain, complex base-band representation. Modeling in complex base-band results in a faster simulation. RF Blockset® is developed mainly to provide an executable specification for RF circuits and it provides an overall system specification for the hardware design of RF components. It also enables the optimization of the overall wireless communication systems at the system level [16].

RF Blockset® has two libraries; mathematical and physical. The mathematical library is just like Simulink®, in fact it is completely compatible with other Simulink® blocks. The main limitation for this mathematical library is that it does not model mismatching effects; it assumes that the characteristic impedance is one ohm and there is no reflection due to mismatching. To overcome this limitation the physical library is implemented. This library uses s-parameters to model mismatching effects in the frequency domain.

To connect blocks from physical library to mathematical library, two special blocks are available in RF Blockset®. These are the input port and output port blocks Fig. 3.2.

3.2.1 Input port and Output port Blocks

Input port and output port blocks provide connection between the physical and mathematical environments. It can be noticed from Fig. 3.2 that the connection between the physical blocks is bi-directional to highlight the fact of the incident and reflected waves. They also provide an interface to enter a new set of simulation parameters. Although the physical environment seems like it is working in the frequency domain, RF Blockset® actually builds an equivalent model in the complex base-band domain. This model is built dynamically, i.e. at run time rather than design time. It then connects this system to Simulink® and runs the simulation. This procedure is explained in the following paragraph.

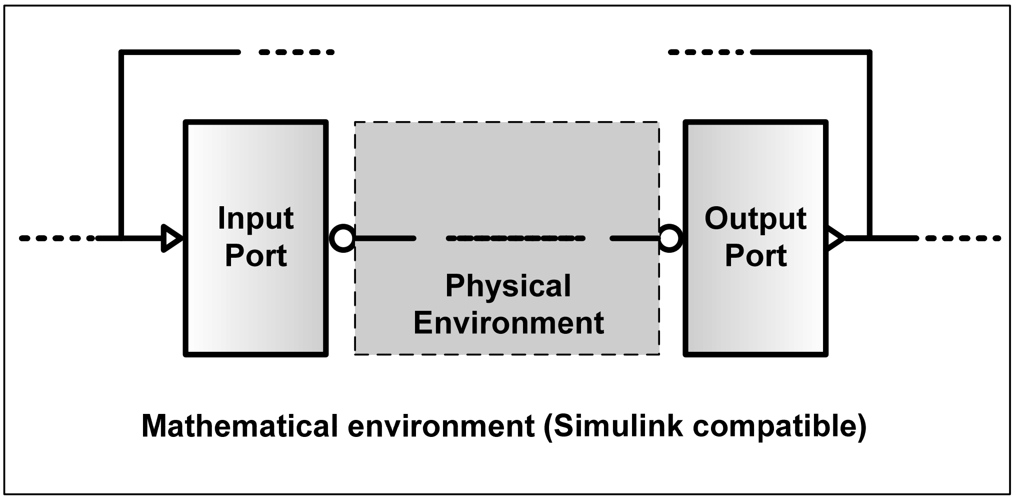


Fig. 3.2 Schematic illustration of the physical and mathematical environments in RF Blockset®

The output port block produces the base-band equivalent time-domain response of an input traveling though series of physical components. The output ports block:

1. Partitions the RF physical components into linear and nonlinear subsystems.
2. Extracts the complex impulse response of the linear subsystems for base-band equivalent modeling of RF linear system.
3. Extracts the nonlinear AM/AM and AM/PM modeling of RF nonlinearity.

As shown in Fig. 3.3, a nonlinear subsystem is implemented by AM/AM and AM/PM nonlinear models built from the nonlinear parameters specified in the physical blocks between the input and output blocks. If a mixer block is included with valid phase noise data then phase noise will be added as shown in the figure. The nonlinear subsystem is further explained in Section 3.2.5.2, and the phase noise modeling is further explained in chapter 4.

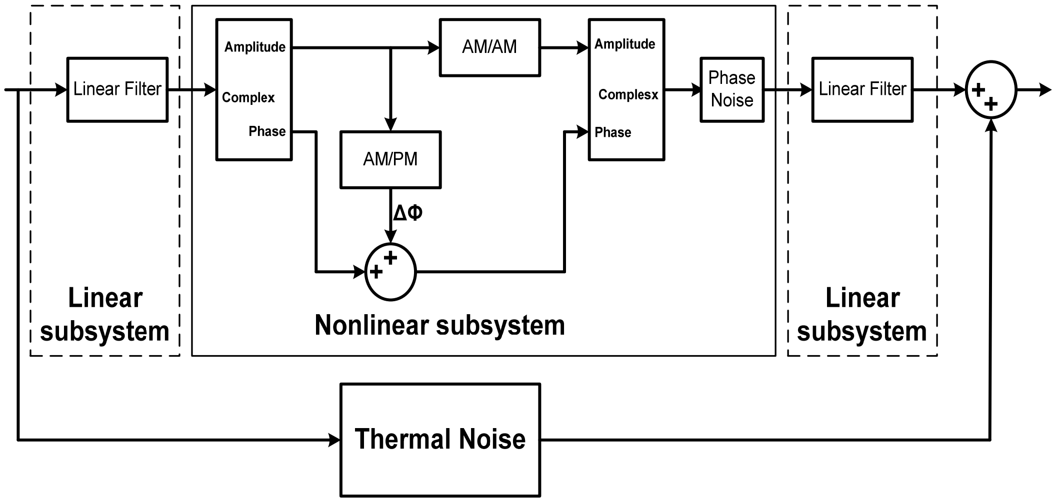


Fig. 3.3 base band complex modeling of physical block between input and output ports

To simulate a model that contains physical blocks, RF Blockset® determines the modeling frequencies of the physical system using parameters in the Input Port block. The modeling frequencies are the frequencies at which the information is taken from the blocks to construct the base-band equivalent model. Then the RF Blockset® determines the block parameter values at those frequencies and uses the information to create a base-band equivalent model for time-domain simulation in Simulink. At the time of simulation output port block uses Input port block parameters to determine the modeling frequencies. These frequencies are in form of vector of N element, where N is Finite impulse response filter length parameter in input port block. The modeling frequencies are a function of the center frequency f_c, Finite impulse response filter length N and the sample time t_s. This process is illustrated in Fig 3.4. [6]

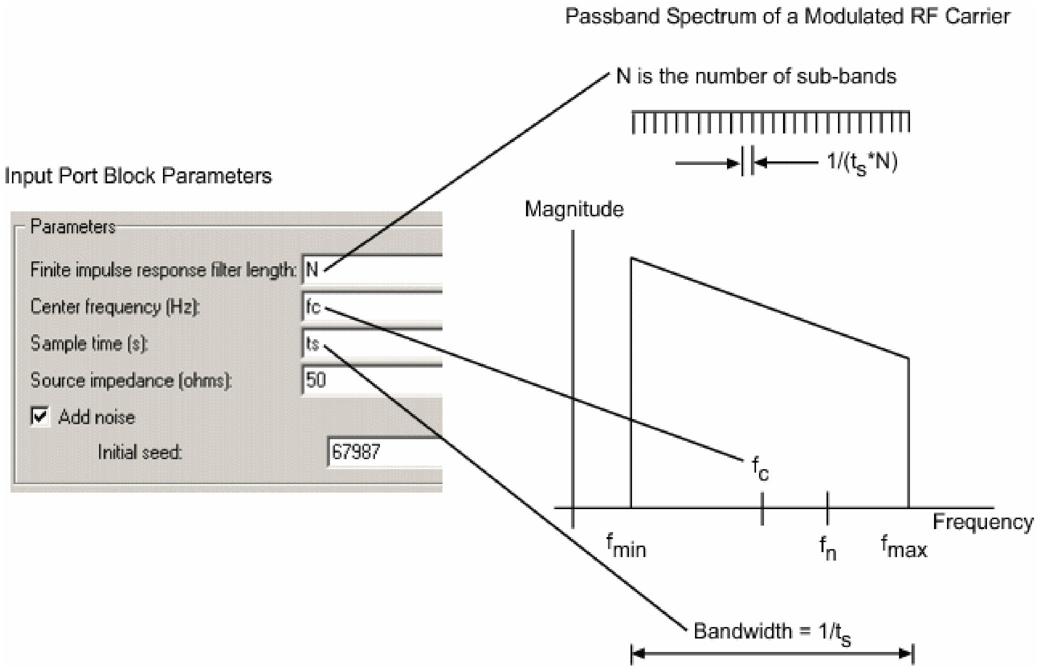


Fig. 3.4 Frequency bandwidth calculation in RF Blockset® [6]

The transfer function is calculated at Pass-band for the frequency calculated and is given by the following equation

$$H(f) = \frac{V_L(f)}{V_S(f)} \tag{3.2}$$

Where V_S and V_L are the source and load voltages, and f represents the modeling frequencies. More specifically

$$H(f) = \frac{S_{21}(1+\Gamma_l)(1-\Gamma_s)}{2(1-S_{22}\Gamma_l)(1-\Gamma_{in}\Gamma_s)} \tag{3.3}$$

where

$$\Gamma_l = \frac{Z_l - Z_0}{Z_l + Z_0}$$

$$\Gamma_s = \frac{Z_s - Z_0}{Z_s + Z_0}$$

$$\Gamma_{in} = S_{11} + \left(S_{12}S_{21}\frac{\Gamma_l}{1 - S_{22}\Gamma_l}\right)$$

and

- Z_s is the source impedance
- Z_l is the load impedance
- S_{ij} are the S-parameters of a two port network

After calculating the transfer function it is transferred to base-band as follows

$$H_{BB}(f) = H_{PB}(f - f_c) \tag{3.4}$$

where

BB is for Base-Band

PB is for Pass-band

Then the base-band impulse response is calculated as follow

$$h_{BB} = \boldsymbol{\mathcal{F}}^{-1}\{H_{BB}(f)\} \tag{3.5}$$

This impulse response is truncated to a length equal to the filter length ***N*** specified in the input port. The base-band transfer function is shown in Fig 3.5 [16].

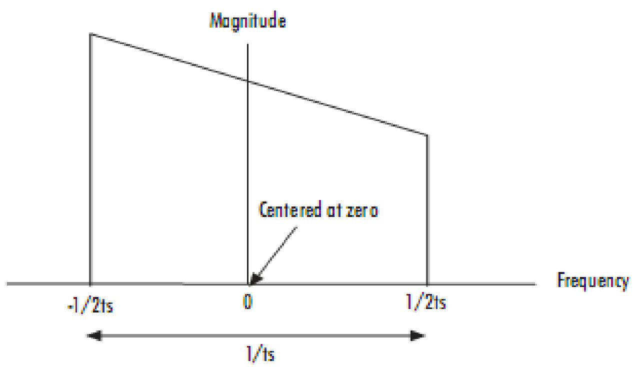


Fig.3.5. Base-band equivalent spectrum

3.2.2 Mismatch's effects modeling in RF Blockset®

RF Blockset® models mismatch using the input s-parameter data, frequency data and characteristic impedance of the amplifier. S-parameter data is first interpolated and extrapolated to the simulation frequency and then used to calculate the incident and reflected waves. This process is explained in the following Section.

3.2.3 Scattering matrix

The scattering matrix (s-matrix) for a certain network describes how an incident wave (or voltage signal) on one port of the network will scatter and be distributed among other ports, Fig. 3.6. An s matrix can be constructed for networks with any number of ports, but since two port networks are the only ones considered in this thesis, only two port S matrices are addressed. Suppose a two port network is assumed with incident and reflected voltage waves V_n^+ and V_n^-respectively, where n is an integer referring to the port number. Then a 2-by-2 s-matrix for this network is written as follow

$$V_1^- = S_{11}V_1^+ + S_{12}V_2^+ \tag{3.6}$$

$$V_2^- = S_{21}V_1^+ + S_{22}V_2^+ \tag{3.7}$$

Or in matrix form

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix} \tag{3.8}$$

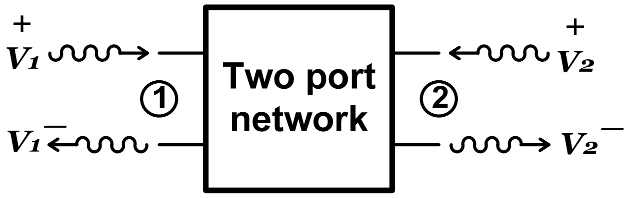


Fig. 3.6 Illustration of S-matrix for a 2-port network

It can be shown that the S matrix is equivalent to other networking matrices such as Z, Y and ABCD , in fact transformation formulas between these matrices do exist [8]. Formulas also exist for combining one or more networks either in cascade or in parallel [8]. The S-matrix is a linear characterization of networks. It encapsulates the voltage/current characteristics of networks linearly (Ohm's law), S-matrices are very useful in RF circuits engineering because they describe the way power flows in the system and enable to optimize for maximum power transfers. The S-parameters are dependent on frequency because they represent networks with frequency dependant components such as inductors and capacitors. Therefore S-matrices are measured for a range of frequencies, and in simulation time after determining the center frequency and bandwidth, the correct set of S-parameters is used.

3.2.4 Modeling of Noise in RF Blockset®

There are three ways to model noise in RF physical blocks. The accuracy of modeling depends on the availability of the noise data. These three different ways are as follow

1. The simplest way is to specify noise figure, noise factor or noise temperature which are equivalent as shown in the following equations

Noise factor (***F***) is defined as follows [7]

$$F = \frac{SNR_{input}}{SNR_{output}} \tag{3.9}$$

Noise figure ***NF*** is the noise factor specified in ***dB***

$$NF = 10\log_{10}(F) \tag{3.10}$$

Noise temperature T_e can be related to noise figure as

$$T_e = (F-1)T_o \tag{3.11}$$

where T_0 is the standard room temperature in Kelvin (290 K)

2. A more accurate way is to specify the noise figure as a function of frequency for the whole frequency range as noise depends upon the operational frequency bandwidth.

3. The third way is to specify the noise data as the Minimum noise figure (NF_{min}), equivalent noise resistance R_n and optimal source admittance Y_{opt}. The RF Blockset® will calculate the noise figure from the noise correlation matrix C_A as follows [19]

$$C_A = 2kT_e \begin{bmatrix} R_n & \frac{NF_{min}-1}{2} - R_n Y_{opt}^* \\ \frac{NF_{min}-1}{2} - R_n Y_{opt} & R_n |Y_{opt}|^2 \end{bmatrix} \tag{3.12}$$

where

K is the Boltzmann's constant

T_e is the noise temperature in Kelvin

The noise factor F is calculated from the correlation matrix as follows

$$F = 1 + \frac{z^* C_A z}{2kT_e Re\{Z_s\}} \tag{3.13}$$

$$z = \begin{bmatrix} 1 \\ Z_s^* \end{bmatrix} \tag{3.14}$$

where

Z_s is the nominal impedance.

z^* is the Herrmitian conjugation of z.

Once the noise figure of each physical block is calculated for all simulation frequencies using one of the above methods, the overall noise figure is calculated from (3.1). The noise power for the whole system is calculated from the following formula

$$Noise\ Power = kT_eB \tag{3.15}$$

where $\boldsymbol{B}$ is the simulation bandwidth

The noise is added to the system as shown in Fig. 3.3.

3.2.5 Modeling of Nonlinearity of LNA

Due to the nonlinearity of solid state transistors, LNA has nonlinear effects but since the power level of the input signal is typically small, memory effects on the output signal are not severe and can be safely ignored. Therefore the static nonlinear model included in the RF Blockset® set is used for the modeling of nonlinearity in LNA.

3.2.5.1 Modeling of nonlinearity in RF Blockset®

The amplifier block in RF Blockset®'s physical library models the nonlinearity of the LNA as follows

- If AM-AM and AM-PM data exists (for example in .AMP file) then AMAM/AMPM nonlinearities are extracted from this data, and the IP3 and the1dB gain compression power are easily extracted from this data.
- If no AM-AM and AM-PM data are included in the source file then the nonlinearities are determined by specifying OIP3 (or IIP3) and the 1dB gain compression point. If however the 1dB gain compression point is not specified then nonlinearity will be computed as follow [16]
    1. Convert the specified value into IIP3 (if needed).
    2. Convert the IIP3 value from decibels to linear units.
    3. Compute a scaling factor, which is equal to 3 divided by the linear IIP3 value.
    4. Apply the scaling factor.
    5. Limit the scaled input to a maximum value of 1 and apply an AM/AM conversions to the magnitude of the scaled signal according to the following function.

$$F_{AM/AM}(x) = x - \frac{x}{3} \tag{3.16}$$

where

x is the magnitude of the scaled signal.

- If the 1dB compression point is specified, then the nonlinearity is computed as follows
    1. The specified third-order intercept value is converted into OIP3 (if needed)

2. The gain, OIP3, and 1dB compression data are converted to linear, unit less values, normalized to 1 volt and the reference impedance *Z0*.

$$G_0 = 10^{\frac{GAIN}{10dB}} \tag{3.17}$$

$$OIP3_0 = 10^{\frac{OIP3}{10dBm}} 10^{-3} Z_0 \tag{3.18}$$

$$P_0 = 10^{\frac{PCOMP}{10dBm}} 10^{-3} Z_0 \tag{3.19}$$

where

GAIN is the amplifier power gain, which is derived from the network parameters.

OIP3 is the output third-order intercept point.

PCOMP is the output power at the 1 dB compression point.

3. Compute the coefficients of the polynomial $F_{AM/AM}(x) = c_1 x + c_3 |x|^2 x + c_5 |x|^4 x$ that determines the AM/AM conversions of the input signal x.

$$c_1 = \sqrt{G_0} \tag{3.20a}$$

$$c_3 = -\frac{c_1^3}{20(IP3_o)} \tag{3.20b}$$

$$c_5 = \frac{c_1^5}{4P_0^2 . 10^{0.2}} \left[\frac{P_0}{OIP3_0} 10^{0.1} - 10^{-0.05} \right] \tag{3.20c}$$

4. The saturation input power is calculated from the following formula

$$A_{sat} = \sqrt{\frac{3.c_3 + \sqrt{9.c_3^2 - 20.c_1.c_5}}{-10.c_5}} \tag{3.21}$$

5. The AM/AM conversions are applied to the input signal.

4 Mixer modeling

4.1 Introduction

According to [7] a mixer can be defined as "a three port device that uses a nonlinear or time –varying element to achieve frequency conversion." Normally in a wireless communication systems all signal processing is done in base-band because it is easy to process low frequency signals. To be able to transmit through the wireless channel however the signal has to be brought to a higher frequency, which is done by modulation and up conversion in the transmitter, this effect has to be undone in the receiver which corresponds to demodulation and down conversion in the receiver. Up and down conversions are done with mixers.

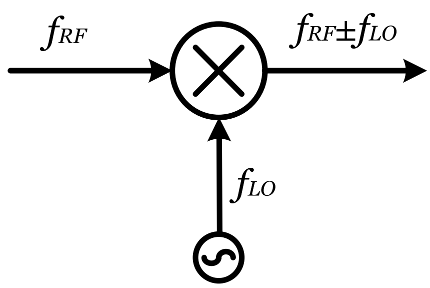


Fig 4.1 A Schematic of a down conversion Mixer.

There are many approaches to model mixers at the system level that depends on which type of impairment the model addresses. Power levels of spurious components are modeled using inter-modulation tables. Nonlinearity which is an inherent phenomenon in mixers is modeled using IIP3 and 1dB compression point as explained in Section 3.2.3. Noise is modeled in the same way as in Section 3.2.2 .The main impairment related to mixers that highly effect the performance of the overall system is the phase noise. Phase noise is explained in following Section together with the algorithm used in RF Blockset® to model it.

4.2 Modeling of Mixers in RF Blockset®

The RF Blockset® provides a complex base-band model for mixers that include phase noise. The oscillator is also included in the mixer block, so the mixer has only input and output ports. The physical mixer model is viewed as a transition to the center frequency of the simulation. If the mixer is set to up convert it will shift the simulation center frequency F_c to $F_c + F_{L0}$ where F_c and F_{L0} are specified by the user and if it is set to down converter its output frequency will be $F_c - F_{L0}$. Other subsequent blocks will then use this new center frequency for choosing the operational S-parameter vector, Fig.4.2

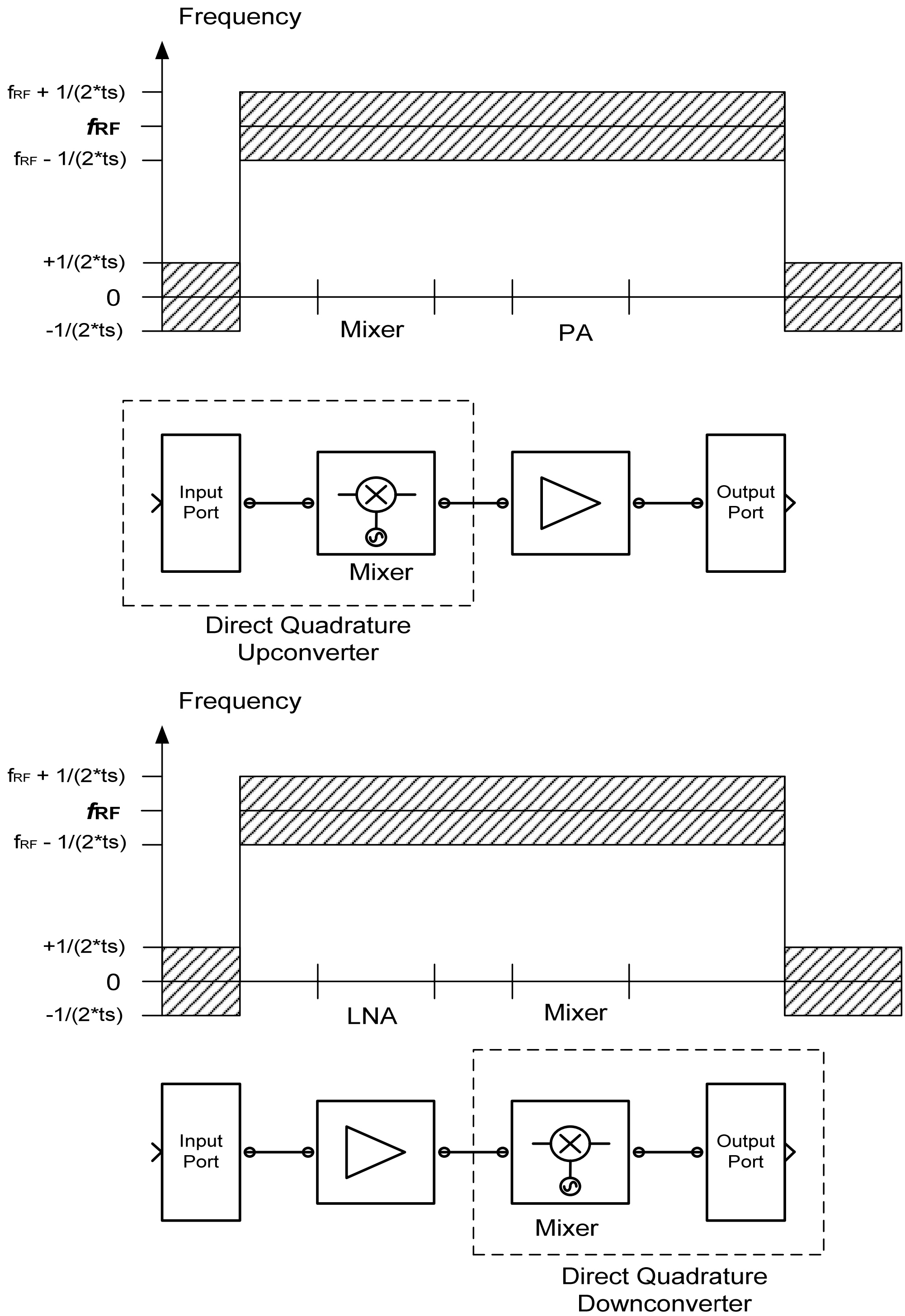


Fig. 4.2 Mixer modeling in RF Blockset® ; up conversion (up) down conversion (down)

4.3 Modeling of Phase Noise

For the mixing process to take place a constant frequency source is needed. The signal of this source is mixed with the input signal to generate up or down converted output signal. The device which produces constant frequency signal for mixers is called oscillator. Ideally an oscillator should produce a pure sinusoidal signal with the desired frequency however in practice pure sinusoidal is not achievable so an actual oscillator signal will encounter both amplitude and frequency fluctuation. The frequency fluctuation is a very serious limitation and can cause severe degradation in performance of the system. This impairment is characterized by "Phase Noise".

Phase noise can be defined as "A short-term random fluctuation in the frequency (Phase) of an oscillator signal" [7]. Phase noise can be modeled with a simple feedback model like, Leeson's Model [15].

In the RF Blockset® however a rather complicated model is used that depends on $1/_{f^{\alpha}}$ power law noise generation [10]. The implementation of which is explained in following Section briefly.

4.3.1 Phase noise modeling in RF Blockset®

The mixer block available in RF Blockset® , adds phase noise in the following order

1. An additive white noise (AWGN) ,which is correlated to the input signal is generated
2. The generated noise is then filtered with a digital filter.

3. The filtered noise is then added to the phase component of the complex base-band input signal
4. This process is demonstrated in Fig. 4.3 [10][16].

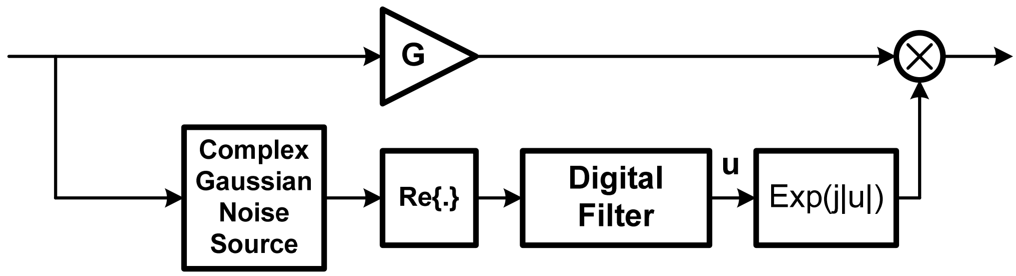


Fig.4.3 phase Noise Modeling in RF Blockset®

5 Analysis of memory-polynomial model

5.1 Introduction

In this chapter the polynomial model explained in [1] and implemented in this thesis is tested and the results are presented and commented. The overall transmitter and receiver were also tested and the results of which are also presented and commented in this chapter.

5.2 Model's analysis

The parameters of the memory-polynomial were extracted from measurement data for a typical WIMAX amplifier[1].

5.2.1 AM/AM and AM/PM Plots

The AM/AM plots are generated by plotting output voltages (or powers) versus input voltages (or powers). AM/PM plots on the other hand are generated by plotting the phase differences between input and output data, versus input voltages (or powers). Only voltage AM/AM and AM/AM are presented in this chapter.

In Fig. 5.1. an AM/AM is plotted for a memory-polynomial model with memory depth $q = 3$ and polynomial of the 5th degree. $n = 3$, on the background a plot for the measured data is also plotted.

[1] This data was provided by the microwave electronic laboratory, in the department of Microtechnology and Nanoscience (MC2), and the communication systems group in the department of signals and systems (S2), Chalmers University of technology. It is for a GaN power amplifier with bandwidth of 3.84 MHz. the measurement was done at 2.1 GHz center frequency and up to 5 times the bandwidth of the amplifier stated above with a maximum power of 15dBm.

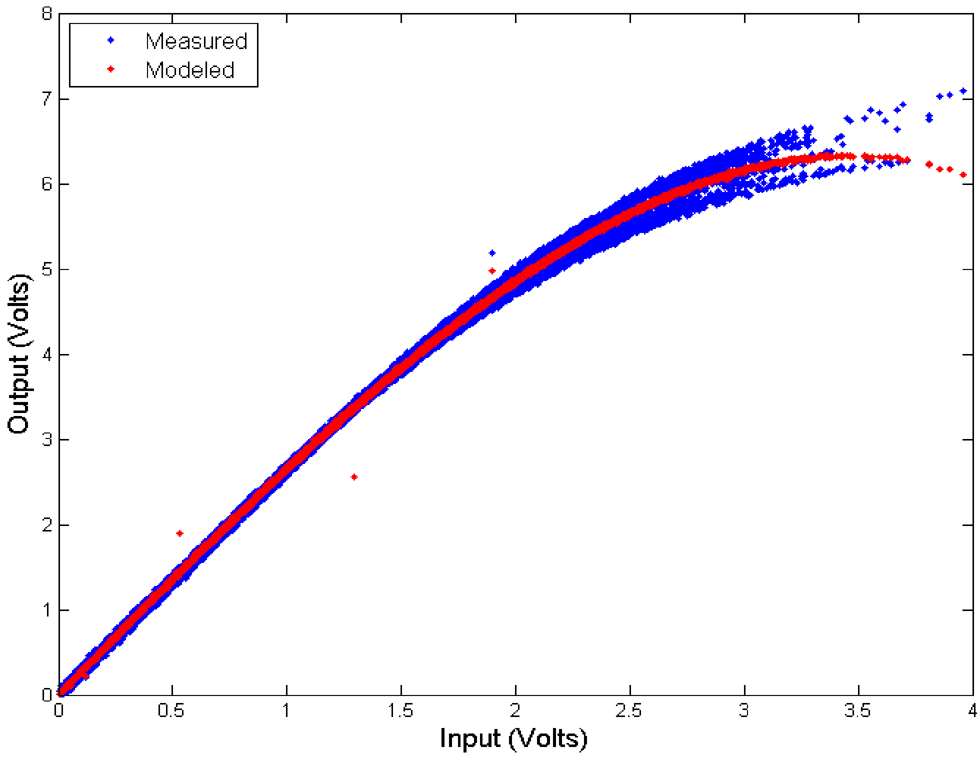


Fig. 5.1. Voltage AM/AM for measured (blue), and modeled (Red) data for $q = 3, n = 3$.

From the figure it is clear that the model fits the data very well at small signals (signals below 2 volts) and it gets less accurate as the operating voltage, or power, increases. That could be a serious limitation for a power amplifier model because it is typically operated in high power levels and the distortion, introduced to the signal, at these power levels are the most important. However, 2 volt is a typically high voltage especially for a mobile device with limited power resources.

The model was investigated along it is two dimensions, which are memory depth q and polynomial order $(2n - 1)$. Fig. 5.2. shows the measured AM/AM data versus the modeled data while the memory depth was kept constant at $q = 0$ and the polynomial orders from the 3rd to the eleventh, that is from $n = 2$ to 6.

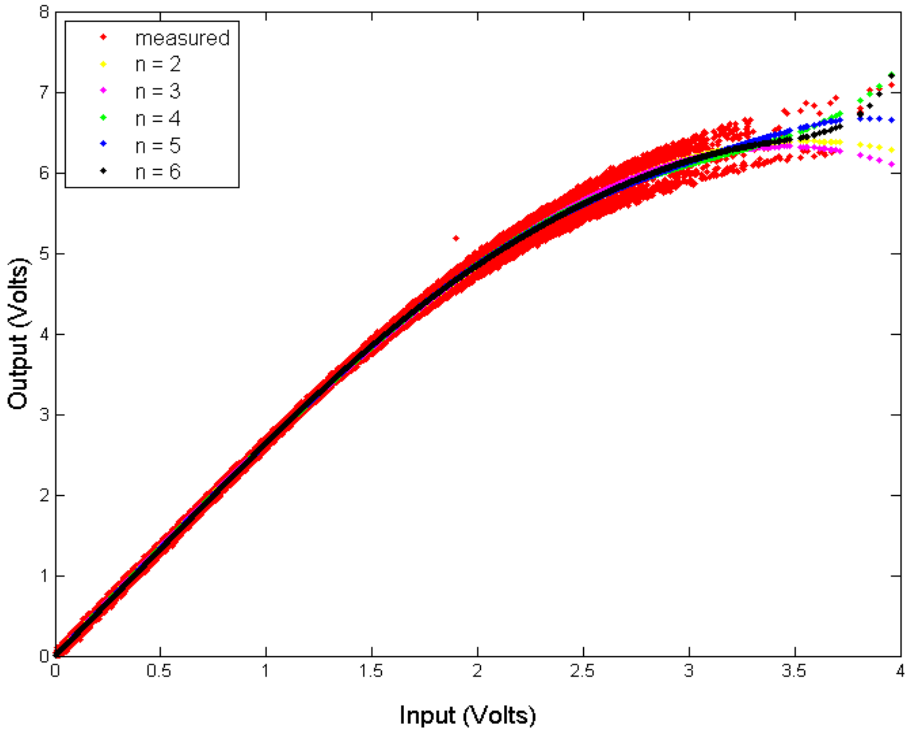


Fig. 5.2. Voltage AM/AM of measured data (red) and modeled data for a memory depth of zero and polynomial order of 3rd (yellow), to eleventh (black).

The order of the polynomial does not affect the accuracy of the model very much as it increased above three [1]. This can also be seen on Fig. 5.2. Therefore the memory order was kept constant at $n = 3$.

The model then was designed with constant polynomial order of three and tunable memory depths that can take any number between zero and six.

It can be concluded from Fig. 5.1 and Fig. 5.2. that the implemented model fits (simulates) the measured amplifier very well if the quality of the data is good.

The memory effect can be observed by plotting the AM/AM for the modeled data for a constant polynomial order of three, and memory depths of zero and six, this is shown in Fig. 5.3. This memory effect can be better observed by applying Gaussian noise data followed by a finite impulse response filter (FIR) and plotting the AM/AM and AM/PM of the model. This is shown in 5.2.1Fig. 5.4. form this figure the spread of the plot is more wild around the memory-less response, this is due to the high signal fluctuation (standard deviation) which triggers memory more rapidly.

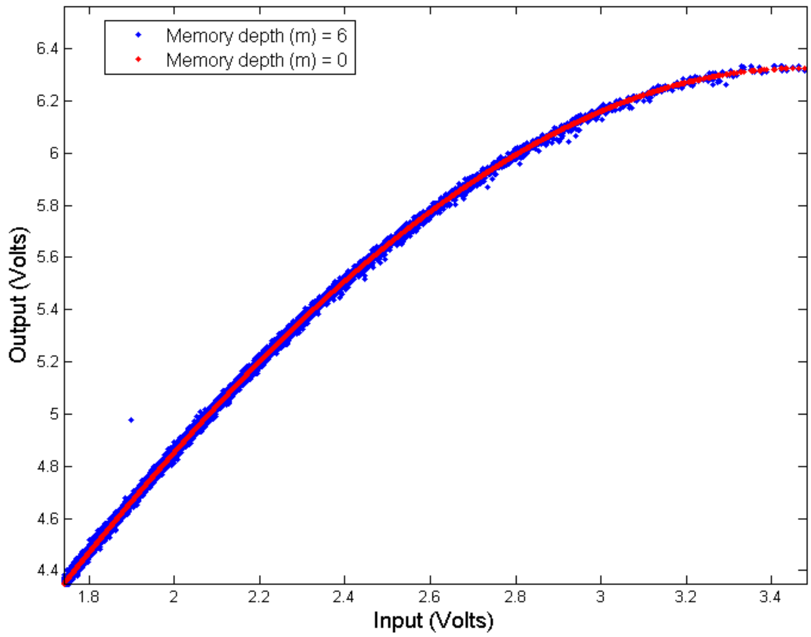
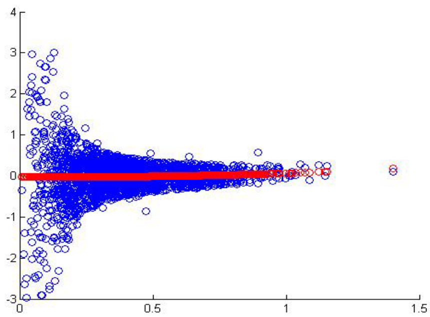
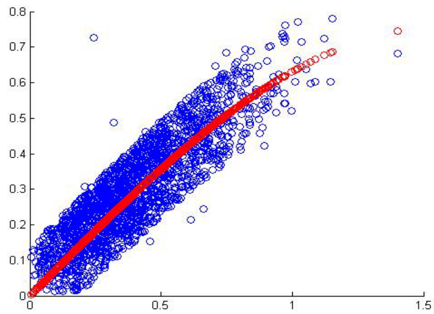


Fig. 5.3. Voltage AM/AM for modeled data for polynomial order 3 and memory depth zero (red) and 6 (blue).



(a)

Fig. 5.4 a) AM/PM response for a Gaussian noise input. The red plots are for a memory-less model and the blue ones are for a model with memory.



(b)

Fig. 5.4 b) AM/AM response for a Gaussian noise input. The red plots are for a memory-less model and the blue ones are for a model with memory.

It is the nature of a polynomial of order n that it has $n-1$ maxima and minima and that it approaches infinity for large positive values, therefore the model is enforced to saturation as shown in Fig. 5.5. The point of saturation was picked after the identification of the model as the point when the slope of the AM-AM curve becomes zero for the first time.

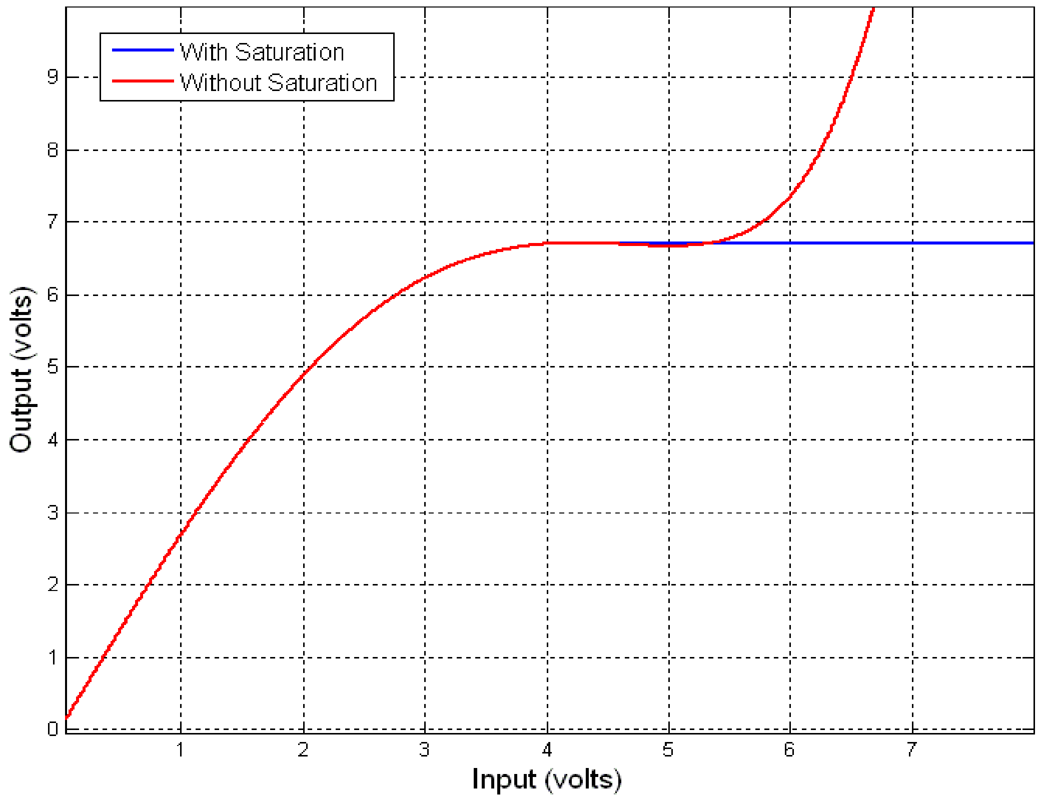


Fig. 5.5. Model response for a ramp input (no memory effect appears).Red: without saturation enforced. Blue: with saturation enforced at maxima.

5.2.2 Multi-tone test

A two tone signal is applied in the form of two pure sinusoidal signals, and the frequency spectrum of the output of the model is plotted in Fig. 5.6. from this figure it is clear that when a memory is introduced in the model an asymmetry between the upper and lower sidebands of the output signal. This asymmetry was claimed to be evidence and a characteristic of systems with memory [3].

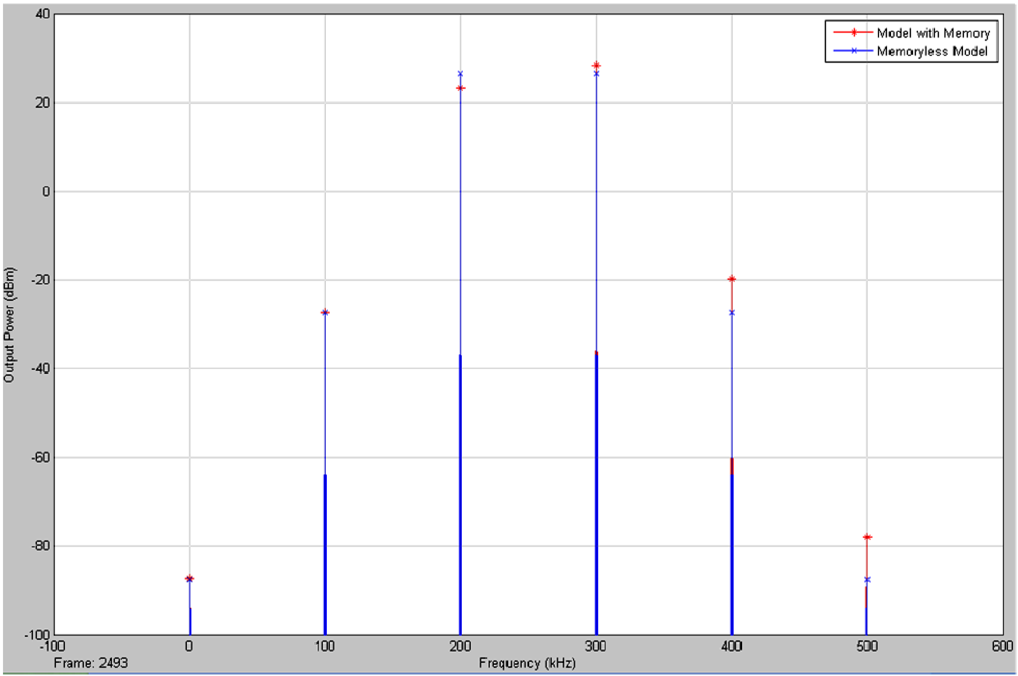


Fig. 5.6. Frequency spectrum of polynomial model output when the input is a two-tone signal and the model has; Red: memory depth six. Blue: memory depth zero.

5.2.3 Response to WCDMA signal

A WCDMA signal was applied to the amplifier and the frequency spectrum of the output is plotted for memory and memory-less models, Fig. 5.7.

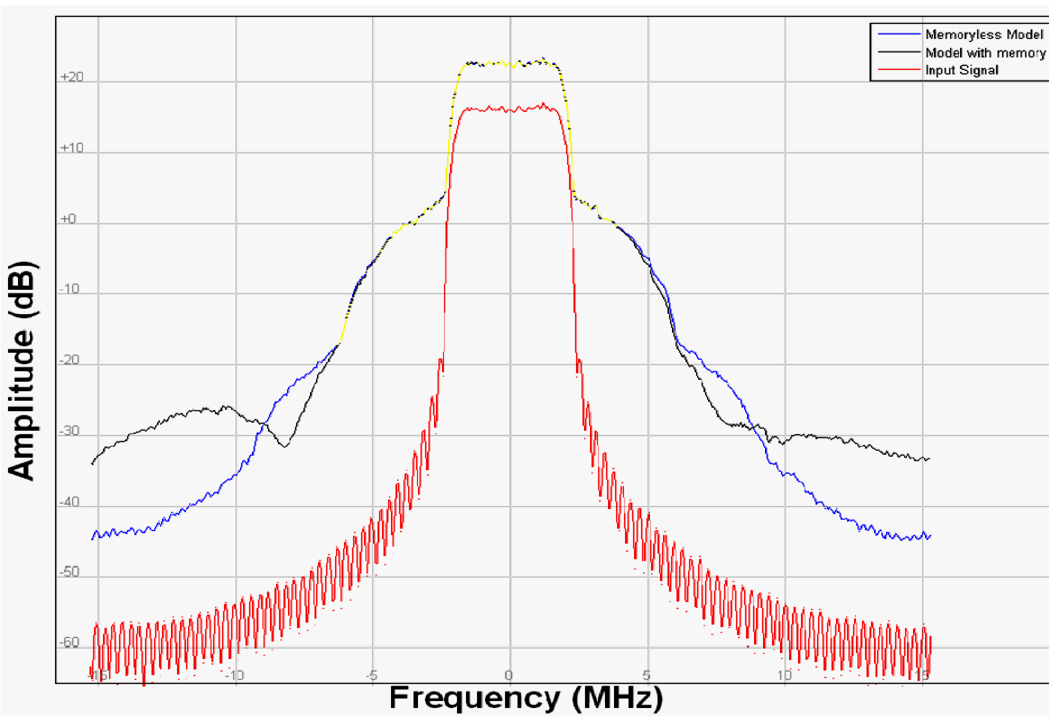


Fig. 5.7. Frequency spectrum for input (red), output of memory-less model (blue) and output of model with memory (black).

The frequency re-growth can be seen clearly in Fig. 5.7. the memory effect, however is not clear but can better be seen by stepping through different values of memory depths, and plot the frequency spectrum in each case Fig. 5.8.

It can be concluded from Fig. 5.8. that as the memory level increases, interference with adjacent channels will become more probable. That complies with theoretical analysis.

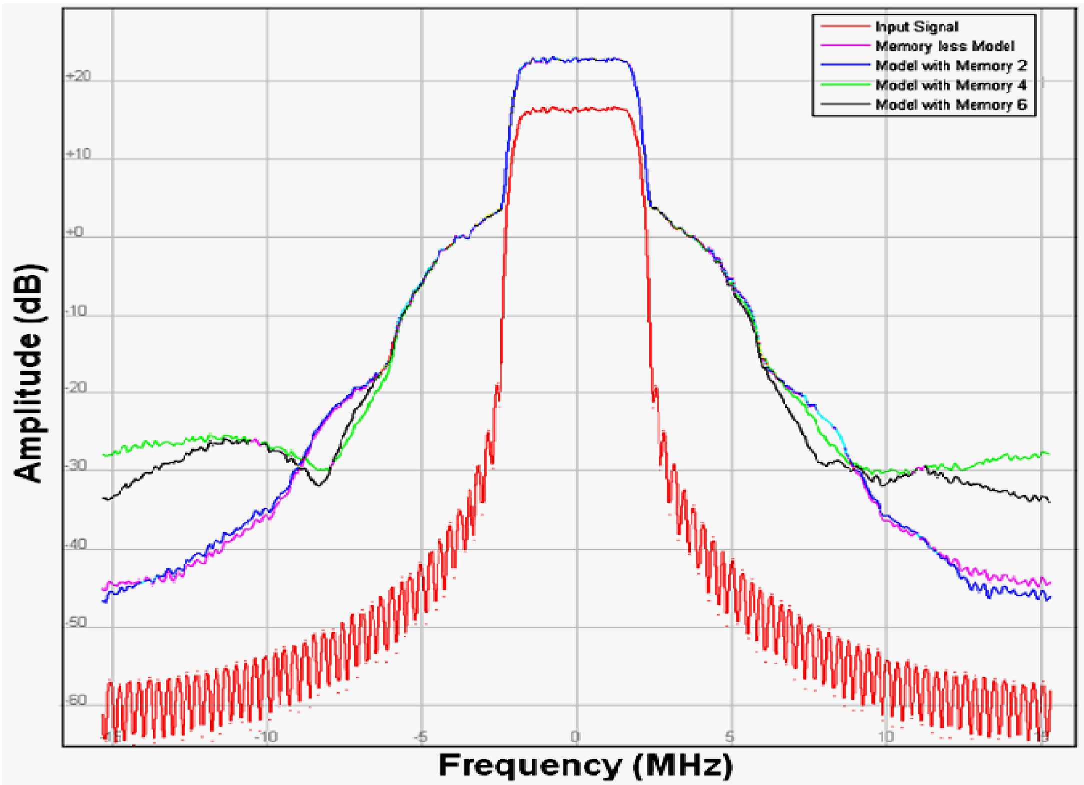


Fig. 5.8. The power spectrum density of: the input signal to the memory-polynomial model (red), and the output of the model, for different memory levels.

6 Conclusions and future work

In this thesis a behavioral models for RF front end components in Simulink® /RF Blockset® are investigated and the algorithms behind them are explained. A Nonlinear model for the power amplifier that takes into account memory effects is implemented in Simulink® /RF Blockset®. Simulink® models are easier to understand and reuse between different applications. Design time in Simulink® is shorter than in Matlab. However, the simulation time in Simulink® is longer than in Matlab, especially for large RF Blockset® models. Also, Simulink® models are less flexible than Matlab models, but very flexible models can be built in Simulink® using S-functions and embedded functions.

The accuracy of the memory-polynomial model is good, however better accuracy can be achieved by using a sparse delay model [1].

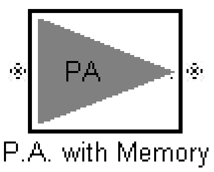
To extend the receiver further, a model for the analog to digital converter can be added to the receiver, this will add the effect of quantization noise and nonlinearity. The effect of the RF filters in the receiver and transmitter can also be added.

Appendix A

Description of implemented Simulink® Models

This is a model for an RF power amplifier. It is built for wideband applications, e.g WCDMA. It models the nonlinearity as well as the memory effects. This model is an implementation of the memory polynomial model.

Power Amplifier Model (P.A. with memory)



Complex base-band nonlinear model of Power amplifier with noise, mismatching and memory

Library

RF Front End

Description

The power amplifier model block is a complex base-band implementation of the memory-polynomial. It includes thermal noise and s-parameters to make it compatible with RF Blockset's physical models Fig. A.1 and Fig. A.2.

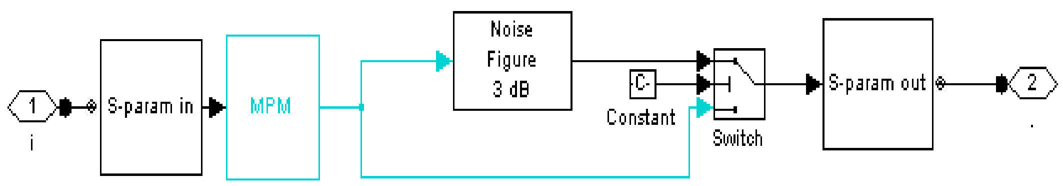


Fig. A.1 Model blocks with S-parameter presentation in Simulink®

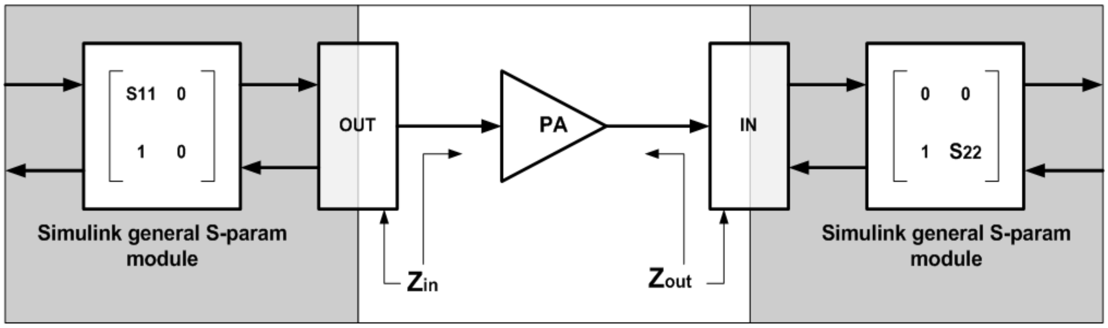


Fig: A.2 Schematic of the model after including S-parameter components.

Building blocks and Parameters identification:

The model is built with blocks from Simulink®, RF Blockset® and communication Blockset®. The process of building the model from memory-polynomial equations is explained in Section 2.3.1, and process of finding coefficient parameter is explained theoretically in Sections 2.3.2, the practical explanation is as follow:

- The input and output data of a typical power amplifier is measured using e.g. load pull method
- This data is converted into IQ or complex base-band data
- MATLAB code given Appendix B is used to find the coefficients of the memory-polynomial model.
- Plug these coefficients in the model as given in the P matrix in the initialization field of the model's mask.

Using model in system

- Make sure that the library "RF Frontend" is in your Matlab's path.
- Open Simulink® browser, the "RF Frontend" library should be there, drag and drop blocks from this library to your model.

Sampling time

In Simulink® blocks putting -1 in the sampling time fields means that the block will inherit the sampling time of the preceding or subsequent blocks.

But in the case of physical RF Blockset® blocks this will not work and a sampling time must be given before running. Getting the sampling time of complicated sources can be very hard in versions before R2007B, a test probe can be used but doesn't always work as this will be a rounded, and the exact sampling time must be used. In the new version of Matlab R2007b this problem can be dealt with as follows:

- Run the model, RF Blockset will gives sampling time error and will state the correct sampling time of preceding block.
- Copy this sampling time to your model's input port.

Dialog Box

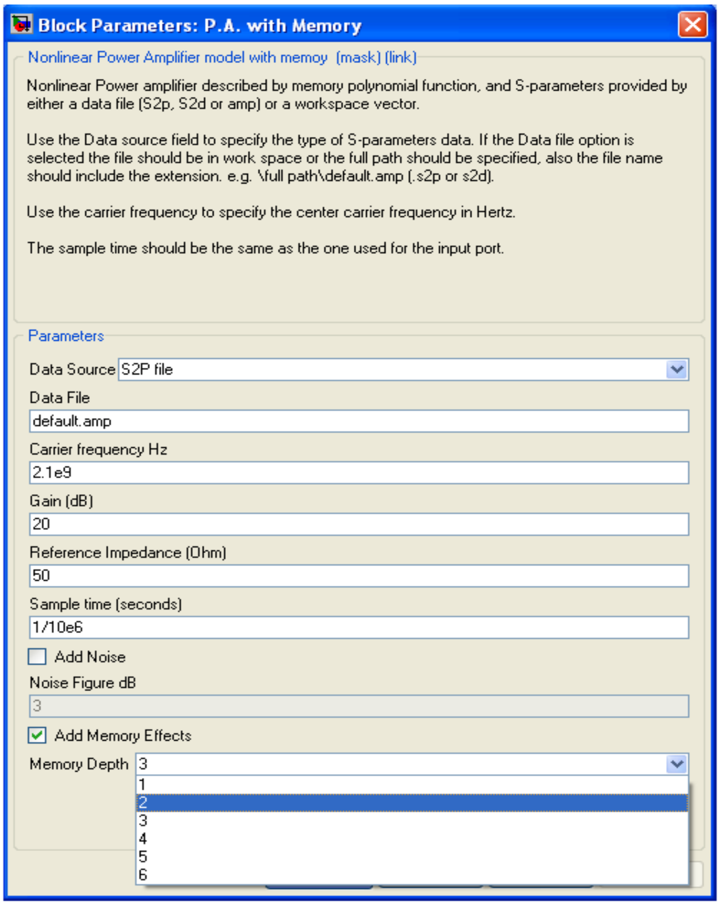


Fig. 2: Dialog box

Data source

This field is used to feed network parameters (S-parameter data).The choices are touchstone files (S2P, S2D or AMP files) and s-parameters vector. It is not tunable.

Data File

If Data source field is set to S2P file, use this field to specify the name of the file that contains the amplifier data. The file name must include the extension. If the file is not in your Matlab path, specify the full path to the file. This field is not tunable.

S-parameters

If the Data source field is set to S-parameter vector, use this field to give S-parameter in vector form; $[S_{11} \quad S_{12}\ ; \quad S_{21} \quad S_{22}]$. This field is not tunable.

Frequency for S-parameters

In case if the Data source field is set to S-parameter vector option, this field gives the value of the frequency corresponding to the network parameters in S-parameter field. This field is not tunable.

Carrier Frequency Hz

This field is used to enter carrier frequency. This field is not tunable.

Gain (dB)

This is the small signal linear gain. This field is not tunable.

Reference impedance (Ohm)

It is the characteristic impedance of the physical block to which the power amplifier is connected. Typical value is 50 ohm.

Sample time (seconds)

This field is to set the time interval between consecutive samples of the input signal. This field is not tunable.

Add Noise

If this field is checked then noise will be added.

Noise figure

When the field Add noise is enabled, then here you can put the scalar value of noise figure in decibel. This field is not tunable.

Add memory effects

If checked this field will enable the addition of memory effects to the model. If it is unchecked then the memory depth will be zero, i.e. memory less polynomial. This field is not tunable.

Memory Effects

If the field "Add memory effects" is enabled here the depth of memory can be selected between "1" and maximum of "6". This field is not tunable.

Appendix B

Matlab Codes to calculate coefficients

Once the input and the output data for the power amplifier are available in IQ format, the following ***code I*** is used to calculate coefficients for specified values of memory depth "m" and odd polynomial order of "n". Only odd components of the polynomial are considered since even components can be easily filtered out in the receiver. "n" in the code refers to the number of polynomial terms and the order of the polynomial is "2n-1". *Code II* is for both even and odd terms

Code I:

```
function a=analyzemodel_mp(x,y,n,m)
% [A,H]=ANALYZEMODEL_MP(X,Y,N,M)Identification of a memory polynomial from
% x and y. N is the order of the nonlinearity, and M the memory depth; if
% omitted, their default values are 6 and 4, respectively.
% Normal usage: A=ANALYZEMODEL_MP(X,Y) identifies the model into the matrix
% a. Each column is the coefficients for the nonlinearities (first col is
% the nonlinearity coeff for delay 0, then for delay 1 etc); each row is
% the filter of each term in the nonlinearity (row 1 is the filter of x,
% row 2 of x^2 etc)
% note that the identified polynomial takes into account only odd
% polynomial terms.
% (c) 2007 Thomas Eriksson thomase@chalmers.se

if nargin<4
   n=6; % order of nonlinearity
   m=4; % order of memory
end
H=geth(x,n,m);
a=pinv(H)*y;
a=reshape(a,n,m+1);
```

```
function H=geth(x,n,m)

N=length(x);
H=[];
for q=0:m
   Hq=zeros(N,n);
   for k=1:n
      for l=(q+1):N
         Hq(l,k)=abs(x(l-q))^(2*k-2)*x(l-q);
      end
   end
   H=[H Hq];
end
```

Code II:

```
function a = analyze_dpd(y, z, K,Q)
%[A,H]=ANALYZE_DPD(X,Y,N,M)  Identification of a memory polynomial from
% x and y. N is the order of the nonlinearity, and M the memory depth.
% Normal usage: A=ANALYZE_DPD(X,Y,N,M) identifies the model into the matrix
% a. Each column is the coefficients for the nonlinearities (first col is
% the nonlinearity coeff for delay 0, then for delay 1 etc); each row is
% the filter of each term in the nonlinearity (row 1 is the filter of x,
% row 2 of x^2 etc)
%note that the identified polynomial contains both even and odd parts.
%(c) 2008 Eyad ARABI

U = getU(y, K,Q);
a = U/z ; % this gives the least square error %% pinv(U)*z can also be used
a = reshape(a,K,Q+1);

function U = getU(y, K, Q)
N = length(y);
U = [];
for q = 0:Q
   Uq = zeros(N, K);
```

```
  for k = 1:K
    for l = (q+1):N
      Uq(l,k) = abs(y(l-q))^(k-1)*y(l-q);
    end
  end
  U = [U Uq];
end
end
```

Matlab code for Memory-polynomial model

The following code is an implementation of the memory-polynomial model. This code is equivalent to the Simulink® model built in Section 2.3.1 of this thesis**.** The Simulink® model and the Matlab code were verified against each other. In this code "**x**" is the complex input vector to the model, "**Y**" is the complex output vector, and "**a**" is the coefficient matrix calculated from above code.

```
function y=runmodel_mp(x,a)
% (c) 2007 Thomas Eriksson thomase@chalmers.se

n=size(a,1); % order of nonlinearity
m=size(a,2)-1; % order of memory
a=reshape(a,prod(size(a)),1);
H=geth(x,n,m);
y=H*a;

function H=geth(x,n,m)
N=length(x);
H=[];
for q=0:m
  Hq=zeros(N,n);
  for k=1:n
    for l=(q+1):N
```

```
        Hq(l,k)=abs(x(l-q))^(2*k-2)*x(l-q);
      end
    end
    H=[H Hq];
End
```

Reference list

[1] Hyunchul Ku, *and J.* Stevenson Kenney, "Behavioral Modeling of Nonlinear RF Power Amplifiers Considering Memory Effects" *IEEE* Trans. on Microwave Theory and Techniques, Vol. 51, No 12, December 2003

[2] Ding L., "Digital Predistortion of Power Amplifiers for Wireless Applications", Ph.D thesis, Georgia Institute of Technology, March 2004.

[3] Magnus Isaksson, David Wisell, and Daniel Rönnow, "A Comparative Analysis of Behavioral Models for RF Power Amplifiers", IEEE Trans on Microwave Theory And Techniques, Vol. 54, no. 1, January 2006

[4] A. Ahmed, "Analysis, Modeling and Linearization of Nonlinearity and Memory Effects in Power Amplifiers Used for Microwave and Mobile Communications", Ph.D thesis, University of Kassel , Germany, March 2005.

[5] Magnus Isaksson, "Radio Frequency Power Amplifiers, Behavioral Modeling, Parameter-Reduction and Digital Predistortion", Ph.D thesis KTH, 2007.

[6] www.mathworks.com

[7] David M. Pozar, "Microwave and RF Design of Wireless Systems",1st edition John Wiley & Sons. - November 2000

[8] David M. Pozar, "Microwave engineering", 2nd edition John Wiley & Sons. 1998

[9] R. E. Collin, "Foundations for microwave engineering," 2nd edition, wily & sons 2000, pp. 248-254.

[10] Kasdin, N.J., "Discrete Simulation of Colored Noise and Stochastic Processes and $(1/f^{\alpha})$; Power Law Noise Generation," *The Proceedings of the IEEE*, May, 1995, Vol. 83, No. 5.

[11] Jie He , Jun Seok Yang, Yongsup Kim and Kim, A.S., "System-Level Time-Domain Behavioral Modeling for A Mobile WiMax Transceiver," *The Proceedings of the IEEE* ,Sept 2006, pp. 138-143.

[12] Ghorbani, A. and M. Sheikhan, "The Effect of Solid State Power Amplifiers (SSPAs) Nonlinearities on MPSK and M-QAM Signal Transmission," *Sixth Int'l Conference on Digital Processing of Signals in Comm.*, 1991, pp. 193-197.

[13] Saleh, A.A.M., "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Trans. Communications*, vol. COM-29, pp.1715-1720, November 1981.

[14] Ding, L. and Zhou, G. T., “A robust predistorter constructed using memory polynomials,” *IEEE Trans*. Veh. Technol., vol. 53, pp. 156–162, Jan. 2004.

[15] D. B. Leeson, “A simple model of feedback oscillator Noise spectrum,” Proc. IEEE, Vol. 54 pp. 329-330, 1966

[16] RF Blockset® User guide (www.mathworks.com)

[17] Pedro, J.C. and Maas, S.A., “A comparative overview of microwave and wireless power-amplifier behavioral modeling approaches,” Microwave Theory and Techniques, *IEEE Transactions*, Volume: 53,pp.1150- 1163 April 2005.

[18] C. Rapp, "Effects of HPA-Nonlinearity on a 4-DPSK/OFDM-Signal for a Digital Sound Broadcasting System", in Proceedings of the Second European Conference on Satellite Communications, Liege, Belgium, Oct. 22-24, 1991, pp. 179-184.

[19] Hillbrand, H. and P.H. Russer, "An Efficient Method for Computer Aided Noise Analysis of Linear Amplifier Networks," *IEEE Transactions on Circuits and Systems*, Vol. CAS-23, Number 4, pp. 235–238, 1976