NTNU

Norwegian University of
Science and Technology

Design and Characterization of a 6 W GaN HEMT Microwave Power Amplifier with Digital Predistortion Linearization

Dragan Mitrevski

Master of Science in Electronics
Submission date: March 2011
Supervisor: Morten Olavsbråten, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

The activity around the subject of power amplifiers have increased considerably in the last decade, both in companies developing radio equipment for wireless services, and in a number of universities and research centers around the world. A power amplifier is of great economic and technical importance, because it accounts for a significant portion of the costs in both development and production of radio equipment, and is a very critical component in today's wireless world. In portable devices, it is the power amplifier that drains the battery the most, and for both hand-held devices and base station equipment (infrastructure) the power amplifier must satisfy several stringent requirements. Technically good power amplifier solutions will thus offer great competitive advantages.

The main topic of this thesis is to characterize a 6 W GaN HEMT transistor from Cree Inc. through load pull simulations and measurements, and implement digital predistortion for linearization. Thus, the tasks will consist of:

- Be familiarized with simulation tools to perform nonlinear simulations.
- Be familiarized with class F/inverse class F power amplifier design techniques.
- Be familiarized with measurement equipment to perform load pull measurements.
- Implement and evaluate different nonlinear black-box methods for characterizing the nonlinear behavior (AM/AM, AM/PM, etc) of the power amplifier in MATLAB.
- Implement routines for digital predistortion in MATLAB.
- Measure the linearity performance of different power amplifiers (AM/AM, AM/PM, ACPR, etc) and compare the performance with and without linearization applied.

Assignment given: 06. October 2010
Supervisor: Morten Olavsbråten, IET

Dragan Mitrevski

Design and Characterization of a 6 W GaN HEMT Microwave Power Amplifier with Digital Predistortion Linearization

Thesis for the degree of Master of Science

Trondheim, March 2011

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics
and Electrical Engineering
Department of Electronics and Telecommunications

NTNU

Norwegian University of Science and Technology

Master's thesis

for the degree of master of science in Electrical Engineering

Faculty of Information Technology, Mathematics and Electrical Engineering

Department of Electronics and Telecommunications

Master's thesis at NTNU, 2011

Abstract

In this thesis, characterization of a 6 W GaN HEMT power amplifier for optimal operating conditions through load pull simulations and measurements is investigated. The purpose is to find source and load impedances to achieve for instance maximum efficiency and maximum output power, and investigate whether the simulated results can be replicated in a measurement setup.

Simulations show that when matching for maximum output power, a peak output power of 13 W is achieved, while in 1 dB compression, an output power of 11.40 W together with a power added efficiency of nearly 67 % is obtained, a remarkable for a 6 W device. Matching for maximum efficiency yields a peak power added efficiency of 78 % and a peak drain efficiency of 90 %, however, in 1 dB compression these levels are 71 % and 74 %, respectively. Load pull measurements are performed with a prototype design, however, due to problems with the software controlling the measurement equipment, the tuning process had to be done manually, which effectively limited the possibility of replicating the simulated results.

With manual tuning in coarse steps, two sets for semi–optimal operating conditions are found. For the first set of impedances, a peak output power greater than 11 W, with a peak power added efficiency of 68 % is obtained, while for the second set of impedances, an output power of 7.40 W with a peak power added efficiency of 70 % is achieved. The measured performance indicate that the simulated results can be replicated once the measurement setup is working optimally.

In addition, the topics of black–box characterization and digital predistortion linearization of microwave power amplifiers are investigated. Microwave power amplifiers obtain maximum operational performance when operated close to the saturation point. However, due to the nonlinear behavior of the device when operated in this region, the output signal will be distorted and cause interference with neighboring channels. Thus, in order to maintain the high operational performance and avoid adjacent channel interference, it is of great importance to characterize the nonlinear behavior of the device, and compensate for the introduced distortion terms. Digital predistortion linearization is considered to be among the most cost–effective linearization schemes as of today, providing flexibility and good performance. The design of a digital predistortion unit amounts to three main steps: deriving a behavioral model of the microwave power amplifier, estimating the model parameters for the inverse characteristic, and the implementation of the preinverse filter. For behavioral modeling, well–known nonlinear models, such as the Volterra series are investigated, and used with an indirect learning architecture to estimate the parameters of the inverse system.

Simulations of the different linearization algorithms show that in an ideal environment, the adjacent channel power ratio can be reduced with up to 25 dB, almost independent of the model used for the predistorter design. However, when used in a real measurement setup, a maximum reduction in adjacent channel power ratio of only 15 dB is obtained. The relatively large difference in performance is most likely caused by a combination of long term memory effects due to semiconductor trapping phenomena, noise, and modeling errors.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of master of science (MSc) at the Department of Electronics and Telecommunications, Norwegian University of Science and Technology (NTNU). The work was carried out in the period October 2010 to March 2011, under the supervision of Associate Professor Morten Olavsbråten, who is with the Department of Electronics and Telecommunications at NTNU.

Acknowledgment

First and foremost I would like to thank my supervisor Morten Olavsbråten, both for introducing me, and giving me the opportunity to work with the topic of RF power amplifiers. His vast, and seemingly never–ending insight within the topic, combined with his enthusiasm, have been a great source for both motivation and inspiration during this work. I would also like to thank head engineer Terje Mathiesen for all his help with the production of the PCBs, together with Tore Landsem and Tore Berg from the engineering workshop for cutting heat sinks, and other mechanical structures that I've asked for. Finally, special thanks to my friend and fellow student Hans Herman Hansen for valuable input on my thesis - and to the rest of my friends I've made during my time at NTNU for five unforgettable years.

Trondheim, Norway, March 2011
Dragan Mitrevski

Contents

List of Figures

List of Tables

Abbreviations

AC	alternating current
ACEPR	adjacent channel error power ratio
ACPR	adjacent channel power ratio
ADC	analog–to–digital converter
ADS	Advanced Design Systems
AIC	Akaike's information criterion
AM/AM	amplitude–to–amplitude conversion
AM/PM	amplitude–to–phase conversion
ANMA	augmented nonlinear moving average
CCDF	complementary cumulative distribution function
CDF	cumulative distribution function
DAC	digital–to–analog converter
DC	direct current
DLA	direct learning architecture
DSP	digital signal processing
DUT	device under test
EVM	error vector magnitude
FIR	finite impulse response
FPGA	field programmable gate array
GaAs	Gallium Arsenide
GaN	Gallium Nitride
GPIB	general purpose interface bus
HB	harmonic balance
HEMT	high electron mobility transistor
ILA	indirect learning architecture
IMD	intermodulation distortion
LMS	least mean square
LPExp	Load Pull Explorer
MAG	maximum available gain
MPT	multi–purpose tuner
NMA	nonlinear moving average
NMSE	normalized mean square error
PA	power amplifier

PAE	power added efficiency
PCB	printed circuit board
PEP	peak–envelope–power
PNA	programmable network analyzer
QAM	quadrature amplitude modulation
RF	radio frequency
RFPA	radio frequency power amplifier
RLS	recursive least square
SDE	spline delay envelope
Si	Silicon
SMA	SubMiniature version A
SOLT	short–open–load–thru
SRF	series resonant frequency
SRRC	square root raised cosine
SVD	singular value decomposition
TEM	transverse electromagnetic
TRL	thru reflect line
VNA	vector network analyzer
VSA	vector signal analyzer
VSG	vector signal generator

1 Introduction

The power amplifier (PA) is a critical component in a wireless communication system. It accounts for a significant portion of the costs in both development and production of radio equipment, making improved power amplifier architectures and design topologies sought-after in the industry. With the later years' rapid increase in the number of wireless communication systems, the already stringent linearity requirements the power amplifier must satisfy have become even tougher. In addition to linearity, requirements regarding power efficiency are also present for some applications.

In portable devices it is the power amplifier that drains the battery the most, while in base station equipment the power amplifiers often require large and expensive cooling systems, increasing the running and total costs for the operator. With the introduction of Gallium Nitride (GaN) high electron mobility transistors (HEMT) in the last decade, the problem of high–efficiency operation seem to have a brighter future. The GaN HEMT technology provides some advantageous properties, such as increased power density, higher operating voltages, and improved thermal properties [2].

These properties combined allows for less complex designs, higher power efficiencies [3], and a reduced requirement for large cooling systems [4]. However, in order to implement a design for high–efficiency operation, an accurate transistor model is required. If an accurate model is unavailable, or if is desirable to validate a nonlinear transistor model, load pull measurements can be carried out [5]. This type of measurements can also be used for finding the optimal operating conditions of the device, which is one of the topics investigated in this study. Although high–efficiency operation is easier achieved with the GaN HEMT technology, the transistors still behave nonlinearly. Thus, for high–efficiency operation with good linearity, additional, external linearization is required. This can either be done in the analog or the digital domain.

Although analog linearization schemes over the years has shown much greater performance compared to digital linearization schemes such as digital predistortion [6, ch. 5–6], the trend is moving towards performing more of the signal processing in the digital domain using digital signal processors (DSP) and/or field programmable gate arrays (FPGA). This also allows for reducing the analog radio frequency (RF) front–end to the necessary minimum, implying cheaper and less complex transmitter systems [7].

Thus, once moving towards the digital domain, digital predistortion becomes the nat-

ural choice for linearization of power amplifiers. Designing a digital predistorter amounts to three main steps: 1) acquiring an accurate behavioral model of the nonlinear system, 2) estimating the parameters for the inverse characteristic, and 3) implement the preinverse filter. Much research within the topic of digital predistortion is put into the two first steps, where decreased model complexity, and robust estimation of the model parameters is desirable [8].

In this study, high–efficiency operation for a discrete 6 W GaN HEMT transistor from Cree Inc. is achieved by characterizing the device thru load pull simulations and measurements. For the load pull simulations, Agilent ADS was used, while for the measurements, source and load impedance tuners from Focus Microwaves were used. As for the issue of amplifier linearity, various nonlinear black–box methods are implemented in MATLAB, and used to model the nonlinaer behavior of different power amplifiers. Based on the behavioral models, the inverse characteristics of the amplifiers are estimated using an indirect learning architecture [9]. The parameters for the inverse characteristic are then used in a digital predistorter design implemented in MATLAB.

1.1 Outline of Thesis

The following presents the organization of the thesis.

Chapter 1 provides a brief introduction to the problem investigated in this thesis, together with the motivation for this work.

Chapter 2 presents preliminary theory for the main topics covered in this work. This includes an overview of the notation used throughout the thesis, a short introduction to basic microwave theory and microwave power amplifiers with focus on class F and inverse class F operation. Also included is a short overview of matrix factorization algorithms and orthogonal polynomials used in this work for improving numerical properties in estimation problems.

Chapter 3 presents the design of the test–board used in the load pull simulations and measurements. An overview of the design technology used for the PCB design is also given, together with an introduction to the concept of source and load pull measurements. Finally, results from simulations and measurements of the simulated and implemented test–board are presented. The results are followed by a discussion and a summary

Chapter 4 presents the topic of black-box behavioral modeling of microwave power amplifiers. The models presented includes the likes of the Volterra series, and other subsets of it, such as the memory polynomial model, and a spline delay envelope model. In addition to the short theoretical analysis given, the performance of the models is validated, both with measured and simulated input/output data. The simulated input/output data is obtained with a cosimulation environment developed specifically for this thesis. Using this environment, the influence of perturbing/detrimental effects such as measurement noise and limitations in the measurement equipment can be neglected. Again, the results presented are followed by a discussion.

Chapter 5 covers the topic of digital predistortion linearization. It includes relevant theory, which again is based on the behavioral models discussed in the previous chapter. The theory is followed by results from linearizing both measured and simulated microwave power amplifiers.[1] Finally, the chapter is finished with a discussion of the results.

Chapter 6 concludes this thesis. Some further topics for future work are also sketched.

[1] The simulated power amplifier is used in the cosimulation test bench mentioned previously.

Preliminaries 2

This chapter provides necessary background information for the analysis presented later in the thesis. The chapter is started by a short note on the notation used throughout the thesis in Section 2.1. Section 2.2 presents basic microwave theory which is used in later chapters of this work. In Section 2.3 a short excerpt of the theory behind microwave power amplifiers, with focus on class F and inverse class F operation, is presented.

The focus is on these two classes of operation because load pull simulations and measurements presented later in the next chapter will mainly be focusing on determining impedances for high–efficiency operation, which is one of the main properties these amplifier classes provide [10–12]. For theory of other power amplifier classes, the reader is referred to [6, 13, 14], where a detailed overview over the most common classes can be found.

Finally, in Section 2.4 and 2.5 a brief introduction to the problem of numerical stability in parameter estimation is presented, together with a few commonly used methods to overcome this problem, e.g. matrix factorization and the use of orthogonal polynomial basis functions.

2.1 Notation

Boldface symbols denote matrices (uppercase) and vectors (lowercase). The transpose, complex conjugate and Hermitian transpose of a matrix $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ are $\boldsymbol{A}^{\mathsf{T}}$, $\boldsymbol{A}^{*}$ and $\boldsymbol{A}^{\mathsf{H}}$, respectively. Here $\mathbb{C}^{M\times N}$ designates the vector space of complex $M \times N$ matrices. Similarily $\mathbb{C}^{M}$ denotes the vector space of complex M-vectors.[1] For matrix $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ composed of L sub–matrices, we define $\underline{\boldsymbol{A}} = [\boldsymbol{A}_1\, \boldsymbol{A}_2\, \ldots\, \boldsymbol{A}_L]$, where $\boldsymbol{A}_l \in \mathbb{C}^{M\times L_i}$ and $N = \sum_i L_i$. Given a matrix $\boldsymbol{A} \in \mathbb{C}^{M\times N}$, we also express it as a set of column vectors, $\boldsymbol{A} = [\boldsymbol{a}_1\, \boldsymbol{a}_2\, \ldots\, \boldsymbol{a}_M]^{\mathsf{T}}$, where $\boldsymbol{a}_m \in \mathbb{C}^{N} \,|\, \forall\, m = 1, \ldots, M$.

Voltages, currents, and impedances are denoted in uppercase V, I, and Z, respectively, with normalized equivalents in v, i, and z. Incident and reflected waves, together with reflection coefficients are defined later in this chapter, with corresponding notation. For the fundamental frequency we use f_0, and for higher–order harmonics kf_0

[1] Vector spaces for $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{Z}$ are denoted in the same manner.

where $k > 1 \in \mathbb{N}$. Operators $\Re\{\cdot\}$ and $\Im\{\cdot\}$ return the real and imaginary part of the argument, respectively. Throughout the thesis, we write $\mathbb{H}$ for linear filters and $f(\cdot)$ nonlinearities respectively. Nonlinearities are considered to be operators $f : \mathbb{C}^M \longrightarrow \mathbb{C}^M$, with their inverse (when existence is assumed) written as $f^{-1}(\cdot)$. Unless stated otherwise, we use $\boldsymbol{x}$ and $\boldsymbol{y}$ for input and output signals respectively, with subscript n or t to denote discrete- or continuous time, respectively. In parameter estimation, the parameters to be estimated are denoted by $\boldsymbol{c} \in \mathbb{C}^P$, where diverging notation will be specified. Already estimated parameters are denoted by a hat, i.e. $\hat{\boldsymbol{c}}$ for an estimated parameter vector.

2.2 Preliminary Microwave Theory

The work done in this thesis deals with microwave power amplifiers, which can be interpreted as active two–port networks. For this reason, the preliminary, and brief, microwave theory in the following has its focus on active two–port networks. This includes subjects such as S parameter characterization, reflection coefficients and measures for stability. For better overview, see [15, ch. 2, 4, 5 and 11].

2.2.1 Active Two–Port Networks

A general, active two–port can be illustrated as in Figure 2.1. The illustration shows an active device terminated with a general source/load impedance, and the corresponding reflections that occur at the source/load. In addition, reflections at the input and output of the device are present. This is because active devices in general have input/output impedances that differ from the termination impedances.

The input/output reflections can be minimized by using passive matching networks (ideally lossless, for power transfer), which can also help in satisfying design specifications, such as bandwidth, gain and input return loss.

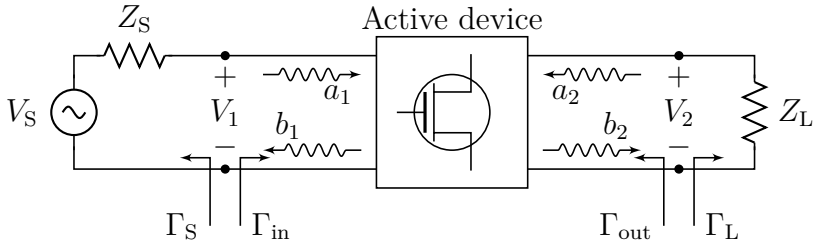


Figure 2.1: A general, active two–port network, with definition of incident and reflected waves at both ports. Z_S and Z_L is the source and load impedance, respectively. The incident and reflected waves (a and b) are, together with the reflection coefficients Γ, defined in the following.

If the transistor is operated in its linear region (small–signal stimuli), the electrical behavior of the two–port can be described by its S parameters, which relate the incident

voltage waves to reflected voltage waves. With reference to Figure 2.1, we define the incident and reflected waves at port k as a_k and b_k, respectively, where

$$a_k = \sqrt{V_k^+ I_k^+} \quad [\sqrt{\mathrm{W}}]\,, \tag{2.1}$$

$$b_k = \sqrt{V_k^- I_k^-} \quad [\sqrt{\mathrm{W}}]\,, \tag{2.2}$$

and $V_k^\pm = (V_k \pm Z_0 I_k)/2$, $I_k^\pm = (V_k \pm Z_0 I_k)/(2Z_0)$, $V_k = V_k^+ + V_k^-$ and $I_k = I_k^+ - I_k^-$. Based on this, the S parameters of the two–port are given as

$$\boldsymbol{S}\boldsymbol{a} = \boldsymbol{b}\,, \tag{2.3}$$

where $\boldsymbol{a} = [a_1\, a_2]^\mathsf{T}$, $\boldsymbol{b} = [b_1\, b_2]^\mathsf{T}$, and $\boldsymbol{S} = \left[[S_{11}\, S_{12}]^\mathsf{T}\ [S_{21}\, S_{22}]^\mathsf{T}\right]^\mathsf{T}$. To determine specific elements of $\boldsymbol{S}$ we use the relationship

$$S_{ij} = \left.\frac{b_i}{a_j}\right|_{a_k=0 \text{ for } k\neq j}\,, \tag{2.4}$$

from which we see that the reflection at port k is the ratio b_k/a_k. Using the definition of the incident and reflected waves, it is easily shown that the reflection coefficient is given by

$$\Gamma_\mathrm{x} = \frac{Z_\mathrm{x} - Z_0}{Z_\mathrm{x} + Z_0}\,, \tag{2.5}$$

where Z_x is the impedance looking into the desired location. The reflection at the source and load are found by replacing Z_x with Z_S and Z_L, respectively.

In order to determine the input/output reflection coefficients of the active device, additional manipulation of the above equations and relations is required. The derivation is not repeated here, however, it can be shown that input/output reflection coefficients are given by (see [15, pp. 537])

$$\Gamma_\mathrm{in} = S_{11} + \frac{S_{21}S_{12}\Gamma_\mathrm{L}}{1 - S_{22}\Gamma_\mathrm{L}}\,, \tag{2.6}$$

and

$$\Gamma_\mathrm{out} = S_{22} + \frac{S_{21}S_{12}\Gamma_\mathrm{S}}{1 - S_{11}\Gamma_\mathrm{S}}\,, \tag{2.7}$$

where the S parameters and the source/load reflection coefficients are as defined above.

2.2.2 Stability

A major concern in power amplifier design is stability. Potentially unstable devices may cause spurious oscillations, which i.e. may result in a change in device properties and a degredation in performance. A device is potentially unstable if it presents an input/output impedance with a negative real part — or equivalently $|\Gamma_\mathrm{in}| > 1$ and/or

$|\Gamma_{\text{out}}| > 1$. As these two parameters depend on the source and load matching networks, the stability of the amplifier depends on Γ_{S} and Γ_{L} [15, pp. 542]. Since Γ_{x} is a function of frequency, the amplifier stability is also frequency dependent — suggesting that stability should be checked at frequencies outside the band of interest as well.

In this work, unconditional stability has been set as a criteria, implying that $|\Gamma_{\text{in}}| < 1$ and $|\Gamma_{\text{out}}| < 1$ for all passive source and load impedances (or equivalently $|\Gamma_{\text{S}}| < 1$ and $|\Gamma_{\text{L}}| < 1$). The measure of stability used is the μ-factor, introduced in [16], and defined as

$$\mu_{\text{load}} = \frac{1 - |S_{11}|^2}{|S_{22} - \Delta S_{11}^*| + |S_{12}S_{21}|}\,, \tag{2.8}$$

$$\mu_{\text{source}} = \frac{1 - |S_{22}|^2}{|S_{11} - \Delta S_{22}^*| + |S_{12}S_{21}|}\,, \tag{2.9}$$

where $\Delta = S_{11}S_{22} - S_{12}S_{21}$. If either μ_{load} or μ_{source} is greater than unity, the network is unconditionally stable. In addition, greater values of μ imply increased stability.

2.3 Microwave Power Amplifiers

As mentioned earlier, the power amplifier is one of the most critical components in a wireless communication system. Its main task is converting dc input power into a significant amount of RF/microwave output power [17], which should be achieved with as little distortion as possible, and preferably with the highest power efficiency obtainable. However, due to its inherent nonlinear behavior, the properties of high–efficiency operation and linear amplification often come at the expense of each other — which also represent one of the main challenges in power amplifier design [5].

In the following, some of the main properties of microwave power amplifiers are presented shortly, together with an introduction to class F and inverse class F mode power amplifiers, which are the two classes of operation that typically offers the best trade–off between high efficiency and linear amplification.

Power Amplifier Gain and Power Efficiency

In this work, the transducer power gain is used as a measure of amplifier gain. It is defined as the ratio of the RF output power ($P_{\text{RF}_{\text{out}}}$) to the RF input power ($P_{\text{RF}_{\text{in}}}$), and depends on both the source and load impedances, thus, accounting for any loss due to reflections in the matching networks.[2] Mathematically, this is equivalent to

$$G_{\text{T}} = P_{\text{RF}_{\text{out}}}/P_{\text{RF}_{\text{in}}}\,. \tag{2.10}$$

For notational simplicity and compactness, this term is referred to as *power gain* and denoted with a simple G throughout the rest of the thesis. As for power efficiency, drain

[2]The RF output power can be interpreted as the power delivered to the load, while the RF input power can be interpreted as the power available from the source.

efficiency and power added efficiency is used as measures. The drain efficiency is defined as the ratio of the RF output power ($P_{\mathrm{RF_{out}}}$) to the dc input power (P_{dc}), or

$$\eta_{\mathrm{dc}} = 100 \cdot P_{\mathrm{RF_{out}}}/P_{\mathrm{dc}} \quad [\%]\,. \tag{2.11}$$

The drawback with this measure is that it does not take into account the incident RF power at the device input. Thus, for a better overview with respect to thermo–dynamics, power added efficiency can be used — which is defined as the ratio of the difference between the RF output power ($P_{\mathrm{RF_{out}}}$) and RF input power ($P_{\mathrm{RF_{in}}}$) to the dc input power (P_{dc}), or

$$\eta_{\mathrm{PAE}} = 100 \cdot (P_{\mathrm{RF_{out}}} - P_{\mathrm{RF_{in}}})\, P_{\mathrm{dc}} \quad [\%]\,, \tag{2.12}$$

From these equations we see that we can express the power added efficiency as

$$\eta_{\mathrm{PAE}} = \eta_{\mathrm{dc}}\,(1 - 1/G) \quad [\%]\,, \tag{2.13}$$

where G is the power gain in linear scale. This relationship implies that for an amplifier with a large gain, these two terms will be almost equal, as the $1/G$ term will converge towards zero as G increases.[3]

Nonlinear Distortion in Microwave Power Amplifiers

In general, microwave power amplifiers are nonlinear devices which will cause distortion to the input signal, both in amplitude and phase. As the output power of the power amplifier reaches saturation, the gain of the power amplifier will drop, leading to gain compression [5]. For single tone inputs, the gain compression will just lead to the generation of harmonic components. For multi tone signals, or digitally modulated signals, the gain compression leads to intermodulation distortion (IMD), which can be described as *the collection of unwanted responses to a combination of signals* [18].

In order to minimize the amplitude distortion, the power amplifier should be operated in its linear operating range, which is up to the power level for which the output power has decreased by 1 dB from the ideal linear characteristic, also referred to as the 1 dB compression point. In addition to gain compression, another source to IMD is memory effects. In an ideal scenario, the power amplifier output is a function of the present input signal only. This is, however, not always the case for a practical scenario, where the present power amplifier output can be a function of both present and previous input signals, and hence deviate from the static case. These deviations are characterized as memory effects, and can be defined as bandwidth–dependent nonlinear effects [19]. The main causes for memory effects are often divided into three categories [14, pp. 256]:

[3] In the literature, there is also a third measure for efficiency defined, however, not as commonly used as the two measures already mentioned. This measure is the overall efficiency, which is defined as the ratio of the RF output power ($P_{\mathrm{RF_{out}}}$) to the sum of the RF input power ($P_{\mathrm{RF_{in}}}$) and dc input power (P_{dc}), which gives the best overview with respect to thermo–dynamics [5, pp. 9].

Dynamic thermal effects (self–heating): The dynamic thermal effects arise due to temperature variations at the top of the chip that modifies the electrical properties of the transistor at the envelope frequency. Such effects are often categorized as long–term memory effects, and can be difficult to model.

Unintentional bias–supply modulation: The bias–supply modulation is caused by unwanted alternating current (ac) voltages that add themselves to the dc supply voltage [20]. This results in amplitude and phase modulation of the output signal such that the sidebands fall within the passband, exactly where the IMD products occur, resulting in decreased linearity. These effects can be minimized by carefully designing the biasing network.

Semiconductor trapping effects: Semiconductor trapping effects are induced by the input signal and vary at a slow rate compared to the modulation speed. In most cases these effects complicate analytical or behavioral modeling.

2.3.1 Class F and Inverse Class F Power Amplifiers

Class F and inverse class F power amplifiers fall in under the category of harmonically tuned power amplifier modes. This means that the basic structure of the amplifier in general is based on another amplifier class, i.e. a class AB, or a class B amplifier, but with additional structures/components used in order to harmonically tune the intrinsic waveforms of the device. The tuning of the waveforms is applied in order to minimize the amount of time for which the active device is presented with a non–negative drain current I_{D} and drain voltage V_{D} simultaneously. Clearly, minimizing this period of time results in an increase in both power efficiency and fundamental output power [14, ch. 6.4].

For class F power amplifiers, this is achieved by using a short load–termination at even–ordered harmonics, and open load–termination at odd–ordered harmonics, as can be seen in Figure 2.2, where the open load–termination at the odd harmonics is obtained by using harmonic resonators. A configuration like this results in the adding of in–phase, odd–ordered voltage components to the fundamental voltage component, reducing the peak–to–peak voltage swing, and the adding of in–phase even–ordered current components to the fundamental current component, resulting in current peaking.

Or with other words, a square waveform for the drain voltage, and a half–wave rectified waveform for the drain current. Ideally, this leads to a power efficiency of 100 % as these waveforms do not overlap at all; however, since only a finite number of harmonics can be used in the generation of the waveforms, there will be some overlap between the current and voltage waveforms, and the power efficiency will decrease. In [21], an equation for calculating the maximum obtainable drain efficiency is given. The equation is a function of the number of odd–ordered harmonics used in the harmonic tuning in class F amplifiers. Using this equation, it is easily seen how rapidly the maximum obtainable drain efficiency decreases. With proper termination of the fundamental and third harmonic, a level of 90.70 % is achievable. In order to approach an efficiency

of 100 %, all odd harmonics up to the eleventh harmonics must be utilized. However, for practical microwave power amplifiers, the output power at the harmonic frequencies greater than the third order harmonic decreases dramatically. Hence, including any higher–order harmonic greater than the third order rarely results in improved efficiency figures [10].

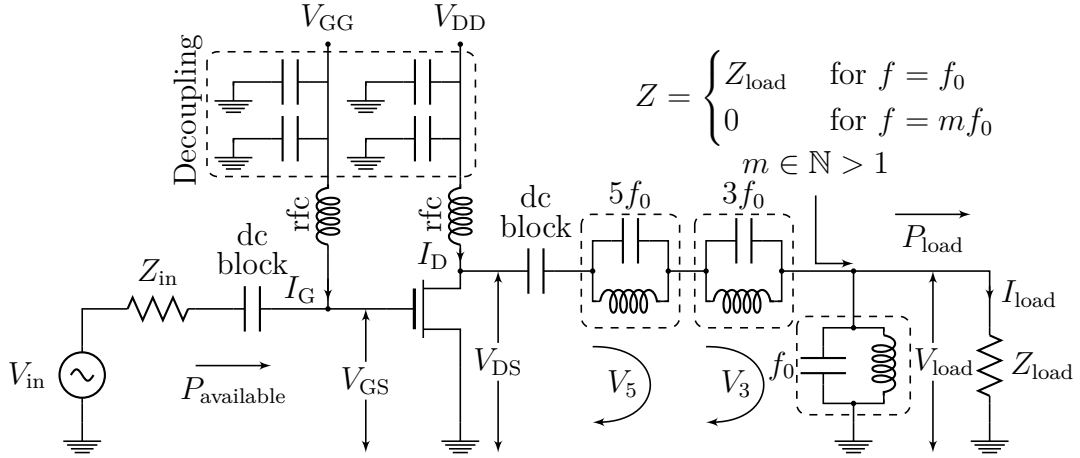


Figure 2.2: Simplified class F PA circuit built around a general HEMT transistor. In the figure, V_{in} is the input signal generator, Z_{in} the input impedance, V_{GG} the gate dc voltage, V_{DD} the drain dc voltage, I_G the gate current, I_D the drain current, V_{GS} the gate–to–source voltage, V_{DS} the drain–to–source voltage, V_5, and V_3 the fifth and third order harmonic voltages, respectively. In addition, rfc is an acronym for radio frequency choke, which basically is a large inductor that for high frequencies acts as an open circuit, and as a short for dc frequencies.

As for the inverse class F amplifier, the terminology is, as the name suggests, inverted compared to a class F amplifier. This implies short load–termination at odd–order harmonics, and open load–termination at even–order harmonics, or equivalently, a square waveform for the drain current, and a half–wave rectified sine for the drain voltage [12], see the illustration in Figure 2.3. In this illustration, the ideal waveforms for both a class F and an inverse class F are depicted, including the effects of the on–state resistance R_{on} has. This resistance is a parasitic, intrinsic resistance, which results in a voltage drop, also known as the knee–voltage V_k, across the transistor when it is conducting [12].[4] The presence of parasitic resistance also causes power to be dissipated by a factor of $R_{on}I_D^2$, in addition to reducing the peak voltage swing, affecting the output properties (i.e. output power and power efficiency) of the amplifier [3].

Since the dissipated power, and the knee–voltage both are functions of I_D, it is clear that an inverse class F power amplifier will outperform a class F when the on–state resistance is of significant value; the half–wave rectified current waveform in the class F PA will always be greater than the square current waveform in the inverse class F PA

[4]The on–state resistance is a function of various intrinsic, parasitic resistances, i.e. source–to–gate resistance R_{sg}, and a drain–to–gate resistance R_{dg} [22].

under the same biasing conditions, or $I_{\mathrm{D}} > I'_{\mathrm{D}}$, where I_{D} and I'_{D} are the drain current of the class F and the inverse class F PA, respectively. In the same manner, inverse class F PAs also have a drawback; the increased peak of the drain voltage. If the biasing conditions and the device technology are not carefully chosen, this may lead to a peak voltage that exceeds the device breakdown, which obviously is undesirable.

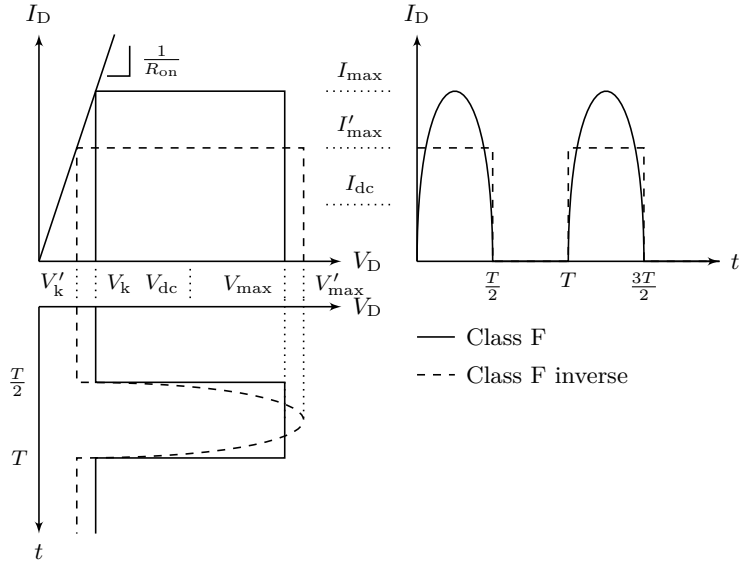


Figure 2.3: Ideal time domain current and voltage waveforms for class F and inverse class F under equal biasing conditions and for same output power. V_{k} denotes the knee voltage, V_{dc} the drain voltage bias, I_{c} the dc current, The parameters for the inverse class F are denoted with a prime.

The waveforms in Figure 2.3 can be analyzed using their Fourier series expansion; for a class F amplifier, the waveforms, which include the effect of the on–state resistance, are given by [12]

$$I_{\mathrm{D}}(\theta) = I_{\mathrm{max}}\left[\frac{1}{\pi} + \frac{1}{2}\sin(\theta) - \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{\cos(2n\theta)}{4n^2-1}\right] \quad [\mathrm{A}], \tag{2.14}$$

$$V_{\mathrm{D}}(\theta) = V_{\mathrm{dc}} - \frac{4(V_{\mathrm{dc}} - V_{\mathrm{k}})}{\pi}\sum_{n=1}^{\infty}\frac{\sin((2n-1)\theta)}{2n-1} \quad [\mathrm{V}], \tag{2.15}$$

where $\theta = \omega t$ and ω is the fundamental frequency of the desired output. For an inverse class F power amplifier, the Fourier series expansion of the waveforms are given by

$$I'_{\mathrm{D}}(\theta) = \frac{I'_{\mathrm{max}}}{2}\left[1 + \frac{4}{\pi}\sum_{n=1}^{\infty}\frac{\sin((2n-1)\theta)}{2n-1}\right] \quad [\mathrm{A}], \tag{2.16}$$

$$V'_{\mathrm{D}}(\theta) = V'_{\mathrm{k}} + (V'_{\mathrm{max}} - V'_{\mathrm{k}})\left(\frac{1}{\pi} - \frac{1}{2}\sin(\theta) + \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{\cos(2n\theta)}{4n^2-1}\right) \quad [\mathrm{V}], \tag{2.17}$$

where

$$V'_{\max} = \pi V_{\mathrm{dc}} - (\pi - 1)\, V'_{\mathrm{k}}\,. \tag{2.18}$$

With these equations at hand, expressions for the output properties of each class (power efficiency, fundamental output power, etc) can be derived. In [12], the authors have derived equations for the power efficiency, output power, and the load resistance, all as a function of the on–state resistance. The results clearly show how the inverse class F mode amplifier outperforms the class F mode amplifier with increasing on–state resistance.

2.4 Matrix Factorizations Algorithms

A widely used approach for solving a linear system of equations is standard least squares methods. The problem is often to find a solution of an overdetermined system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{C}^{M\times N}$, $\boldsymbol{x} \in \mathbb{C}^N$, and $\boldsymbol{b} \in \mathbb{C}^M$ with $M > N$. Ideally, we would like to find the vector $\boldsymbol{x}$ that results in a zero–residual, however, in general there is no such solution [23, pp. 77].[5] Thus, we turn to the least squares solution, which is to find the vector $\boldsymbol{x}$ that minimizes the ℓ_2-norm of the residual $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$, or equivalently

$$\arg\min_{\boldsymbol{x}\in\mathbb{C}^N} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2\,, \tag{2.19}$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm (Euclidean norm), defined as

$$\|\boldsymbol{x}\|_2 = \left(\sum_{i=1}^{N} |x_i|^2\right)^{1/2}\,. \tag{2.20}$$

The simplest solution of (2.19), given $\boldsymbol{A}$ is of full–rank, is found by solving the normal equations

$$\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{b}\,, \tag{2.21}$$

which yields the unique solution

$$\boldsymbol{x} = \left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{b}\,, \tag{2.22}$$

where $\left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{H}}$ is known as the pseudoinverse of $\boldsymbol{A}$, denoted by $\boldsymbol{A}^{\dagger}$.[6] Although the solution in (2.22) is easily derived, computing it directly by numerical computations may lead to undesirable effects, depending on the condition number of $\boldsymbol{A}$.

The condition number of a matrix $\boldsymbol{A}$ describes the sensitivity of the solution $\boldsymbol{x}$ to perturbations in the data, and is thus an indication on the accuracy of the linear equation solution. It is defined as

[5] For zero residual, $\exists\, \boldsymbol{b} \in \mathrm{range}\,(\boldsymbol{A}) : \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, which rarely is the case since $\dim(\boldsymbol{b}) = M \times 1$ and $\mathrm{range}\,(\boldsymbol{A}) \leq N$, where $\mathrm{range}\,(\boldsymbol{A}) = \left\{\boldsymbol{y} \in \mathbb{C}^M : \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \text{ for some } \boldsymbol{x} \in \mathbb{C}^N\right\}$.

[6] $\mathrm{rank}\,(\boldsymbol{A}) = \dim(\mathrm{range}\,(\boldsymbol{A}))$, where $\dim(\cdot)$ denotes the dimension of $(\cdot)$.

$$\operatorname{cond}(\boldsymbol{A}) = \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\|, \tag{2.23}$$

however, if $\|\cdot\|$ in (2.23) is the ℓ_2-norm, or if $\boldsymbol{A}$ is rectangular, the condition number generalizes to [24, pp. 130]

$$\operatorname{cond}(\boldsymbol{A}) = \frac{\max \sigma(\boldsymbol{A})}{\min \sigma(\boldsymbol{A})}, \tag{2.24}$$

where $\sigma(\boldsymbol{A})$ is the set of singular values of $\boldsymbol{A}$. If $\operatorname{cond}(\boldsymbol{A})$ is small, $\boldsymbol{A}$ is said to be well–conditioned; if $\operatorname{cond}(\boldsymbol{A})$ is large, $\boldsymbol{A}$ is ill–conditioned.[7] Returning to the original problem in (2.19), we see that the solution given in (2.22) depends on the matrix product $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$. Recalling that the singular values of $\boldsymbol{A}$ are defined as

$$\sigma(\boldsymbol{A}) = \sqrt{\lambda(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})}, \tag{2.25}$$

where $\lambda\left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right)$ are the eigenvalues of $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$, it is easily seen that the singular values of $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ are found by

$$\begin{aligned}
\sigma\left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right) &= \sqrt{\lambda\left(\left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right)^{\mathsf{H}}\left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right)\right)}, && (2.26)\\
&= \lambda\left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right), && (2.27)\\
&= \left(\sigma(\boldsymbol{A})\right)^2, && (2.28)
\end{aligned}$$

which implies

$$\operatorname{cond}\left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right) = \left(\operatorname{cond}(\boldsymbol{A})\right)^2, \tag{2.29}$$

a result suggesting that if $\boldsymbol{A}$ is ill–conditioned, one should try to solve the linear system by other methods than just directly computing the solution in (2.22).[8] When the system is ill–conditioned, matrix factorization should be applied in order to solve the system with matrices that provide improved numerical properties in terms of accuracy and stability (see [23–25] for an overview of methods available for this purpose). The advantage of matrix factorization is even more important if the system is rank–deficient; in such scenarios there exist an infinite amount of solutions. If $\boldsymbol{x}$ is a minimizer of the residual, so is $\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{n} \in \operatorname{null}(\boldsymbol{A})$.[9] Thus, we got a convex set of minimizers

$$\mathcal{S} = \left\{\boldsymbol{x} \in \mathbb{C}^N \,:\, \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \min\right\}, \tag{2.30}$$

implying that $\mathcal{S}$ contains an element $\boldsymbol{x}_{\mathrm{LS}}$ which is the least ℓ_2-norm solution of the rank–deficient problem (see [26, pp. 256–257] for proof). The least ℓ_2-norm solution

[7] The two terms *small* and *large* are relative, however, the smaller $\operatorname{cond}(\boldsymbol{A})$ is, the better. For $\boldsymbol{A}$ singular, $\operatorname{cond}(\boldsymbol{A}) = \infty$.

[8] The eigenvalues of a square matrix $\boldsymbol{A} \in \mathbb{C}^{N\times N}$ are the N roots of the characteristic polynomial $p(\lambda) = \det(\lambda\boldsymbol{I}_N - \boldsymbol{A})$, where $\boldsymbol{I}_N$ is the identity matrix of dimension $N \times N$.

[9] $\operatorname{null}(\boldsymbol{A}) = \left\{\boldsymbol{x} = \mathbb{C}^N \,:\, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\right\}$.

can be obtained by any complete orthogonal factorization, such as the singular value decomposition, or QR factorization with column pivoting [26, pp. 255]. Since both these methods are frequently used in the numerical computations of this work, they are both briefly presented in the following.

QR Factorization

The purpose of QR factorization is to factorize a matrix $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ into the product of two matrices, an unitary matrix $\tilde{\boldsymbol{Q}} \in \mathbb{C}^{M\times N}$ and an upper triangular matrix $\tilde{\boldsymbol{R}} \in \mathbb{C}^{N\times N}$, in order to improve the numerical properties when solving a linear system of equations.

Definition 1 (Reduced form QR factorization [24]). *Given $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ where we express the successive spaces spanned by the colums $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N$ of $\boldsymbol{A}$ as*

$$\langle \boldsymbol{a}_1 \rangle \subseteq \langle \boldsymbol{a}_1, \boldsymbol{a}_2 \rangle \subseteq \langle \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3 \rangle \subseteq \ldots \,, \tag{2.31}$$

then we want to construct a sequence of orthonormal vectors $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots$ that span these successive spaces.[10] *With other words, the sequence $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots$ should have the property*

$$\langle \boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_i \rangle = \langle \boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_i \rangle \,, \qquad i = 1, \ldots, N \,, \tag{2.32}$$

which amounts to

$$\left[\begin{array}{c|c|c|c} & & & \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_N \\ & & & \end{array}\right] \cdot \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N} \\ & r_{21} & & \\ & & \ddots & \vdots \\ & & & r_{NN} \end{bmatrix} , \tag{2.33}$$

where the diagonal terms r_{ii} are nonzero. Then, due to the invertibility of the upper–left $i \times i$ block of $\tilde{\boldsymbol{R}}$ we can express the column vectors $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_i$ as a linear combination of the columns of $\boldsymbol{A}$, or

$$\boldsymbol{A} = \tilde{\boldsymbol{Q}}\tilde{\boldsymbol{R}} \,. \tag{2.34}$$

If $\boldsymbol{A}$ is rank–deficient, or close to rank–deficient, QR factorization with column pivoting should be used instead of regular QR factorization. This method uses the factorization $\boldsymbol{A\Pi} = \tilde{\boldsymbol{Q}}\tilde{\boldsymbol{R}}$, where $\boldsymbol{\Pi}$ is a permutation matrix, often chosen such that the diagonal elements of $\tilde{\boldsymbol{R}}$ are non–increasing; $|r_{11}| \geq |r_{22}| \geq \ldots \geq |r_{NN}|$.[11] Applying QR factorization to the system $\boldsymbol{Ax} = \boldsymbol{b}$ gives us the solution $\boldsymbol{x}_{\mathrm{LS}} = \tilde{\boldsymbol{R}}^{-1}\tilde{\boldsymbol{Q}}^{\mathsf{H}}\boldsymbol{b}$, which can be solved directly, or with i.e. back substitution.

[10] The notation $\langle\cdot\rangle$ indicates the subspace spanned by the vectors inside the brackets.

[11] This is the method used in MATLAB's built–in '\' operator, provided by the LAPACK library [27, pp. 32–33].

Singular Value Decomposition

The singular value decomposition (SVD) is a method for decomposing a matrix $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ into a product of three matrices which provice improved numerical properties when solving a linear system of equations. The SVD is considered to be one of the most powerful tools in linear algebra, and is defined as below.

Definition 2 (Reduced form SVD [28]). *Given $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ with singular values $\sigma(\boldsymbol{A}) = \{\sigma_1\, \sigma_2\, \ldots\, \sigma_N\}$, there exist two unitary matrices $\tilde{\boldsymbol{U}} \in \mathbb{C}^{M\times N}$ and $\tilde{\boldsymbol{V}} \in \mathbb{C}^{N\times N}$ such that the matrix*

$$\tilde{\boldsymbol{\Sigma}} = \tilde{\boldsymbol{U}}^{\mathsf{H}} \boldsymbol{A} \tilde{\boldsymbol{V}} \in \mathbb{C}^{N\times N} \tag{2.35}$$

is diagonal, or $\tilde{\boldsymbol{\Sigma}} = \operatorname{diag}(\sigma(\boldsymbol{A}))$. Thus any $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ is unitarily equivalent to a diagonal matrix

$$\boldsymbol{A} = \tilde{\boldsymbol{U}} \tilde{\boldsymbol{\Sigma}} \tilde{\boldsymbol{V}}^{\mathsf{H}}. \tag{2.36}$$

Consider the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Applying SVD to $\boldsymbol{A}$ and solving for $\boldsymbol{x}$ gives us the solution

$$\boldsymbol{x} = \tilde{\boldsymbol{V}} \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{U}}^{\mathsf{H}} \boldsymbol{b}. \tag{2.37}$$

Comparing (2.37) with (2.22), we see that the pseudoinverse of $\boldsymbol{A}$ can also be expressed as

$$\boldsymbol{A}^{\dagger} = \tilde{\boldsymbol{V}} \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{U}}^{\mathsf{H}} \in \mathbb{C}^{N\times M}, \tag{2.38}$$

where the scaling factors for the contributions from $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{V}}$ to $\boldsymbol{A}^{\dagger}$ originate from $\tilde{\boldsymbol{\Sigma}}^{-1}$. To guarantee numerical stability, only the singular values of $\boldsymbol{A}$ greater than some arbitrary tolerance should be used when computing (2.38) — so we avoid amplyfying values in (2.38) due to $\tilde{\boldsymbol{\Sigma}}^{-1} = \operatorname{diag}(1/\sigma(\boldsymbol{A}))$. Hence, even when $\boldsymbol{A}$ is rank–deficient, solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with SVD will yield the least ℓ_2-norm solution of all solutions — because we zero out the singular values smaller than our tolerance, e.g. if $\operatorname{rank}(\boldsymbol{A}) = r < N$, then $\tilde{\boldsymbol{\Sigma}}^{-1} = \operatorname{diag}(1/\sigma_1\, 1/\sigma_2\, \ldots\, 1/\sigma_r\, 0\, \ldots\, 0) \in \mathbb{C}^{N\times N}$, yielding $\boldsymbol{x}_{\mathrm{LS}} = \boldsymbol{A}^{\dagger}\boldsymbol{b}$.

2.5 Orthogonal Polynomials

As with matrix factorization, orthogonal polynomials can used in order to improve numerical properties in the estimation of model parameters for i.e. system identification [29, 30], or linearization [30, 31]. The orthogonal polynomial sequences are defined as below, and are available in many different forms [32, 33].

Definition 3 (Classical formulation [32]). *Let $[a, b]$ be a finite or infinite open interval on $\mathbb{R}$. An orthogonal polynomial is a set of polynomial sequences $\{p_n(x)\}$, $n \in \mathbb{N}_0$ that is orthogonal on $[a, b]$ with respect to a weighting function $w(x)$, or equivalently*

$$\int_a^b w(x)p_m(x)p_n(x)\mathrm{d}x = \delta_{mn}c_n\,,$$

where δ_{mn} is the Kronecker delta function, and c_n is a constant equal to unity for orthonormal $\{p_n(x)\}$.[12]

By using orthogonal polynomials in behavioral modeling, or digital predistortion linearization of microwave power amplifiers, the condition number for the regression matrix used in the linear system of equations can be reduced substantially, leading to a numerically robust estimation of the model parameters [34, pp. 144–145]. For this work, a small subset of the available orthogonal polynomial basis functions is implemented in MATLAB. Table 2.1 presents an overview over the basis functions implemented, with short comments specifying which type of the polynomial basis functions if more than one set exists (e.g. both regular and reversed Bessel polynomials exist).

[12] $\delta_{mn} = 1$ for $m = n$, and 0 otherwise.

Table 2.1: Orthogonal polynomial basis functions implemented in MATLAB to improve numerical properties of model parameter estimation procedures.

Name	Comments	Limits	Recurrence relation	Initial values
Bessel, B_n	reversed	$[0, \infty)$	$B_n(x) = \sum_{k=0}^{n} \frac{(2n-k)!}{(n-k)!k!} \frac{x^k}{2^{n-k}}$	$B_0(x) = 1, B_1(x) = x + 1$
Chebyshev, T_n	first kind	$[-1, 1]$	$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$	$T_0(x) = 1, T_1(x) = x$
Chebyshev, U_n	second kind	$[-1, 1]$	$U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x)$	$U_0(x) = 1, U_1(x) = 2x$
Hermite, H_n	probabilist	$(-\infty, \infty)$	$H_{n+1}(x) = xH_n(x) - nH_{n-1}(x)$	$H_0(x) = 1, H_1(x) = x$
Hermite, $\tilde{H}_n$	physicist	$(-\infty, \infty)$	$\tilde{H}_{n+1}(x) = 2x\tilde{H}_n(x) - 2n\tilde{H}_{n-1}(x)$	$\tilde{H}_0(x) = 1, \tilde{H}_1(x) = 2x$
Laguerre, L_n		$[0, \infty)$	$L_{n+1}(x) = \frac{1}{n+1}\left((2n+1-x)\,L_n(x) - nL_{n-1}(x)\right)$	$L_0(x) = 1, L_1(x) = -x + 1$
Legendre, $\tilde{L}_n$		$(-1, 1)$	$\tilde{L}_{n+1}(x) = \frac{1}{n+1}\left((2n+1)\,x\tilde{L}_n(x) - n\tilde{L}_{n-1}(x)\right)$	$\tilde{L}_0(x) = 1, \tilde{L}_1(x) = x$
Unnamed 1, $O_n^{(1)}$	see [30]	$[0, 1]$	$O_n^{(1)}(x) = \sum_{k=1}^{n} (-1)^{k+n} \frac{(k+n)!}{(k-1)!(k+1)!(n-k)!} x$	$O_0^{(1)}(x) = 1, O_1^{(1)}(x) = x$
Unnamed 2, $O_n^{(2)}$	see [31]	$[0, 1]$	$O_n^{(2)}(x) = \sum_{k=0}^{n} (-1)^{n-k} \frac{1}{k!} \binom{n}{k} x$	$O_0^{(2)}(x) = 1, O_1^{(2)}(x) = x - 1$

Design and Measurements of Microwave Power Amplifiers 3

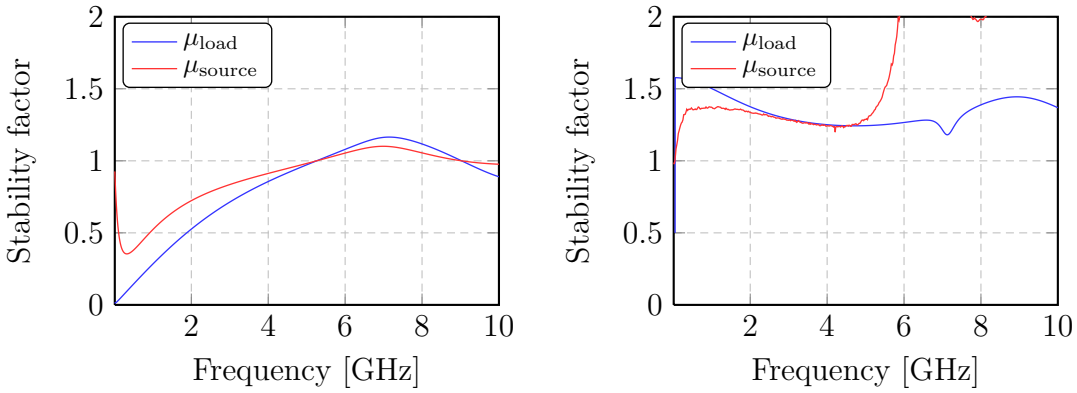
This chapter presents the design procedure of the test–board to be used in the load pull measurement setup. After this, a short introduction to the concept of source and load pull is given, followed by results from simulations and measurements of the designed and produced test–board. The chapter ends with a discussion and a summary.

3.1 Test–Board Design for Load Pull Characterization

The first step in the design process of the test–board for load pull characterizations, was to decide proper biasing conditions for the device. Results from the dc/bias simulations based on the nonlinear transistor model provided by Cree Inc. is given in Appendix H. As high–efficiency operation is desired, a small drain current is necessary; however, without sacrificing too much gain. The data sheet from Cree Inc. recommended a drain voltage of 28 V, which also was used. The trade–off was made based on studying and comparing the IV curves of the device, and curves for the maximum available gain (MAG) vs. drain and gate voltage, see Figure H.1a and H.1b. Based on the simulated results for the device a drain current of 11 mA was chosen, which together with the recommended drain voltage resulted in a MAG of approximately 17 dB.

Since the test–board was to be used with bias tees, no biasing network was designed. Hence, for the following simulations, ideal dc blocks and RF chokes were used in ADS while simulating stability. Initial simulations showed that the device was potentially unstable at all frequencies below 5 GHz. Thus, additional loss had to be introduced at the frequencies for which $\mu_{\text{load}} < 1$ or $\mu_{\text{source}} < 1$. The first attempt for stabilization was to just use a parallel RC high–pass filter in a series connection at gate. However, since the device was potentially unstable to up to 5 GHz, it was difficult to find components that introduced enough loss at all required frequencies without affecting the MAG too much. If a biasing network had been used in the design, the stabilization circuitry could have been done in the same manner as Cree recommends in the data sheet for the device; with a parallel RC high–pass filter in a series connection at gate together

with a series resistance in the gate bias network. However, this was not an option, so in addition to the high–pass filter, a series RC low–pass filter in shunt had to be used. Using this combination of stabilization network resulted in an unconditionally stable amplifier. Figure 3.1a and 3.1b show the stability factors before and after stabilization. Having done this, the design was completed, and the only step left was to add lines of the same length as the lines used in the TRL calibration kit design on the input and output of the DUT, see Appendix C.1. A layout for the final design of the test–board is presented in Appendix F.



(a) Before stabilization.

(b) After stabilization.

Figure 3.1: Simulated stability of test–board before and after stabilization.

3.2 Design Technology

In this section, the design technology used for the discrete PCB design is presented. This includes a short overview of the transistor, the capacitors, and the substrate used. Since the resistors used have no label, these are excluded from the following overview. They are however regular 0603–size surface mount resistors following the IEC 60063 standard.[1]

3.2.1 Transistor

The transistor used in this work is CGH4006P, a 6 W Gallium Nitride (GaN) high electron mobility transistor (HEMT) from Cree Inc. [35]. Over the last decade, GaN HEMTs have become more and more popular for microwave applications, and especially in base stations, where the properties of the GaN technology can be utilized to their full potential [2]. Some of the properties the technology provides is a relatively high power

[1]IEC is an acronym for International Electrotechnical Commission, an international standards organization.

density, which has been shown to be one–order of magnitude greater than its counterparts in Gallium Arsenide (GaAs) and Silicon (Si) [36]. Hence, for the same output power, a ten-times device size reduction can be realized using GaN HEMT technology, which effectively also results in lower input and output capacitances [3]. The increased power density is a result of a wide band gap. GaN HEMTs can also tolerate high peak operating voltages, which for i.e. commercial base station systems can either remove, or reduce the need for voltage conversion for low operating voltage devices, and thus reduce overall complexity. The high operating voltage also results in higher efficiency and reduced power requirement - which simplifies cooling [2]. The latter property is of great importance for base station operators, as running costs in general represent the biggest expense of the total cost for a base station. A detailed overview over many of the advantages GaN HEMTs provide can be found in [2].

3.2.2 Capacitors

The capacitors used in this work are ceramic multilayer capacitors from Johanson Technology's R14S series, which are designed for ultra-high Q and microwave applications. The capacitors exhibit negative–positive zero temperature (NP0) characteristics, meaning that the capacitance is relatively constant regardless of any variation in temperature [37]. Although designed for high–Q performance and minimum loss, the presence of parasitics is unavoidable. Figure 3.2 illustrates a simplified lumped-element equivalent circuit for a chip capacitor, where the parasitics are included.

Due to the contacts and the ceramic layers, a parasitic inductance, L_s in the illustration, will be present. The contacts and the electrode resistance will also give rise to a parasitic series resistance, R_s in the illustration [38, pp. 178]. Since the component is mounted on a microstrip, there will also be external parasitic capacitive effects to ground at both contacts, denoted with C_{g_1} and C_{g_2} in the illustration. The parasitic shunt capacitance, C_p in the illustration, represents the capacitive parallel-plate effect due to the electric field between the two electrodes. The impedance for the capacitance between the two electrodes is given by

$$Z_c = \frac{1}{\frac{1}{\mathrm{j}\,\omega C_\mathrm{p}} + \left(R_\mathrm{s} + \mathrm{j}\,\omega L_\mathrm{s} + \frac{1}{\mathrm{j}\,\omega C}\right)^{-1}} \quad [\Omega]\,, \tag{3.1}$$

and from this expression it is obvious that the capacitor will have both series and parallel resonance frequencies, where the series and the first parallel resonance frequencies are the dominant [38, pp. 167]. At frequencies below the series resonant frequency (SRF) it behaves as a capacitor. However, above the SRF, and below the first parallel resonance frequency, its total reactance is inductive and it behaves as an inductor. Hence, it is important to choose a capacitor that behaves as desired in the frequency band of interest. It should be noted that if minimal loss at a specific frequency is desired, the capacitor chosen should have its SRF at this frequency. From (3.1) we see that the capacitor will be represented by a small resistance, R_s, at the SRF because $C \gg C_\mathrm{p}$.[2] For the

[2] At SRF, $\omega L_\mathrm{s} = 1/(\omega C)$.

Johanson capacitors, the data sheet states that $R_\mathrm{s} \in [0.01, 1]\ \Omega \forall\, f \in [0.1, 4]$ GHz [37, pp. 15].

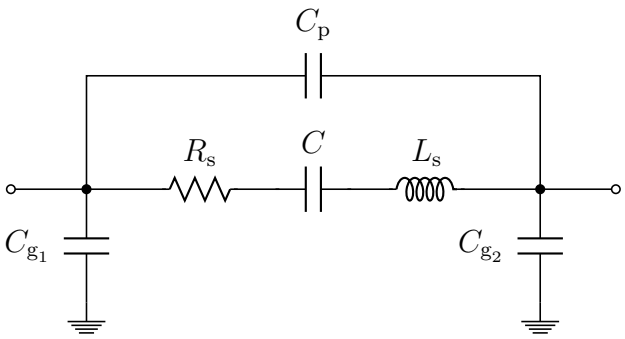


Figure 3.2: Simplified lumped-element equivalent circuit for a series mounted chip capacitor on a microstrip line.

3.2.3 Substrate

The substrate used for the PCB prototype designs is a Rogers 4003 low loss glass reinforced hydrocarbon/ceramic laminate [39]. The main substrate parameters, specified in the substrate data sheet, are summarized in Table 3.1. For the dielectric constant, the data sheets specifies one value recommended for circuit simulations (3.55), and one value for the process specification (3.38 ± 0.05), which can be confusing. Due to the ambiguous specification, the dielectric constant representing the physical value after routing a PCB was estimated, see Appendix C.1 for the procedure. The new substrate parameters based on this estimation are given in Table 3.2. In the tables, ϵ_r denotes the dielectric constant, and $\tan(\delta)$ the dielectric loss tangent.

Table 3.1: Substrate parameters for RO4003 with reference to Figure 3.3.

Parameter	ϵ_r	h [mm]	t [µm]	$\tan(\delta)$	$W_{50\,\Omega}$ [mm]
Value	$3.38 \pm 0.05/3.55$	0.51	36	0.00	1.14/1.10

Table 3.2: Modified substrate parameters for RO4003 with reference to Figure 3.3.

Parameter	ϵ_r	h [mm]	t [µm]	$\tan(\delta)$	$W_{50\,\Omega}$ [mm]
Value	3.30	0.51	36	0.00	1.15

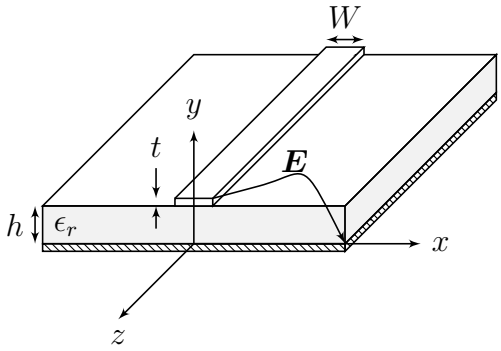


Figure 3.3: Microstrip on lossy substrate.

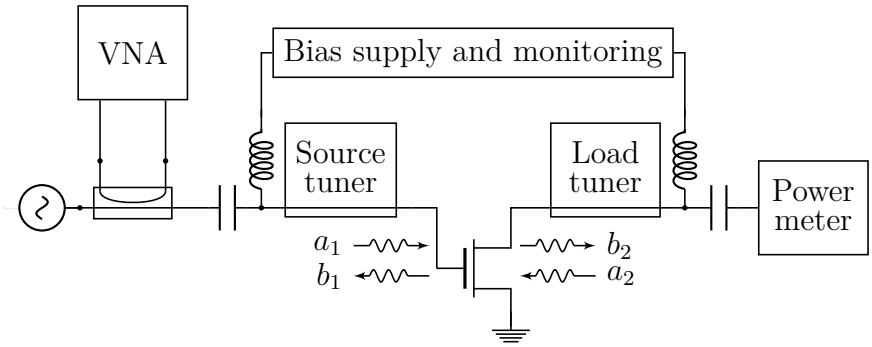
3.3 Load and Source Pull

Load/source pull is a measurement technique where the device–under–test (DUT) is presented with a whole range of load/source impedances together with a large signal input signal, while measuring the performance of the DUT, such as power gain, power efficiency and output power. The technique is suited for finding optimal operating conditions of a transistor for satisfying design specifications, as well as being suited for verifying nonlinear transistor models [40].
When the measurements are carried out in the frequency domain, a setup like the one illustrated in Figure 3.4a is often used. The illustration shows an example of an advanced classic load/source pull setup, where a load and source tuner are placed as close as possible to the DUT. The tuners are then used to provide the whole range of source/load impedances presented to the DUT, while the large signal input signal is generated at an external source/signal generator, often being subject to boosting in a driver PA in front of the source tuner. To measure the RF signals in the setup a vector network analyzer (VNA) is used at the source-side, together with a power meter at the load-side. In order to characterize and analyze the DUT, the measured data from the VNA and the power meter must be combined with the biasing currents and voltages, and the S parameters for the tuners. The S parameters of the tuners will differ for each realized source or load impedance, meaning that they are a function of the tuner settings [40].

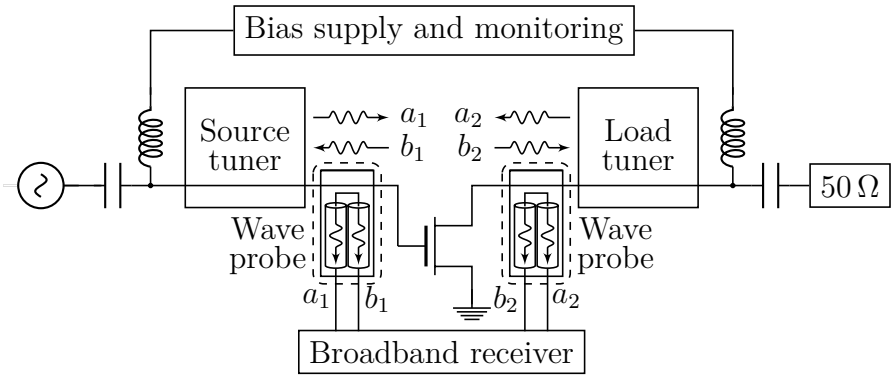
These functions are predetermined by a time consuming tuner calibration procedure. For performing load/source pull measurements the DUT in the time domain, a modern setup like the one illustrated in Figure 3.4b can be used. In this setup the RF signals are sensed with wave probes placed between the DUT and the tuners, allowing the incident and reflected waves (a and b waves) at the DUT terminals to be sensed. The wave probes, which basically can be viewed as directional couplers with low coupling factor that introduce low insertion loss [41], are connected to a broad-banded receiver.[3]

For the measurements carried out in this work, a combination of a classical and modern setup was used, see the illustration of set setup in Figure E.3. The SWAP-X402 receiver was not fully operational at the time of the measurements, so in order to measure

[3]At the microwave lab at NTNU the receiver is a SWAP-X402 from VTD [42].



(a) Schematic of classical load pull setup.



(b) Schematic of modern load pull setup.

Figure 3.4: Schematic of classical and modern load pull setups.

the input and output power levels, a power meter which measured the power level at the input of the source tuner and at the output of the load tuner was used. Together with monitoring of the dc supply, this was sufficient for the most common large signal measurements, e.g. measuring power gain, output power, and power efficiency.

3.4 Small Signal Measurements and Simulations

Prior to characterizing the large signal performance of the designs, small signal measurements were carried out, e.g. the S parameters were measured. For this purpose a setup like the one illustrated in Figure E.1 was used (Appendix E). Note the additional stabilization circuit placed between the dc supply and the gate side of the DUT. During initial measurements the network analyzer lost the phase-lock every other second most likely because the DUT was oscillating. To confirm instability, the DUT was connected to a Rohde & Schwarz FSQ40 Vector Signal Analyzer (VSA), and the output spectrum was analyzed. Immediately after biasing the DUT, several spikes in the frequency range 10 MHz to 30 MHz were seen on the screen, confirming that the DUT was oscillating and required stabilization. As a precaution, the DUT was designed with additional microstrip structures that could be used for stabilization if needed as a last

resort. See illustration of the design layout in Figure F.1, Appendix F.

However, using these additional structures would result in a different physical, and electrical length from the SMA connector on the gate side to the modified DUT, compared to the length used in the design of the TRL calibration kits. A difference in length would affect the accuracy in the calibration for the load-pull setup, and therefore it was decided to try to stabilize the DUT with an external stabilization circuit prior to altering the DUT design. The stabilization circuit designed consisted mainly of decoupling capacitors, and a resistor to provide a desirable impedance at low frequencies. See Figure 3.5 for an illustration of the circuit schematic.

To confirm that the stabilization circuit did not affect the performance at higher frequencies, the transmission thru the bias tee at the gate side of the DUT was measured with, and without the stabilization circuit attached. As can be seen from Figure 3.6a and Figure 3.6b, the stabilization circuit primarily introduces loss at low frequencies, leaving the transmission at higher frequencies unaltered (i.e. at 2 GHz, the additional loss is less than 0.05 dB, which is neglectable). It should be noted that since the reference plane was at the coaxial cables from the PNA, both scenarios (with, and without stabilization circuit attached) were expected to show a high-pass behavior. This is because the dc-block capacitor inside the bias tee represents a large impedance at low frequencies, becoming smaller with increasing frequency. Figure 3.6b shows that the attenuation at the lowest frequencies with the stabilization circuit attached is greater than when not using the stabilization circuit, indicating that the stabilization circuit works as desired.

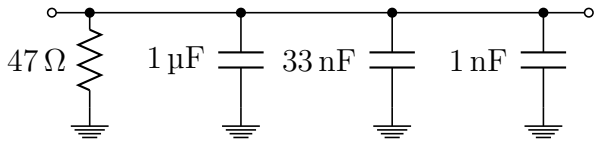
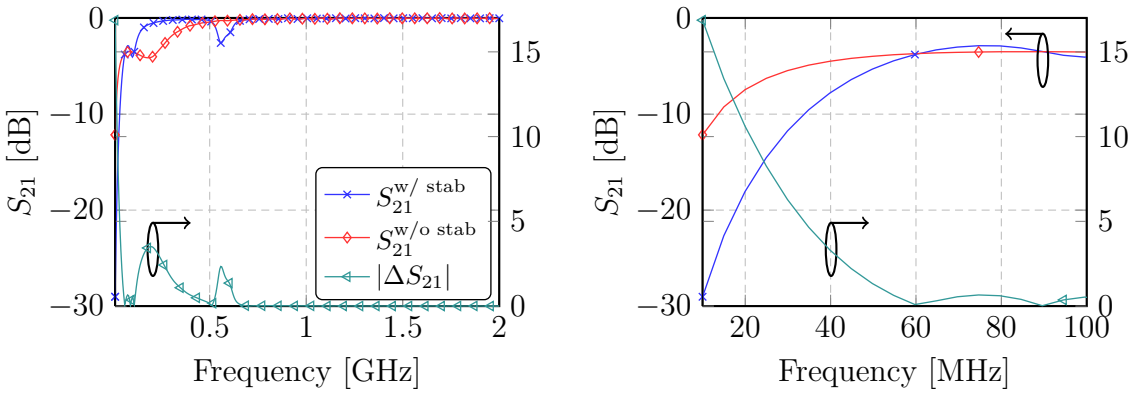


Figure 3.5: Simplified circuit schematic for the stabilization circuit designed. The components are soldered on a microstrip line, which has a total length of 40 mm. The dc supply is on the left-hand side, and the RF network at the right-hand side of the circuit.

3.4.1 S parameters for Design with $\epsilon_r = 3.55$

With the DUT stabilized, the S parameters were measured. The measurement setup used is illustrated in Figure E.1. For the measurements, the reference plane was at the SMA-connectors at the input/output of the DUT, so that any perturbation of the signals in the bias tees and all other components was included in the calibration.[4] In Figure 3.7a – 3.7d the measured and simulated forward mode parameters of the DUT, with no input/output matching, are presented. Due to the two attenuators, the signal level in reverse mode was heavily attenuated prior to reaching the DUT, which resulted in invalid results for the reverse mode parameters (output reflection coefficient and reverse transmission), e.g. the measurement of these parameters just showed noise.

[4]It should also be emphasized that the bias tees are designed for $f \in [1.5, 4.5]$ GHz, implying that measurements outside this frequency range should be taken with a grain of salt.



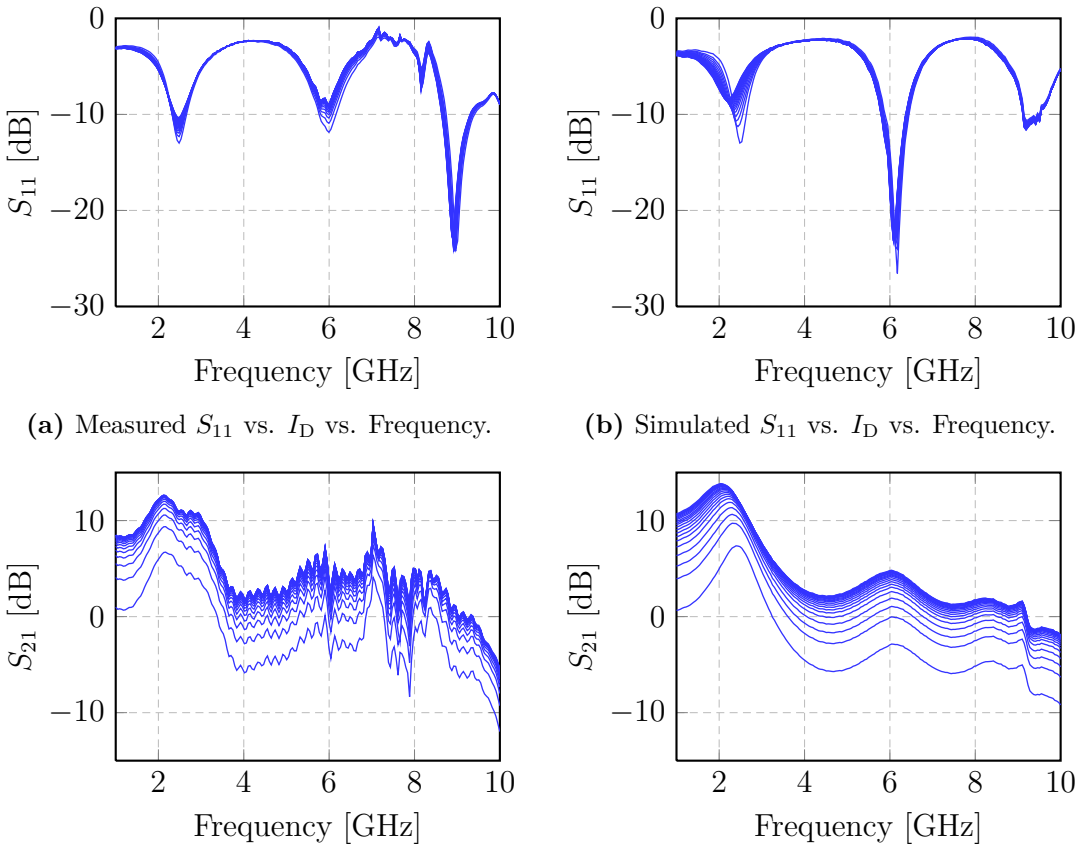
(a) $f \in [0.01, 2]$ GHz. **(b)** $f \in [10, 100]$ MHz.

Figure 3.6: Transmission thru the bias tee at the gate side, and how connecting the stabilization circuit affects the transmission with respect to loss at both low and high frequencies. In the figures $|\Delta S_{21}| = |S_{21}^{\text{w/ stab}} - S_{21}^{\text{w/o stab}}|$, with all parameters in dB.

As for the forward mode measurements, the figures confirm that the measured S parameters to some extent agree with the simulated ones. The measured input return loss is slightly better than the simulated for almost all biasing conditions, in addition to being closer to the desired center frequency. On the other hand, the measured small signal gain is slightly lower than the simulated one. The simulations show a small signal gain of about 16 dB when the drain current is set to 100 mA, whereas in the measurements the gain under equal biasing conditions is only 12 dB. The measured small signal gain do, however, agree well with the value given in the transistor data sheet [35], where it is stated that the typical value under similar biasing conditions at 2 GHz is 13 dB.

3.4.2 *S* parameters for Design with $\epsilon_r = 3.30$

The same measurement setup was used for measuring this design as the previous one. In Figure 3.8a – 3.8d the measured and simulated forward mode S parameters without input/output matching of the DUT are presented. Again, due to two attenuators used at the DUT output, the measured reverse mode parameters were useless, showing only noise. Hence, these are left out, together with the simulated ones as no comparison can be made. From the figures showing the forward mode parameters, we see that this design shows a slightly better gain at 2 GHz compared to the other design, which most likely is due to the impedance of the lines, which is closer to 50 Ω than the other design. Otherwise, we see again that the measured S parameters matched the simulated values relatively good, indicating that the nonlinear transistor model from Cree Inc. is accurate.



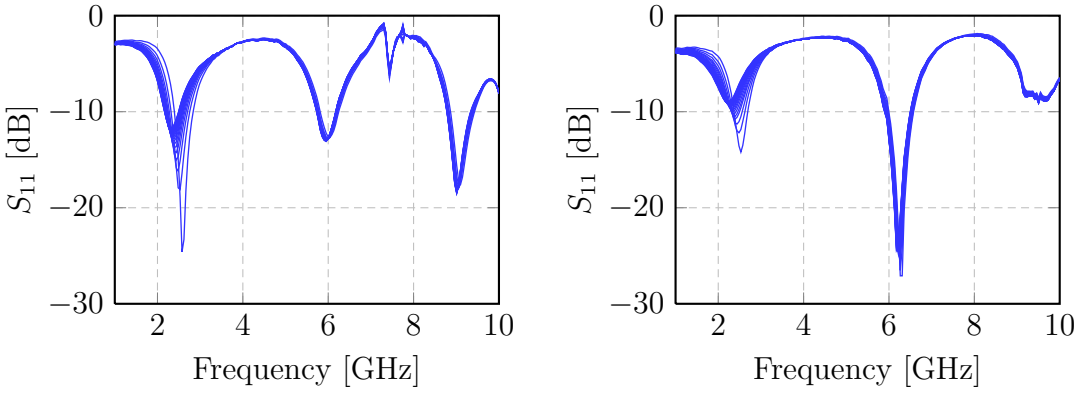
(a) Measured S_{11} vs. I_{D} vs. Frequency.

(b) Simulated S_{11} vs. I_{D} vs. Frequency.

(c) Measured S_{21} vs. I_{D} vs. Frequency.

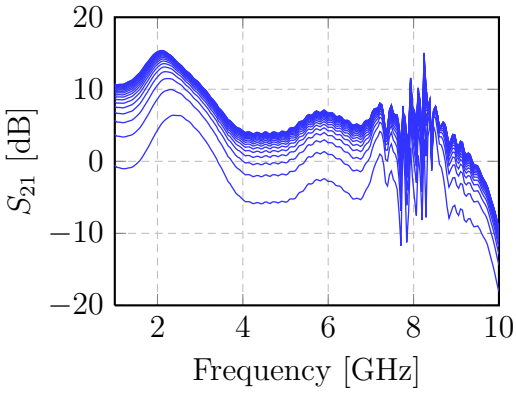
(d) Simulated S_{21} vs. I_{D} vs. Frequency.

Figure 3.7: Comparison of measured and simulated S parameters for design with $\epsilon_{\mathrm{r}} = 3.55$. In the measurements $I_{\mathrm{D}} \in [5, 100]\,\mathrm{mA}$ and $f \in [1, 10]\,\mathrm{GHz}$. The purpose of the plot is to show that the measured S parameters to some degree correspond with the simulated ones, but not comparing them for each and every biasing point. For this reason the S parameters under the different biasing conditions are plotted without line markers to distinguish them. See Figure G.2 for a 3D contour-plot of the same measurements if it is desirable to see the effect of the different biasing conditions.

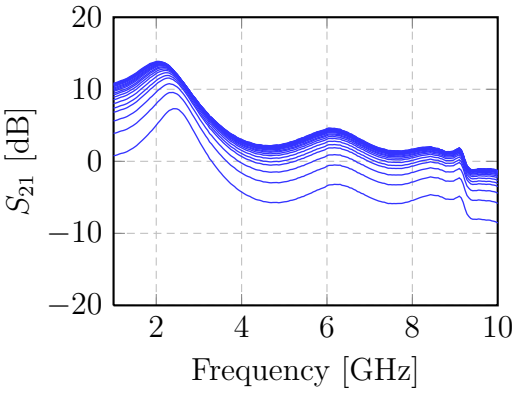


(a) Measured S_{11} vs. I_{D} vs. Frequency.

(b) Simulated S_{11} vs. I_{D} vs. Frequency.



(c) Measured S_{21} vs. I_{D} vs. Frequency.



(d) Simulated S_{21} vs. I_{D} vs. Frequency.

Figure 3.8: Comparison of measured and simulated S parameters for design with $\epsilon_{\mathrm{r}} = 3.30$. In the measurements $I_{\mathrm{D}} \in [5, 100]\,\mathrm{mA}$ and $f \in [1, 10]\,\mathrm{GHz}$. The purpose of the plot is to show that the measured S parameters to some degree correspond with the simulated ones, but not comparing them for each and every biasing point. For this reason the S parameters under the different biasing conditions are plotted without line markers to distinguish them. See Figure G.3 for a 3D contour-plot of the same measurements if it is desirable to see the effect of the different biasing conditions.

3.5 Large Signal Simulations and Measurements

In this section the results from the large signal simulations and measurements are presented and commented. All the simulations are based on a harmonic balance (HB) simulation in ADS, which analyze the analog RF circuit in both the frequency- and time domain, and use linear and nonlinear elements from this analysis to iteratively converge to a steady state solution at the fundamental frequency, and a predetermined number of harmonics [43]. The different sources and load impedances used in the HB simulation are found thru source and load pull simulations using a test bench implemented in [5, pp. 50–54], which to some degree imitates the functionalities of the multi-purpose tuners (MPT) from Focus Microwaves. The MPTs use a waveguide and three probes which can be moved horizontally and vertically in order to set the reflection coefficient at f_0, $2f_0$, and $3f_0$ independently. Biasing conditions are $V_\mathrm{D} = 28\,\mathrm{V}$ and $I_\mathrm{D} = 11\,\mathrm{mA}$ under all simulations, with other words deep class AB operation.

3.5.1 Large Signal Performance without Matching

In order to see the full effect of matching for optimal operating conditions, it is of interest to see how the device performs without matching, e.g. the device is presented with $50\,\Omega$ on both the input and output. From Figure 3.9a we see that even with no matching, acceptable power efficiencies can be obtained, peaking 50 %. More surprising is the output power, which peaks 10 W for both designs. This can suggest that the source and load impedances for maximum output power are located close to $50\,\Omega$ in the Smith chart. On the other hand, the gain curve shows that the power amplifier is quite nonlinear under the current biasing conditions. The shape of the gain curve is similar to what is common to see for power amplifiers biased close to class B operation [14, ch. 3], which this device is.

Table 3.3: Key figures for simulated performance with no input/output matching.

Design	η_PAE [%]		η_dc [%]		G [dB]		P_o [dBm]	
ϵ_r	$\eta_\mathrm{PAE}^\mathrm{max}$	$\eta_\mathrm{PAE}^{P_{1\,\mathrm{dB}}}$	$\eta_\mathrm{dc}^\mathrm{max}$	$\eta_\mathrm{dc}^{P_{1\,\mathrm{dB}}}$	G^max	$G^{P_{1\,\mathrm{dB}}}$	$P_\mathrm{o}^\mathrm{max}$	$P_\mathrm{o}^{P_{1\,\mathrm{dB}}}$
3.55	50.64	50.30	58.24	55.94	10.82	9.79	40.06	39.49
3.30	50.85	50.34	58.65	56.06	10.74	9.75	40.07	39.45

3.5.2 Load and Source Pull Simulations

In order to validate the nonlinear transistor model, four sets of reflection coefficients $\mathbf{\Gamma}^{(i)}$ for different output properties have been found. The different impedance sets consists of a source impedance at the fundamental, and load impedances at three different frequencies (f_0, $2f_0$, and $3f_0$), and can be found in Table 3.4. A graphical representation of the reflection coefficients in a Smith chart is given in Figure 3.4b. For the different sets,

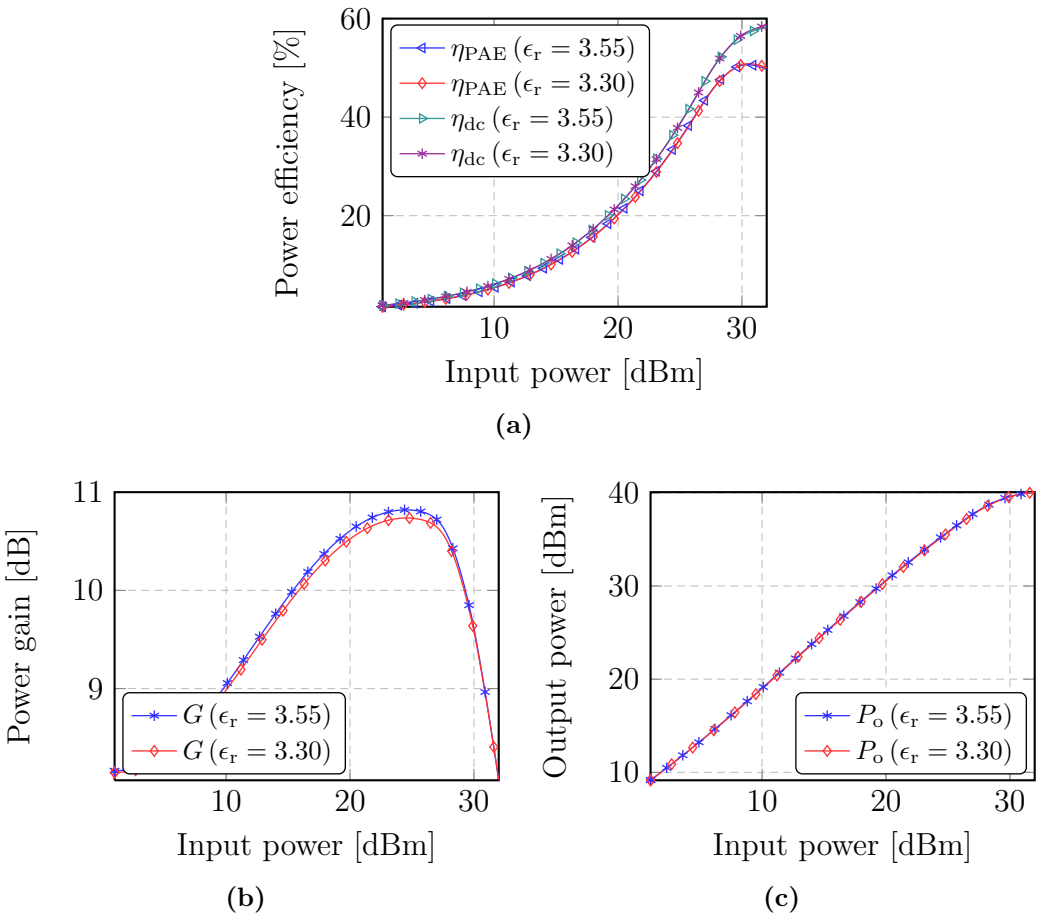


Figure 3.9: Simulated large signal properties for both designs without input/output matching.

different goals for the optimization were used. I.e. maximize PAE while maintaining an output power greater than some desired level, or a power gain variation over an input power range that is limited to i.e. 1 dB. In the following, the large signal properties obtained for each set of impedances are presented together with the optimization specifications used to find the set of impedances. All four sets were found by simulating the design with $\epsilon_{\mathrm{r}} = 3.30$, and then used directly together with the design for comparison.

Table 3.4: Sets of reflection coefficients based on load/source-pull simulations in ADS.

Set	$\Gamma_{\mathrm{S}}(f_0)$	$\Gamma_{\mathrm{L}}(f_0)$	$\Gamma_{\mathrm{L}}(2f_0)$	$\Gamma_{\mathrm{L}}(3f_0)$
$\boldsymbol{\Gamma^{(1)}}$	$0\angle 0°$	$0.56\angle 159.92°$	$0.99\angle 167.67°$	$0.99\angle 270.97°$
$\boldsymbol{\Gamma^{(2)}}$	$0.49\angle 152.08°$	$0.56\angle 171.58°$	$0.99\angle 182.87°$	$0.99\angle 261.04°$
$\boldsymbol{\Gamma^{(3)}}$	$0.67\angle 147.31°$	$0.63\angle 142.35°$	$0.99\angle 184.19°$	$0.99\angle 247.03°$
$\boldsymbol{\Gamma^{(4)}}$	$0.62\angle 122.83°$	$0.55\angle 151.25°$	$0.99\angle 176.40°$	$0.99\angle 253.63°$

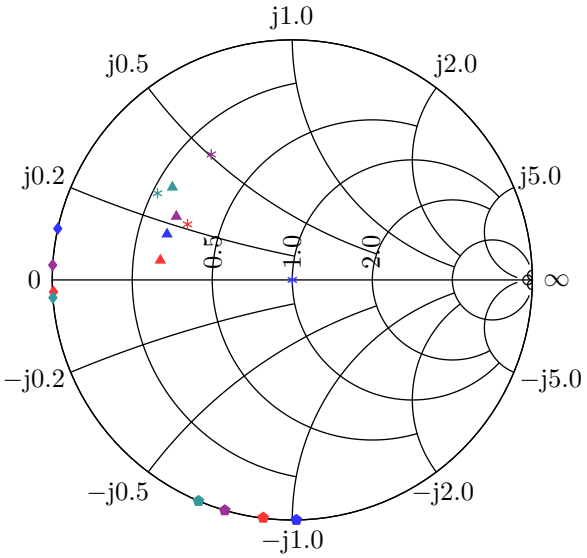


Figure 3.10: Impedances for various optimal output properties for CGH4006P found by load/source-pull simulations. Asterisk denotes $\Gamma_S(f_0)$, triangle denotes $\Gamma_L(f_0)$, diamond denotes $\Gamma_L(2f_0)$, and pentagon denotes $\Gamma_L(3f_0)$. Blue denotes $\mathbf{\Gamma^{(1)}}$, red denotes $\mathbf{\Gamma^{(2)}}$, teal denotes $\mathbf{\Gamma^{(3)}}$, and violet denotes $\mathbf{\Gamma^{(4)}}$. See Table 3.4 for details for each reflection coefficient.

Large Signal Performance when Matching to $\mathbf{\Gamma}^{(1)}$

For this simulation the optimization goals were specified in order to achieve a good overall performance, e.g. relatively flat gain, high power efficiency, and output power, while keeping the source impedance at 50 Ω. The reason for keeping the source impedance at 50 Ω was basically simplicity; to see if good performance could be achieved without the need of an input matching network. As for the optimization goals, the gain variation for the input power range $P_{in} \in [10, 28]$ dBm was set to a maximum of 1 dB.[5] In addition, an output power of at least 9 W at 1 dB compression was desired, so this level was specified as a minimum value. With these limits defined, a gradient based optimization was run in ADS to maximize power added efficiency.

The key figures achieved for both designs are summarized in Table 3.5, with plots for the different properties in Figure 3.11a – 3.11c. From the simulation results it is evident that the design with $\epsilon_r = 3.55$ shows improved performance compared to the other design. This is most likely due to the decreased line width, which results in less capacitance per unit length, and thus slightly smaller loss. This behavior is however not expected to be the same in the measurements, as the substrate has a physical ϵ_r which cannot be tweaked to make different 50 Ω line widths. We also see that both designs to some certain extent satisfy the optimization criteria; the design with $\epsilon_r = 3.55$ shows a

[5]Based on the simulated power gain close to 11 dB without input/output matching a power gain of 14 dB seemed achievable, therefore the interval for the power gain in the optimization was set to $G \in [13, 14]$ dB

gain variation that just exceeds the 1 dB limit, however, the excessive gain variation is so small that it is still within an acceptable level. The output power of both designs is slightly less than that the desired 9 W at 1 dB compression, however, both are still in the region of 8.50 W, which is considered good for a 6 W device. As for the power efficiency, both designs achieve a power added efficiency greater than 70 % at 1 dB compression, a promising result. A drain efficiency closing in on 90 % also indicates that the device clearly operates in a class F/inverse class F mode. Although the simulated results for the set $\mathbf{\Gamma}^{(1)}$ are good, it should be emphasized that designing an output matching network that realize the desired reflection coefficients at the harmonic frequencies can be a non-trivial task when they are located so close to unity (at the edge of the Smith chart). This applies to all impedance sets found in the simulations.

Table 3.5: Key figures for simulated performance when matching to $\mathbf{\Gamma}^{(1)}$.

Design	η_{PAE} [%]		η_{dc} [%]		G [dB]		P_{o} [dBm]	
ϵ_{r}	$\eta_{\text{PAE}}^{\text{max}}$	$\eta_{\text{PAE}}^{P_{1\,\text{dB}}}$	$\eta_{\text{dc}}^{\text{max}}$	$\eta_{\text{dc}}^{P_{1\,\text{dB}}}$	G^{max}	$G^{P_{1\,\text{dB}}}$	$P_{\text{o}}^{\text{max}}$	$P_{\text{o}}^{P_{1\,\text{dB}}}$
3.55	76.15	70.60	86.34	74.05	14.31	13.31	40.25	39.21
3.30	75.98	71.02	86.24	74.73	14.08	13.04	40.28	39.34

Large Signal Performance when Matching to $\mathbf{\Gamma}^{(2)}$

For this simulation, the main goal was to maximize the output power, while maintaining a power added efficiency greater than 65 % in 1 dB compression. Again, the gain variation was limited to 1 dB over the same input power range as in the simulations for $\mathbf{\Gamma}^{(1)}$, however, since an increased output power was expected, the min/max limits for the gain were adjusted to $G \in [14, 15]$ dB. A gradient based optimization was run to derive the optimal impedances. They key results obtained for both power amplifier designs are presented in Table 3.6, with figures for the whole input power range used presented in Figure 3.12a – 3.12c. As we can see, a maximum output power greater than 41.20 dBm, which corresponds to 13.10 W, is achieved. This is over twice the rated output power for the device, a remarkable result, even for a simulation. In addition, a power added efficiency greater than the desired 65 % is achieved. Otherwise, the two designs show quite similar results.

Table 3.6: Key figures for simulated performance when matching to $\mathbf{\Gamma}^{(2)}$.

Design	η_{PAE} [%]		η_{dc} [%]		G [dB]		P_{o} [dBm]	
ϵ_{r}	$\eta_{\text{PAE}}^{\text{max}}$	$\eta_{\text{PAE}}^{P_{1\,\text{dB}}}$	$\eta_{\text{dc}}^{\text{max}}$	$\eta_{\text{dc}}^{P_{1\,\text{dB}}}$	G^{max}	$G^{P_{1\,\text{dB}}}$	$P_{\text{o}}^{\text{max}}$	$P_{\text{o}}^{P_{1\,\text{dB}}}$
3.55	70.43	66.97	76.75	69.76	15.01	13.98	41.28	40.48
3.30	69.90	66.91	76.22	69.85	14.79	13.76	41.29	40.56

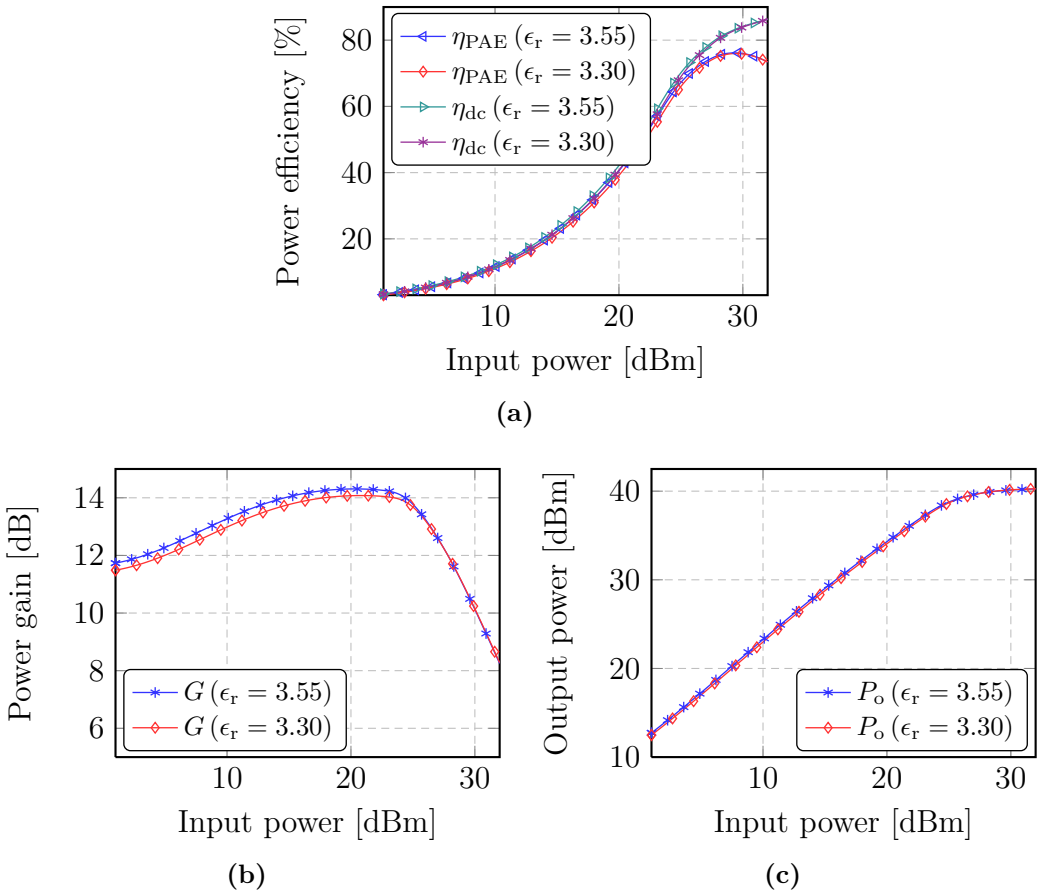


Figure 3.11: Simulated large signal properties for both designs matched to the impedances given in $\mathbf{\Gamma}^{(1)}$. Optimization goals were specified to achieve the best combined overall performance without the need for an input matching network.

Large Signal Performance when Matching to $\mathbf{\Gamma}^{(3)}$

The main goal of this simulation was to maximize the peak power added efficiency. The only limitation specified was the gain variation, which again was set to 1 dB over the same input power range as before, with $G \in [13.50, 14.50]$ dB specified as the min/max values. The key results achieved are summarized in Table 3.7, with figures showing the complete input power range used in the simulations in Figure 3.13a – 3.13c. Again both designs show promising results, achieving a power added efficiency of 78 %, and a drain efficiency greater than 90 %. Since the load impedance for maximum output power and maximum power efficiency lies somewhat far from each other (see Smith chart in Figure 3.10), the output power achieved when maximizing the power efficiency decreases rapidly. We see that an output power of 38.70 dBm is achieved for both designs, several dBs lower than in the case of matching to $\mathbf{\Gamma}^{(2)}$. However, this level still corresponds to more than 7 W, which is better than the rated output power for the device.

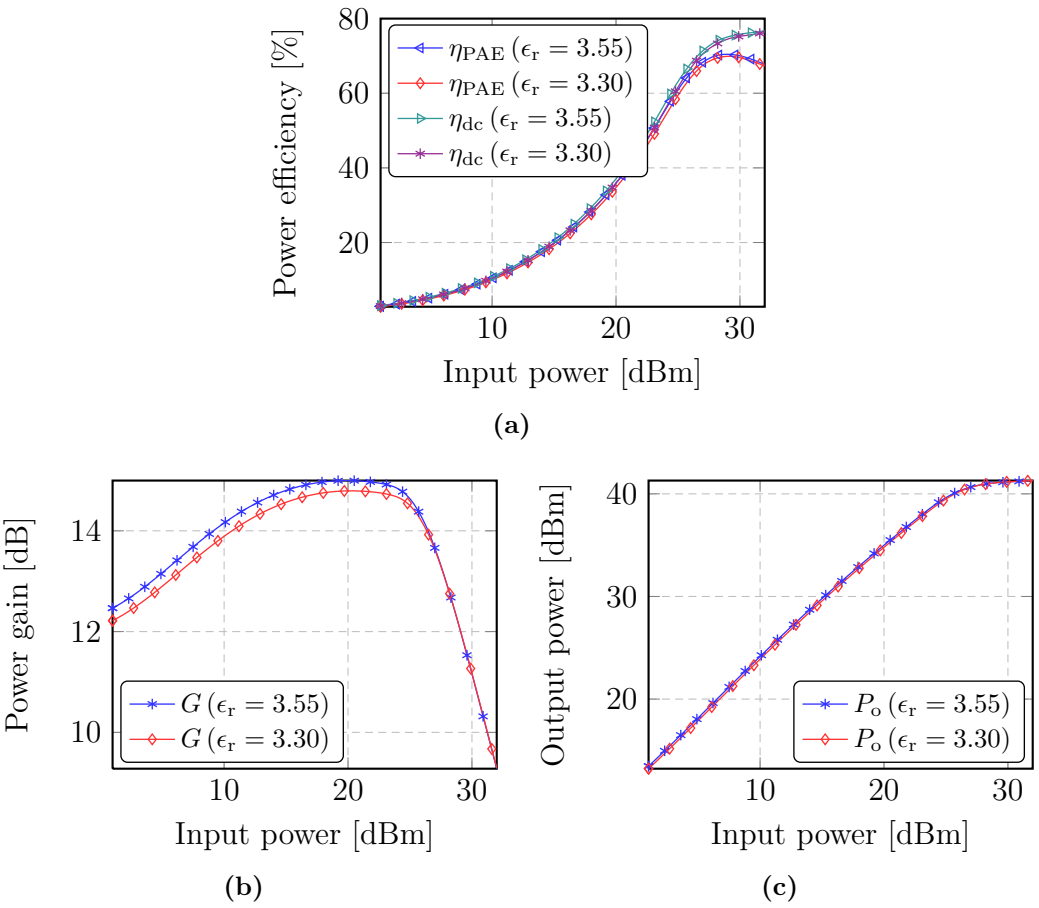


Figure 3.12: Simulated large signal properties for both designs matched to the impedances given in $\mathbf{\Gamma}^{(2)}$. Optimization goals were specified to achieve maximum output power while the peak power added efficiency was greater than 65 %.

Table 3.7: Key figures for simulated performance when matching to $\mathbf{\Gamma}^{(3)}$.

Design	η_{PAE} [%]		η_{dc} [%]		G [dB]		P_{o} [dBm]	
ϵ_{r}	$\eta_{\mathrm{PAE}}^{\mathrm{max}}$	$\eta_{\mathrm{PAE}}^{P_{1\,\mathrm{dB}}}$	$\eta_{\mathrm{dc}}^{\mathrm{max}}$	$\eta_{\mathrm{dc}}^{P_{1\,\mathrm{dB}}}$	G^{max}	$G^{P_{1\,\mathrm{dB}}}$	$P_{\mathrm{o}}^{\mathrm{max}}$	$P_{\mathrm{o}}^{P_{1\,\mathrm{dB}}}$
3.55	78.31	63.31	90.58	66.24	14.57	13.55	38.71	36.25
3.30	78.04	63.95	90.52	66.99	14.45	13.43	38.74	36.43

Large Signal Performance when Matching to $\mathbf{\Gamma}^{(4)}$

In this simulation the main goal was primarily maximum power added efficiency, as with the simulations for $\mathbf{\Gamma}^{(3)}$, however while trying to achieve a maximum output power of 10 W. Setting this minimum value for the output power clearly effects the achieved power efficiency. The peak values for the power efficiencies have decreased with a few percentage points compare to $\mathbf{\Gamma}^{(3)}$, however, the obtained values in 1 dB compression are

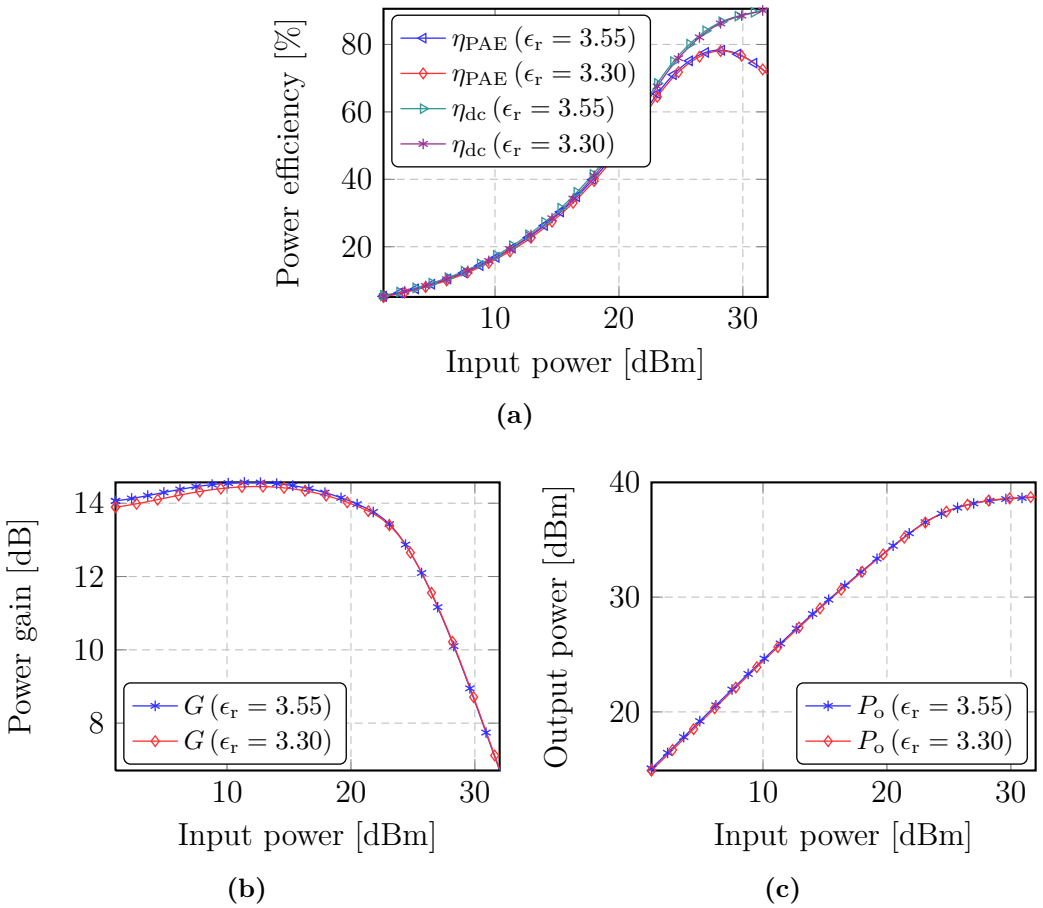


Figure 3.13: Simulated large signal properties for both designs matched to the impedances given in $\mathbf{\Gamma}^{(3)}$. Optimization goals were specified to achieve maximum power added efficiency.

several percentage points greater — which is due to the slower decreasing gain, as can be seen in Figure 3.14b. This effect is also seen for the output power; the peak value is almost 1 dB greater, and in 1 dB compression an increase of almost 2 dB is observed. Although the optimization goal with respect to the output power is not satisfied, the obtained peak output power is not much smaller. Given the good power efficiency, the overall performance that was desired prior to the optimization is achieved. The obtained results also indicate that the device operates in a class F or inverse class F mode. All key figures for this set of impedances can be found in Table 3.8.

3.5.3 Load and Source Pull Measurements

For the load pull measurements, the setup illustrated in Figure E.3 was used. All the equipment was intended to be controlled with the Load Pull Explorer (LPExp), provided by Focus Microwaves. In order to obtain proper and correct measurements, the software requires the user to provide S parameter files for all component blocks

Table 3.8: Key figures for simulated performance when matching to $\mathbf{\Gamma}^{(4)}$.

Design	η_{PAE} [%]		η_{dc} [%]		G [dB]		P_{o} [dBm]	
ϵ_{r}	$\eta_{\text{PAE}}^{\text{max}}$	$\eta_{\text{PAE}}^{P_{1\,\text{dB}}}$	$\eta_{\text{dc}}^{\text{max}}$	$\eta_{\text{dc}}^{P_{1\,\text{dB}}}$	G^{max}	$G^{P_{1\,\text{dB}}}$	$P_{\text{o}}^{\text{max}}$	$P_{\text{o}}^{P_{1\,\text{dB}}}$
3.55	77.39	70.32	87.43	73.84	14.25	13.21	39.87	38.51
3.30	77.29	70.28	87.52	73.94	14.09	13.06	39.90	38.56

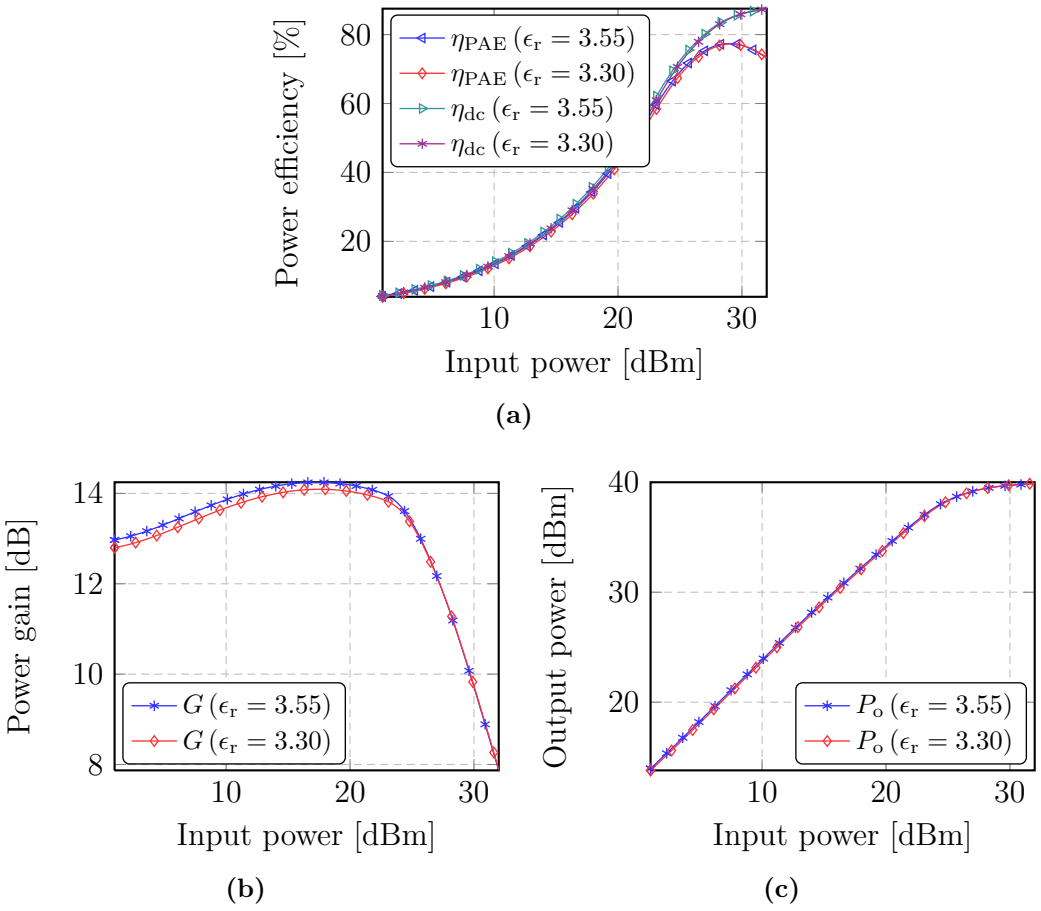


Figure 3.14: Simulated large signal properties for both designs matched to the impedances given in $\mathbf{\Gamma}^{(4)}$. Optimization goals were specified to achieve maximum power added efficiency, while trying to achieve a peak output power greater than 10 W.

in the measurement setup. This means that i.e. S parameters of the the input block, consisting of all the components between the output of the second driver PA and the input of the source tuner with respect to Figure E.3, had to be measured and provided to LPExp. Hence, both the input and output blocks were measured, for both paths in the coupler, e.g. from input port to the coupled port, and the input port to the output port - resulting in a total of four S parameter files for the input and output blocks.

In addition, the software requires S parameter files for the connection between the

tuners and the DUT, e.g. from the output of the source tuner to the input of the DUT, and the output of the DUT to the input of the load tuner. Since the DUT has an extra line length between the SMA connector and the desired reference plane which was used in the simulations, the provided S parameter files for this connection had to include the effects of the additional line, a seemingly straight forward measurement. However, measuring this turned out to be more troublesome than expected.

The input of the wave probes is an APC-7 (7 mm) connector, and the output a standard SMA connector (3.50 mm), which implies that without a 7 mm/3.50 mm calibration kit, the wave probes can not be measured directly. Thus, the VNA was calibrated with an APC-7 TRL kit from Focus Microwaves, and measurements were carried out of the cascaded connection of the wave probes and the TRL thru connection.[6] The measured S parameters represented now $\boldsymbol{M} = \boldsymbol{W}\boldsymbol{T}\tilde{\boldsymbol{W}}$, where $\boldsymbol{W}$ denotes the input wave probe, $\boldsymbol{T}$ the thru connection of the TRL kit, and $\tilde{\boldsymbol{W}}$ the output wave probe.[7]

The next step was to analyze and de–embed the measured S parameters $\boldsymbol{M} \in \mathbb{C}^{2\times 2}$, which had to be split into two equal S parameter matrices $\boldsymbol{M}^{\text{split}}$, one representing the connection from the output of the source tuner to the DUT, and one representing the connection between the output of the DUT to the input of the load tuner. Since the wave probes and the thru connection of the TRL kit introduces little or no reflections, M_{11} and M_{22} were both set to zero. The connection is also, in theory, a symmetric connection, which allowed for setting

$$|M_{12}^{\text{split}}| = \sqrt{\frac{|M_{12}| + |M_{21}|}{2}}\,,$$
$$|M_{21}^{\text{split}}| = \sqrt{\frac{|M_{12}| + |M_{21}|}{2}}\,,$$

and

$$\arg\left(M_{12}^{\text{split}}\right) = \frac{1}{2}\left(\frac{\arg\left(M_{12}\right) + \arg\left(M_{21}\right)}{2}\right) \quad [\text{rad}]\,,$$
$$\arg\left(M_{21}^{\text{split}}\right) = \frac{1}{2}\left(\frac{\arg\left(M_{12}\right) + \arg\left(M_{21}\right)}{2}\right) \quad [\text{rad}]\,,$$

where the square root of the magnitude is used to split the total measured loss into two equal contributions. Unfortunately, Focus Microwaves could not provide a driver for the power supply available at the microwave lab at NTNU, which restricted the type of measurements that could be performed with LPExp. In addition, the communication with the power meter timed out due to driver problems. To solve this, the power meter was operated in GPIB emulation mode, where it emulated a HP438A power

[6] The TRL thru connection here is from the TRL calibration kit designed for this work. The TRL kit based on $\epsilon_r = 3.30$ was used.

[7] If calculating the equation, ABCD parameters must be used.

meter, solving the problem, however, in a suboptimal way.[8] As for the power supply, an attempt was made on writing an own driver in C++, but with no luck as the available version of Microsoft's Visual Studio differed from the one that Focus recommended and based their driver walk-thru on.[9] Thus, LPExp was chosen for determining impedances for maximum power gain and output power. Measurements for power efficiency were performed using MATLAB, which communicated properly with both the power supply, and with the power meter in its native mode.

After settling down with the current solution, measurements were carried out with LPExp in order to locate the impedances that yielded maximum gain. The results from the first load pull measurement indicated additional problems in the measurement setup; multiple reflection coefficients (>5) yielded maximum gain, and especially values in the outer region of the Smith chart. Maximum gain in the outer region of the Smith chart indicates that the device can be potentially unstable; however, the external stabilization circuit was used in the measurement setup, and with no indication of oscillations when viewing the output spectrum on the spectrum analyzer, it was clear that the results from LPExp did not make any sense at all. Several new load pull measurements were carried out, however, the result kept repeating itself.

Even when measuring the thru connection of the TRL kit, a large gain was measured with LPExp, which obviously is incorrect. In order to ensure that the tuners presented the correct reflection coefficients to the DUT, they were connected to a HP8510C VNA and measured. The VNA was used to verify the reflection coefficients specified in LPExp. Hence, this was not the problem in the measurement setup. After additional testing the problem was not resolved. MATLAB was therefore chosen to control all the measurement equipment, because when communicating with the different equipment in the setup, MATLAB obtained the most reasonable results. The drawback with this solution was that MATLAB at the time being could not control the tuners, since a module in LPExp for external control from MATLAB was not provided by Focus Microwaves. Hence, the tuners had to be tuned manually for the measurements, meaning that the magnitude and phase of each of the three probes in both tuners had to be set manually.

With twelve free variables, this amounted to a time–consuming process — hence, the focus was turned to finding impedances for maximum gain. Prior to random tuning, the sets of impedances found in the load pull simulations were tested, but with no luck. The performance when specifying these reflections in LPExp was far from expected, and indicating additional problems in the setup. This could either be caused by the approach for de–embedding the S parameters of the two blocks between the tuners and the DUT, or by any other unknown problem. Anyhow, since no solution to the problem was found, manual tuning was the only option left. For this procedure, a random phase was chosen initially, and then the magnitude of the reflection coefficient was increased

[8]LPExp came with a driver for this equipment, which worked together with the ML2438A power meter when used in emulation mode.

[9]The version available version was Visual Studio 2010, whereas Focus had used Visual Studio 2005 in their driver guide. When trying to use their packages with Visual Studio 2010 error messages warned about outdated libraries and such.

in steps of 0.01. Going from 0.00 to 0.99, a region of magnitudes for which the gain was higher than for the other values tested was found.

Once this region of magnitudes was found, the phase was increased in steps of five degrees, starting at the initial phase, going a round in the Smith chart, and back to start. With this, a region for the phase for which the gain was higher was also found. After tuning to this region, the procedure started over, searching within the region with finer steps for the magnitude and the phase. Once a good reflection coefficient for the load at the fundamental was found, the same thing was repeated for the second and third harmonic, while keeping the reflection at the fundamental locked. Initially, the harmonic loads were swept in the outer region of the Smith chart, however, this did not yield any good results, so in the end it became more of random procedure.

For instance, the third harmonic was locked to a random reflection coefficient while sweeping the second harmonic. Using this procedure, the sets of impedances presented in Table 3.9 were found. $\mathbf{\Gamma}^{(5)}$ is included just for comparing the simulated and measured performance without matching, whereas $\mathbf{\Gamma}^{(6)}$ and $\mathbf{\Gamma}^{(7)}$ both yielded good performance with respect to output power and power efficiency, respectively. One limiting factor with respect to the obtained results that was discovered during the measurements was a voltage drop on drain. Although the measured drain voltage at the input of the bias tee was the desired 28 V, the measured drain voltage at the transistor terminal was for some input power levels more than 1 V lower than desired, altering the biasing conditions. It was, however, not enough time to make a workaround for this problem. Hence, in the results presented later, a drain voltage of 28 V is assumed for all input powers in the calculation of the drain and power added efficiency levels.

In the following, the obtained results are presented and evaluated. Due to the problems with the measurement setup described above, it is obvious that except for the case where all reflection coefficients are set to 50 Ω, a direct comparison of the performance achieved in the simulations and the measurements cannot be done. Recall also that when the term power gain is used, it is actually the transducer power gain that is measured; thus, for the measurement setup used for the load pull measurements, the input power P_{in} is defined as the power measured at the input of the source tuner, while the output power P_{out} is defined as the power measured at the output of the load tuner. In this way, the loss due to reflections in the tuners is also accounted for in the calculations.

Table 3.9: Sets of reflection coefficients based on load/source-pull measurements.

Set	$\Gamma_{\text{S}}(f_0)$	$\Gamma_{\text{L}}(f_0)$	$\Gamma_{\text{L}}(2f_0)$	$\Gamma_{\text{L}}(3f_0)$
$\mathbf{\Gamma^{(5)}}$	$0\angle 0°$	$0\angle 0°$	$0\angle 0°$	$0\angle 0°$
$\mathbf{\Gamma^{(6)}}$	$0\angle 0°$	$0.394\angle 120.70°$	$0.25\angle 180°$	$0.25\angle 270°$
$\mathbf{\Gamma^{(7)}}$	$0\angle 0°$	$0.555\angle 90.00°$	$0\angle 0°$	$0\angle 0°$

Large Signal Performance when Matching to $\mathbf{\Gamma}^{(5)}$

Comparing the measured results to the results obtained in the simulations with no input/output matching (see Table 3.3), we see that for the design with $\epsilon_r = 3.30$, an improved performance is achieved. The peak power added efficiency is three percentage points greater than in the simulations, and in 1 dB compression the power added efficiency is still one percentage point greater. In addition, the peak power gain is almost 1 dB greater in the measurements. Again, the surprising figure is the output power, which peaks almost 11 W, which is very good without matching. However, in 1 dB compression it is slightly less than 5 W, so the good peak is obtained when operated far into compression. As for the design with $\epsilon_r = 3.55$, we now see the real effect of an incorrect 50 Ω line width. In the validation of the TRL calibration kits, it was found that line widths for this card represented close to 51.50 Ω, which obviously affects the performance. This also confirms the fact that the results obtained in the simulations with this design are somewhat invalid; since ADS relies on correct user specified data for the substrate calculations, specifying incorrect data implies that multiple line widths can represent a 50 ohm line, depending on the parameters specified. However, for the real substrate, its parameters are given and cannot be changed, thus, only one line width will represent 50 Ω exactly. The key figures obtained for both designs are given in Table 3.10, with a graphical representation in Figure 3.15a – 3.15c.

Table 3.10: Key figures for measured performance when matching to $\mathbf{\Gamma}^{(5)}$.

Design	η_{PAE} [%]		η_{dc} [%]		G [dB]		P_{o} [dBm]	
ϵ_r	$\eta_{\text{PAE}}^{\max}$	$\eta_{\text{PAE}}^{P_{1\,\text{dB}}}$	$\eta_{\text{dc}}^{\max}$	$\eta_{\text{dc}}^{P_{1\,\text{dB}}}$	$G^{\max}$	$G^{P_{1\,\text{dB}}}$	$P_{\text{o}}^{\max}$	$P_{\text{o}}^{P_{1\,\text{dB}}}$
3.55	40.37	30.65	54.22	35.26	9.88	8.84	38.80	34.79
3.30	53.17	51.26	64.32	56.27	11.59	10.51	40.34	36.96

Large Signal Performance when Matching to $\mathbf{\Gamma}^{(6)}$

This impedance set represents the set for which maximum output power is obtained in the measurements. As can be seen from Table 3.11, the peak output power for the design with $\epsilon_r = 3.30$ peaks 11 W. However, this is again far into compression — in 1 dB compression the measured output power is slightly below 5 W, smaller than desired, but still acceptable considering the sub-optimal tuning of the load impedances. The peak power added efficiency for this set of impedances is measured to slightly less than 68.50 %, an uplifting result. In 1 dB compression it has decreased to 60 %. A power gain greater than 13 dB is also achieved, which is considered as a good figure. For obtained results for this set of impedances are somewhat comparable with the simulated results, but still not just as good. With automatic tuning it is likely that impedances in the region of those found manually could result in much better performance. For the design with $\epsilon_r = 3.55$ the measured results are not equally good as for the other design, but we see an improvement for all output properties compared to the ones measured for $\mathbf{\Gamma}^{(5)}$.

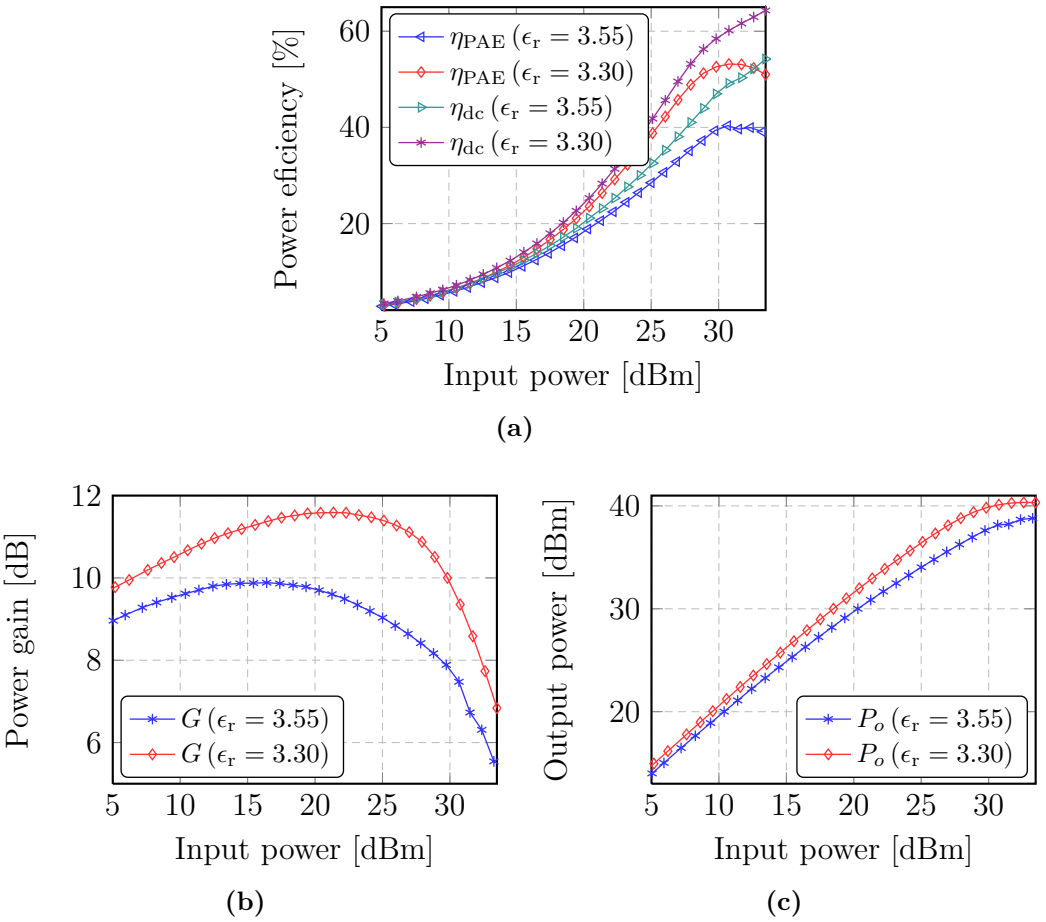


Figure 3.15: Measured large signal properties for both designs matched to the impedances given in $\mathbf{\Gamma}^{(5)}$.

Table 3.11: Key figures for measured performance when matching to $\mathbf{\Gamma}^{(6)}$.

Design	η_{PAE} [%]		η_{dc} [%]		G [dB]		P_{o} [dBm]	
ϵ_{r}	$\eta_{\mathrm{PAE}}^{\mathrm{max}}$	$\eta_{\mathrm{PAE}}^{P_{1\,\mathrm{dB}}}$	$\eta_{\mathrm{dc}}^{\mathrm{max}}$	$\eta_{\mathrm{dc}}^{P_{1\,\mathrm{dB}}}$	G^{max}	$G^{P_{1\,\mathrm{dB}}}$	$P_{\mathrm{o}}^{\mathrm{max}}$	$P_{\mathrm{o}}^{P_{1\,\mathrm{dB}}}$
3.55	59.69	55.22	75.94	62.60	10.26	9.28	39.76	38.11
3.30	68.49	60.89	80.89	64.83	13.14	12.17	40.45	36.55

Large Signal Performance when Matching to $\mathbf{\Gamma}^{(7)}$

The performance for this set of impedances yields the highest peak power efficiencies measured, for both designs. For the design with $\epsilon_{\mathrm{r}} = 3.30$, a peak power added efficiency of 70 % is measured in 3 dB compression. However, for 1 dB compression, only 59 % is obtained. The measured power gain is also the highest measured, peaking 13.30 dB. But again the output power disappoints in 1 dB compression; only a value of 4.10 W is measured. As for the other design, matching to these impedances yields the best

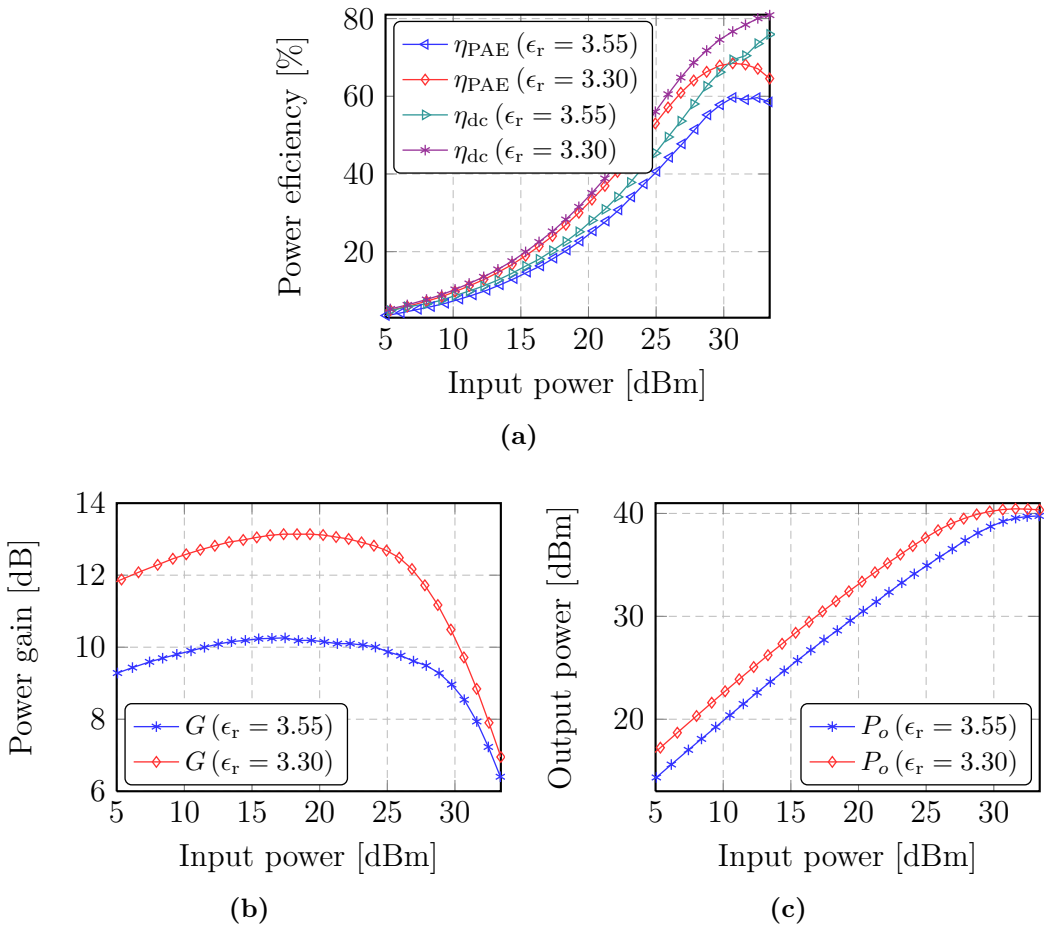


Figure 3.16: Measured large signal properties for both designs matched to the impedances given in $\mathbf{\Gamma}^{(6)}$.

measured performance for this design. Although the measured gain is almost 2 dB lower than for the other design, it is flatter and decreases slower for higher input power levels; thus, improved power efficiency and output power is measured in 1 dB compression. The key values of the measurements are summarized in Table 3.12, with a graphical representation for the whole measured input power range given in Figure 3.17a – 3.17c.

Table 3.12: Key figures for measured performance when matching to $\mathbf{\Gamma}^{(7)}$.

Design	η_{PAE} [%]		η_{dc} [%]		G [dB]		P_{o} [dBm]	
ϵ_{r}	$\eta_{\mathrm{PAE}}^{\mathrm{max}}$	$\eta_{\mathrm{PAE}}^{P_{1\,\mathrm{dB}}}$	$\eta_{\mathrm{dc}}^{\mathrm{max}}$	$\eta_{\mathrm{dc}}^{P_{1\,\mathrm{dB}}}$	G^{max}	$G^{P_{1\,\mathrm{dB}}}$	$P_{\mathrm{o}}^{\mathrm{max}}$	$P_{\mathrm{o}}^{P_{1\,\mathrm{dB}}}$
3.55	67.14	63.05	88.16	69.84	11.16	10.12	38.60	37.05
3.30	70.13	59.71	86.31	63.46	13.39	12.28	38.67	36.19

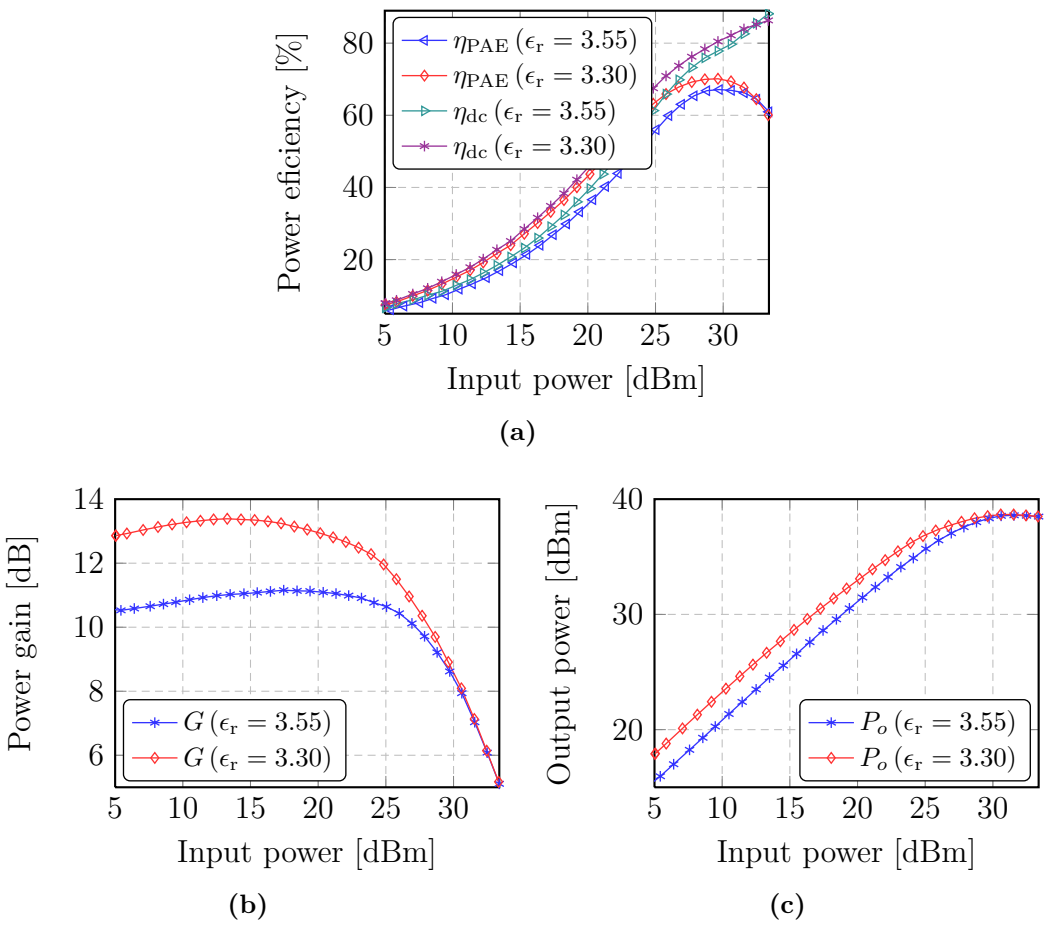


Figure 3.17: Measured large signal properties for both designs matched to the impedances given in $\mathbf{\Gamma}^{(7)}$.

3.6 Summary

This chapter addressed the design of a test–board made for source and load pull simulations and measurements. Prior to measurements, load pull simulations carried out in ADS showed that the 6 W device has the potential of delivering an output power greater than 13 W if operated far into compression. Simulations also showed that for reasonable operating levels, i.e. around 1 dB compression, output power levels in the region of 11.50 W are obtainable. When matching for maximum power efficiency, a drain efficiency of 90 % was achieved in 5 dB compression, indicating that the device is operating in class F or inverse class F mode. For maximum efficiency around 1 dB compression, power added efficiency levels in the region of 70 % were achieved, confirming that the device has a great potential for high–efficiency operation.

During initial measurements of the implemented test–board the power amplifier was oscillating, so an external stabilization circuit was made in order to ensure stable operation. With the device stabilized, load pull measurements were carried out. However, due to problems with the measurement setup and the software controlling the measure-

ment equipment, the tuning of the MPTs had to be done manually, which limited the possibility for verifying the simulated results. With manual tuning, impedances which resulted in an output power greater than 11 W were found, together with impedances that resulted in a power added efficiency of 70 %. Again, these figures were measured when operating in 4 dB to 5 dB compression. Around 1 dB compression the figures were not as good as in the simulations, but since these results were achieved with manual tuning of the equipment it is likely that the simulated results can be replicated if all the measurement equipment works optimally. The main factor that speaks against replication of the results is the location of the reflection coefficients at the harmonic frequencies in the Smith chart. When operating the MPTs in multi-harmonic mode (using three probes to set reflections at three different frequencies), the maximum achievable magnitude of the reflection coefficients is not much greater than 0.95 due to loss in the tuners. This can therefore limit how close to the simulated results the measurements can get. If the simulated results can be replicated in the load pull measurements, and for full power amplifier design, then the results will be comparable with the highest levels reported for such devices with an output power greater than 5 W [44].

Black-Box Modeling of Microwave Power Amplifiers 4

The first step in the design process of a digital predistorter is to derive an accurate model of the dynamic nonlinear system. Accuracy is vital for the linearization performance of the predistortion unit; the more accurate the model of the nonlinear system is, the better the linearization will be. There are many models available for characterizing a microwave power amplifier, ranging from models which describes the system on transistor level (using equivalent circuits of the active device, its package parasitics and the rest of the circuitry around it), to higher-level parametric black-box models, which describe the dynamic nonlinear system solely based on the observed input/output data. A large class of such behavioral models, with focus on microwave power amplifiers, is covered in [34], with additional literature in [45–47].

In this chapter a set of parametric nonlinear models for microwave power amplifiers is presented. The models discussed range from the most widely investigated model for nonlinear dynamic systems, the Volterra series [48–50], to subsets of the Volterra series, such as the Wiener- and Hammerstein models [45, 46, 51], and the memory polynomial- and spline delay envelope (SDE) models [52–55]. It is not the scope of this thesis to present a detailed mathematical analysis of these models, however, the different models are presented such that their performance can be easily evaluated and compared with respect to a set of suitable figure merits presented in the chapter.

Finally, the performance of the presented models is evaluated against both simulated and measured input/output data from different microwave power amplifiers. The simulation environment used consists of Agilent's simulation tool for RF, microwave and signal integrity applications, Advanced Design Systems (ADS), and Mathworks' MATLAB®.[1] The advantage with this type of environment is that no limitations in the measurement equipment need to be taken into account, such as limited resolution and bandwidth in analog-to-digital converters (ADCs). In addition, perturbing effects such as measurement noise and other limitations in the equipment which will corrupt the signal will not have any effect on the result. However, a simulation environment like described in Appendix B will give an overview of the performance under under ideal conditions, which are not in exact accordance with reality. Hence, in order to avoid making conclusions

[1] See Appendix B for a detailed description of the setup used, and a signal flow chart.

based on incorrect assumptions, the different models are also tested with measured input/output data of a physical device. The chapter ends with a discussion of the obtained results.

4.1 The Volterra Series

The Volterra series is a functional power series introduced by Vito Volterra in 1887 [48], and was first used in nonlinear system theory by Norbert Wiener in 1947 [56]. It has the continuous time input/output relation given by [49, pp. 77–80]

$$y(t) = h_0 + \sum_{p=1}^{\infty} \int \cdots \int h_p\left(t, \tau_1, \tau_2, \ldots, \tau_p\right) x(\tau_1) x(\tau_2) \cdots x(\tau_p)\, \mathrm{d}\tau_1\, \mathrm{d}\tau_2 \cdots\, \mathrm{d}\tau_p\,, \tag{4.1}$$

where $y(t)$ and $x(t)$ are the continuous time input and output signals, respectively, $h_p(\cdot)$ for $p \geq 1$ are the continuous time Volterra kernels, and h_0 is the constant zeroth-order kernel. In 1910, Maurice Fréchet proved that a set of Volterra functionals is complete [57], implying that every continuous functional of a signal $x(t)$ could be approximated with arbitrary precision as a sum of a finite number of Volterra functionals in $x(t)$ [50, pp. 15], a generalization of the Stone-Weierstrass theorem.[2]

Although being complete, the series has its limitations when there are discontinuities, or strong nonlinearities in the system [59]. However, for RF power amplifiers, commonly classified as mildly nonlinear systems under normal operating conditions, the Volterra series is a great tool for behavioral modeling, and has been widely used for this purpose during the past decades.[3] Even though (4.1) can be used to model many nonlinear dynamic systems, it has the disadvantage of being an infinite series. Thus, for practical reasons, a truncated and causal Volterra series is considered in the following. The input/output relation of the truncated series is given by

$$\begin{aligned} \tilde{y}(t) &= \sum_{p=1}^{2P+1} \int \cdots \int \tilde{h}_p\left(\tau_1, \ldots, \tau_p\right) \tilde{x}\left(\tau_1\right) \cdots \tilde{x}\left(\tau_p\right)\, \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_p\,, \\ &= \sum_{p=1}^{2P+1} \int \tilde{h}_p\left(\boldsymbol{\tau}_p\right) \prod_{i=1}^{p} \tilde{x}\left(t - \tau_i\right)\, \mathrm{d}\boldsymbol{\tau}_p\,, \end{aligned} \tag{4.2}$$

where $\boldsymbol{\tau}_p = [\tau_1 \ \ldots \ \tau_p]^{\mathsf{T}}$ is the time-arguments for the p-dimensional kernel $\tilde{h}_p\left(\tau_1, \ldots, \tau_p\right)$, and $\mathrm{d}\boldsymbol{\tau}_p = \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_p$ is used for notational simplicity. Again the input and output signals are denoted by $x(t)$ and $y(t)$, respectively, and the tilde mark denotes that the variables are real-valued bandpass. The complex baseband representation of (4.2) is

[2] The Stone-Weierstrass theorem states that every function $f(x)$ can be approximated with arbitrary precision as a sum of a finite number of polynomials in x [58].

[3] A search for *volterra amplifier* on IEEE Xplore reveals the amount of attention the series has gotten for this purpose — and this only for IEEE publications.

$$y(t) = \sum_{p=0}^{P-1} \int h_{2p+1}\left(\boldsymbol{\tau}_{2p+1}\right) \prod_{i=1}^{p+1} x\left(t-\tau_i\right) \prod_{i=p+2}^{2p+1} x^*\left(t-\tau_i\right) \, \mathrm{d}\boldsymbol{\tau}_{2p+1} \,, \tag{4.3}$$

where the notation is as in (4.2).[4] Although complex baseband input/output representation now is established, we seek a discrete time complex baseband relationship in order to design the digital predistorter. In the digital predistortion system considered in this work, the output is sampled after attenuation and down-conversion, before it is used together with the input to estimate the Volterra kernels.

However, sampling the output at twice the bandwidth can be challenging because of the spectral broadening that occurs due to the nonlinear amplification of the input signal. In [7] it is stated that sampling an output signal to be used in a digital predistortion system may for some amplifiers require ADCs with a resolution of typically 12 to 14 bits supporting a dynamic range of up to 70 dB, which will increase both cost and power consumption. For system identification, however, it can be shown that it is sufficient to sample the output with the same rate used to sample the input [1, 60, 61]. If the input signal is assumed to be band-limited to $[-B, B]$, and together with the output signal sampled at the Nyquist rate $T = \frac{1}{2B}$, it can be shown that the discrete time input/output relation of the $(2P-1)$th order Volterra series is given by [7, pp. 16 – 18]

$$y[n] = \sum_{p=0}^{P-1} \sum_{\boldsymbol{m}_{2p+1} \in \mathbb{Z}} h_{2p+1}\left[\boldsymbol{m}_{2p+1}\right] \prod_{i=1}^{p+1} x\left[n-m_i\right] \prod_{i=p+2}^{2p+1} x^*\left[n-m_i\right] \,, \tag{4.4}$$

where $h_{2p+1}\left[\boldsymbol{m}_{2p+1}\right]$ is the discrete time equivalent of $h_{2p+1}\left(\boldsymbol{\tau}_{2p+1}\right)$ in (4.3).[5] The input and output signals are again denoted by $x[n]$ and $y[n]$, respectively, while the vector $\boldsymbol{m}_{2p+1} = \left[m_1 \, m_2 \, \ldots \, m_{2p+1}\right]^{\mathsf{T}}$ represents the delay elements for the individual kernels.

With this relationship established, we have the situation depicted in Figure 4.1, where $\mathcal{V}^c$ denotes the continuous time Volterra system defined by the continuous time kernels, and $\mathcal{V}^d$ denotes the discrete time Volterra system defined by the discrete time kernels [1].

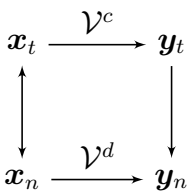


Figure 4.1: Commutative diagram showing the input/output relationship between continuous time and discrete time Volterra series [1].

[4] See Appendix A for the derivation and the assumptions made for making the relationship valid.

[5] By band-limiting the input to $I = [-B, B]$, and assuming that the spectral components outside I are sufficiently small, it can be assumed that the output also is band-limited, in a hypercube $C = I \times I \times \ldots \times I$, because the kernels outside C will not be excited by the input signal [7, pp. 17].

The diagram shows that if the Nyquist condition is satisfied, the paths $\boldsymbol{x}_t \xrightarrow{\mathcal{V}^c} \boldsymbol{y}_t \longrightarrow \boldsymbol{y}_n$ and $\boldsymbol{x}_t \longrightarrow \boldsymbol{x}_n \xrightarrow{\mathcal{V}^d} \boldsymbol{y}_n$ are equivalent. This is because the input $\boldsymbol{x}_t$ uniquely determines, and is uniquely determined by, its samples [1].

4.1.1 Parameter Estimation for the Volterra Series

From (4.4) it can be seen that the output of the Volterra series is linear in parameters with the kernels. This implies that the estimation of the kernels can be treated as a standard least squares problem, estimating the coefficients in a linear system.

Since we only consider odd-ordered terms, the regression matrix $\underline{\boldsymbol{H}}$ is composed in a way which reduces the number of coefficients required considerably, compared to a full Volterra series. As can be seen in the derivation of (4.4) in Appendix A, the kernels are symmetric in the first $k+1$ and last k components. This allows us to disregard the terms with equal signal products from the regression matrix; i.e. if we consider a fifth order kernel ($k = 2$) with a memory depth $m_5 = 1$, the signal products

$$x[n]x[n]x[n-1]x^*[n]x^*[n-1] \quad \text{and} \quad x[n]x[n-1]x[n]x^*[n]x^*[n-1]$$

are equal, implying that one of them can be ignored in the composition of the regression matrix.[6] After disregarding a copy of all the symmetric terms in the composition of the fifth order kernel with a one-tap memory, the kernel will, in matrix form for readability, look like

$$\tilde{\boldsymbol{h}}_5^{\mathsf{T}} = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&0&1&1&1&1&1&1 \\ 0&0&0&1&1&1&1&1&1&1&1&1 \\ 0&0&1&0&0&1&0&0&1&0&0&1 \\ 0&1&1&0&1&1&0&1&1&0&1&1 \end{bmatrix}^{\mathsf{T}},$$

where non-zero elements indicate a one-tap memory, i.e. $\tilde{\boldsymbol{h}}_5^{(3)} = [0\,0\,0\,1\,1]^{\mathsf{T}}$ equals the signal product $x[n]x[n]x[n]x^*[n-1]x^*[n-1]$. The number of coefficients for each kernel of odd-order p can be shown to be on the form [62]

$$\nu_p = \binom{m_p + \tilde{p}}{\tilde{p}} \binom{m_p + \tilde{p} + 1}{\tilde{p} + 1}, \tag{4.5}$$

where m_p is the memory depth for the kernel of order p, and $\tilde{p} = (p-1)/2$ is used for notational simplicity. For the same example as above, the number of coefficients for the fifth order kernel will be 12, as can be seen by the number of columns in $\tilde{\boldsymbol{h}}_5^{\mathsf{T}}$. The total number of coefficients for the Volterra series of order $2P-1$ in (4.4) is

[6] It should be emphasized that in order to avoid linearly dependent columns in the regression matrix, these terms must be disregarded in the composition of the matrix. With linearly dependent columns, the regression matrix will suffer from rank deficiency, which may lead to poor results when solving the linear system.

$$\nu_{\text{tot}} = \sum_{\substack{p=1 \\ p \text{ odd}}}^{2P-1} \nu_p \in \mathbb{N}\,. \tag{4.6}$$

With the composition of the parameter vector established, we express the input/output relation in matrix form, which is more convenient for solving the least squares problem. The parameter vector for the model of order $2P-1$ consisting of only odd orders is given by

$$\boldsymbol{c} = \left[\boldsymbol{h}_1^{\mathsf{T}}\, \boldsymbol{h}_3^{\mathsf{T}}\, \ldots\, \boldsymbol{h}_{2P-1}^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{C}^{\nu_{\text{tot}}}\,. \tag{4.7}$$

The corresponding regression matrix $\underline{\boldsymbol{H}} \in \mathbb{C}^{N\times\nu_{\text{tot}}}$ is composed of multiple sub-matrices $\boldsymbol{X}_p \in \mathbb{C}^{N\times\nu_p}$, each associated with the respective kernel vector $\boldsymbol{h}_p \in \mathbb{C}^{\nu_p}$,

$$\underline{\boldsymbol{H}} = [\boldsymbol{X}_1\, \boldsymbol{X}_3\, \ldots\, \boldsymbol{X}_{2P-1}]\,. \tag{4.8}$$

Here the sub-matrices $\boldsymbol{X}_p$ consists of the signal products for the respective order

$$\boldsymbol{X}_p = \left[\boldsymbol{x}_{p,n}\, \boldsymbol{x}_{p,n-1}\, \ldots\, \boldsymbol{x}_{p,n-(N-1)}\right]^{\mathsf{T}}\,, \tag{4.9}$$

where each vector $\boldsymbol{x}_{p,n} \in \mathbb{C}^{\nu_p}$ is given by

$$\begin{aligned}
\boldsymbol{x}_{p,n} = \Big[& \underbrace{x[n]\cdots x[n]}_{\tilde{p}+1}\underbrace{x^*[n]\cdots x^*[n]}_{\tilde{p}}\ldots \\
& \underbrace{x[n]\cdots x[n]}_{\tilde{p}+1}\underbrace{x^*[n]\cdots x^*[n-(m_p-1)]}_{\tilde{p}}\ldots \\
& \underbrace{x[n]\cdots x[n]}_{\tilde{p}+1}\underbrace{x^*[n-(m_p-1)]\cdots x^*[n-(m_p-1)]}_{\tilde{p}}\ldots \\
& \underbrace{x[n-(m_p-1)]\cdots x[n-(m_p-1)]}_{\tilde{p}+1}\underbrace{x^*[n-(m_p-1)]\cdots x^*[n-(m_p-1)]}_{\tilde{p}}\Big]^{\mathsf{T}},
\end{aligned} \tag{4.10}$$

with $\tilde{p}$ as defined above, and m_p the memory length of the pth order kernel (p odd). If we express the Volterra series as a linear system

$$\boldsymbol{y}_n = \underline{\boldsymbol{H}}\boldsymbol{c}\,, \tag{4.11}$$

where $\boldsymbol{y}_n = [y[0]\ \ldots\ y[N-1]]^{\mathsf{T}}$ is the measured output signal, $\underline{\boldsymbol{H}}$ and $\boldsymbol{c}$ the regression matrix and parameter vector as defined above, respectively, the kernels can be estimated by

$$\hat{\boldsymbol{c}} = \left(\underline{\boldsymbol{H}}^{\mathsf{H}}\underline{\boldsymbol{H}}\right)^{-1}\underline{\boldsymbol{H}}^{\mathsf{H}}\boldsymbol{y}_n\,. \tag{4.12}$$

After the kernels are estimated, the model output is given by

$$\hat{\boldsymbol{y}}_n = \underline{\boldsymbol{H}}\hat{\boldsymbol{c}}\,. \tag{4.13}$$

4.2 The Wiener Model

A widely used subset of the Volterra series for system identification is the Wiener model, consisting of a linear filter $\mathbb{H}[\cdot]$ followed by a static nonlinearity $f(\cdot)$, see Figure 4.2. The Wiener model used in this work is not the original model proposed by Wiener in 1958 [46, pp. 16–49], but a a simplified model from [63] that requires less parameters compared to the orignal model, which is of the same complexity as the Volterra series.[7] The complex baseband input/output relation for the model is given by

$$y_{\mathrm{W}}[n] = \mathbb{H}\left[x[n]\right] \sum_{k=0}^{K-1} a_k \phi_k \left(\left|\mathbb{H}\left[x[n]\right]\right|\right) , \tag{4.14}$$

where $y_{\mathrm{W}}[n]$ and $x[n]$ is the output and input signal, respectively, K is the maximum polynomial order, $\{a_k\}_{k=0}^{K-1}$ the parameters of the nonlinear function $f(\cdot)$, and $\phi_k(\cdot)$ is a basis function of own choice, i.e. Hermitian polynomials. The filter $\mathbb{H}[\cdot]$ is a linear finite impulse-response (FIR) filter of order M, given by

$$\mathbb{H}\left[x[n]\right] = \sum_{m=0}^{M-1} h_m x[n-m] . \tag{4.15}$$

From (4.14) it is easily seen that the output is nonlinear-in-parameters with respect $\{h_m\}_{m=0}^{M-1}$, and linear-in-parameters with respect to $\{a_k\}_{k=0}^{K-1}$. This implies that estimating the model parameters will be more troublesome than i.e. for the Volterra series, which is linear-in-parameters for all model parameters. Unless various assumptions regarding the system are made, i.e. separability or transformation of the parameters, the linear filter coefficients can only be derived by nonlinear estimation techniques [64, pp. 254–260], such as techniques based on the Levenberg-Marquardt algorithm [65], or other iterative/direct search algorithms.[8]

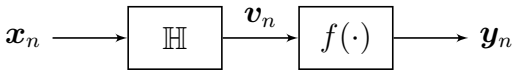


Figure 4.2: Block schematic for the general Wiener model.

4.2.1 Parameter Estimation for the Wiener Model

Since nonlinear estimation is out of the scope for this thesis, a simplied estimation approach, based on a two-step procedure [7, pp. 21–22], is used. In order to use this method, two sets of sampled input/output data $(\boldsymbol{x}_n, \boldsymbol{y}_n)$ are required. The procedure

[7] The original Wiener model consists of Wiener's orthogonal G-functionals, derived from the Volterra functionals through a Gram-Schmidt orthogonalization procedure [51, pp. 45 – 76].

[8] Determining a transform of the parameters $\boldsymbol{\theta}$ in the manner of $\boldsymbol{\alpha} = \boldsymbol{g}(\boldsymbol{\theta})$ is often very difficult, and in most cases such a transform with one-to-one mapping does not exist [64, pp. 255].

assumes that the linear part of the Wiener model is a stable FIR filter of order M, and based on this assumption a linear estimate of the total system is made. Let

$$\boldsymbol{c}_1 = [h_1\, h_2\, \ldots\, h_M]^{\mathsf{T}} \in \mathbb{C}^M\,, \tag{4.16}$$

and

$$\underline{\boldsymbol{H}} = [\boldsymbol{X}_1\, \boldsymbol{X}_2\, \ldots\, \boldsymbol{X}_M] \in \mathbb{C}^{N\times M}\,, \tag{4.17}$$

$$\boldsymbol{X}_m = \left[\boldsymbol{x}_{m,n}\, \boldsymbol{x}_{m,n-1}\, \ldots\, \boldsymbol{x}_{m,n-(N-1)}\right]^{\mathsf{T}} \in \mathbb{C}^N\,, \tag{4.18}$$

with $\boldsymbol{x}_m = x[n-m]$. Then the first step is to estimate $\hat{\boldsymbol{c}}_1$ by solving

$$\hat{\boldsymbol{c}}_1 = \left(\underline{\boldsymbol{H}}^{\mathsf{H}}\underline{\boldsymbol{H}}\right)^{-1}\underline{\boldsymbol{H}}^{\mathsf{H}}\boldsymbol{y}_n\,, \tag{4.19}$$

where $\boldsymbol{y}_n = [y[0]\, \ldots\, y[N-1]]^{\mathsf{T}}$ is the sampled output data.[9] After $\hat{\boldsymbol{c}}_1$ is estimated, the intermediate signal $\boldsymbol{v}_n$, with respect to Figure 4.2, is estimated by $\hat{\boldsymbol{v}}_n = \mathbb{H}\left[\boldsymbol{x}_n; \hat{\boldsymbol{c}}_1\right]$, where we now use a new set of input data $\boldsymbol{x}_n$. With an estimate of the intermediate signal available, we go on to the second step of the estimation procedure. Let

$$\boldsymbol{c}_2 = [a_1\, a_2\, \ldots\, a_K]^{\mathsf{T}} \in \mathbb{C}^K\,, \tag{4.20}$$

and

$$\underline{\boldsymbol{F}} = [\boldsymbol{V}_1\, \boldsymbol{V}_2\, \ldots\, \boldsymbol{V}_K] \in \mathbb{C}^{N\times K}\,, \tag{4.21}$$

$$\boldsymbol{V}_k = \left[\hat{\boldsymbol{v}}_{k,n}\, \hat{\boldsymbol{v}}_{k,n-1}\, \ldots\, \hat{\boldsymbol{v}}_{k,n-(N-1)}\right]^{\mathsf{T}} \in \mathbb{C}^N\,, \tag{4.22}$$

with $\hat{\boldsymbol{v}}_{k,n} = \hat{v}[n]\phi_k\left(|\hat{v}[n]|\right)$. The parameters of the static nonlinearity are then estimated by a standard least squares approach

$$\boldsymbol{c}_2 = \left(\underline{\boldsymbol{F}}^{\mathsf{H}}\underline{\boldsymbol{F}}\right)^{-1}\underline{\boldsymbol{F}}^{\mathsf{H}}\boldsymbol{y}_n\,, \tag{4.23}$$

where $\boldsymbol{y}_n = [y[0]\, \ldots\, y[N-1]]^{\mathsf{T}}$ now is the sampled output data belonging to the second set of input data. The model output is found by $\hat{\boldsymbol{y}}_n = \underline{\boldsymbol{F}}\hat{\boldsymbol{c}}_2$.

4.3 The Hammerstein Model

Similar as in the case of the Wiener model, the Hammerstein model is also a subset of the Volterra series. The model is consists of a static nonlinearity, $f(\cdot)$, followed by a linear FIR filter, $\mathbb{H}[\cdot]$, as depicted in Figure 4.3. Although the Hammerstein is closely related to the Volterra series and the Wiener model, it lacks some of the capabilities of these

[9] In (4.16) – (4.22) the index is shifted by one in order to follow conventional notation from linear algebra; hence, $m \in \{1, M\}$, and $k \in \{1, K\}$.

models. Due to the order of the blocks in the model, the model output will not consist of nonlinear memory, as is the case for the other two models. However, this has not restricted the use of the model, with good results, for both modeling and linearization of microwave power amplifiers [66–69]. The complex-baseband input/output relation of the model is given by

$$y_H[n] = \mathbb{H}\left[\sum_{k=0}^{K-1} a_k x[n]\phi_k\left(|x[n]|\right)\right], \tag{4.24}$$

where $x[n]$ and $y[n]$ are the input and output signals, respectively, M the length of the FIR filter, K the maximum polynomial order, and $\phi_k(\cdot)$ again a polynomial basis function of own choice, e.g. Chebyshev polynomials. The filter $\mathbb{H}[\cdot]$ is as given in (4.15). As with the Wiener model, the model output is only linear-in-parameters with to a subset of the model parameters. It is linear with respect to the coefficients $\{h_m\}_{m=0}^{M-1}$, and nonlinear-in-parameters with respect to the coefficients $\{a_k\}_{k=0}^{K-1}$.

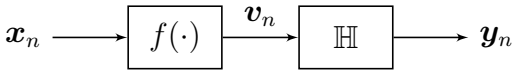


Figure 4.3: Block schematic for the general Hammerstein model.

4.3.1 Parameter Estimation for the Hammerstein Model

As mentioned previously, the Hammerstein model is only partly linear-in-parameters with a subset of the model coefficients. This implies that estimating the model parameters requires esimation procedures like the ones mentioned in Section 4.2.1. Again, since nonlinear estimation and iterative/direct search algorithms for estimation problems are not in the scope of this work, an approach similar to the two-step procedure used in the parameter estimation for the Wiener model is adopted for the Hammerstein system. Similarly as for the Wiener model, two sets of sampled input/output data $(\boldsymbol{x}_n, \boldsymbol{y}_n)$ are required. The first step is to estimate the intermediate signal $\boldsymbol{v}_n$, with respect to Figure 4.3, with a standard least squares approach. Let

$$\boldsymbol{c}_1 = [a_1\, a_2\, \ldots\, a_K]^{\mathsf{T}} \in \mathbb{C}^K, \tag{4.25}$$

and

$$\underline{\boldsymbol{H}} = [\boldsymbol{X}_1\, \boldsymbol{X}_2\, \ldots\, \boldsymbol{X}_K] \in \mathbb{C}^{N\times K}, \tag{4.26}$$

$$\boldsymbol{X}_k = \left[\boldsymbol{x}_{k,n}\, \boldsymbol{x}_{k,n-1}\, \ldots\, \boldsymbol{x}_{k,n-(N-1)}\right]^{\mathsf{T}} \in \mathbb{C}^N, \tag{4.27}$$

where $\boldsymbol{x}_{k,n} = x[n]\phi_k\left(|x[n]|\right)$. We find the polynomial parameters $\hat{\boldsymbol{c}}_1$ by solving

$$\hat{\boldsymbol{c}}_1 = \left(\underline{\boldsymbol{H}}^{\mathsf{H}}\underline{\boldsymbol{H}}\right)^{-1}\underline{\boldsymbol{H}}^{\mathsf{H}}\boldsymbol{y}_n\,, \tag{4.28}$$

where $\boldsymbol{y}_n = [y[0]\,y[1]\,\ldots\,y[N-1]]^{\mathsf{T}}$ belongs to the first set of sampled output data, and $\underline{\boldsymbol{H}}$ is defined as above. When $\hat{\boldsymbol{c}}_1$ is available, we estimate the intermediate signal by passing the second set of sampled input data through the nonlinear function, or $\hat{\boldsymbol{v}}_n = f\left(\boldsymbol{x}_n; \hat{\boldsymbol{c}}_1\right)$. For the second step, let

$$\boldsymbol{c}_2 = [h_1\,h_2\,\ldots\,h_M]^{\mathsf{T}} \in \mathbb{C}^M\,, \tag{4.29}$$

and

$$\underline{\boldsymbol{F}} = [\boldsymbol{V}_1\,\boldsymbol{V}_2\,\ldots\,\boldsymbol{V}_M] \in \mathbb{C}^{N\times M}\,, \tag{4.30}$$

$$\boldsymbol{V}_m = \left[\hat{\boldsymbol{v}}_{m,n}\,\hat{\boldsymbol{v}}_{m,n-1}\,\ldots\,\hat{\boldsymbol{v}}_{m,n-(N-1)}\right]^{\mathsf{T}} \in \mathbb{C}^N\,, \tag{4.31}$$

where $\hat{\boldsymbol{v}}_{m,n} = \hat{v}[n-m]$. The linear filter coefficietns are then found by solving

$$\hat{\boldsymbol{c}}_2 = \left(\underline{\boldsymbol{F}}^{\mathsf{H}}\underline{\boldsymbol{F}}\right)^{-1}\underline{\boldsymbol{F}}^{\mathsf{H}}\boldsymbol{y}_n\,, \tag{4.32}$$

where $\boldsymbol{y}_n = [y[0]\,y[1]\,\ldots\,y[N-1]]^{\mathsf{T}}$ now is the sampled output data that belongs to the second set of input/output data, and $\underline{\boldsymbol{F}}$ is as defined above. The model output is then found by the relation $\hat{\boldsymbol{y}}_n = \underline{\boldsymbol{F}}\hat{\boldsymbol{c}}_2$.

4.4 Memory Polynomial Model

The memory polynomial model is, as the Wiener- and Hammerstein models, a subset of the Volterra series, and is derived by only considering the diagonal elements of the Volterra kernels in (4.4). This implies that the off-diagonal kernels are considered to be zero, e.g. $h_{2p+1}\left[\boldsymbol{m}_{2p+1}\right] = 0\,\forall\, m_1 \neq m_3 \neq \ldots \neq m_{2p+1}$. Applying this to (4.4), the Volterra series reduces to the memory polynomial model, with the complex baseband input/output relationship given by

$$y_{\mathrm{MP}}[n] = \sum_{m=0}^{M-1}\sum_{p=0}^{P-1} a_{m,p}x[n-m]|x[n-m]|^{2p}\,, \tag{4.33}$$

where $y[n]$ is the output signal, $x[n]$ the input signal, M the maximum sample delay, $P-1$ the maximum polynomial order considered, and $\{a_{m,p}\}$ the model parameters/kernels. Evaluating this expression, we see that the envelope terms only consist of even orders, e.g. $|\cdot|^{2p}$ (equivalently meaning that $x[n-m]|x[n-m]|^{2p}$ is the result of an odd number of signal prodcts). This follows the conventional understanding that only odd-ordered power amplifier nonlinearities will produce in-band distortion in an communication system, which to some degree is mathematically proved in the Appendix of [70]. The authors of [70, 71] have also investigated the effect of adding odd-ordered envelope terms to black-box models, e.g. $|\cdot|^p$, and showed that including these may improve the

modeling of the power amplifier in terms of reduced maximum polynomial order $P-1$. Hence, in the rest of this section, and thesis, we consider a slightly modified memory polynomial model, using the input/output relationship given by

$$\tilde{y}_{\mathrm{MP}}[n] = \sum_{m=0}^{M-1} \sum_{p=0}^{P-1} a_{m,p} x[n-m] \phi_p \left(|x[n-m]| \right) , \tag{4.34}$$

where the notation is mostly as before, with the exception of $\phi_p(\cdot)$, which represent a polynomial basis of own choice. If it is desired to model higher order nonlinear terms, i.e. $P-1=11$, the drawback of this model is that it will require an increased amount of coefficients compared to (4.33), which definitely is contributing to the fact that (4.34) shows improved modeling results in terms of normalized mean square error (nmse) compared to (4.33).[10] The original model consists of $\nu_{\mathrm{tot}} = M(P+1)/2$ coefficients for P odd, and $\nu_{\mathrm{tot}} = MP/2$ coefficients for P even. On the other hand, the modified model consists of $\nu_{\mathrm{tot}} = MP$ coefficients for all P — almost twice the number of the original model for any nonlinear order. A simple comparison of the two different models is given in Table 4.1, where it is seen that the modified model results in improved modeling in terms of nmse, but at the expense of increased complexity.

Another interesting figure is the condition number of the data matrix. The condition number of the original model is several powers of ten greater compared to the modified model, which can be explained with the maximum polynomial order; the highest order for the original model is $2(P-1)=8$, and $P-1=4$ for the modified model. Polynomials are often ill-posed for high orders, and the condition number of the data matrix, for both models, will suffer from this fact. Although the modified model has a lower condition number, a value in the region of 1.00×10^6 can not be said to be small - which is desired, as discussed in Section 2.4.

Table 4.1: Comparison of original and modified memory polynomial model.

Model	M	P	Coeffs.	$\mathrm{cond}(\boldsymbol{X})$	$\mathrm{nmse}\,(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]
$\boldsymbol{y}_{\mathrm{MP}}$	5	5	15	2.69×10^9	-31.22
$\tilde{\boldsymbol{y}}_{\mathrm{MP}}$	5	5	25	3.88×10^6	-57.36

In addition to the modified model in (4.34), several other modifications of the memory polynomial model are published in literature. This includes variations such as the triangular memory polynomial model [72] - a model that reduces the envelope order for increasing memory length, and thus requires less coefficients, or for instance a generalized memory polynomial model [52] - which includes additional terms specifically introduced to capture long-term memory effects and envelope cross-products, and thus requires a great amount of extra coefficients.[11] For flexibility, the model implemented in this work

[10] See Section 4.6 for the definition of the nmse.

[11] The memory polynomial model is in some cases referred to as the nonlinear moving average (NMA) model [54]. There also exist an extension of the NMA model which includes cross-product envelope

is based on (4.34), but includes the opportunity to operate as the triangular memory polynomial model so comparisons between the two can be made, i.e. with respect to reduced complexity vs. modeling capabilities.

4.4.1 Parameter Estimation of the Memory Polynomial Model

As the Volterra series, the memory polynomial model is linear in parameters, meaning that the model coefficients can be estimated by any robust least squares approach. To use a notation that is in accordance with conventional linear algebra, we shift the indexing with one sample in (4.34), resulting in $m \in \{1, M\}$, and $p \in \{1, P\}$. The envelope order will now be given by $|\cdot|^{p-1}$. Let

$$\boldsymbol{c} = \left[\boldsymbol{c}_1^{\mathsf{T}}\, \boldsymbol{c}_2^{\mathsf{T}}\, \ldots\, \boldsymbol{c}_M^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{C}^{\nu_{\text{tot}}}\,, \tag{4.35}$$

$$\boldsymbol{c}_m = \left[a_{m,1}\, a_{m,2}\, \ldots\, a_{m,P}\right]^{\mathsf{T}} \in \mathbb{C}^{\nu_{\text{P}}}\,, \tag{4.36}$$

and

$$\underline{\boldsymbol{H}} = \left[\boldsymbol{X}_1\, \boldsymbol{X}_2\, \ldots\, \boldsymbol{X}_M\right] \in \mathbb{C}^{N \times \nu_{\text{tot}}}\,, \tag{4.37}$$

$$\boldsymbol{X}_m = \left[\boldsymbol{x}_{m,p,n}\, \boldsymbol{x}_{m,p,n-1}\, \ldots\, \boldsymbol{x}_{m,p,n-N}\right]^{\mathsf{T}} \in \mathbb{C}^{N \times \nu_{\text{P}}}\,, \tag{4.38}$$

where

$$\boldsymbol{x}_{m,p} = \left[x[n-m]|x[n-m]|^0\, x[n-m]|x[n-m]|\, \ldots\, x[n-m]|x[n-m]|^{P-1}\right]^{\mathsf{T}} \in \mathbb{C}^{\nu_{\text{P}}}\,. \tag{4.39}$$

If we express (4.34) as a linear system of equations, $\boldsymbol{y}_n = \underline{\boldsymbol{H}}\boldsymbol{c}$, the coefficient vector $\hat{\boldsymbol{c}}$ that minimizes the least square error is given by

$$\hat{\boldsymbol{c}} = \left(\underline{\boldsymbol{H}}^{\mathsf{H}}\underline{\boldsymbol{H}}\right)^{-1}\underline{\boldsymbol{H}}^{\mathsf{H}}\boldsymbol{y}_n\,, \tag{4.40}$$

where $\boldsymbol{y}_n = [y[0]\, \ldots\, y[N-1]]^{\mathsf{T}}$ is the sampled output data, and $\underline{\boldsymbol{H}}$ is defined as above. The model output is then given by $\hat{\boldsymbol{y}}_n = \underline{\boldsymbol{H}}\hat{\boldsymbol{c}}$.

4.5 Spline Delay Envelope Model

The spline delay envelope (SDE) model is an approximation to the Volterra series, where cross-term products also are considered, in contrast to i.e. the memory polynomial model. It was proposed in [55], with the purpose of modeling nonlinear microwave power

terms, called the augmented nonlinear moving average (ANMA) model. This model can be interpreted as an generalized memory polynomial model.

amplifiers with improved numerical properties. The improved properties are related to the basis function used in the model; splines. A spline is a function $\mathcal{C}$ defined piecewise, often by low-order polynomials, on an interval $\mathcal{I} = [a, b] \in \mathbb{R}$ [73, pp. 100].[12] Given $K+1$ points, or knots, ρ_j in $\mathcal{I}$, where $\rho_0 < \rho_1 < \ldots < \rho_K$, on each subinterval $[\rho_j, \rho_{j+1}]_{j=0}^{K-1}$, a low-degree polynomial of order n, $P_n(\rho)$, is used to represent the spline function, or

$$\mathcal{C}(\rho) = P_n^{(j)}(\rho), \quad \rho_j \leq \rho < \rho_{j+1},\ j = 0, \ldots, K-1\,. \tag{4.41}$$

For smoothness, the spline function, and its $n-1$ first derivatives are required to be continuous in all the knot points [75, pp. 51–58].

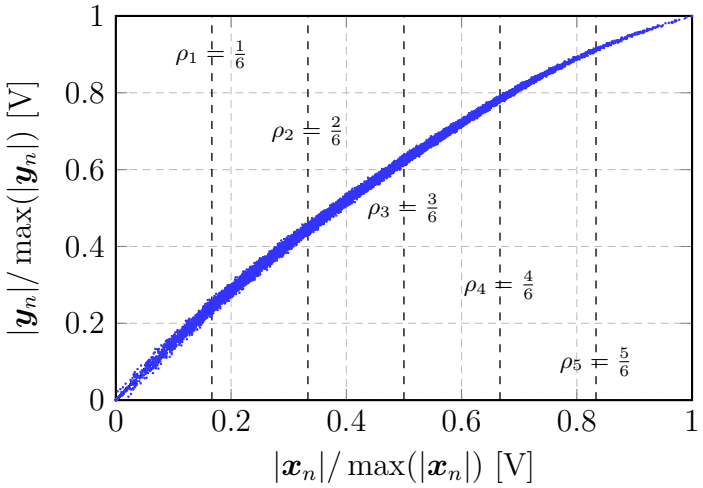


Figure 4.4: AM/AM characteristic of RFPA divided into six segments for generating a spline with seven knot points. The first and the last knot points, $\rho_0 = 0$ and $\rho_7 = 1$, are not depicted.

With the basics of spline functions presented, we turn back to the SDE model; it's complex baseband input/output relation is given by

$$y_{\text{SDE}}[n] = \sum_{q=0}^{Q-1} \sum_{l=0}^{L-1} x[n-q]\mathcal{C}^{q,l}\left(|x[n-l]|\right)\,, \tag{4.42}$$

with the cubic spline function of polynomial order $n = 3$

$$\mathcal{C}^{q,l}\left(|x[n-l]|\right) = \sum_{j=1}^{K_{q,l}-1} a_{j,q,l}||x[n-l]| - \rho_j|^3 + \sum_{\substack{i=0 \text{ if } l=0 \\ i=1 \text{ if } l\neq 0}}^{3} b_{i,q,l}|x[n-l]|^i\,, \tag{4.43}$$

where $K_{q,l}$ is the (q,l)th element of $\boldsymbol{K} \in \mathbb{N}_0^{Q\times L}$, a matrix holding the number of segments the amplitude of the input signal should be split into for each memory tap, $\{a_{j,q,l}\, b_{i,q,l}\}$

[12] Due to the low-order polynomials, spline functions will not suffer from the same numerical properties that high-order polynomials suffer from, i.e. Runge's phenomenon for certain functions [74].

are the complex coefficients to be estimated, and ρ_j are the knots representing the borders between each interval. If $K_{q,l}$ in the upper limit of j in (4.43) is zero, a spline is not be computed. The complexity of the model in terms of number of coefficients is given by

$$\nu_{\text{tot}} = \sum_{q=0}^{Q-1} \sum_{l=0}^{L-1} (K_{q,l} - 1 + \delta)\,, \qquad \text{for } K_{q,l} > 0\,, \tag{4.44}$$

where

$$\delta = \begin{cases} 4 & \text{if } l = 0\,, \\ 3 & \text{if } l \neq 0\,. \end{cases} \tag{4.45}$$

This expression differs from the one proposed in [55, pp. 100], but it can easily be verified to be correct with a simple example.

Example 1 (Number of coefficients) If $\boldsymbol{K} = \text{diag}\,(3, 3, 2, 2)$, the formula in [55] yields $\nu_{\text{tot}} = 22$, whereas (4.44) gives $\nu_{\text{tot}} = \text{tr}\,(\boldsymbol{K}) + \text{tr}\,(\text{diag}\,(3, 2, 2, 2)) = 19$, that also agrees with MATLAB simulations of (4.42) without using any of the two closed form expressions for pre-allocating the number of coefficients.[13]

4.5.1 Parameter Estimation for the Spline Delay Envelope Model

In the following we shift the indexing by one, so $l \in \{1, L\}$ and $q \in \{1, Q\}$, which agrees better with conventional indexing in linear algebra. Let

$$\boldsymbol{c} = \left[\boldsymbol{c}_{1,1}^{\mathsf{T}}\, \boldsymbol{c}_{1,2}^{\mathsf{T}}\, \ldots\, \boldsymbol{c}_{1,L}^{\mathsf{T}}\, \ldots\, \boldsymbol{c}_{Q,1}^{\mathsf{T}}\, \ldots\, \boldsymbol{c}_{Q,L}^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{C}^{\nu_{\text{tot}}}\,, \tag{4.46}$$

where

$$\boldsymbol{c}_{q,l} = \begin{cases} \left[a_{1,q,l}\, \ldots\, a_{K_{q,l}-1,q,l}\, b_{0,q,l}\, \ldots\, b_{3,q,l}\right]^{\mathsf{T}} \in \mathbb{C}^{K_{q,l}+3} & \text{if } l = 1\,, \\ \left[a_{1,q,l}\, \ldots\, a_{K_{q,l}-1,q,l}\, b_{1,q,l}\, \ldots\, b_{3,q,l}\right]^{\mathsf{T}} \in \mathbb{C}^{K_{q,l}+2} & \text{if } l \neq 1\,, \end{cases} \tag{4.47}$$

and

$$\underline{\boldsymbol{H}} = [\boldsymbol{X}_{1,1}\, \boldsymbol{X}_{1,2}\, \ldots\, \boldsymbol{X}_{1,L}\, \ldots\, \boldsymbol{X}_{Q,1}\, \ldots\, \boldsymbol{X}_{Q,L}] \in \mathbb{C}^{N \times \nu_{\text{tot}}}\,, \tag{4.48}$$

where

$$\boldsymbol{X}_{q,l} = \begin{cases} [\boldsymbol{x}_{q,l,n}\, \boldsymbol{x}_{q,l,n-1}\, \ldots\, \boldsymbol{x}_{q,l,n-N}]^{\mathsf{T}} \in \mathbb{C}^{N \times K_{q,l}+3} & \text{if } l = 1\,, \\ [\boldsymbol{x}_{q,l,n}\, \boldsymbol{x}_{q,l,n-1}\, \ldots\, \boldsymbol{x}_{q,l,n-N}]^{\mathsf{T}} \in \mathbb{C}^{N \times K_{q,l}+2} & \text{if } l \neq 1\,. \end{cases} \tag{4.49}$$

For $l = 1$ we have

[13] Given $\boldsymbol{A}^{n \times n}$, $\text{tr}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ii}$. The matrix trace is only defined for square matrices.

$$\boldsymbol{x}_{q,l} = \Big[x[n-q]|x[n-l]| - \rho_1|^3 \; \dots \; x[n-q]|x[n-l]| - \rho_{K_{q,l}-1}|^3 \\ x[n-q]|x[n-l]|^0 \; \dots \; x[n-q]|x[n-l]|^3\Big]^{\mathsf{T}} \in \mathbb{C}^{K_{q,l}+3}\,, \tag{4.50}$$

whereas for $l \neq 1$

$$\boldsymbol{x}_{q,l} = \Big[x[n-q]|x[n-l]| - \rho_1|^3 \; \dots \; x[n-q]|x[n-l]| - \rho_{K_{q,l}-1}|^3 \\ x[n-q]|x[n-l]|^1 \; \dots \; x[n-q]|x[n-l]|^3\Big]^{\mathsf{T}} \in \mathbb{C}^{K_{q,l}+2}\,. \tag{4.51}$$

Again, expressing the system in (4.42) in matrix form, $\boldsymbol{y}_n = \underline{\boldsymbol{H}}\boldsymbol{c}$, we can easily solve for the coefficient vector $\hat{\boldsymbol{c}}$ that minimizes the squared error in the least square problem. The least sqaure solution will be given by

$$\hat{\boldsymbol{c}} = \left(\underline{\boldsymbol{H}}^{\mathsf{H}}\underline{\boldsymbol{H}}\right)^{-1}\underline{\boldsymbol{H}}^{\mathsf{H}}\boldsymbol{y}_n\,, \tag{4.52}$$

where $\boldsymbol{y}_n = [y[0] \; \dots \; y[N-1]]^{\mathsf{T}}$ is the sampled output data vector, and $\underline{\boldsymbol{H}}$ is as defined above. Again, the model output is given by $\hat{\boldsymbol{y}}_n = \underline{\boldsymbol{H}}\hat{\boldsymbol{c}}$.

4.6 Model Validation

Choosing which behavioral model to use amounts to compromising between different properties, such as complexity, accuracy, robustness, and so on. Which property that should be favored depends on the application for the behavioral model, e.g. if the model is to be used in the design of a digital predistortion unit for a handset, complexity will be a limiting factor as processing power is limited in such applications. On the other hand, if the predistortion unit is designed for a base station application, accuracy and robustness are typically more important than complexity. In this section the models presented in this chapter are validated with respect to a set of figure of merits. The models are tested with both simulated and measured sets of input/output data — allowing us to quantify the effect of the limitations and noise in the measurement setup.

4.6.1 Figure of Merits for Model Validation

One of the most common figure of merits used in system identification, and estimation theory in general, is the normalized mean square error (nmse), defined as

$$\mathrm{nmse}\left(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n\right) = 10\log_{10}\left(\frac{\|\boldsymbol{y}_n - \hat{\boldsymbol{y}}_n\|_2^2}{\|\boldsymbol{y}_n\|_2^2}\right) \quad [\mathrm{dB}]\,, \tag{4.53}$$

where $\boldsymbol{y}_n \in \mathbb{C}^N$ is the sampled system output, $\hat{\boldsymbol{y}}_n \in \mathbb{C}^N$ is the estimated model output, and $\|\cdot\|_2^2$ is the squared Euclidean norm. The nmse gives an overall view of how close the model output is to the measured system output, however, without taking into account

the number of model parameters or the data length. If it is desirable to take these variables into account, a modified mean square error criteria can be used, e.g. Akaike's final prediction error criterion [76], or the minimum description length criterion [77], which both add a penalty term to the modeling error for an increased number of model parameters, and data length. Although using these gives a better overview of the modeling capabilities of each model structure, they are not used as a validation criteria in this work. Instead, the adjacent channel error power ratio (acepr) is used together with the nmse. The acepr is defined as the ratio of the power in the error signal in an adjacent channel to the power of the desired signal in the main channel [78], or

$$\text{acepr}\left(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n\right) = 10\log_{10}\left(\frac{\int_{\text{adj.}} |E(f)|^2 \, \mathrm{d}f}{\int_{\text{main ch.}} |Y(f)|^2 \, \mathrm{d}f}\right) \quad [\text{dB}]\,, \tag{4.54}$$

where $E(f)$ is the discrete Fourier transform of $\boldsymbol{e}_n = \boldsymbol{y}_n - \hat{\boldsymbol{y}}_n$, and $Y(f)$ the discrete Fourier transform of $\boldsymbol{y}_n$. With this measure, the out-of-band modeling can be quantified in a better way than just using the nmse, which only gives an overview of the total modeling performance. The lower the values of both these figures are, the better the model structure is at modeling the behavior of the microwave power amplifier. Although it is desirable to minimize both these terms, it is not given that the model structure that achieves the best value for i.e. nmse will also achieves the best acepr value. This is due to the fact that the majority of the signal power is located in the main channel, and at frequencies outside this band the signal power decreases rapidly with increasing frequency distance from the center of the main channel. Thus, acepr is a good complementary measure to use next to nmse, especially if several model structures achieve almost similar nmse. In such cases, acepr can be deciding with respect to concluding which model that yields the most accurate behavior of the power amplifier.

4.7 Modeling of Simulated and Measured Microwave Power Amplifiers

In this section the modeling performance of each model presented previously in the chapter is evaluated, based on captured input/output data from both simulated and measured power amplifiers. This allows us to compare the performance in an ideal environment with no limitations in measurement equipment, measurement noise, and other perturbing effects, to an environment where all the mentioned limitations are present and are affecting the end results. The power amplifiers used are: 1) a 1 W GaAs pHEMT PA at 3.40 GHz, designed in [79], 2) a simulated 6 W GaN HEMT PA at 3.40 GHz, summarized in Appendix D, and 3) a 6 W GaN HEMT PA at 2 GHz, designed in [80].[14] For the simulated amplifier, the setup described in Appendix B is used to generate the set of input/output data, whereas for the measured amplifiers, the setup illustrated in Figure E.2 is used to acquire the input/output data. The input

[14] GaAs is an acronym for Gallium Arsenide, and pHEMT an acronym for pseudomorphic high electron mobility transistor.

signal for all scenarios is a 16-QAM signal generated in MATLAB, consisting of 1000 symbols, and filtered with a square-root-raised-cosine filter (SRRC) using a roll-off factor of $r = 0.22$. Prior to filtering and oversampling, the bandwidth of the signal is 3.84 MHz, however, after oversampling with a rate of 16, the bandwidth of the signal into the DAC in the signal generator is 61.44 MHz. A filter delay of 12 samples is also used in the pulse-shaping filter in MATLAB. The peak-to-average power ratio (papr) of the input signal is about 6.50 dB, and can be found by using the relation

$$\text{papr}(x) = 10\log_{10}\left(\frac{\max|\boldsymbol{x}_n|^2}{\mathbb{E}\left[|\boldsymbol{x}_n|^2\right]}\right) \quad [\text{dB}]\,, \tag{4.55}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. If signals with no well defined peak value are used in the modeling approach, e.g. OFDM signals (for which a great amount of research is put into papr reducing techniques [81]), the papr should be derived in a different manner — i.e. only using data samples with an amplitude value within a specific threshold, as illustrated in Figure 4.5. In the figure, the complementary cumulative distribution function (CCDF) of the envelope for a 16-QAM signal is plotted vs. papr. The red line shows the papr of the signal if only the data samples that are within the 99 % level of the cumulative distribution function (CDF) for the signal envelope are used in the calculation. Since a well defined peak amplitude exists for M-QAM signals, the difference between the two numbers will be small for this case, as can be seen in the plot.

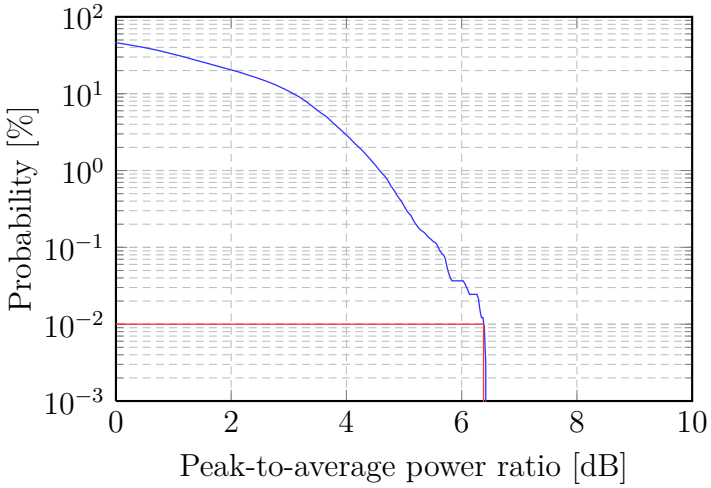


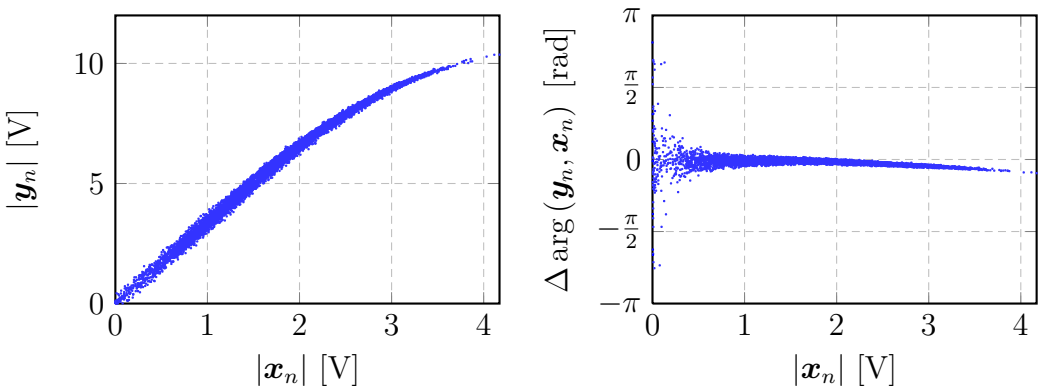
Figure 4.5: CCDF for the amplitude of a 16-QAM signal generated in MATLAB.

In the following sections, plots showing the amplitude-to-amplitude (AM/AM) and amplitude-to-phase (AM/PM) conversion of each PA used in the black-box modeling are presented. The AM/AM conversion describes the amplitude distortion on the output due to the amplitude variations on the input, while the AM/PM describes the phase distortion on the output due to the amplitude variations on the input. For the AM/PM conversion we use the difference in phase between the output and input, or

$$\Delta \arg(\boldsymbol{y}_n, \boldsymbol{x}_n) = \arg(\boldsymbol{y}_n) - \arg(\boldsymbol{x}_n) \quad [\text{rad}], \tag{4.56}$$

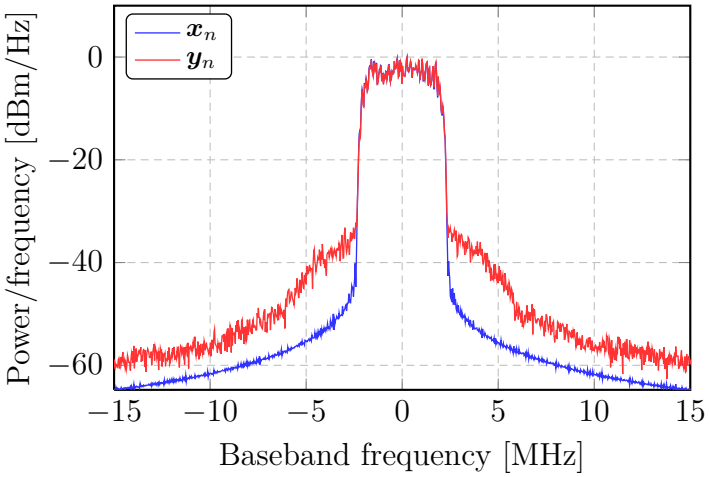
where $\boldsymbol{y}_n$ denotes the sampled output data vector, $\boldsymbol{x}_n$ denotes the sampled input data vector, and $\arg(\cdot)$ returns the argument of the complex vector. In all AM/AM and AM/PM plots, the voltage/amplitude levels on the axes are given with reference to 50 Ω.

4.7.1 Modeling of a 1 W GaAs pHEMT Class AB PA



(a) AM/AM conversion.

(b) AM/PM conversion.



(c) Spectrum.

Figure 4.6: Measured AM/AM and AM/PM characteristics, and output spectrum for the 1 W GaAs pHEMT power amplifier with an average input power of 16 dBm. Biasing is $V_{\text{D}} = 10\,\text{V}$ and $I_{\text{D}} = 100\,\text{mA}$.

The 1 W PA modeled in this section has a power gain of approximately 10 dB, and delivers just below 30 dBm in 1 dB compression. The measured AM/AM and AM/PM conversion of the PA is plotted in Figure 4.6a and 4.6b, together with its output spectrum

given in Figure 4.6c. The figures show the PAs nonlinear behavior when excited with a signal with an average input power of 16 dBm, implying peak values well into compression since the papr of the input signal is greater than 6 dB. For simplicity, the modeling of the PA is only performed with this input power level. It is, however, expected that the performance of the different algorithms will decrease with increasing input power.[15] Also, since the signal generator is capable to deliver output powers up to 26 dBm in overrange, no driver PA is used for testing the 1 W PA design.

Table 4.2: Modeling results for the Volterra series with increasing model order and static kernels. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 1 W PA.

m_{V}	P	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
[1 1]	3	2	1.61×10^1	−28.85	−44.67	−44.33
[1 1 1]	5	3	2.50×10^2	−29.01	−45.65	−44.95
[1 1 1 1]	7	4	3.97×10^3	−29.16	−46.73	−45.64
[1 1 1 1 1]	9	5	7.12×10^4	−29.22	−47.58	−46.30
[1 1 1 1 1 1]	11	6	1.50×10^6	−29.24	−47.93	−46.50
[1 1 1 1 1 1 1]	13	7	3.42×10^7	−29.25	−47.93	−46.49

Table 4.2 presents the modeling results for the Volterra series with increasing model order, and static kernels. In the table, $m_{\mathrm{V}} = [m_1\, m_3\, \ldots\, m_P]$ denotes the length of the Volterra kernels in each dimension. It is evident that for the static case, increasing the model order does not improve the modeling results significantly with respect to nmse. On the other hand, with respect to acepr, an increased model order results in a 3 dB improvement compared to the lowest order tested, however, at the expense of an greatly increased condition number for the data matrix. Although the 13th order model shows the best overall modeling results, the solution found for this order will be much more sensitive to perturbations in the data than the solution found for i.e. the 9th order model. Since the improvement in modeling when using a model order greater than $P = 9$ is decreasing, the model order is now fixed to $P = 9$, and the kernel length in each dimension is increased linearly in order to see how this affects the modeling results, see Table 4.2. Choosing a model order of $P = 9$ can also be justified by using Akaike's information criterion (aic), defined as [82]

$$\mathrm{aic} = 2(N+1) + (n+1)\left[\ln\left(2\pi \cdot \frac{\mathrm{rss}}{n+1}\right) + 1\right], \tag{4.57}$$

where N is the number of data points, n the polynomial degree, and rss the residual sum of squares given by

[15] When in saturation, the peak amplitudes of the input signal will be clipped almost as in a hard limiter. For such nonlinear systems, the Volterra series, and the other models presented will all have problems modeling the system accurately.

$$\text{rss} = \sum_{i=0}^{N} \left(y_i - p(x_i)\right)^2 \ . \tag{4.58}$$

With the criterion it is possible to estimate the optimal order for fitting the AM/AM conversion of the PA with a polynomial sequence $p_n(x)$.[16] Using this function, a polynomial order of $n = 10$ is found for the given set of input/output data, suggesting that either a 9th or a 11th order Volterra series could be optimal for modeling the envelope nonlinearity. From the results in Table 4.3 we also see that delaying the input signal with only one sample results in improved modeling with respect to nmse. The results suggests that either; 1) memory effects are present in the PA, or 2) that the time alignment algorithm used to align the input/output signals prior to the identification process is suboptimal, or 3) a combination of both. For alignment of the signals, the cross correlation between the input and output is used to find the optimal integer delay between the two, or

$$d_{\text{int}} = \arg\max_{d_{\text{int}}} \left\{ r_{\boldsymbol{x}_n, \boldsymbol{y}_n} \left(d_{\text{int}} \right) \right\} \ . \tag{4.59}$$

Even though this method results in an optimal integer delay, it might be that the optimal delay between the input and output is a fractional number.[17] Further, we see that after increasing the tap length of the linear kernel with one tap, an additional increase to two taps does not improve the modeling significantly. On the other hand, increasing the kernel length for the cubic kernel results in an improvement of approximately 1.70 dB, at the expense of a slight increase in both the number of model parameters and condition number. We also see that increasing the tap length for higher order kernels does not result in great improvements, showing almost unchanged values for both nmse and acepr. When modeling the PA with the other behavioral models presented earlier in the chapter, the results are quite similar to the results obtained with the Volterra series, see Tables I.1 – I.5 for results acquired with the memory polynomial model using different model settings, and Table 4.5 for the spline delay envelope model.[18] For the memory polynomial model we see that in the static case, the results are almost identical to the Volterra series. Even when increasing the kernel length to one tap, the same increase in modeling performance is seen. With a single-tap kernel length, a 4th order memory polynomial model achieves almost identical results to a 9th order Volterra series. And this with a regression matrix whose condition number is almost 200 times smaller than for the Volterra series — implying that the estimated parameters are much less sensitive to perturbations in the data. The results also show that switching from a regular polynomial basis function to an orthogonal polynomial basis function can help impoving the

[16] There is a freely available MATLAB function *polydeg* at `http://www.biomecardio.com/matlab/polydeg.html` that can be used for this purpose.

[17] If it is desirable to look for such a number, the technique described in [83] can be applied, which involves sinc interpolation of the input signal vector and the cross correlation vector prior to estimating a fractional delay d_{frac}.

[18] The Wiener and Hammerstein models are not used for modeling this PA as only one set of input/output data was saved at the time the measurements were carried out.

Table 4.3: Modeling results for the Volterra series with fixed model order $P = 9$, and linearly increasing kernel lengths. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 1 W PA.

m_{V}	P	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
[1 1 1 1 1]	9	5	7.12×10^4	−29.22	−47.58	−46.30
[2 1 1 1 1]	9	6	1.01×10^5	−37.60	−47.44	−46.63
[3 1 1 1 1]	9	7	1.41×10^6	−37.61	−47.42	−46.64
[2 2 1 1 1]	9	11	4.08×10^6	−39.32	−51.14	−50.68
[2 2 2 1 1]	9	22	1.98×10^8	−39.39	−51.08	−50.98
[2 2 2 2 1]	9	41	9.66×10^9	−39.49	−51.54	−51.33
[2 2 2 2 2]	9	70	2.61×10^{12}	−39.55	−51.54	−51.51

numerical properties further by reducing the condition number. However, this will, for some basis functions, be at the expense of reduced accuracy in the modeling. The main results obtained with this model are summarized in Table 4.4. When modeling with the spline delay envelope model we see the same trend as for the other two models. With a static kernel the nmse does not get any better than approximately −29 dB. Again, increasing the kernel length to a single tap, an improvement of nearly 10 dB is achieved. A further increase in kernel length and the delay for the envelope term does not improve the results significantly. In Table 4.5 the main results achieved with this model are summarized. We see that the best modeling of the 1 W PA with respect to both nmse and acepr is achieved with the this model. However, both the number of complex coefficients and the condition number of the regression matrix is greater than desired.

Table 4.4: Main modeling results for the memory polynomial model of order $P = 6$ and with a kernel length of two taps ($M = 2$). For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. Rows marked with a * denote that only even-ordered polynomials have been used. The modeled power amplifier is the measured 1 W PA.

Type/basis	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
reg/$O_n^{(1)}$	12	2.20×10^4	−39.60	−52.21	−51.98
reg/$O_n^{(2)}$	12	1.01×10^4	−39.60	−52.21	−51.98
tri/$O_n^{(2)}$	10	4.03×10^3	−39.59	−52.18	−51.95
tri/L_n	10	4.03×10^3	−39.59	−52.18	−51.95
reg*/$O_n^{(2)}$	6	4.96×10^1	−38.05	−48.34	−48.45
reg*/H_n	6	9.44×10^2	−38.98	−50.45	−50.61
reg*/L_n	6	4.96×10^1	−38.05	−48.34	−48.45
tri*/H_n	5	9.44×10^2	−38.98	−50.45	−50.61

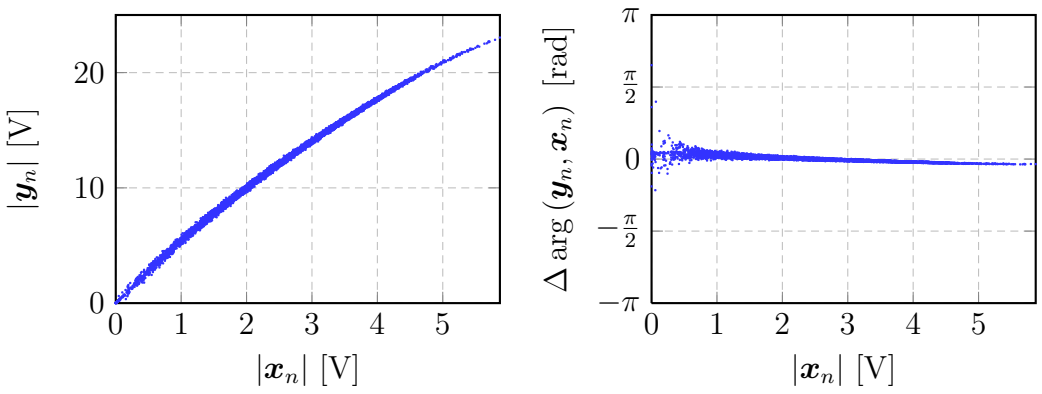
Table 4.5: Main modeling results for the spline delay envelope model with fixed number of envelope segments $\max_{i,j} \boldsymbol{K}_{i,j}$, and a kernel length of four taps ($Q = 4$). The segment matrix is varied between being diagonal, upper triangular, and lower triangular. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 1 W PA.

Type/$[M\,Q\,L]$	dim $(\boldsymbol{c})$	cond $(\underline{\boldsymbol{H}})$	nmse $(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	acepr $(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
diag/[5 4 4]	29	1.15×10^5	-39.91	-53.24	-53.14
upper/[5 4 4]	71	5.89×10^8	-40.01	-53.14	-52.99
lower/[5 4 4]	74	4.49×10^9	-40.01	-53.25	-53.20

4.7.2 Modeling of a Simulated 6 W GaN HEMT Class AB PA

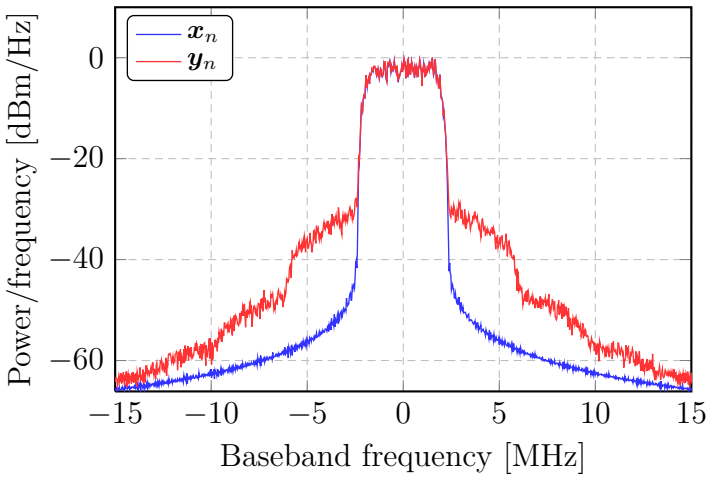
Figure 4.7a and 4.7b shows the AM/AM and the AM/PM conversion for the simulated 6 W power amplifier with an average input power level of 18 dBm. The output spectrum for the same case is given in Figure 4.7c. Again, the Volterra series is tested for modeling, and a model structure with a static kernel and increasing order is applied first. Using static kernels, the modeling performance is approximately 6 dB better than for the measured 1 W power amplifier, as expected. Once again we see that increasing the order, while keeping the kernels static, does not improve the modeling performance significantly with respect to nmse. However, increased order have a huge impact on the acepr — the higher the order, the better the acepr. The difference between a 3rd order model and a 13th order model is almost 15 dB, which most likely is the result of an ideal environment with no limitations or noise. When using (4.57), the optimal nonlinear order for polynomial fitting is found to be $n = 8$.

Since the improvement in nmse is so small when going from $P = 7$ to $P = 9$, the model order is now fixed to $P = 7$, and the kernel length in each dimension is again linearly increased in order to see how this affects the modeling. See Table 4.6 for a complete overview of the obtained results. When increasing the kernel length with a single tap we see the same jump of 10 dB in modeling performance as we did in in the modeling of the 1 W power amplifier. This suggests that (4.59) is working as desired, and that it is more likely that the increase in kernel length improves modeling because concentrated memory effects are present in the system. We also see the same behavior as before when increasing the kernel lengths to a length of two or greater, both for the linear kernel and the higher-order kernels. Hence, a single-tap kernel seems to be adequate for modeling the memory in the system. Otherwise, we see, as expected, that the modeling results based on simulated data are better compared to those based on measurements, yielding an improvement of 6 dB. When increasing the model order to an order greater than $P = 7$ even better results are obtained as long as the kernels are not static. I.e., for a Volterra system with $m_{\mathrm{V}} = [3\,2\,2\,2\,1\,1\,1]$, an nmse of -57.85 dB is achieved, with acepr values of -66.59 dB and -66.20 dB for the lower and upper adjacent channels, respectively. However, the penalty for the improved modeling performance is an increase in the number of model parameters (dim $(\boldsymbol{c}) = 44$), and a rank-deficient



(a) AM/AM conversion.

(b) AM/PM conversion.



(c) Spectrum.

Figure 4.7: Simulated AM/AM and AM/PM characteristics, and output spectrum of a 6 W GaN HEMT power amplifier with an average input power of 18 dBm. Biasing is $V_{\mathrm{D}} = 28\,\mathrm{V}$ and $I_{\mathrm{D}} = 11\,\mathrm{mA}$.

system ($\mathrm{rank}(\underline{\boldsymbol{H}}) = 41 < (\min\dim(\underline{\boldsymbol{H}})) = 44$).

When applying the Wiener and Hammerstein models, which as formulated in Chapter 4 supports the use of orthogonal polynomials without increasing the complexity of the implementation, good modeling results based on numerically robust estimations are obtained. Table 4.7 shows the results obtained using these models. The most interesting result with these models is the great impact the use of orthogonal polynomials have. When used with the memory polynomial model for modeling the 1 W power amplifier, the results achieved were not quite that good, and if improved, it was for several model settings at the expense of the modeling performance. Otherwise, we see that under ideal conditions, the Wiener and Hammerstein models outperform the Volterra series — with respect to both modeling performance, complexity, and numerical properties. Even though they outperform the Volterra series, the spline delay envelope still achieves the

Table 4.6: Modeling results for the Volterra series with fixed model order $P = 7$, and linearly increasing kernel lengths. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the simulated 6 W PA.

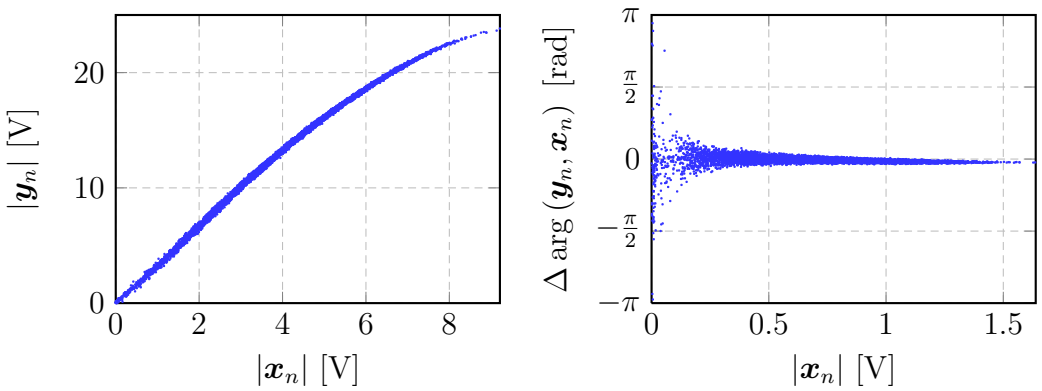
m_{V}	P	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
[1 1 1 1]	7	4	1.70×10^4	−35.02	−52.39	−52.10
[2 1 1 1]	7	5	3.70×10^4	−45.59	−52.46	−51.95
[3 1 1 1]	7	6	5.61×10^5	−45.69	−52.43	−51.98
[2 2 1 1]	7	10	8.86×10^5	−46.38	−53.62	−53.24
[2 2 2 1]	7	21	2.27×10^7	−46.59	−53.93	−53.52
[2 2 2 2]	7	40	2.74×10^9	−46.72	−54.09	−53.69

best overall modeling results showing nmse values as low as −76 dB. The complexity is much greater than for any other model tested, however, the data set used in the identification is more than 20 times the number of model parameters, which according to system identification theory [63] can be considered as the lower limit in order to avoid over-fitting and uncertainty in the estimated kernels, implying that the model parameters estimated are valid. An overview of the main results obtained with the spline delay envelope model is presented in Table 4.8.

4.7.3 Modeling of a 6 W GaN HEMT Class AB PA

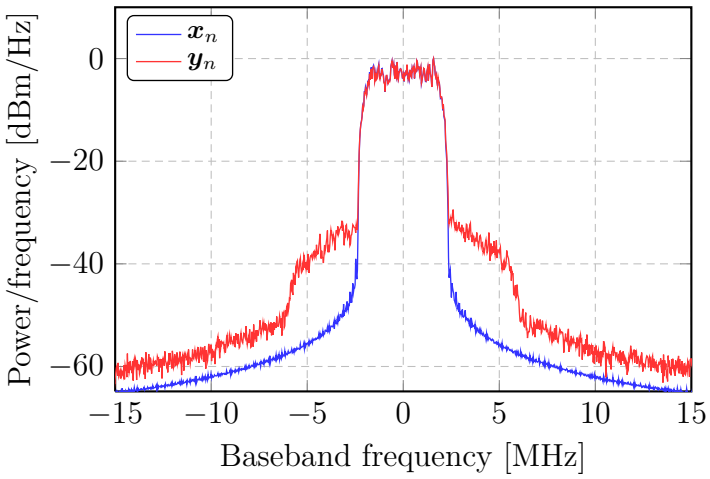
Figure 4.8a and 4.8b shows the AM/AM and the AM/PM conversion for the measured 6 W power amplifier with an average input power level of 8 dBm to the driver PA, which provides approximately 15 dB power gain. Again, the same procedure used for the two previous power amplifiers is used; first, the models are tested with increasing model order and static kernels, and then the model order that yields the best modeling results is used for estimating the optimal kernel length. The results obtained show the same results as for the other amplifiers; once adding a single tap on the linear kernel, the modeling results improves with approximately 10 dB. However, from this point on, a further increase in kernel length does not improve the modeling performance significantly. So a kernel length of 1 to 2 is used in the various models tested.

The main results obtained are presented in Table 4.9. In the table, $m_{\mathrm{V}}\,[m_1\, m_3\, \ldots\, m_P]$ denotes a Pth-order Volterra series, where m_p is the kernel length of the pth dimension, $m_{\mathrm{S}}[M\, Q\, L]$ denotes the spline delay envelope model, were $M = \max_{i,j} \boldsymbol{K}_{i,j}$, Q the kernel length, and L the envelope delay, $m_{\mathrm{W}}[P\, M]$ and $m_{\mathrm{H}}[P\, M]$ denote a Pth-order Wiener- and Hammerstein model, repsectively, both with kernel length M. From the results we see that the Hammerstein model with Chebyshev polynomials models the power amplifier behavior the best, both for nmse and acepr. The use of Chebyshev polynomials are of great significance for the numerical properties of the estimation procedure with the Hammerstein model; using them allows us to increase the model order to $P = 22$ and still have a regression matrix for which the condition number is smaller than 8, which



(a) AM/AM conversion.

(b) AM/PM conversion.



(c) Spectrum.

Figure 4.8: Measured AM/AM and AM/PM characteristics, and output spectrum of a 6 W GaN HEMT power amplifier with an average input power of 23 dBm. Biasing is $V_{\mathrm{D}} = 28\,\mathrm{V}$ and $I_{\mathrm{D}} = 50\,\mathrm{mA}$.

is quite remarkable. Using the same polynomial basis functions for the Wiener model does not result in the same numerical properties in terms of regression matrix condition number, however, the condition numbers achieved are still several orders of magnitude better than the ones of the Volterra series and the spline delay envelope model.

Table 4.7: Modeling results for the Wiener and Hammerstein models with different model settings (nonlinear order, kernel length, basis functions, and so on). Wiener models are denoted $m_{\mathrm{W}}[P\,M]$, where P denotes the order, and M the kernel length. Hammerstein models are denoted $m_{\mathrm{H}}[P\,M]$, with the same meaning for P and M. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the simulated 6 W PA.

Model	Basis/orders	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
$m_{\mathrm{W}}[7\,1]$	$O_n^{(1)}$/even	5	1.44×10^2	−35.25	−55.40	−55.24
$m_{\mathrm{W}}[7\,5]$	$O_n^{(1)}$/even	9	1.45×10^2	−45.82	−59.34	−58.84
$m_{\mathrm{W}}[15\,5]$	$O_n^{(1)}$/even	13	7.47×10^2	−46.50	−62.89	−62.68
$m_{\mathrm{W}}[15\,5]$	$O_n^{(2)}$/even	13	1.05×10^3	−46.70	−64.25	−64.03
$m_{\mathrm{W}}[5\,5]$	L_n/even	8	2.48×10^2	−44.91	−56.88	−56.51
$m_{\mathrm{W}}[5\,5]$	T_n/even	8	3.18×10^4	−45.48	−58.55	−58.16
$m_{\mathrm{W}}[5\,5]$	$\tilde{L}_n$/even	8	2.48×10^5	−45.48	−58.55	−58.16
$m_{\mathrm{W}}[5\,5]$	$\tilde{L}_n$/even	8	2.93×10^1	−45.48	−58.55	−58.16
$m_{\mathrm{H}}[7\,2]$	$O_n^{(1)}$/all	9	3.83×10^5	−55.47	−69.24	−61.84
$m_{\mathrm{H}}[7\,2]$	$O_n^{(1)}$/even	6	6.53×10^1	−54.63	−66.50	−61.22
$m_{\mathrm{H}}[7\,7]$	$O_n^{(1)}$/even	11	6.53×10^1	−56.11	−65.39	−64.25
$m_{\mathrm{H}}[7\,7]$	$O_n^{(2)}$/odd	11	3.14×10^2	−56.09	−65.37	−64.21
$m_{\mathrm{H}}[15\,7]$	$O_n^{(2)}$/even	15	4.20×10^3	−57.60	−68.33	−66.42
$m_{\mathrm{H}}[7\,7]$	P_n/all	14	5.76×10^6	−43.69	−53.44	−53.38
$m_{\mathrm{H}}[4\,2]$	T_n/all	6	4.76×10^1	−52.31	−60.97	−58.97
$m_{\mathrm{H}}[5\,2]$	U_n/all	7	2.43×10^2	−55.15	−67.92	−61.76
$m_{\mathrm{H}}[5\,2]$	$\tilde{L}_n$/all	7	3.92×10^2	−55.60	−69.90	−62.14

Table 4.8: Main modeling results for the spline delay envelope model with different model settings. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the simulated 6 W PA.

Type/$[M\,Q\,L]$	dim $(\boldsymbol{c})$	cond $(\underline{\boldsymbol{H}})$	nmse $(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	acepr $(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
upper/[15 7 7]	265	8.99×10^7	-75.21	-86.11	-86.01
lower/[20 7 7]	315	1.73×10^9	-76.19	-86.98	-87.08
lower/[5 2 2]	13	6.52×10^3	-60.64	-72.30	-72.92

Table 4.9: Main modeling results achieved with various model structures. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 6 W PA.

Model type	Settings	dim ($\boldsymbol{c}$)	cond ($\underline{\boldsymbol{H}}$)	nmse ($\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n$) [dB]	acepr ($\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n$) [dB]	
$m_{\mathrm{V}}[3\,2\,2]$		21	4.66×10^4	-40.41	-50.01	-50.19
$m_{\mathrm{V}}[3\,2\,2\,1]$		22	5.01×10^5	-41.39	-51.52	-51.51
$m_{\mathrm{V}}[2\,1\,1\,1\,1]$		6	4.48×10^5	-40.33	-49.43	-48.41
$m_{\mathrm{V}}[3\,2\,2\,1\,1]$		23	6.03×10^6	-42.16	-53.40	-53.18
$m_{\mathrm{S}}[8\,2\,2]$	diag	15	7.12×10^4	-42.32	-55.55	-55.26
$m_{\mathrm{S}}[15\,2\,2]$	diag	27	1.23×10^6	-42.33	-55.58	-55.26
$m_{\mathrm{S}}[15\,2\,2]$	lower	26	2.99×10^5	-42.34	-55.62	-55.31
$m_{\mathrm{W}}[5\,5]$	$\tilde{L}_n$/even	8	3.27×10^5	-40.21	-50.23	-49.64
$m_{\mathrm{W}}[7\,2]$	$\tilde{L}_n$/even	6	7.01×10^8	-41.29	-52.39	-51.90
$m_{\mathrm{W}}[7\,2]$	$O_n^{(1)}$/even	6	2.73×10^2	-40.40	-50.39	-49.98
$m_{\mathrm{W}}[18\,2]$	$O_n^{(1)}$/even	11	6.97×10^2	-41.76	-54.11	-53.43
$m_{\mathrm{W}}[10\,2]$	$O_n^{(1)}$/all	12	4.19×10^3	-42.16	-55.33	-54.43
$m_{\mathrm{W}}[10\,2]$	$O_n^{(2)}$/all	12	1.42×10^4	-42.16	-55.33	-54.43
$m_{\mathrm{W}}[20\,2]$	$O_n^{(2)}$/even	12	3.16×10^3	-41.90	-54.53	-53.72
$m_{\mathrm{W}}[20\,2]$	$O_n^{(2)}$/odd	12	4.07×10^2	-41.69	-54.20	-53.52
$m_{\mathrm{H}}[5\,2]$	$\tilde{L}_n$/all	7	7.95×10^2	-42.07	-55.21	-53.93
$m_{\mathrm{H}}[5\,2]$	$\tilde{L}_n$/even	5	3.00×10^1	-40.13	-50.27	-49.57
$m_{\mathrm{H}}[5\,2]$	T_n/even	5	7.11×10^0	-39.95	-50.04	-49.68
$m_{\mathrm{H}}[5\,2]$	U_n/even	5	3.44×10^1	-39.95	-50.04	-49.68
$m_{\mathrm{H}}[22\,2]$	U_n/even	13	1.72×10^1	-42.08	-55.23	-53.89
$m_{\mathrm{H}}[22\,2]$	U_n/odd	13	7.42×10^0	-42.06	-55.16	-53.88
$m_{\mathrm{H}}[22\,5]$	U_n/odd	16	7.42×10^0	-42.93	-55.06	-54.09
$m_{\mathrm{H}}[22\,5]$	T_n/odd	16	3.60×10^1	-42.94	-55.09	-54.10

4.8 Summary

This chapter has addressed black-box modeling of microwave power amplifiers with various nonlinear models, such as the Volterra series. The different model structures presented in the chapter were tested with both measured and simulated power amplifiers, and the results clearly show the difference of modeling a device for which its input/output data is obtained from an ideal simulation environment versus a device for which its input/output data is obtained from a real measurement setup. Limitations in measurement equipment together with measurement noise effectively limits the accuracy of the model. If a too high model order is chosen when modeling based on measured data, noise-fitting is a potential outcome. In the idealized simulation environment this is not a problem as noise is not present, unless added deliberately.

Otherwise, the obtained results show that for the power amplifiers modeled, the eventual memory effects present seem to be concentrated; kernel lengths greater than 1 to 2 taps just increase the model complexity without improving the modeling performance significantly. *Eventual* is used because it is difficult to classify whether or not such effects are present. However, it should be noted that this can also indicate that limitation is in the models, and that they just do not manage to model the memory effects, which can be the case for long term memory due to trapping effects in the semiconductor. If it is desirable to confirm the presence of such effects, and how they affect the system, there exist better, but much more complex methods for this purpose [19, ch. 6]. In general, the presence of memory effects becomes more obvious with increased input signal bandwidth and/or high-power devices, so it might be that longer kernels would be required in the model structures if i.e. signals from the 802.11n standard, which supports bandwidths up to 40 MHz, were used in the modeling. The results also show that using higher-order Volterra series do not yield better modeling results than lower-complexity models, such as the memory polynomial model, or simplified Wiener and Hammerstein models.

Another interesting observation is the numerical properties of the spline delay envelope model. The use of cubic splines in the computation of the regression matrix should improve the numerical properties of the estimation, however, the results clearly show that if good modeling is desired with this model structure, it will be at the expense of poor numerical properties. The best numerical properties are achieved using the Hammerstein and Wiener models together with orthogonal polynomial sequences. Of the different types tested, the Chebyshev polynomials, both the first and second kind, show the best numerical properties together with the two unnamed types (see Table 2.1).

Digital Predistortion Linearization 5

Linearization techniques for nonlinear microwave power amplifiers have been around for decades, ranging from analog techniques such as feedforward linearization [6, ch. 5] and Cartesian loop feedback correction [6, ch. 4], to digital techniques such as digital predistortion linearization [8, 52, 84]. As processing power has become cheaper and more powerful over the last two decades, mainly due to the great advances in digital signal processing (DSP), digital predistortion linearization has become one of, if not the most cost efficient linearization technique available for microwave power amplifiers. The main principles behind the technique are similar to those of Cartesian loop linearization, however in the digital domain. The overall goal is to design a block which compensates for nonlinear effects present in the power amplifier in digital baseband, allowing us to utilize digital signal processing techniques to achieve great precision [7, pp. 7]. Prior to designing a digital predistorter, a model of the nonlinear microwave power amplifier is often required in order to estimate its inverse. Deriving a good behavioral model for the nonlinear system is often the most difficult step in the design process of a digital predistorter. However, after such a model has been found, the next step is to estimate the model parameters, either by an direct approach with least squares methods, or by an adaptive/iterative approach using i.e. an adaptive filter such as the least mean square (LMS) filter [85, ch. 5 – 6], or the recursive least square (RLS) filter [85, ch. 9]. For this work, a direct approach has been chosen; mainly due to its simplicity, however, also because an adaptive approach for online linearization would require implementation on a field programmable gate array (FPGA), a time-consuming process for someone with limited knowledge of FPGA/digital design. In the following, preliminary theory for digital predistortion linearization is presented, together with simulated and measured results acquired with two of the same power amplifiers used for the behavioral modeling in the previous chapter. Finally, the chapter is ended with a discussion of the achieved results.

5.1 Formulating the Linearization Problem

Linearization of microwave power amplifiers can be represented graphically as in Figure 5.1. The digital predistortion unit $\mathbb{T}[\cdot]$ predistorts the input signal $\boldsymbol{x}_n$ so the input signal $\boldsymbol{u}_n$ to the power amplifier $\mathbb{P}[\cdot]$ follows the inverse characteristic of the power amplifier. In the ideal case, $\mathbb{T}[\cdot]$ represents the exact inverse transfer characteristic of the nonlinear power amplifier, leaving us with $\boldsymbol{y}_n = \boldsymbol{d}_n$, where $\boldsymbol{d}_n = \mathbb{L}[\boldsymbol{x}]$ is the desired linear response. If only linear amplification with a voltage gain g is desired, $\mathbb{L}[\boldsymbol{x}] = g\boldsymbol{x}_n$. In order to achieve $\boldsymbol{y}_n = g\boldsymbol{x}_n$, $\mathbb{T}[\cdot]$ must represent the exact inverse of $\mathbb{P}[\cdot]$.

In general this is not the case, as only an estimated model of the nonlinear system is available — for nonlinear systems an analytical solution for the inverse rarely exists because the requirements for bijection seldom are satisfied. However, for many nonlinear systems, and especially microwave power amplifiers, local solutions for the inverse exist; this implies that for a restricted set of input signals around an operating point $\boldsymbol{x}_n^{(0)}$, the microwave power amplifier output can be linearized so we have $\boldsymbol{y}_n \approx \boldsymbol{d}_n$, or

$$\left\| \mathbb{P}\left[\mathbb{T}\left[\boldsymbol{x}_n\right]\right] - \mathbb{L}\left[\boldsymbol{x}_n\right] \right\| < \delta \quad \text{if} \quad \boldsymbol{x}_n \in \left\{ \boldsymbol{x}_n \middle| \|\boldsymbol{x}_n^{(0)} - \boldsymbol{x}_n\| < \epsilon \right\}, \tag{5.1}$$

where ϵ and δ are small, positive constants.

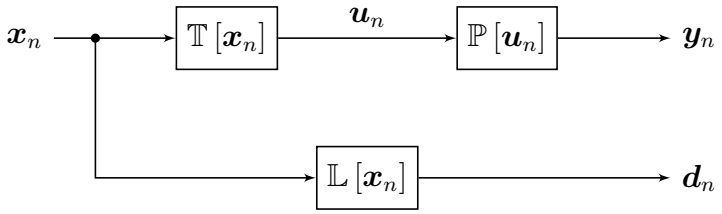


Figure 5.1: Predistortion linearization problem of a nonlinear microwave power amplifier.

There exist several methods for estimating the inverse characteristic of a nonlinear system; for systems that can be described by Volterra systems, a technique known as the pth-order inverse, presented by Martin Schetzen in [49, ch. 7] and [86], can be applied. Using this method, all the nonlinear effects up to the pth order are compensated for, given that the inverse of the first-order kernel of the Volterra system is causal and stable. Besides compensating for nonlinear effects up to the pth order, the technique also introduce new, higher order distortion terms from the $(p+1)$th order and upwards.

This, together with the high computational complexity is an obvious drawback.[1] Hence, for estimating the inverse of the microwave power amplifier with less complexity, an approach known as the indirect learning architecture (ILA), presented in [9], is used in this work. Using this method, the inverse characteristic of the microwave power amplifier is identified by using its output signal to predict its input signal, a postdistortion process. Martin Schetzen also proved in [49, ch. 7] that a general pth-order postinverse of a

[1] To get a glimpse of the complexity, see [87], where the pth-order inverse technique was applied to predistort a traveling wave tube (TWT) power amplifier used in wavelet packet division multiplexing.

Volterra system is identical to the pth-order preinverse, allowing us to use estimated coefficients of a postinverse filter in the design of a preinverse filter (predistorter) [52]. A block-schematic illustrating how this method can be implemented is given in Figure 5.2a. In order to use this method, it is required to assume a specific structure of the postinverse filter — the better the assumed structure agrees with the actual and unknown inverse, the better the linearization performance. Referring to Figure 5.2a, the error $\boldsymbol{e}_n$ between the postinverse output signal, and the power amplifier input signal is given by

$$\boldsymbol{e}_n = \mathbb{T}\left[\boldsymbol{x}_n\right] - \mathbb{T}\left[\frac{1}{g}\boldsymbol{y}_n\right], \tag{5.2}$$

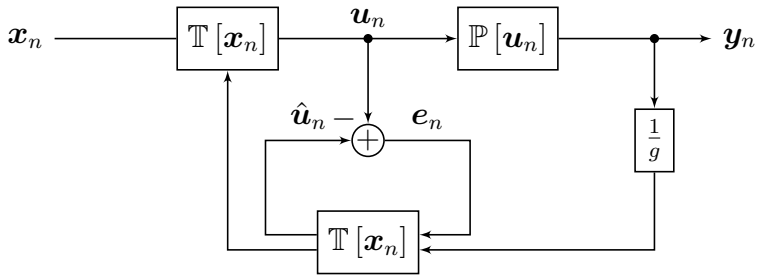
$$= \boldsymbol{u}_n - \hat{\boldsymbol{u}}_n, \tag{5.3}$$

where g is the desired linear gain.[2] If the assumed structure of the postinverse filter represents the exact inverse characteristics of $\mathbb{P}\left[\cdot\right]$, the error will be equal to zero and perfect linearization is achieved, or $\boldsymbol{y}_n = g\boldsymbol{x}_n$. However, since the assumed postinverse filter structure in general will differ from the exact inverse of the nonlinear system to some extent, some error will be present. Therefore the output of the microwave power amplifier after predistorting its input is given by

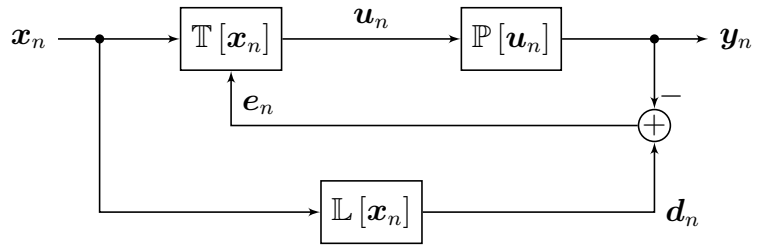
$$\boldsymbol{y}_n = g\mathbb{T}^{-1}\left[\mathbb{T}\left[\boldsymbol{x}_n\right] - \boldsymbol{e}_n\right], \tag{5.4}$$

where $\mathbb{T}^{-1}\left[\cdot\right]$ denotes the inverse of the predistorter unit. Another commonly used method for identifying the coefficients for a predistorter is given in Figure 5.2b, see [88, 89]. This method is often referred to as the direct learning architecture (DLA). As for the ILA technique, a model of the nonlinear system is not required, however, a model structure of the predistorter unit must be assumed. Using this assumption, the coefficients of the assumed inverse model can be estimated in a similar way as for the ILA (see the next section). The main difference is that instead of estimating the postinverse filter and then copy its filter coefficients to the preinverse filter, the preinverse filter is estimated directly by a feedback error signal $\boldsymbol{e}_n$ using adaptive algorithms. And as with the ILA, the better the assumed models structure agrees with the exact and unknown inverse characteristic, the better the linearization performance. The main advantage with both these methods is that they are not bounded to a specific model structure — so general algorithms for estimation of the inverse system can be derived.

[2] It should be emphasized that desired linear gain g cannot be greater than what the actual linear gain of the microwave power amplifier is.



(a) Identification of predistortion unit without knowledge of the nonlinear PA.



(b) Identification of predistortion unit without knowledge of the nonlinear PA.

Figure 5.2: Digital predistortion learning architectures.

5.2 Estimation Algorithm for Digital Predistortion Linearization

As mentioned in the prelude of this chapter, a digital predistortion unit can be estimated by both direct least squares methods, and with adaptive/iterative methods using i.e. adaptive filters. The direct least squares methods have been adopted in this work, both due to their simplicity, but also because good linearization performance can be achieved using such methods, e.g. 15 dB to 20 dB improvement in acpr [52]. Since the learning architectures discussed in the previous section are chosen for the estimation procedure of the predistorter unit, a general algorithm that can be used with all the model structures presented in Chapter 4 is given in the following. Given that the system input is $\boldsymbol{u}_n$, and the system output is $\boldsymbol{y}_n$, the first step is to estimate the output of the postfilter by solving

$$\hat{\boldsymbol{u}}_n = \underline{\boldsymbol{H}}_{\boldsymbol{y}_n} \boldsymbol{c}_{\text{post}} \,, \tag{5.5}$$

where $\underline{\boldsymbol{H}}_{\boldsymbol{y}_n} \in \mathbb{C}^{N \times \nu_{\text{tot}}}$ denotes the regression matrix for any of the models presented in the previous chapter composed with $\boldsymbol{y}_n$ as the model input, and $\boldsymbol{c}_{\text{post}} \in \mathbb{C}^{\nu_{\text{tot}}}$ is the coefficient vector for the postinverse filter. The error between the actual power amplifier input, and the estimated power amplifier input is given by

$$\boldsymbol{e}_n = \boldsymbol{u}_n - \hat{\boldsymbol{u}}_n \,. \tag{5.6}$$

The least squares solution that minimizes $\|\boldsymbol{e}_n\|_2^2$ is given by

$$\hat{\boldsymbol{c}}_{\text{post}} = \left(\underline{\underline{\boldsymbol{H}}}_{\boldsymbol{y}_n}^{\mathsf{H}} \underline{\underline{\boldsymbol{H}}}_{\boldsymbol{y}_n}\right)^{-1} \underline{\underline{\boldsymbol{H}}}_{\boldsymbol{y}_n}^{\mathsf{H}} \boldsymbol{u}_n \,. \tag{5.7}$$

Once $\hat{\boldsymbol{c}}_{\text{post}}$ is found, the predistorted power amplifier input signal in the forward path is found by solving

$$\boldsymbol{u}_n = \underline{\underline{\boldsymbol{H}}}_{\boldsymbol{x}_n} \hat{\boldsymbol{c}}_{\text{post}} \,, \tag{5.8}$$

where $\underline{\underline{\boldsymbol{H}}}_{\boldsymbol{x}_n} \in \mathbb{C}^{N \times \nu_{\text{tot}}}$ now denotes the regression matrix for the actual model structure composed with $\boldsymbol{x}_n$. In (5.5) – (5.7) we have exchanged the original input signal $\boldsymbol{x}_n$ with $\boldsymbol{u}_n$. This is because that $\boldsymbol{x}_n$ will only be the actual power amplifier input signal if the estimation algorithm is run once. If run two or more times, the power amplifier input signal will for all the successive iterations be $\boldsymbol{u}_n$, where the input signal is adjusted according to the estimation error for the present iteration. This is done by updating $\boldsymbol{c}_{\text{post}}$ for each iteration, where the update process can either be done partially by a Gauss-Newton algorithm in order to force some continuity in the updating process, or by just replacing the previous estimate of $\boldsymbol{c}_{\text{post}}$ with the new [52]. If updated partially, this amounts to

$$\boldsymbol{c}_{\text{post}}^{(l+1)} = \boldsymbol{c}_{\text{post}}^{(l)} + \mu \left(\underline{\underline{\boldsymbol{H}}}_{\boldsymbol{y}_n}^{\mathsf{H}} \underline{\underline{\boldsymbol{H}}}_{\boldsymbol{y}_n}\right)^{-1} \underline{\underline{\boldsymbol{H}}}_{\boldsymbol{y}_n}^{\mathsf{H}} \boldsymbol{e}_n \tag{5.9}$$

where $\boldsymbol{c}_{\text{post}}^{(0)}$ is the initial estimate of the predistorter coefficient vector, and μ is the step-size that regulates how much the present error vector can affect the coefficient vector update. If $\mu = 1$ and $\boldsymbol{c}_{\text{post}}^{(0)} = \boldsymbol{0}$, the algorithm converges to the least squares solution (5.7) in one iteration. If $\mu < 1$, the coefficient vector of iteration $l+1$ will depend on the coefficient vector of iteration l, forcing some continuity in the updating process. Forcing continuity in the updating process has the advantage of robustness; since it is the error vector $\boldsymbol{e}_n$ that drives the weight update, a solution that minimizes the average error is always obtained [52]. However, the drawback is the increased computational complexity. If the last term of (5.9) is solved with i.e. SVD, a great amount of calculations are required for each iteration, which obviously is non-ideal for applications where processing power is limited, or time usage must be at a minimum.

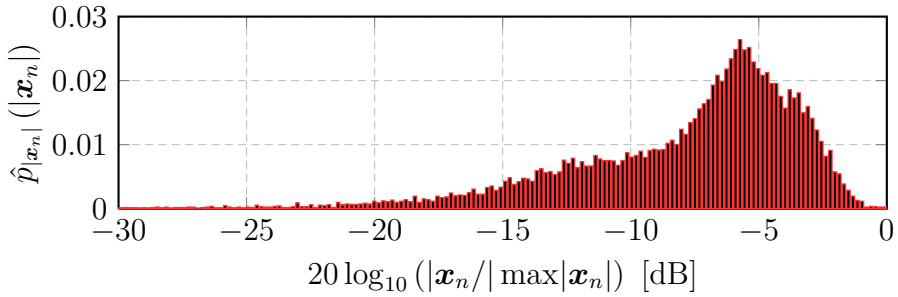
5.3 Limitations of Digital Predistortion Linearization

Although the concept of digital predistortion linearization shows many advantageous properties, there are some drawbacks which should be considered. Besides the fact that increased complexity requires more processing power, there are also limitations present due to other components in a transmitter chain, such as the specifications of the DAC. In theory, once the inverse characteristics of the power amplifier have been found, it is just to predistort the signal and excite the power amplifier with it in order to achieve good linearization. In practice, it is not as simple as this. Once predistorting the signal, both the bandwidth and the papr of the signal increases. If the bandwidth of the original

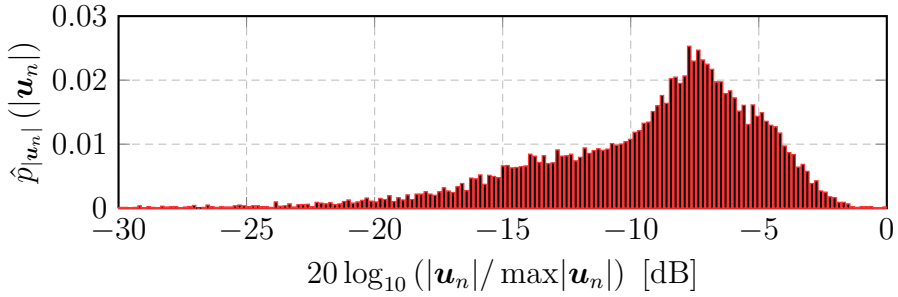
input signal prior to oversampling is 5 MHz, the bandwidth of the predistorted signal will be $P \times 5$ MHz using a predistorter of nonlinear order P. After oversampling, i.e. with a rate of 16, the DAC is required to convert a signal of $P \times 80$ MHz, which easily can exceed the maximum bandwidth the DAC supports if a high enough nonlinear order P is used in the predistorter. With other words, the DAC in the system is effectively limiting the model structures that can be used in the predistorter design, and a trade-off must be made in terms of linearization accuracy versus the complexity/cost of the system.

DACs that support high bandwidths ($B \geq 400$ MHz) with good resolution (≥ 14 bit) are expensive components, and they are usually very power hungry. Thus, if predistortion is used for operating the power amplifier in high-efficiency operation, there is a possibility that the expenses absorb the profits. In addition, the papr of the predistorted signal, as mentioned, increase. This is because the predistorter tries to compensate for the compression the signal experience in the power amplifier, no matter how far into compression the signal peaks are. This effect can be seen in Figure 5.3a and 5.3b.

Prior to predistortion, the signal has a papr of approximately 6.50 dB. After predistortion, the papr is increased to about 8 dB, which corresponds to the level of compression the peaks of the original input signal experienced when exciting the power amplifier. Now, if the predistorted signal is applied to the power amplifier without backing off the input power, the result will be similar to a hard-limiter, where the peaks of the predistorted signal will be hard-clipped, and result in a more severe nonlinearity. Thus, depending on how close to saturation the power amplifier is operated, input back-off must be used together with predistortion in order to avoid hard-clipping.



(a) Before predistortion.



(b) After predistortion.

Figure 5.3: Normalized envelope distribution of 16-QAM signal before and after digital predistortion. The model structure of the predistorter used to generate $\boldsymbol{u}_n$ is a memory polynomial model with $P = 7$ and $M = 4$.

5.4 Validation of Linearization Performance

In this section the different model structures discussed in Chapter 4 are tested together with the ILA technique for digital predistortion linearization of a simulated and a measured power amplifier. For the simulations, the test-bench for the ADS/MATLAB cosimulation is used (see Appendix B), whereas for the measurements, the large signal measurement setup presented in Appendix E.2 is used. In this way, the performance of the different model structures in combination with the ILA technique can be evaluated under both ideal and non-ideal conditions. Prior to presenting the results, the validation criteria used are given. The input signal used in all simulations and measurements is computed with the same parameters as mentioned in Section 4.7.

5.4.1 Validation Criteria for Linearization Performance

As with the behavioral modeling in the previous chapter, the nmse is used as a validation criterion. With reference to Figure 5.1, the desired signal is now the linearly amplified version of the input signal, or $\boldsymbol{d}_n = g\boldsymbol{x}_n$. In addition, the nmse of the magnitude and the phase is used as well, e.g.

$$\mathrm{nmse}_{\mathcal{A}}\left(\boldsymbol{d}_n, \boldsymbol{y}_{n_{\mathrm{PD}}}\right) = 10\log_{10}\left(\frac{\left\|\left|\boldsymbol{d}_n\right| - \left|\boldsymbol{y}_{n_{\mathrm{PD}}}\right|\right\|_2^2}{\left\|\left|\boldsymbol{d}_n\right|\right\|_2^2}\right) \quad [\mathrm{dB}]\,, \tag{5.10}$$

and

$$\mathrm{nmse}_{\phi}\left(\boldsymbol{d}_n, \boldsymbol{y}_{n_{\mathrm{PD}}}\right) = 10\log_{10}\left(\frac{\left\|\arg\left(\boldsymbol{d}_n\right) - \arg\left(\boldsymbol{y}_{n_{\mathrm{PD}}}\right)\right\|_2^2}{\left\|\arg\left(\boldsymbol{d}_n\right)\right\|_2^2}\right) \quad [\mathrm{dB}]\,. \tag{5.11}$$

Together with the different variations of the nmse, the adjacent channel power ratio (acpr) is used. It is defined as the ratio of the total power in an adjacent channel to the total power in the main channel, and is usually expressed in dB relative to carrier (dBc) [5]. Thus, it represents a good measure for out-of-band linearity of microwave power amplifiers. For a signal $\boldsymbol{y}_n$, the acpr is given by

$$\mathrm{acpr}\left(\boldsymbol{y}_n\right) = 10\log_{10}\left(\frac{\int_{\mathrm{adj.}} \left|Y(f)\right|^2 \,\mathrm{d}f}{\int_{\mathrm{main\ ch.}} \left|Y(f)\right|^2 \,\mathrm{d}f}\right) \quad [\mathrm{dBc}]\,, \tag{5.12}$$

where $Y(f)$ is the discrete Fourier transform of $\boldsymbol{y}_n$. If it is desirable to validate the in band linearization performance, figure of merits like the error vector magnitude (EVM), which is the distance between the desired and the actual signal vectors, can be used. This is however not done in this work; hence, the nmse is used to classify the overall performance, whereas the acpr is used to classify the out-of-band performance.

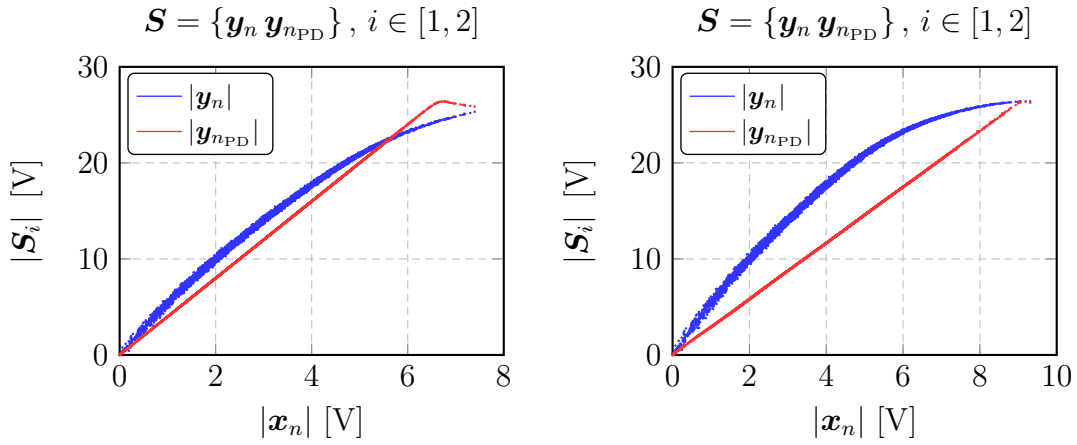
5.4.2 Linearization of a Simulated 6 W GaN HEMT PA at 3.40 GHz

For the simulated 6 W power amplifier, the results of testing two of the previously discussed model structures are presented; the Volterra series and the memory polynomial model. Since the environment now is ideal with no noise present, increased nonlinear order for the model structure will not result in noise-fitting; hence, for the Volterra series, a model structure $m_{\mathrm{V}}[3\,3\,2\,2\,1\,1]$ is chosen, whereas for the memory polynomial model, the model structure chosen is $m_{\mathrm{MP}}[9\,5]$. The simulated results (nmse and acpr) are presented in Figure 5.5a – 5.5f, and in Figure 5.6a, and 5.6b.

As we see from the plots showing the overall nmse, both models yield a linearization performance for which the error between the linearized output signal and the desired output signal is less than -40 dB. Just as expected, the closer to saturation the PA is operated, the worse the linearization performance (see Figure D.2 for large signal performance of PA). With an average input power of 17 dBm, the peaks of the 16-QAM signal are in approximatley 1.50 dB compression. If operated closer to saturation, predistorting the input signal will only lead to hard clipping unless the input power is backed off, as can be seen in Figure 5.4. Otherwise, we see from Figure 5.5c – 5.5f that phase compression is more accurately linearized than the envelope compression.

This is due to the relatively linear phase characteristics of the device without linearization applied. However, the reason for the sudden jumps of 15 dB in nmse_{ϕ} is

unknown. But despite these jumps, the phase characteristic of the linearized output signal relatively close to the desired one. For linearization performance in terms of acpr, we see that both models yield very good results for all input powers used in the simulation. The memory polynomial shows the best performance of the two models presented, and for some input powers the acpr is reduced with more than 25 dB.



(a) Withouth input back-off. **(b)** With input back-off.

Figure 5.4: Linearization of PA operated well into saturation. For the case without input back-off, the average input power is 21 dBm, whereas for the case with input back-off, the average input power is 23 dBm.

5.4.3 Linearization of a Measured 6 W GaN HEMT PA at 2 GHz

First, the digital predistortion algorithms are tested on the 6 W GaN HEMT power amplifier with it being biased in deep class AB. In Figure 5.7a – 5.7d the AM/AM and AM/PM conversion of the power amplifier, with and without predistortion applied, is presented. Here $\boldsymbol{x}_n$ is the original input signal, $\boldsymbol{u}_n$ the predistorted input signal, $\boldsymbol{y}_n$ and $\boldsymbol{y}_{n_{\mathrm{PD}}}$ the output signal with, and without predistortion applied, respectively. From the AM/AM conversions, we see that the power amplifier shows a nonlinear behavior, compressing the peaks of the input signal. Using digital predistortion, the AM/AM response becomes more linear, however, the peaks are again compressed, which can suggest that the power amplifier either is driven too close to saturation that linearizing it will not improve the linearity of the peak values significantly, or that the model structure assumed for the postinverse filter deviates more than expected from the true, unknown inverse of the power amplifier. As for the AM/PM conversion, the power amplifier itself shows a surprisingly linear phase compression for the given input power level. Given the nonlinear behavior seen in the AM/AM conversion, a more nonlinear behavior is expected for the phase compression, especially for the highest peaks. The

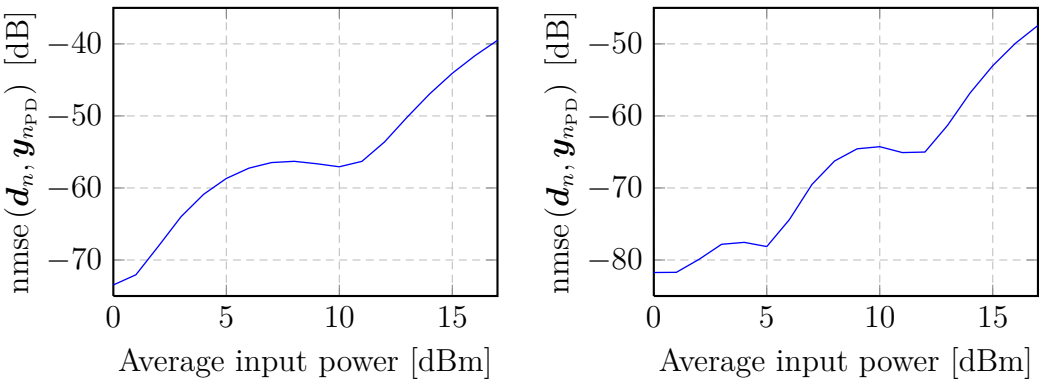
reason for the samples being scattered is most likely noise in the measurement setup, however, it can also be memory effects in the power amplifier itself that cause this effect. After applying linearization the phase compression seemingly improves a bit, showing a slightly more linear phase compression. However, due to the highly scattered data, any specific conclusion is difficult to draw. Figure 5.8a and 5.8b shows the output spectrum with and without linearization applied. Here we can see that applying linearization indeed improves the linearity of the power amplifier, resulting in reduced acpr.

However, the obtained results are not as good as the best results reported in literature [52, 72]. Without linearization applied, the acpr is calculated to be $-35.26\,\text{dBc}$ and $-34.87\,\text{dBc}$ for the lower and upper adjacent channels, respectively. Applying linearization, the results presented in Table 5.1 are obtained. From these results we see that when applying linearization, the acpr is reduced with 10 dB to 12 dB, depending on the algorithm used for linearization. The model structures used for linearization in the AM/AM and AM/PM plots are among the two that shows the best linearization performance with respect to the phase of the predistorted signal and the desired output signal. Although showing the best results achieved, the achieved performance is still far from the expected and desired performance.

Of the various combinations tested for the memory polynomial model, the model structure using Legendre polynomials as basis functions with a maximum polynomial order of $P = 2$ and a single-tap kernel yields the best linearization results with respect to acpr. The model with these settings also achieves the best $\text{nmse}_{\mathcal{A}}$ value of all the memory polynomial models tested. However, with respect to the phase compression, this combination of model settings do not yield any good linearization at all, only achieving $\text{nmse}_{\phi} \approx -4\,\text{dB}$. The poor performance can be a result of comparing the ideal MATLAB generated signal, linearly amplified with the desired voltage gain, and the linearized output signal directly. Since the signal will be corrupted by noise, and experience some phase delay in the VSG and the cables prior to the driver PA, the signal entering the driver PA is not the same as the ideal signal generated in MATLAB.

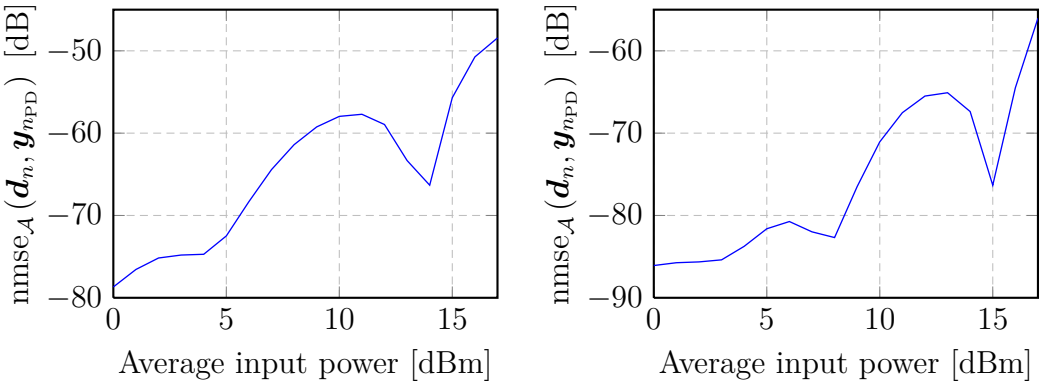
Hence, it could be interesting to see if improved performance is achieved if the driver PA input signal is sampled, i.e. by just connecting the VSG output directly to the VSA input, using the same cable that the VSG normally uses to connect to the driver PA, and then use this signal in the predistortion algorithm as the original input signal to the driver PA. The algorithms are also tested with the power amplifier biased closer to class B as well ($V_{\text{D}} = 28\,\text{V}$ and $I_{\text{D}} = 18\,\text{mA}$). With the new biasing conditions quite similar linearization performance was surprisingly achieved, even when driving the power amplifier harder (average input power to driver PA was 9 dBm). Without linearization, the acpr is $-34.63\,\text{dBc}$ and $-34.42\,\text{dBc}$ for the lower and upper adjacent channel, respectively. Using a 3rd order memory polynomial model with a kernel length of $M = 5$, the acpr is reduced to $-44.75\,\text{dBc}$ and $-43.98\,\text{dBc}$ for the lower and upper adjacent channels, respectively. Not the same amount of reduction, but still approximately a 10 dB improvement. When backing off the input power to 6 dBm, the improvement seen in acpr with the same model structure is almost 15 dB. The other model structures presented in Chapter 4 are tested as well, but for some reason none of them yields good linearization results. This indicates that the MATLAB implementation of the predistortion algorithm

for these models needs some more tweaking. Based on the modeling results obtained with the same models when used for black-box modeling, it was expected that they would result in quite good linearization, at least similar to results obtained with the two models that worked properly.



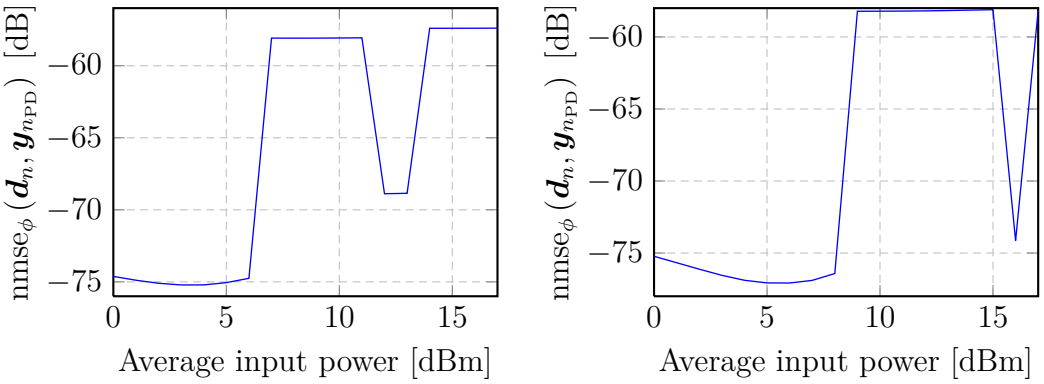
(a) $m_{\mathrm{V}}[3\,3\,2\,2\,1\,1]$.

(b) $m_{\mathrm{MP}}[9\,5]$.



(c) $m_{\mathrm{V}}[3\,3\,2\,2\,1\,1]$.

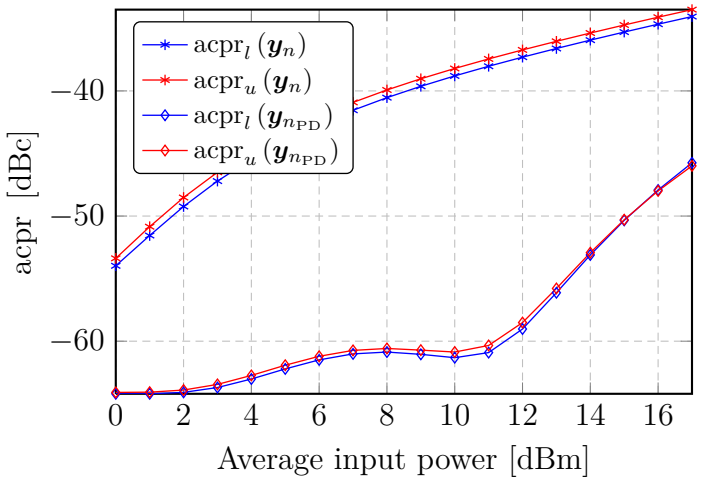
(d) $m_{\mathrm{MP}}[9\,5]$.



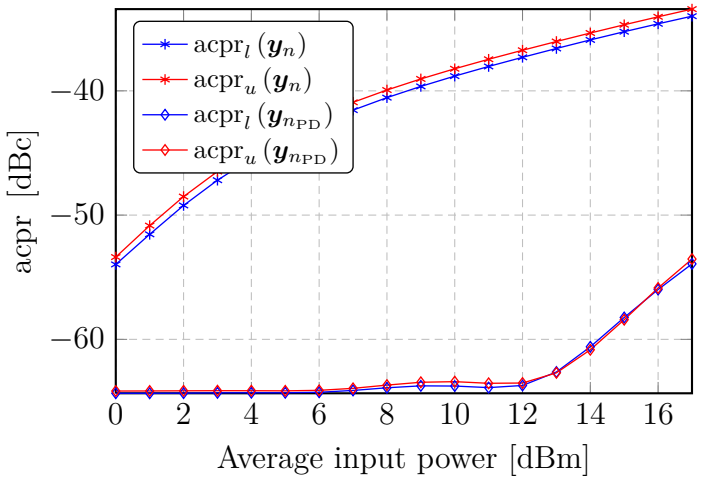
(e) $m_{\mathrm{V}}[3\,3\,2\,2\,1\,1]$.

(f) $m_{\mathrm{MP}}[9\,5]$.

Figure 5.5: Achieved linearization results (nmse) for both Volterra series and memory polynomial model. The PA used is the simulated 6 W PA biased with $V_{\mathrm{D}} = 28\,\mathrm{V}$ and $I_{\mathrm{D}} = 11\,\mathrm{mA}$.

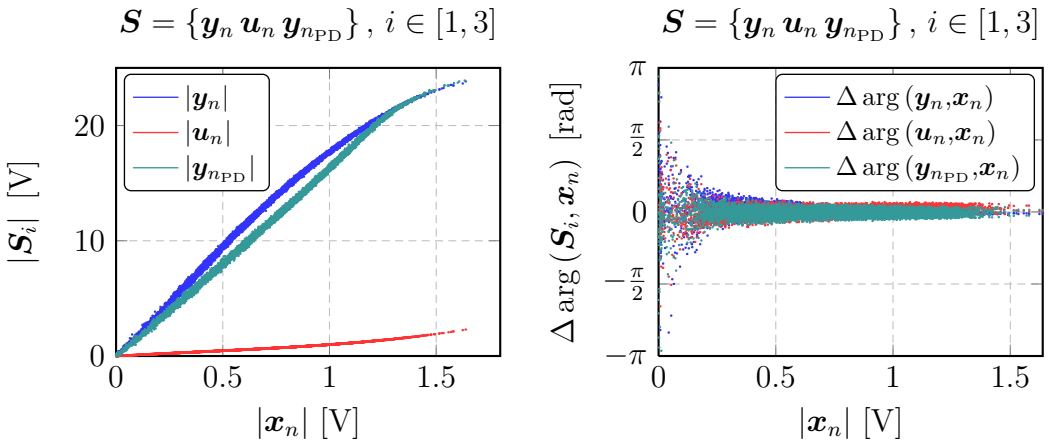


(a) $m_{\mathrm{V}}[3\,3\,2\,2\,1\,1]$.



(b) $m_{\mathrm{MP}}[9\,5]$.

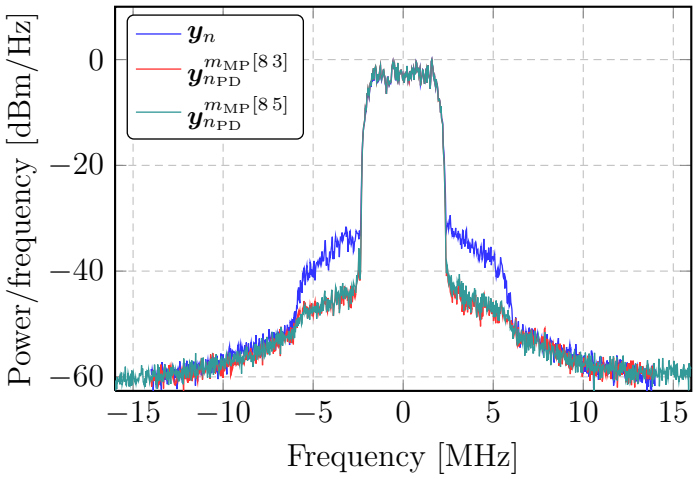
Figure 5.6: Achieved linearization results (acpr) for both Volterra series and memory polynomial model. The PA used is the simulated 6 W PA biased with $V_{\mathrm{D}} = 28\,\mathrm{V}$ and $I_{\mathrm{D}} = 11\,\mathrm{mA}$.



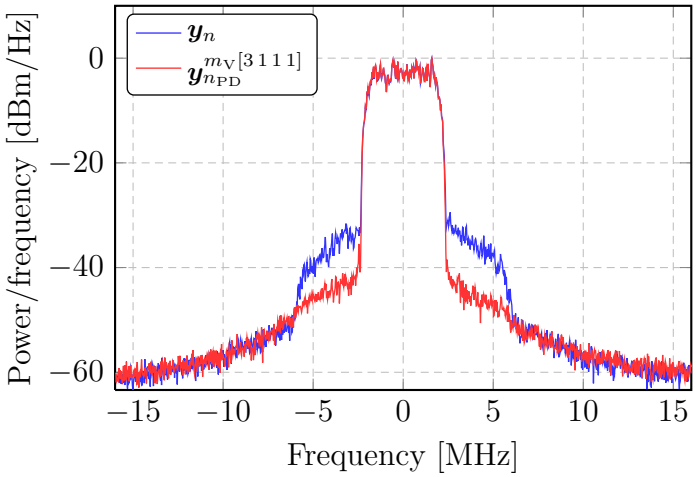
(a) AM/AM. **(b)** AM/PM.

(c) AM/AM. **(d)** AM/PM.

Figure 5.7: Digital predistortion of 6 W PA biased with $V_{\mathrm{D}} = 28\,\mathrm{V}$ and $I_{\mathrm{D}} = 50\,\mathrm{mA}$. The model structure chosen for the predistorter is memory polynomial model with $P = 8$ and $M = 3$ for a) and b), whereas a Volterra series of order $P = 7$ with the kernel lengths [3 1 1 1] for c) and d). The average input power to the driver PA is 8 dBm for both cases.



(a) Memory polynomial predistorter. The polynomial basis functions for the red trace is the Legendre polynomials, whereas for the teal trace it is the Laguerre polynomials. In both cases, only even orders are used in the composition of the orthogonal polynomial sequence.



(b) Volterra series predistorter.

Figure 5.8: Output spectrum with and without predistortion using different model structures. The power amplifier is biased with $V_{\mathrm{D}} = 28\,\mathrm{V}$ and $I_{\mathrm{D}} = 50\,\mathrm{mA}$, and the average input power to the driver PA is 8 dBm for all cases.

Table 5.1: Linearization performance for memory polynomial predistorter with various model settings. The average input power to the driver PA is 8 dBm.

Type/basis/orders	P	M	nmse [dB]	$\text{nmse}_{\mathcal{A}}$ [dB]	nmse_{ϕ} [dB]	acpr_{l} [dBc]	acpr_{u} [dBc]
reg/L_n/even	8	3	−11.89	−35.43	−14.50	−45.64	−45.64
reg/$\tilde{L}_n$/even	8	5	−11.93	−28.29	−20.68	−45.30	−44.92
reg/$\tilde{L}_n$/even	3	5	−8.22	−34.59	−5.78	−45.91	−44.70
reg/$\tilde{L}_n$/even	2	5	−2.10	−32.24	−3.51	−46.17	−45.13
reg/$\tilde{L}_n$/all	2	1	−3.90	−37.72	−4.27	−46.44	−46.17
reg/$\tilde{L}_n$/all	2	0	−6.01	−25.77	−7.35	−46.06	−46.71

5.5 Summary

Digital predistortion linearization of microwave power amplifiers has been addressed in this chapter. The model structures presented in the previous chapter were all tested with the ILA technique for estimating the inverse structure of the power amplifier, both in an ideal simulation environment, and a real measurement setup. Results from the ideal environment again show the superior performance that can be achieved when no limitations in the measurement equipment are present, as expected.

In the simulations, the acpr was for some input power levels reduced with 25 dB, whereas the maximum reduction achieved in the measurements was 15 dB. The measured linearization perfomance is comparable to results presented in the literature [52, 72], however, since different types of microwave power amplifiers and measurement setups are used, it is difficult to conclude if the obtained results are better or not. Otherwise, we have seen that even if the theory behind digital predistortion linearization is quite straight forward, implementing a predistorter that works as desired is not a trivial task.

Although all the models presented in the previous were tested, only two of them worked as desired. Tweaking of the MATLAB implementation for these models did not lead to desired results either. The biggest problem experienced was the huge increase in papr; once a signal had been predistorted, its papr value could for some models become much greater than the expected value, which for the input power level the PA was operated at would result in an increase of maybe 1 dB to 3 dB, and not 10 dB. Thus, it seems like measurement noise or other limitations in the measurement setup have a much bigger impact on the estimation of the inverse characteristic for some specific nonlineer models than expected.

Conclusions 6

In this work a discrete 6 W GaN HEMT transistor from Cree Inc. has been characterized through load pull simulations and measurements. For the simulations, a nonlinear transistor model provided by Cree Inc. was used together with a load pull setup in Agilent ADS to determine source and load impedances for optimal operating conditions. Further, two prototype PCB designs on a Rogers 4003 substrate were made, only including the transistor and two RC filters for unconditional stability mounted on a PCB.

Load pull simulations showed that the device is capable of delivering an output power greater than 13 W when operated far into compression. However, in 1 dB compression, the output power for the corresponding sets of impedances was in the region of 11.50 W, together with a power added efficiency just below 70 % and a power gain of 13 dB, a remarkable result for a 6 W device. Simulations also showed that when matching for maximum efficiency, the device can operate at a drain efficiency greater than 90 % in 5 dB compression. For operation in 1 dB compression, the maximum power added efficiency level achieved was in the region of 71 %.

The measured output properties of the device did not match the simulated. With load pull measurements, the peak output power measured was slightly greater than 11 W, however, in 4 dB compression. As for power added efficiency, the peak levels measured were in the region of 70 %, but again far into compression. In 1 dB compression, the measured results were not comparable to the simulated ones at all. This is likely due to the sub–optimal tuning of the source and load impedance tuners, which was a result of the problems experienced with the load pull measurement setup.

Since the peak levels measured were relatively comparable to the simulated results, and because S parameter measurements and simulations agreed well, it is likely that the results from the load pull simulations are achievable, and can be replicated with a fully working measurement setup. Thus, CGH4006P from Cree Inc. shows great potential for high–efficiency operation while delivering above rated output power.

Black-box behavioral modeling and linearization by digital predistortion have also been addressed in this work. Various models for nonlinear and dynamic microwave power amplifiers have been implemented in MATLAB, and evaluated under both ideal and non-ideal conditions through simulations and measurements, respectively. Based on

simulations and measurements, it was shown that limiting effects in the measurement setup, together with other perturbing effects, such as noise, have great impact on both the modeling and linearization performance.

The obtained results show that all the implemented models achieve nearly the same modeling performance, suggesting that for behavioral modeling of microwave power amplifiers, low–complexity models such as the memory polynomial model, and the simplified Wiener and Hammerstein models used in this work are sufficient for achieving good results. As for linearization performance, the memory polynomial model obtained the best performance in terms of reduced adjacent channel power ratio.

However, the measured linearization performance was not as good as simulated. This is likely due to long term memory effects which are caused by trapping phenomena in the semiconductor [14]. Such effects are difficult to model accurately, and will account for the reduced modeling and linearization performance. Depending on the power amplifier input drive level, the memory polynomial model achieved an adjacent channel power ratio reduction in the range 10 dB to 15 dB, which is comparable with results published in the literature [52, 72].

6.1 Further Topics for Future Work

In the following, some future topics related to the work carried out in thesis are provided.

Further Characterization of CGH4006P with Load Pull and Time Domain Measurements

* An obvious topic for further work is load pull measurements with automatic tuning of the loads, allowing for peak searching of various output properties. In this way it can be shown whether simulated results can be replicated or not. If similar results as in the simulations can be achieved, it would be interesting to make a full power amplifier design with input/output matching networks and see if the device can achieve similar performance [44] as other GaN HEMT power transistors from Cree Inc. .

* All load pull simulations and measurements in this work are performed with single tone signals. This does not give a complete overview of the transistors capabilities for practical applications, as in todays communication standards, modulated signals are commonly used. Hence, load pull measurements should be carried out with modulated signals as well, i.e. as in [90].

* A SWAP-X402 receiver, which allows for time domain waveform measurements, is available in the load pull measurement setup. Combining the capabilities of the SWAP-X402 receiver and the load pull measurement equipment may allow for many interesting methods for nonlinear transistor modeling, i.e. estimation of transistor output parasitics and intrinsic waveforms, as proposed in [91].

Further Development of Measurement Setup for Load Pull and Time Domain Measurements

For the load pull measurement setup depicted in Figure E.3, most operations must, as for now, be carried out in Focus Microwaves' Load Pull Explorer (LPExp). Although the software is developed for controlling all measurement equipment that is added to its equipment list, it still lacks much of the data processing capabilities that i.e. MATLAB provides. Since the majority of the measurement equipment at the lab already can be controlled by MATLAB through a GPIB interface, it would be advantageous to control the tuners and LPExp from MATLAB as well. Hence, algorithms for this purpose should be developed for MATLAB. The same thing applies for the SWAP-X402 receiver. Much of its functionalities are per today only available in SciLab, so these should be ported to MATLAB as well.[1]

Improvement of Measurement Setup for Large Signal Measurements

This topic is not directly research related, but still a suggestion for how the measurement setup can be improved in order to allow for extended digital bandwidth in measurements. The setup, as of today, uses a FSQ40 VSA for capturing raw I/Q data at the receiver end. The FSQ40 supports a maximum, equalized RF bandwidth of 28 MHz for raw I/Q data [92, pp. 4.275], which in some cases is a too small bandwidth, considering the fact that nonlinear distortion will spread the signal out in frequency. In [93] the authors present a method for increasing the digital bandwidth of a VSA, using the same type of equipment that is available at the microwave lab at NTNU. The technique is called frequency stitching, and is a further development of the method proposed in [94]. Using this technique, the authors achieved a digital bandwidth of 144 MHz, which should be sufficient for most microwave power amplifier measurements. Additional information of the technique can also be found in [95].

Further Development of Cosimulation Test Bench for ADS/MATLAB Simulations

The test bench as it is now allows for simulations of analog RF networks designed in ADS with modulated signals generated in MATLAB. With the library of DSP blocks in ADS for digital designs, the test bench can be extended to allow for simulations of i.e. online adaptive predistortion algorithms. In addition, noise sources can be added to make the environment more realistic and comparable with a real measurement environment. In this way time can be saved in the process of testing the linearization algorithms under mode realistic conditions prior to using them in an experimental setup.

[1] SciLab is a free software for numerical computations, see `http://www.scilab.org/` for more information.

References

[1] R. J. Martin, "Volterra System Identification and Kramer's Sampling Theorem," *IEEE Trans. Signal Process.*, vol. 47, no. 11, pp. 3152–3155, Nov. 1999.

[2] U. K. Mishra, P. Parikh, and Y.-F. Wu, "AlGaN/GaN HEMTs - An Overview of Device Operation and Applications," *Proc. IEEE*, vol. 90, pp. 1022–1031, Jun. 2002.

[3] D. Schmelzer and S. I. Long, "A GaN HEMT Class F Amplifier at 2 GHz With > 80 % PAE," in *Compound Semiconductor Integrated Circuit Symposium, 2006. CSIC 2006. IEEE*, Nov. 2006.

[4] U. K. Mishra, S. Likun, T. E. Kazior, and Y.-F. Wu, "GaN-Based RF Power Devices and Amplifiers," *Proc. IEEE*, vol. 96, no. 2, pp. 287–305, Feb. 2008.

[5] M. Ubostad, "Linearization and Efficiency Enhancement of RF Power Amplifiers based on Advanced Measurements and Transistor Modeling," Ph.D. dissertation, Norwegian University of Science and Technology, Dec. 2010.

[6] P. B. Kenington, *High Linearity RF Amplifier Design*, ser. Artech House Microwave Library. Artech House, 2000.

[7] E. Aschbacher, "Digital Pre-Distortion of Microwave Power Amplifiers," Ph.D. dissertation, The Vienna University of Technology, Sep. 2005.

[8] E. Aschbacher, M. Y. Cheong, P. Brunmayr, M. Rupp, and T. I. Laakso, "Prototype Implementation of Two Efficient Low-Complexity Digital Predistortion Algorithms," *EURASIP Journal on Advances in Signal Processing*, 2008.

[9] C. Eun and E. J. Powers, "A New Volterra Predistorter Based on the Indirect Learning Architecture," *IEEE Trans. Signal Process.*, vol. 45, no. 1, p. 223, Jan. 1997.

[10] F. H. Raab, "Class-f Power Amplifiers With Maximally Flat Waveforms," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 11, p. 2007, Nov. 1997.

[11] ——, "Maximum Efficiency and Output of Class-f Power Amplifiers," *IEEE Trans. Microw. Theory Tech.*, vol. 49, no. 6, p. 1162, Jun. 2001.

[12] Y. Y. Woo, Y. Yang, and B. Kim, "Analysis and Experiments for High-efficiency Class-F and Inverse Class-F Power Amplifiers," *IEEE Trans. Microw. Theory Tech.*, vol. 54, pp. 1969–1974, May 2008.

[13] S. C. Cripps, *Advanced Techniques in RF Power Amplifier Design*, 1st ed., ser. Artech House Microwave Library. Artech House, 2002.

[14] ——, *RF Power Amplifiers for Wireless Communications*, 2nd ed., ser. Artech House Microwave Library. Artech House, 2002.

[15] D. M. Pozar, *Microwave Engineering*, 3rd ed. Wiley, 2005.

[16] M. L. Edwards and J. H. Sinsky, "A New Criterion for Linear 2-Port Stability Using a Single Geometrically Derived Parameter," *IEEE Trans. Microw. Theory Tech.*, vol. 40, no. 12, pp. 2303–2311, Dec. 1992.

[17] F. H. Raab, P. Asbeck, S. Cripps, P. B. Kenington, Z. B. Popovic, N. Pothecary, J. F. Sevic, and N. O. Sokal, "Power Amplifiers and Transmitters for RF and Microwave," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no. 3, p. 814, mar 2002.

[18] G. Breed, "IMD Characteristics and Performance Issues in Wide Bandwidth Systems," *High Frequency Electronics*, vol. 4, no. 3, pp. 50–53, Mar. 2005.

[19] J. Vuolevi and T. Rahkonen, *Distortion in RF Power Amplifiers*, 1st ed., ser. Artech House Microwave Library. Artech House, 2003.

[20] M. Franco, A. Guida, A. Katz, and P. Herczfeld, "Minimization of Bias-Induced Memory Effects in UHF Radio Frequency High Power Amplifiers With Broadband Signals," in *Radio and Wireless Symposium, 2007 IEEE*, Long Beach, CA, Jan. 2007, pp. 369–372.

[21] J. D. Rhodes, "Output Universality in Maximum Efficiency Linear Power Amplifiers," *International Journal of Circuit Theory and Applications*, vol. 31, no. 4, pp. 385–405, Jul. 2003.

[22] N. V. Drozdovski and R. H. Caverly, "GaN-Based High Electron-Mobility Transistors for Microwave and RF Control Applications," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no. 1, Jan. 2002.

[23] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 1st ed. Society for Industrial and Applied Mathematics, 1996.

[24] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, 1st ed. Society for Industrial and Applied Mathematics, 1997.

[25] Åke Björck, *Numerical Methods for Least Squares Problems*, 1st ed. Society for Industrial and Applied Mathematics, 1996.

[26] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[27] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. Society for Industrial and Applied Mathematics, 1999.

[28] A. Ben-Isreal and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed., ser. CMS Books in Mathematics. Springer, 2003.

[29] W. K. Jenkins, C. W. Therrien, and X. Li, "Orthogonal Polynomial-based Nonlinear Adaptive Filters," in *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*, vol. 1, Aug. 2001.

[30] R. Raich, H. Qian, and G. T. Zhou, "Orthogonal Polynomials for Power Amplifier Modeling and Predistorter Design," *IEEE Trans. Veh. Technol.*, vol. 53, no. 5, pp. 1468–1479, Sep. 2004.

[31] X. Wu, J. Shi, and H. Chen, "On the Numerical Stability of RF Power Amplifier's Digital Predistortion," in *Communications, 2009. APCC 2009. 15th Asia-Pacific Conference on*, Oct. 2009, p. 430.

[32] G. Szegö, *Orthogonal Polynomials*, 4th ed. American Mathematical Society, Dec. 1939.

[33] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th ed. Dover Publications, 1972.

[34] D. Schreurs, M. O'Droma, A. A. Goacher, and M. Gadringer, *RF Power Amplifier Behavioral Modeling*, 1st ed., ser. The Cambridge RF and Microwave Engineering Series. Cambridge University Press, 2009.

[35] *CGH40006P 6 W RF Power GaN HEMT: Data Sheet*, Cree Inc., May 2010, revision 1.0, online.

[36] Y. F. Wu, A. Saxler, M. Moore, R. P. Smith, S. Sheppard, P. M. Chavarkar, T. Wisleder, U. K. Mishra, and P. Parikh, "30-W/mm Gan HEMTs by Field Plate Optimization," *IEEE Electron Device Lett.*, vol. 25, no. 3, p. 117, Mar. 2004.

[37] *Multi-Layer High-Q Capacitors*, Johanson Technology, Feb. 2011.

[38] I. J. Bahl, *Lumped Elements for RF and Microwave Circuits*, 1st ed., ser. Artech House Microwave Library. Artech House, 2003.

[39] *RO4000 Series High Frequency Circuit Materials*, Rogers Corporation, May 2010, data sheet.

[40] J. Verspecth, F. D. Groote, and J.-P. Teyssier, "Advanced Time Domain Load-pull Technique for Characterization of Microwave Power Transistors," in *ESA Microwave Technology and Techniques Workshop 2008*, May 2008.

[41] F. D. Groote, J. Verspecht, C. Tsironis, D. Barataud, and J. P. Teyssier, "An Improved Coupling Method for Time Domain Load-Pull Measurements," in *ARFTG Conference Digest, 2005. Spring 2005. 65th*, Jun. 2005, p. 4.

[42] *View it in Time Domain: SWAP-X402*, VTD, Dec. 2009, product presentation, online.

[43] Agilent, "Guide to Harmonic Balance Simulation in ADS," online, Sep. 2006.

[44] P. Saad, H. M. Nemati, M. Thorsell, K. Andersson, and C. Fager, "An Inverse Class-F GaN HEMT Power Amplifier With 78% PAE at 3.5 GHz," in *Microwave Conference, 2009. EuMC 2009. European*, Rome, Sep. 2009, p. 496.

[45] W. J. Rugh, *Nonlinear System Theory: The Volterra/Wiener Approach*, 1st ed. The John Hopkins University Press, 1981.

[46] N. Wiener, *Nonlinear Problems in Random Theory.* John Wiley & Sons, 1958.

[47] M. Väth, *Volterra and Integral Equations of Vector Functions*, 1st ed. Marcel Dekker, 2000.

[48] V. Volterra, *Theory of Functionals and of Integral and Integro-Differential Equations.* New York: Dover Publications, 1959, reprint.

[49] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems.* Krieger Publishing Company, 2006, reprint.

[50] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*, 1st ed., ser. Wiley Series in Telecommunication and Signal Processing. John Wiley & Sons, 2000.

[51] T. Ogonfunmi, *Adaptive Nonlinear Systems: The Volterra and Wiener Model Approaches*, 1st ed., ser. Springer Series on Signals and Communication Technology. Springer, 2007.

[52] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A Generalized Memory Polynomial Model for Digital Predistortion of RF Power Amplifiers," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[53] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "A Robust Digital Baseband Predistorter Constructed Using Memory Polynomials," *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 159–165, Jan. 2004.

[54] P. L. Gilabert, D. D. Silveira, G. Montoro, M. E. Gadringer, and E. Bertran, "Heuristic Algorithms for Power Amplifier Behavioral Modeling," *IEEE Microw. Wireless Compon. Lett.*, vol. 17, no. 10, Oct. 2007.

[55] N. Safari, "Linearization and Efficiency Enhancement of Power Amplifiers Using Digital Predistortion," Ph.D. dissertation, Norwegian University of Science and Technology, 2008.

[56] N. Wiener, "Response of a Nonlinear Device to Noise," Radiation Laboratory, M.I.T, Tech. Rep. 129, Apr. 1942.

[57] M. R. Fréchet, "Sur les Functionnels Continues," *Ann. Sci. Ecole Normal Sup.*, vol. 27, pp. 193–219, 1910.

[58] E. Bishop, "A Generalization of the Stone-Weierstrass Theorem," *Pacific Journal of Mathematics*, vol. 11, pp. 777–783, 1961.

[59] V. J. Mathews, "Adaptive Polynomial Filters," *IEEE Signal Process. Mag.*, vol. 8, no. 3, Jul. 1991.

[60] W. A. Frank, "Sampling Requirements for Volterra System Identification," *IEEE Signal Process. Lett.*, vol. 3, no. 9, pp. 266–268, Sep. 1996.

[61] J. Tsimbinos and K. V. Lever, "Input Nyquist Sampling Suffices to Identify and Compensate Nonlinear Systems," *IEEE Trans. Signal Process.*, vol. 46, no. 10, pp. 2833–2837, Oct. 1998.

[62] A. S. Tehrani, H. Cao, S. Afsardoost, T. Eriksson, M. Isaksson, and C. Fager, "A Comparative Analysis of the Complexity/Accuracy Tradeoff in Power Amplifier Behavioral Models," *IEEE Trans. Microw. Theory Tech.*, vol. 58, no. 6, p. 1510, Jun. 2010.

[63] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice Hall, Jan. 1999.

[64] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, ser. Prentice Hall Signal Processing Series. Prentice Hall, 1993, vol. 1.

[65] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Maths*, vol. 11, no. 2, pp. 431–441, Jun. 1963.

[66] P. Gilabert, G. Montoro, and E. Bertran, "On the Wiener and Hammerstein Models for Power Amplifier Predistortion," in *Microwave Conference Proceedings, 2005. APMC 2005. Asia-Pacific Conference Proceedings*, vol. 2, Dec. 2005, p. 4.

[67] L. Ding, R. Raich, and G. T. Zhou, "A Hammerstein Predistortion Linearization Design Based on the Indirect Learning Architecture," in *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, vol. 3, 2002, pp. 2689–2692.

[68] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "Memory Polynomial Predistorter Based on the Indirect Learning Architecture," in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, Nov. 2002, pp. 967–971.

[69] F. M. Ghannouchi, F. Taringou, and O. Hammi, "A Dual Branch Hammerstein-Wiener Architecture for Behavior Modeling of Wideband RF Transmitters," in *Microwave Symposium Digest (MTT), 2010 IEEE MTT-S International*, May 2010, p. 1692.

[70] L. Ding and G. T. Zhou, "Effects of Even-Order Nonlinear Terms on Predistortion Linearization," in *Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop. Proceedings of 2002 IEEE 10th*, Oct. 2002, pp. 1–6.

[71] ——, "Effects of Even-Order Nonlinear Terms on Power Amplifier Modeling and Predistortion Linearization," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 1, pp. 156–162, Jan. 2004.

[72] C. Yu, Y. Liu, and S. Li, "Triangular Memory Polynomial Predistorter," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, Beijing, Sep. 2009, pp. 1–4.

[73] F. W. J. Olver, D. W. Lozier, R. F. Biosvert, and C. W. Clark, *NIST Handbook of Mathematical Functions*, 1st ed. Cambridge University Press, Jul. 2010.

[74] C. Runge, "Über Empirische Funktionen und die Interpolation Zwischen Äquidistanten Ordinaten," *Zeitschrift für Mathematik und Physik*, vol. 46, pp. 223–243, 1901, in German.

[75] C. de Boor, *A Practical Guide to Splines*, 1st ed., ser. Applied Mathematical Sciences, Vol. 27. Springer Verlag, 2001, revised edition.

[76] A. Hirotugu, "Fitting Autoregressive Mmodels for Prediction," *Annals of the Institute of Statistical Mathematics*, vol. 21, no. 1, pp. 243–247, 1969.

[77] J. Rissanen, "Modeling by Shortest Data Description," *Automatica*, vol. 14, no. 5, pp. 465 – 471, 1978.

[78] M. Isaksson, D. Wisell, and D. Ronnow, "A Comparative Analysis of Behavioral Models for RF Power Amplifiers," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 1, pp. 348–359, Jan. 2006.

[79] D. Mitrevski, "The Design of a Class AB Power Amplifier with Digital Predistortion," Mar. 2010, Unpublished Specialization-Project Thesis.

[80] E. H. Hennie, "Analyse og Konstruksjon av en Harmonisk Tunet 6 W GaN Effektforsterker," Master's thesis, Norwegian University of Science and Technology, Mar. 2011, in Norwegian.

[81] S. S. Jeng and J. M. Chen, "Efficient PAPR Reduction in OFDM Systems Based on a Companding Technique With Trapezium Distribution," *IEEE Trans. Broadcast.*, no. 99, p. 1, Feb. 2011.

[82] H. Akaike, "A New Look at the Statistical Model Identification," *IEEE Trans. Autom. Control*, vol. 19, no. 6, pp. 716–723, Dec. 1974.

[83] H. Cao, A. S. Tehrani, H. M. Nemati, C. Fager, T. Eriksson, and H. Zirath, "Time Alignment in a Dynamic Load Modulation Transmitter Architecture," in *Microwave Conference, 2009. EuMC 2009. European*, Oct. 2009, p. 1211.

[84] R. Raich, H. Qian, and G. T. Zhou, "Digital Baseband Predistortion of Nonlinear Power Amplifiers Using Orthogonal Polynomials," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 6, Apr. 2003, pp. 689–92.

[85] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice Hall, 2002.

[86] M. Schetzen, "Theory of Pth-order Inverses of Nonlinear Systems," *IEEE Trans. Circuits Syst.*, vol. 23, no. 5, pp. 285–291, May 1976.

[87] K.-F. To, P. C. Ching, and K. M. Wong, "Compensation of Amplifier Nonlinearities on Wavelet Packet Division Multiplexing," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 4, May 2001, p. 2669.

[88] E. Abd-Elrady and L. Gan, "Direct and Indirect Learning Methods for Adaptive Predistortion of IIR Hammerstein Systems," *E&I: Elektrotechnik und Informationstechnik*, vol. 125, no. 4, pp. 126 – 131, Apr. 2008.

[89] L. Gan, "Adaptive Digital Predistortion of Nonlinear Systems," Ph.D. dissertation, Graz University of Technology, Jun. 2009.

[90] M. Ubostad and M. Olavsbråten, "Analysis of an RF Power Amplifier for Envelope Tracking Based on Load-pull Data," in *Microwave Measurement Symposium, 2009 74th ARFTG*, Broomfield, CO, Dec. 2009, pp. 1–4.

[91] M. Olavsbråten, M. Ubostad, and E. B. Mogstad, "Transistor Output Parasitics and Intrinsic Waveforms Estimation by Time-Domain Waveforms Measurements and Load-Pull," in *IEEE Power Amplifier Symposium*, Sep. 2010.

[92] *FSQ40 Signal Analyzer: Operating Manual*, Rohde & Schwarz, Aug. 2010, online.

[93] D. Wisell, D. Ronnow, and P. Handel, "A Technique to Extend the Bandwidth of an RF Power Amplifier Test Bed," *IEEE Trans. Instrum. Meas.*, vol. 56, pp. 1488–1494, Aug. 2007.

[94] S. R. Velazquez, T. Q. Nguyen, and S. R. Broadstone, "Design of Hybrid Filter Banks for Analog/digital Conversion," *IEEE Trans. Signal Process.*, vol. 46, no. 4, p. 956, Apr. 1998.

[95] D. Wisell, "Measurement Techniques for Characterization of Power Amplifiers," Ph.D. dissertation, KTH School of Electrical Engineering, Dec. 2007.

[96] D. Hummels and R. Gitchell, "Equivalent Low-Pass Representations for Bandpass Volterra Systems," *IEEE Trans. Commun.*, vol. 28, pp. 140–142, Jan. 1980.

[97] R. Raich and G. T. Zhou, "On the Modeling of Memory Nonlinear Effects of Power Amplifiers for Communication Applications," in *Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop. Proceedings of 2002 IEEE 10th*, Oct. 2002, pp. 7–10.

[98] Agilent, "Applying Error Correction to Network Analyzer Measurements," online, Mar. 2002, Application Note 1287-3.

[99] ——, "Applying the 8510 TRL Calibration for Non-Coaxial Measurements," online, May 2001, Product Note 8510-8A.

[100] K. C. Gupta, R. Garg, I. Bahl, and P. Bhartia, *Microstrip Lines and Slotlines*, 2nd ed., ser. Artech House Antennas and Propagation Library. Artech House, 1996.

[101] *Agilent PNA Microwave Network Analyzers: Data Sheet*, Agilent, Oct. 2008, online.

[102] *SMU200A Vector Signal Generator: Operating Manual*, Rohde & Schwarz, Nov. 2010, online.

[103] E. B. Mogstad, "Analyse og Konstruksjon av en Klasse B Effektforsterker i GaN Teknologi," Master's thesis, Norwegian University of Science and Technology, Jul. 2010, in Norwegian.

Derivation of Input/Output Relation for Complex Baseband Volterra Series

A

The input/output relation for the truncated, real–valued bandpass Volterra series is given by

$$\tilde{y}(t) = \sum_{p=1}^{2P+1} \int \tilde{h}_p\left(\boldsymbol{\tau}_p\right) \prod_{i=1}^{p} \tilde{x}\left(t-\tau_i\right) \, \mathrm{d}\boldsymbol{\tau}_p \,. \tag{A.1}$$

where $\tilde{x}(t)$ is the input signal, $\tilde{y}(t)$ is the output signal, $\boldsymbol{\tau}_p = [\tau_1 \, \ldots \, \tau_p]^{\mathsf{T}}$ is the argument of p–dimensional kernel $\tilde{h}_p\left(\tau_1, \ldots, \tau_p\right)$, and $\mathrm{d}\boldsymbol{\tau}_p = \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_p$ is used for notational simplicity. If we in (A.1) express the real–valued bandpass input signal centered at f_0 as

$$\tilde{x}(t) = \frac{1}{2}\Big(x(t)\exp\left(\mathrm{j}2\pi f_0 t\right) + x^*(t)\exp\left(-\mathrm{j}2\pi f_0 t\right)\Big) \,, \tag{A.2}$$

with $x(t)$ as the equivalent complex baseband signal, we get

$$\begin{aligned}\tilde{y}(t) = \sum_{p=1}^{2P+1} \frac{1}{2^p} \int \tilde{h}_p\left(\boldsymbol{\tau}_p\right) \prod_{i=1}^{p} \Big[& x\left(t-\tau_i\right)\exp\left(\mathrm{j}2\pi f_0\left(t-\tau_i\right)\right) \\ & + \; x^*\left(t-\tau_i\right)\exp\left(-\mathrm{j}2\pi f_0\left(t-\tau_i\right)\right)\Big] \, \mathrm{d}\boldsymbol{\tau}_p \,.\end{aligned} \tag{A.3}$$

In the following we only consider symmetric kernels, implying that the argument of the kernel $\tilde{h}_p\left(t_1, \ldots, t_p\right)$ can be permuted in any order without affecting the output signal [7]. This is because every asymmetric kernel can, without loss of generality, be converted into a symmetric kernel [49, pp. 80–81]. We also assume that the input signal is band limited to $[-B, B]$, and that the carrier frequency is significantly greater than the maximum envelope frequency of the input signal, or $f_0 \gg B$. This assumption will, when modeling microwave power amplifiers, often hold [34, pp. 31]. Since we are only interested in signals centered at f_0, the only combination of signal products

$$\prod_{i=1}^{p} \frac{1}{2} \left(x\left(t-\tau_i\right) \exp\left(\mathrm{j}2\pi f_0 \left(t-\tau_i\right)\right) + x^*\left(t-\tau_i\right) \exp\left(-\mathrm{j}2\pi f_0 \left(t-\tau_i\right)\right)\right) \tag{A.4}$$

for an arbitrary p that satisfies this condition is the combination which consist of $k+1$ regular terms and k complex–conjugated terms [96]

$$\underbrace{x\left(t-\tau_1\right) \cdots x\left(t-\tau_{k+1}\right)}_{k+1} \underbrace{x^*\left(t-\tau_{k+2}\right) \cdots x^*\left(t-\tau_{2k+1}\right)}_{k} \qquad \text{for } 2k+1=p \in \mathbb{N}\,. \tag{A.5}$$

This outcome is seen from the argument of the resulting exponential function, which determines the signal frequency. For the signal to be centered at f_0, the argument of the exponential function must be on the form $\mathrm{j}2\pi f_0 \left(t+g\left(\tau\right)\right)$, which can only be achieved if p is odd.[1] If p is even, the argument of the exponential function consists of frequency components at either dc or higher even-ordered harmonics that are outside the region of interest, and are assumed to be suppressed perfectly by the output zonal filter $\mathcal{I}$, as depicted in Figure A.1.

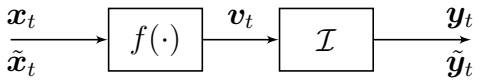


Figure A.1: Zonal filtering in complex baseband representation of nonlinear pass band system. Signals denoted with tilde are complex baseband signals.

From this we can express the equivalent baseband output for the $(2p+1)$th component as

$$y_{2p+1}(t) = \int h_{2p+1}\left(\boldsymbol{\tau}_{2p+1}\right) \prod_{i=1}^{p+1} x\left(t-\tau_i\right) \prod_{i=p+2}^{2p+1} x^*\left(t-\tau_i\right) \, \mathrm{d}\boldsymbol{\tau}_{2p+1}\,, \tag{A.6}$$

where the equivalent baseband kernel is given by

$$h_{2p+1}\left(\boldsymbol{\tau}_{2p+1}\right) = \left(\frac{1}{2}\right)^{2p} \binom{2p+1}{p} \tilde{h}_{2p+1}\left(\boldsymbol{\tau}_{2p+1}\right) \exp\left(-\mathrm{j}2\pi f_0 \left(\sum_{i=1}^{p+1} \tau_i - \sum_{i=p+2}^{2p+1} \tau_i\right)\right)\,. \tag{A.7}$$

The equivalent complex baseband output for the Volterra series of order $2P-1$ centered at f_0 will then be given as [97]

$$y(t) = \sum_{p=0}^{P-1} \int h_{2p+1}\left(\boldsymbol{\tau}_{2p+1}\right) \prod_{i=1}^{p+1} x\left(t-\tau_i\right) \prod_{i=p+2}^{2p+1} x^*\left(t-\tau_i\right) \, \mathrm{d}\boldsymbol{\tau}_{2p+1}\,. \tag{A.8}$$

[1] In this context $g(\tau)$ represent some arbitrary function of the delay τ.

ADS/MATLAB Cosimulation Test Bench Schematic

B

To validate the algorithmic performance of the different behavioral models and linearization algorithms, a cosimulation test bench has been designed in ADS. The test bench allows for passing information between MATLAB and ADS with ease.

It provides advantages such as the possibility to validate the performance of an model/algorithm in an environment with no measurement noise, or any other limitations in the measurement equipment, such as a finite bit–depth in the ADCs. The signal flow chart for the test bench is depicted in Figure B.1, with the name of the different blocks in ADS given outside the dashed lines.

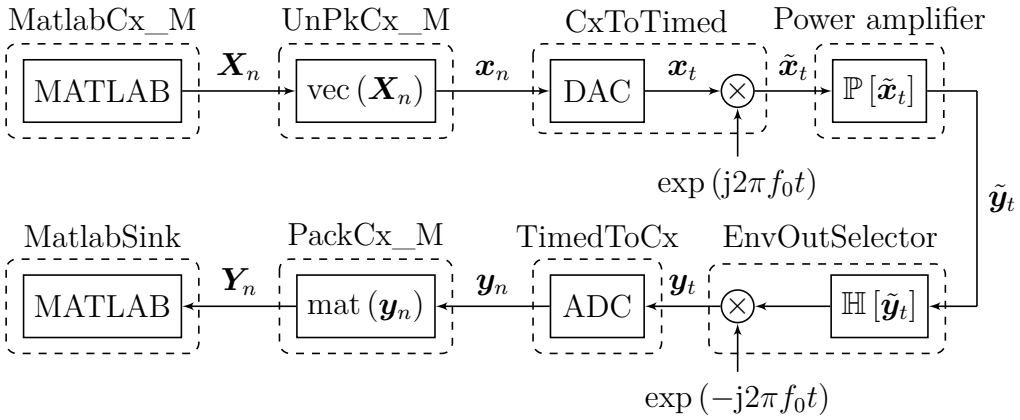


Figure B.1: Signal flow chart for ADS/MATLAB cosimulation test bench.

The MATLAB source block in ADS returns the output as a matrix $\boldsymbol{X}_n$, even though the routine used to generate the signal in MATLAB returns the output as a vector. As the DAC block only accepts inputs in vector format, the complex baseband discrete time input signal $\boldsymbol{X}_n \in \mathbb{C}^{M\times N}$ is vectorized to $\boldsymbol{x}_n \in \mathbb{C}^{MN} = \mathrm{vec}\,(\boldsymbol{X}_n)$.[1]

In ADS, the DAC also performs an upconversion to passband if provided a center frequency f_0, resulting in a continuous time passband signal $\tilde{\boldsymbol{x}}_t$, which is the analog

[1]Given $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, $\mathrm{vec}\,(\boldsymbol{A}) = [a_{1,1}\,\ldots\,a_{m,1}\,a_{1,2}\,\ldots\,a_{m,2}\,\ldots\,a_{1,n}\,\ldots\,a_{m,n}]^{\mathsf{T}} \in \mathbb{C}^{mn}$.

RF network input signal. The output of the power amplifier $\tilde{\boldsymbol{y}}_t = \mathbb{P}[\tilde{\boldsymbol{x}}_t]$ is then passed through a zonal filter $\mathbb{H}[\cdot]$, which can be considered as an ideal bandpass filter that selects the spectral zone of interest, and suppress all other frequencies perfectly.[2] In addition, the block that performs this filtering in ADS also performs a downconversion to complex baseband. The complex baseband output signal $\boldsymbol{y}_t$ is then digitized in the ADC, and transformed back to the initial matrix format in ADS. Thus, the discrete time signal sent back to MATLAB is in ADS recognized as $\boldsymbol{Y}_n \in \mathbb{C}^{M\times N}$, whereas in MATLAB it is $\boldsymbol{y}_n \in \mathbb{C}^M$.

[2]The spectral zone specified must be a multiple of the center frequency f_0.

TRL Calibration of Non-Coaxial Two-Port Networks

C

In a test setup for measuring microwave networks, imperfections that perturb the results are inevitable. These are often related to the test equipment, or the setup itself, and give rise to different types of errors. Some of these errors are repeatable and predictable, making them possible to remove mathematically, whereas other errors can only be reduced to a certain level due to an unpredictable/random behavior. See [98] for a detailed overview of the type of errors present, and how they are classified.

For an uncalibrated test setup, the scenario will be like the one depicted in Figure C.1. The primary measurement plane will differ from the reference plane of the DUT, leading to measurements that suffer from additional losses and phase delays [15, pp. 193]. The origin of these effects are, amongst others, increased signal path in a dispersive media, and the transition from a full transverse electromagnetic (TEM) wave in the coaxial cable, to a quasi TEM wave on the microstrip. In the illustration, all such perturbing effects are assumed to be embedded in the two–port error boxes placed between the actual measurement plane, and the desired reference plane of the DUT.[1]

In the illustration, the ABCD parameters of the DUT and the measured parameters are denoted by $\left(\begin{smallmatrix} A' & B' \\ C' & D' \end{smallmatrix}\right)$ and $(\begin{smallmatrix} A^m & B^m \\ C^m & D^m \end{smallmatrix})$, respectively. The purpose of a calibration procedure is to characterize the error–boxes prior to the DUT measurements, allowing us to derive the actual ABCD/S parameters of the DUT from the measured data. There are several calibration methods available, such as one–port calibration, short–open–load–thru (SOLT) two–port calibration, and thru–reflect–line (TRL) calibration.

The first two mentioned depend on three or more known standards/loads (i.e. open, short and matched load) to characterize the error boxes, whereas TRL only use three connections for the same purpose. All three types mentioned have their pros and cons (see [15, pp. 193]), but in the following we only focus on TRL calibration, the method employed in the work of this thesis. By using the three connections thru, reflect, and line, the error boxes (or the full 12–term error model [98]) will be fully characterized, allowing

[1] Each error box is characterized by its respective ABCD/S parameters.

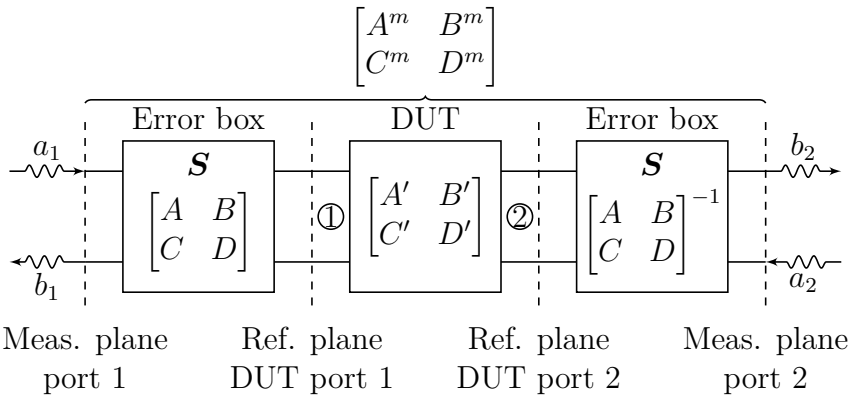


Figure C.1: Block diagram for measurement of a two–port network.

us to derive the actual ABCD/S parameters of the DUT. In short, the connections should satisfy (see [99, pp. 15] for detailed overview):

Thru: made by connecting the two ports directly together at the desired reference plane, representing a zero–length thru. Thus, $S_{21} = S_{12} = 1$ at 0 degrees and $S_{11} = S_{22} = 0$.

Reflect: requires a load with large reflection coefficient Γ_{L} (ideally $\Gamma_{\mathrm{L}} = 1$), which will be determined during calibration. This can typically be an open or a short for microstrip measurements.[2] To achieve good results, it is essential that the reflection coefficient is equal on both ports. In addition, $\arg(\Gamma_{\mathrm{L}})$ must be known within $\pm\lambda/4$.

Line: made by inserting a line with characteristic impedance Z_{c} equal to the system impedance Z_0. The difference in electrical length between the thru and the line must be in the range of 20 to 160 degrees. Hence, with a zero–length thru, the electrical length of the line must be in the specified range. It is also essential that $Z_{\mathrm{c}} = Z_0$ to achieve good results.

In order to determine the ABCD/S parameters of the error boxes, a set of equations is required. These equations can be found by applying the different connections at the reference plane of the DUT, and measure the S parameters for each respective connection at the measurement planes (with reference to Figure C.1). When applying each connection, the block schematic of the measurement, with its corresponding signal flow graph, can be represented as depicted in Figure C.2 – C.4, for the thru, reflect, and line connection, respectively. The derivation of the error correcting equations, based on the signal flow graphs illustrated below, is given in [15, pp. 194–196], so it is not repeated here. However, with reference to Figure C.1, the ABCD parameters of the DUT can easily be determined if the error boxes are characterized. If we, for notational

[2] Using a short will introduce inductive effects, whereas an open will radiate energy [99, pp. 5].

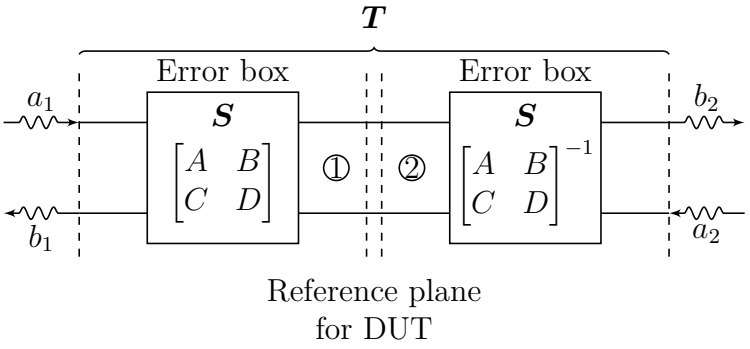
simplicity, denote the ABCD parameters of the error boxes, the DUT, and the measured data by $\boldsymbol{E}$, $\boldsymbol{D}$, and $\boldsymbol{M}$, respectively, the measured data can be expressed as

$$\boldsymbol{M} = \boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^{-1}\,. \tag{C.1}$$

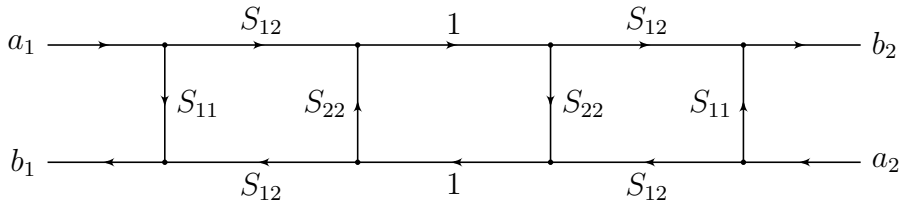
With basic matrix operations we can isolate the ABCD parameters of the DUT and get the relationship

$$\boldsymbol{D} = \boldsymbol{E}^{-1}\boldsymbol{M}\boldsymbol{E}\,. \tag{C.2}$$

In the illustrations below, the measured S parameters of the different connections are denoted by $\boldsymbol{T}$, $\boldsymbol{R}$, and $\boldsymbol{L}$, for the thru, reflect, and the line connection, respectively. The incident and reflected waves are defined as before. In Figure C.4, γ is the complex propagation constant defined as $\gamma = \alpha + \mathrm{j}\beta$, where α is the attenuation constant in Np/m, and β the phase constant in rad/m.

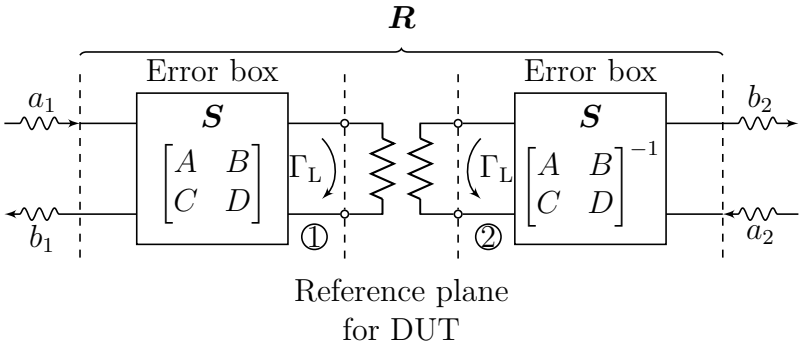


(a) Block diagram.

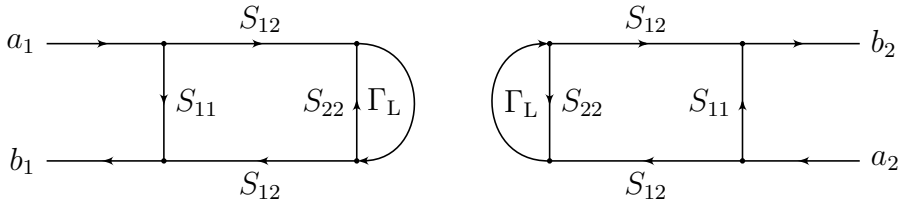


(b) Signal flow graph.

Figure C.2: Block diagram and signal flow graph for thru connection.

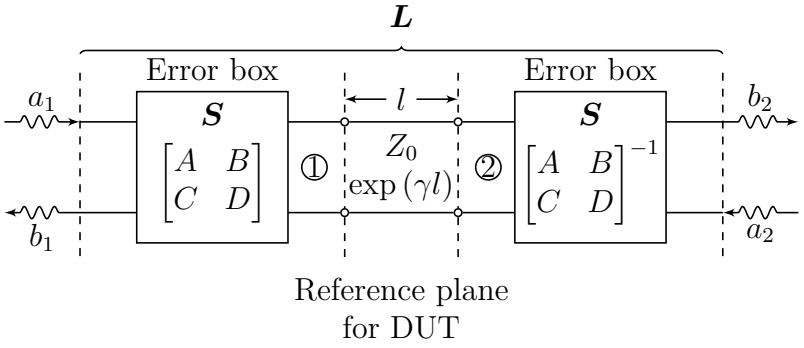


(a) Block diagram.

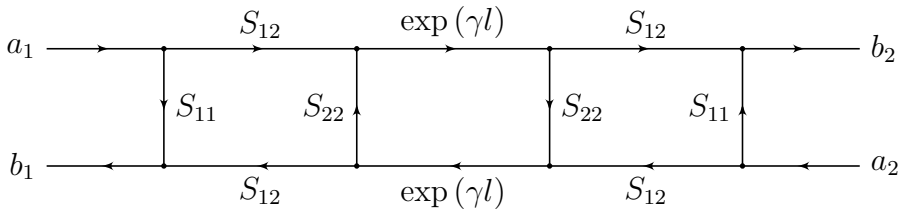


(b) Signal flow graph.

Figure C.3: Block diagram and signal flow graph for reflect connection.



(a) Block diagram.



(b) Signal flow graph.

Figure C.4: Block diagram and signal flow graph for line connection.

C.1 Design and Validation of TRL Calibration Kits

In this section the design and validation procedure of the TRL calibration kits designed are presented respectively.

C.1.1 Design of TRL Calibration Kits

The design of a TRL calibration kit is a straight forward procedure, where only the criteria mentioned in the previous section must be satisfied. In Figure C.5 an illustration of a typical microstrip TRL calibration kit is presented, where m_l is the length between the SMA connector and the desired reference plane, and n_l is the length of the line connection. This length specifies the bandwidth of the calibration kit; hence, if a bandwidth greater than the 8:1 ratio a single line provides, multiple lines of different length, as in multi–line TRL, should be used [99, pp. 16].

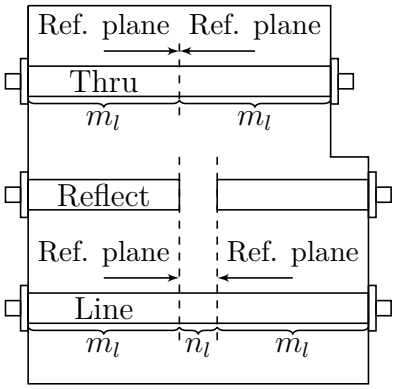


Figure C.5: Illustration of the designed TRL calibration kits.

For this work, two TRL calibration kits were designed. The first was designed based on the assumption that $\epsilon_r = 3.55$, implying a line width of 1.09 mm. Since it was unclear that if this value of ϵ_r was the correct one, the S parameters of the kit were measured and compared to the ones of the ideal TRL kit simulated in ADS.[3] The measured parameters indicated presence of undesired reflections, showing a reflection coefficient greater than expected. It was assumed that this was caused by an incorrect line impedance, together with parasitic effects in the coaxial–to–microstrip transition in the SMA connectors (from full TEM in the coaxial cable to quasi–TEM on the microstrip). In order to replicate the perturbing effects, an equivalent circuit model as the one illustrated in Figure C.6 was used in ADS [100, pp. 33–37]. The illustration includes a parasitic capacitance to ground, and a series parasitic inductance, together with resistances for each component to account for losses. Using this model, the values for the parasitic effects presented in Table C.1 were estimated by a simple optimization/tuning procedure in ADS, and

[3]The substrate data sheet recommends $\epsilon_r = 3.55$ to be used for circuit simulations, and $\epsilon_r = 3.38$ for other purposes [39].

thereafter used to estimate a value for ϵ_r that resulted in an improved match between simulations and measurements. The new value of ϵ_r was found to be around 3.30, slightly less than one of the values specified in the substrate data sheet. With this value for ϵ_r, the difference in width for a 50 Ω line was approximately 19.53 µm.

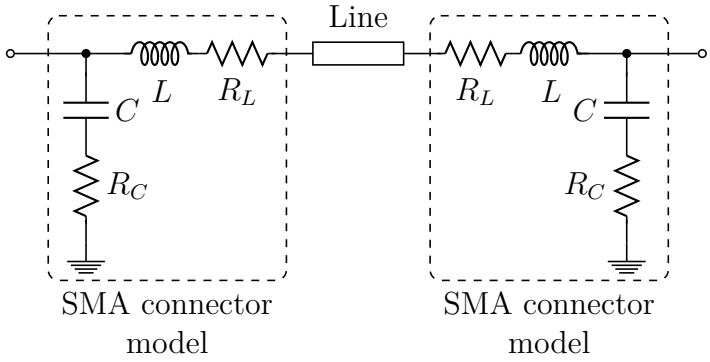


Figure C.6: Equivalent circuit model for a coaxial–to–microstrip transition.

Table C.1: Estimated values for parasitics in coaxial–to–microstrip connection.

Component	C [pF]	R_C [Ω]	L [nH]	R_L [Ω]
Value	0.14	3.70	0.43	0.56

C.1.2 Validation of TRL Calibration Kits

With the more exact value for ϵ_r at hand, a new TRL kit with equal line lengths as the first one, but with increased line width, was produced and used for further calibration of other equipment. However, before calibrating the equipment, both calibration kits were validated using WinCal, a software from Cascade Microtech® that can be used for this purpose.[4] In order for WinCal to provide a good validation and calculate the correct error terms, some initial parameters for i.e. the line length and phase velocity must be provided. To provide a good initial guess for the phase velocity, an approximate value for the relative dielectric constant ϵ_{eff} is needed. This can either be calculated using an approximate formula [38, pp. 430], or i.e. with ADS' LineCalc tool. Using LineCalc, ϵ_{eff} was estimated to be in the region of 2.55 to 2.75 for a set of frequencies, for both TRL kits. Hence, $\epsilon_{eff} = 2.65$ and $v_p = 184.15$ µm/ps was provided to WinCal as initial guesses for these values. In addition to get accurate error terms for the TRL kit, it was desired to estimate the line impedance and the effective dielectric constant. However, for this

[4]The software is originally intended for on–wafer measurements with probes, however, using it for a discrete microstrip structure is also possible.

purpose WinCal required a guess for the capacitance per unit length of the microstrip line. This was found by combining

$$v_{\mathrm{p}} = \frac{c_0}{\sqrt{\epsilon_{\mathrm{eff}}}} = \frac{1}{\sqrt{LC}} \quad [\mathrm{m/s}] \tag{C.3}$$

and

$$Z_0 = \sqrt{\frac{L}{C}} \quad [\Omega]\,, \tag{C.4}$$

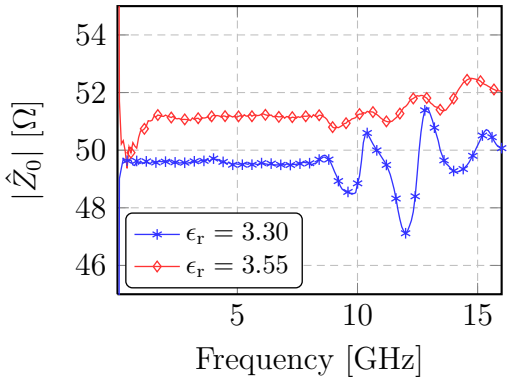
where c_0 is the speed of light in vacuum, C and L the capacitance and inductance per unit length of the line, respectively.[5] Combining these equations results in

$$C = \frac{1}{Z_0 v_{\mathrm{p}}} \quad [\mathrm{F/m}]\,, \tag{C.5}$$

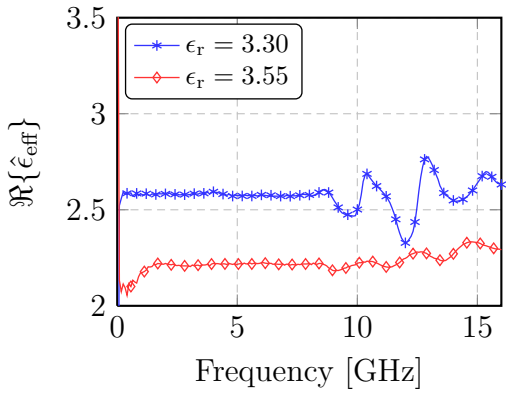
where Z_0 now is the assumed line impedance. Using $Z_0 = 50\,\Omega$ and $v_{\mathrm{p}} = 184.15\,\mu\mathrm{m/ps}$ in (C.5) yields $C \approx 108.60\,\mathrm{pF/m}$, which was provided to WinCal when validating the kit with $\epsilon_{\mathrm{r}} = 3.30$. For the kit with $\epsilon_{\mathrm{r}} = 3.55$, the value of C was decreased to $97\,\mathrm{pF/m}$, as reduced line width implies less capacitance per unit length, and increased phase velocity. After measuring the three different standards in the TRL calibration, WinCal estimated Z_0 and ϵ_{eff}, as shown in Figure C.7a – C.7c. In Figure C.7a we see that the line impedance for both kits is close to the desired $50\,\Omega$, however, the kit based on $\epsilon_{\mathrm{r}} = 3.30$ is closer, showing a deviation of approximately $0.50\,\Omega$ over a wide frequency band.

We also see that $\hat{\epsilon}_{\mathrm{eff}}$ for the kit designed with $\epsilon_{\mathrm{r}} = 3.30$ is closer to the region already estimated by LineCalc in ADS. Figure C.8a – C.8d shows the S parameters for the reflect connection of both calibration kits, measured with correction on in the VNA. It is evident that the kit designed with $\epsilon_{\mathrm{r}} = 3.30$ is the most accurate, agreeing most with the criteria for a TRL calibration kit presented in the previous section. In all figures one can also see a strange behavior from about 10 GHz and above. This is believed to be caused by the parasitic resonances in the SMA connector, and the transition to microstrip.

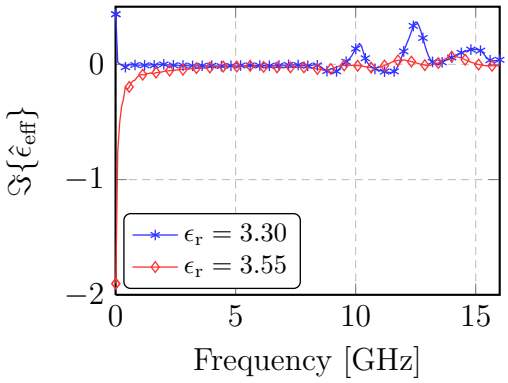
[5]The above equations assume a lossless line; however, WinCal only required an initial guess, and for this purpose it was assumed that the error made would not affect the end result significantly.



(a) Estimated characteristic impedance of microstrip lines.

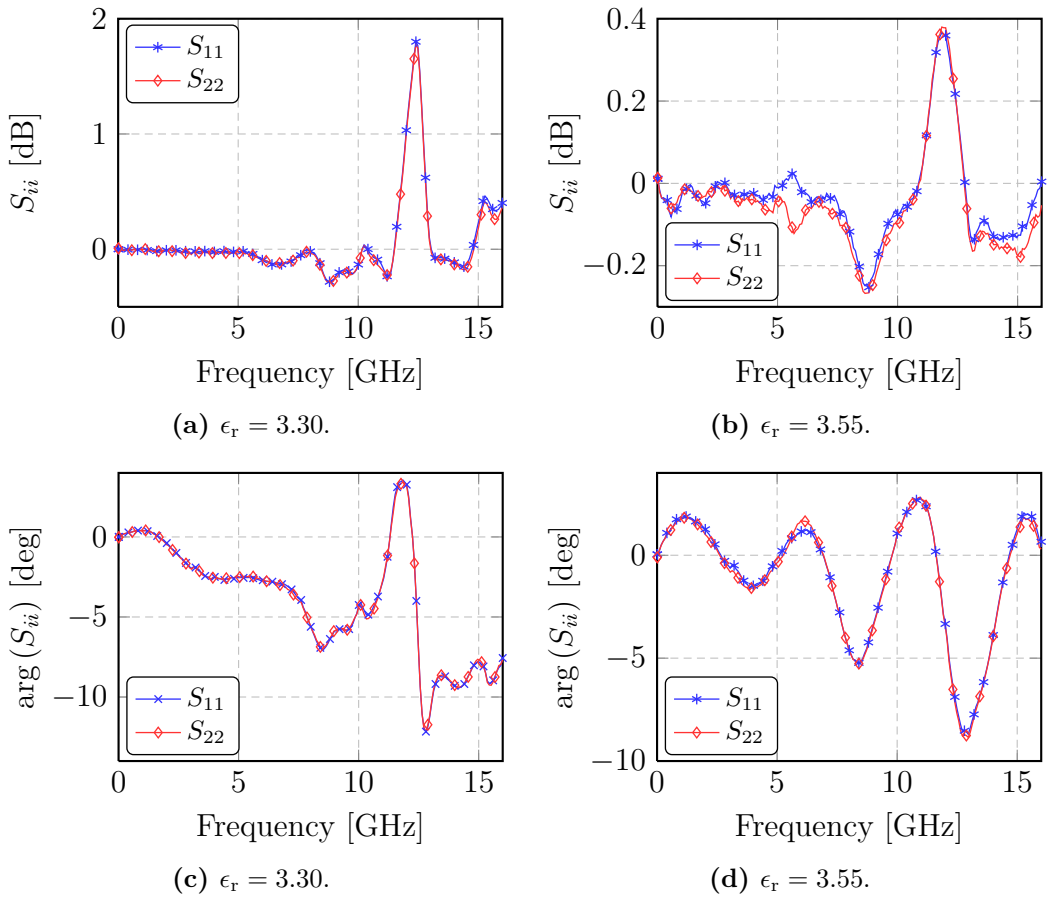


(b) Real part of $\hat{\epsilon}_{\text{eff}}$.



(c) Imaginary part of $\hat{\epsilon}_{\text{eff}}$.

Figure C.7: Estimated line parameters for both TRL calibration kits.



(a) $\epsilon_r = 3.30$. **(b)** $\epsilon_r = 3.55$.

(c) $\epsilon_r = 3.30$. **(d)** $\epsilon_r = 3.55$.

Figure C.8: Reflection coefficients for TRL reflect connection of both produced TRL kits.

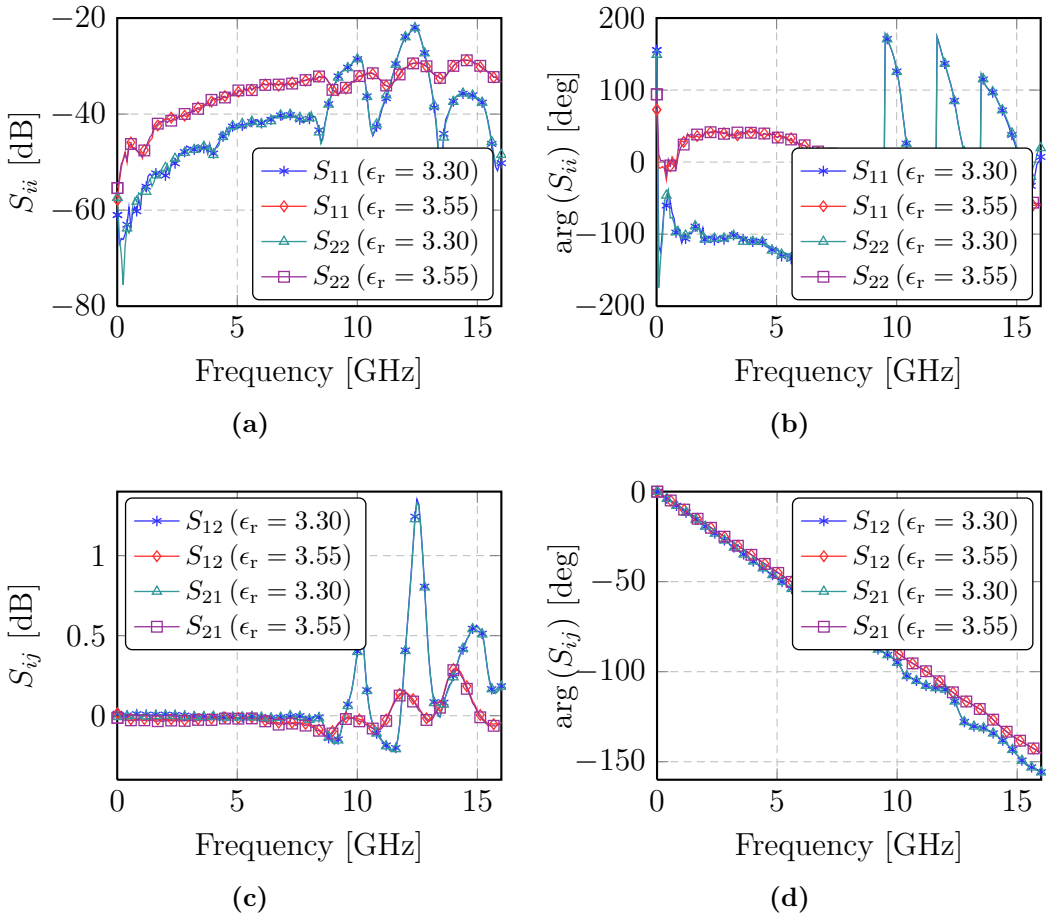


Figure C.9: Reflection and transmission coefficients for TRL line connection of both produced TRL kits.

Simulated Class AB Power Amplifier Design

D

This appendix presents the main properties of a simulated 6 W class AB power amplifier used in the ADS/MATLAB cosimulation setup for testing modeling and linearization. The design process for this amplifier design is similar to the one described in [79] and is thus left out in the following. In order to keep things short, only the most essential data from the small- and large signal simulations is presented as figures. The transistor used in this design is the same as in the class F/inverse class F design discussed earlier in the thesis, however, since this design was made at the beginning of this study, the center frequency differs from the one used in the other designs ($f_0 = 3.40\,\text{GHz}$ vs. $f_0 = 2\,\text{GHz}$). Biasing conditions are otherwise the same ($V_\text{D} = 28\,\text{V}$ and $I_\text{D} = 11\,\text{mA}$). Figure D.1a – D.1c shows the most important small signal properties of the PA design, whereas Figure D.2a – D.2c shows the most important large signal properties of the design. The active device was first stabilized with a parallel high–pass RC filter in a series connection at gate, together with a resistor in the biasing network at gate for introducing loss at the lowest frequencies.

Since maximum gain was desired, the components in the stabilization network were found by running a gradient based optimization in ADS, where the optimization criteria were set to a minimum MAG value of 13 dB and unconditional stability ($\mu_{\text{source/load}} \geq 1$). After stabilizing the circuit, the input and output matching networks were designed based on a large signal S parameter simulation in ADS. Again, an optimization was run with optimization goals specifying the minimum PAE, output power, and input return loss desired throughout the band of interest ($f_\text{BW} = [3.35, 3.45]\,\text{GHz}$) at the input power level for which the output was expected to be in 1 dB compression.[1] The optimization goals for the large signal S parameter optimization are specified in Table D.1.

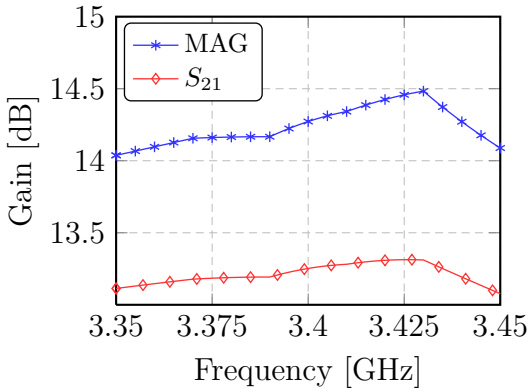
From Figure D.2a – D.2c we see that a power gain of 12 dB is achieved, together with a PAE of slightly less that 65 % at 1 dB compression. The output power at this input power level is approximately 37.20 dBm, also less than the desired value. Although

[1] The expected input power level for the 1 dB compression point was derived based on the small signal MAG value of the PA with only a stabilization circuit attached. With a MAG of 14 dB, it was assumed that a power gain of 13 dB could be achieved. Hence, with a desired output power of 6 W, at least 12 dB power gain at $P_\text{in} \approx 26\,\text{dBm}$ was required.

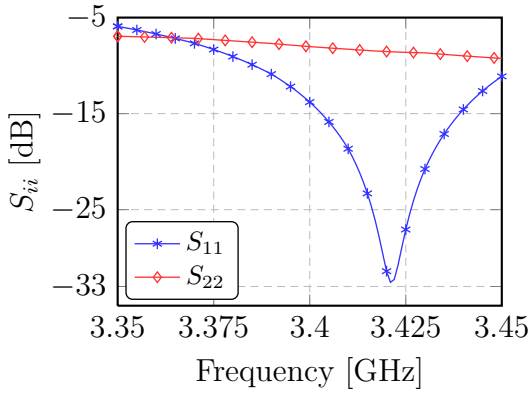
the optimization did not yield as high values as desired, the achieved results were still considered acceptable.

Table D.1: Optimization criteria for optimization at expected input power level for $P_{1\,\mathrm{dB}}$.

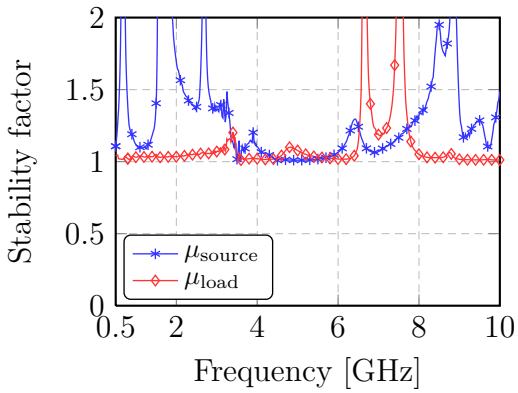
Property	min P_{out} [dBm]	min η_{PAE} [%]	min G [dB]	max S_{11} [dB]
Value	37.78	65	12	−10



(a) Small signal gain and MAG after matching both input and output.

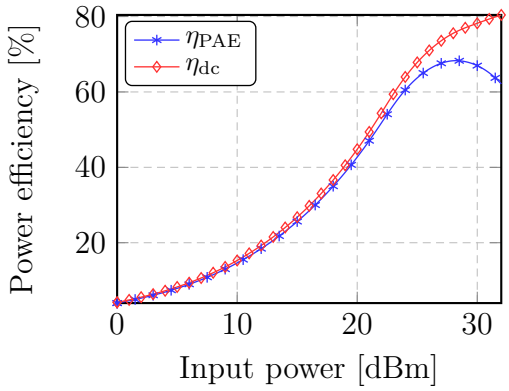


(b) Input and output return loss.

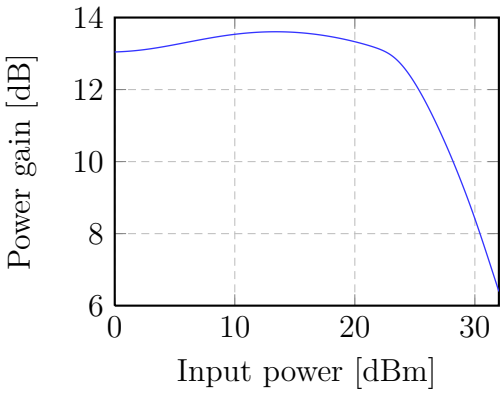


(c) Stability factor after matching.

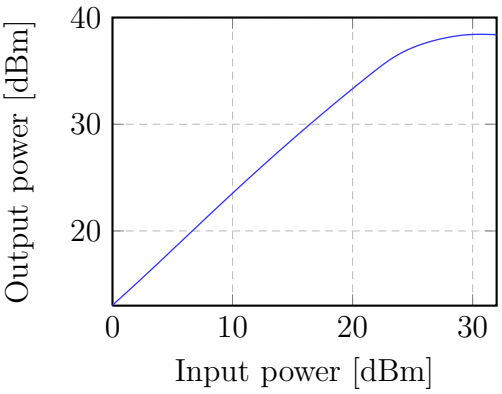
Figure D.1: Simulated small signal performance for the 6 W class AB PA at 3.40 GHz after matching for max. PAE.



(a) Power efficiency.



(b) Power gain.



(c) Output power.

Figure D.2: Simulated large signal performance for the 6 W class AB PA at 3.40 GHz after matching for max. PAE

Measurement Setup E

In this appendix the measurement setups used for the small- and large signal measurements, together with the load pull measurements are presented. For the small signal measurements, the DUT is the power amplifier design used in the load/source pull setup, whereas for the large signal measurements, a full power amplifier design based on the same transistor is used. The large signal measurement setup presented in this appendix is therefore only used to acquire input/output data for testing the modeling and linearization performance of the algorithms presented in Chapter 4 and Chapter 5, respectively.

E.1 Small Signal Measurement Setup

Figure E.1 illustrates the small signal measurement setup used in order to measure the S parameters of the power amplifier design for the load/source pull setup. The S parameters are measured with an Agilent E8364B PNA network analyzer [101].[1] Due to oscillatory behavior for the DUT, a stabilization circuit ($\mathbb{S}$) providing the necessary loss, is placed in between the dc supply and the bias tee on the gate side of the DUT. In order to measure the drain current accurately a Fluke 177 True RMS Multimeter ($\mathbb{M}$), is series connected between the dc supply and the bias tee on the drain side of the DUT.

Since the DUT potentially can deliver several watts, two attenuators are used between the DUT's RF output and port 2 on the PNA, to ensure that the power level is below the maximum input power of 30 dBm for the PNA. The first attenuator ($\mathbb{A}_1$) is a 10 dB Huber+Suhner attenuator which tolerates up to 10 W, whereas the second attenuator ($\mathbb{A}_2$) is a 10 dB attenuator from the same manufacturer which tolerates up to 2 W, hence the order of the two components. For all measurements the reference and measurement plane is in between the bias tees and the DUT's input/output.

[1] PNA is an acronym for *programmable network analyzer*.

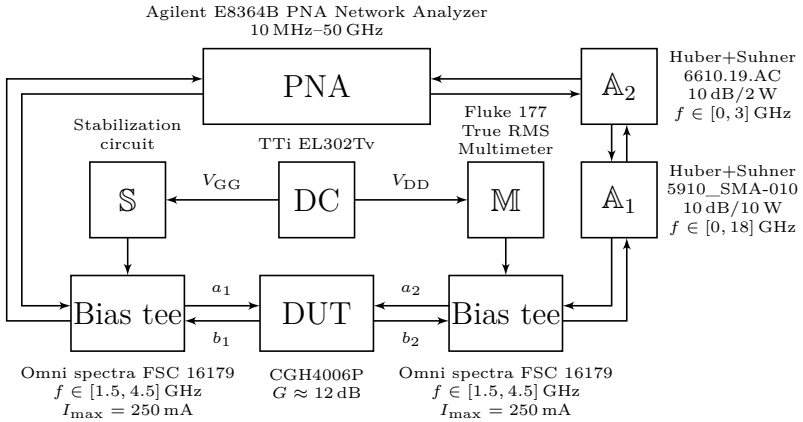


Figure E.1: Measurement setup for small signal measurements.

E.2 Large Signal Measurement Setup

Figure E.2 illustrates the large signal measurement setup used in this work. The modulated signal is generated in MATLAB, and then uploaded to a Rohde & Schwarz SMU200A vector signal generator (VSG) through a GPIB interface.[2] If the sampling frequency of the signal is less than the system clock of 100 MHz, the VSG resamples the waveform to the system clock, before passing it to the DAC [102, pp. 388–389].

The DAC operates at a sample rate of 400 MHz with 4x interpolation. When converted to an analog waveform $\boldsymbol{x}_t$, the signal is passed through a low pass filter to limit the RF bandwidth to a maximum of 80 MHz, before being modulated to passband by direct upconversion. The RF signal $\tilde{\boldsymbol{x}}_t$ is then filtered by a re–tunable bandpass filter [102] to suppress harmonic components. The maximum output peak–envelope–power (PEP) of the signal generator is 19 dBm (26 dBm overrange), thus a driver PA is used to reach the input power level required to drive the 6 W PA into saturation. The driver PA is a 10 W GaN HEMT amplifier (CGH40010) designed in [103] for class B operation, which, however, in this setup is operated in a class AB, providing a fairly linear power gain of 15 dB. The PEP of the 6 W PA will be in the range of 39 dBm to 40 dBm, almost 10 dB greater than the maximum allowed level into the Rohde & Schwarz FSQ40 vector signal analyzer (VSA). To operate within this limit, an attenuator is used prior to the VSA. In the VSA, the analog signal is first downconverted to an intermediate frequency (IF) of 20.40 MHz, before being filtered by an analog bandpass IF–filter with a tunable bandwidth. After filtering, the IF signal is digitized using an ADC with a sampling rate of 81.60 MHz, and a resolution of 14 bit [92, pp. 666–667]. The digitized IF signal is then low–pass–filtered prior to a reduction in sampling rate by resampling and decimation, which will help avoiding aliasing products due to the decimation [92]. The signal is then passed back to the PC with MATLAB for further processing.

[2]GPIB is an acronym for *general purpose interface bus*, and is an IEEE standard (IEEE–488) for short–range digital communication.

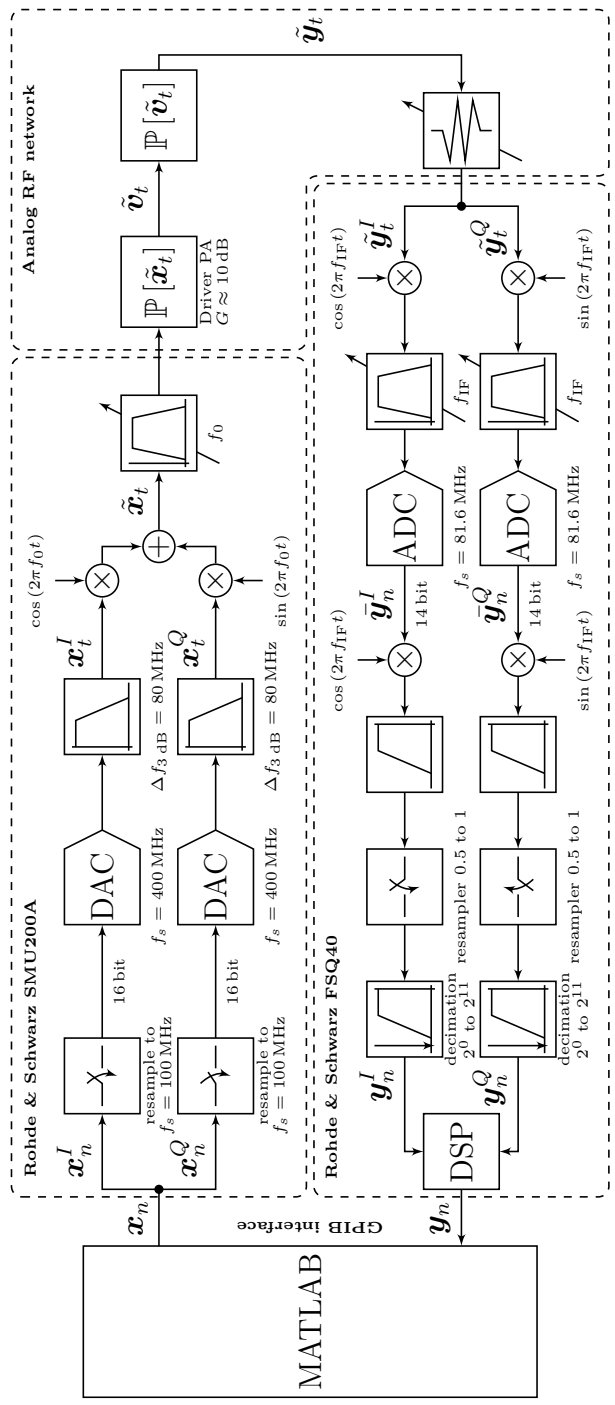


Figure E.2: Measurement setup for large signal measurements.

E.3 Load Pull Measurement Setup

Figure E.3 depicts the the load pull measurement setup used for this work. In contrast to the large signal measurements, the SMU200A is in this setup used to generate a single tone sinusoid. To reach the power levels required to drive the DUT into saturation, two driver PAs, providing a total transducer power gain of approximately 30 dB, are used. After these, a circulator is used in order to ensure that any reflections from the components following the circulator are directed away from the signal path with the driver PAs. Following the circulator is a high directional coupler, coupling a portion (-13 dB) of the input signal to a path leading to the input power sensor. Due to its maximum allowed input power of 20 dBm, an additional attenuator $\mathbb{A}_1$ is used prior to the power sensor. On the direct path, after the coupler, a bias tee follows.

Since the DUT is designed without biasing networks, biasing tees are required in the setup to provide dc to the DUT. Following the bias tee is the source tuner $\mathbb{T}_1$ and a waveprobe $\mathbb{W}_1$ for measuring the a_1 and b_1 waves on the device input.[3] These two blocks represent the input matching network of the DUT, implying that the available power from the source to the DUT is the power level at the source tuner input. After the DUT, the output network is somewhat reciprocal of the input network. A waveprobe $\mathbb{W}_2$ for measuring the a_2 and b_2 waves at the device output, a load tuner $\mathbb{T}_2$ for specifying the desired load impedances at different frequencies, and additional attenuators ($\mathbb{A}_2$, $\mathbb{A}_3$, and $\mathbb{A}_4$) to ensure that the input power levels to the output power sensor and the signal analyzer are within the maximum limits of 20 dBm and 30 dBm, respectively.

For calculating large signal properties of the DUT, we define the power delivered to the load as the power of the load tuner output, so any eventual power loss in the tuner and waveprobe is included in the calculations. From the illustration it is also clear that most measurement equipment is controlled by MATLAB through GPIB communication. The exception is the source and load tuners, that use a TCP/IP communication protocol, and are controlled by Focus Microwaves' Load Pull Explorer (LPExp). So one first specifies the desired loads in LPExp, and then proceed with measurements carried out in MATLAB.[4]

[3]The waveprobes couples portions of the incident and reflected waves to the SWAP–X402, which then process' the data.

[4]In the long run, also the tuners will be controlled by MATLAB, which have the possibility to control LPExp through an ActiveX server.

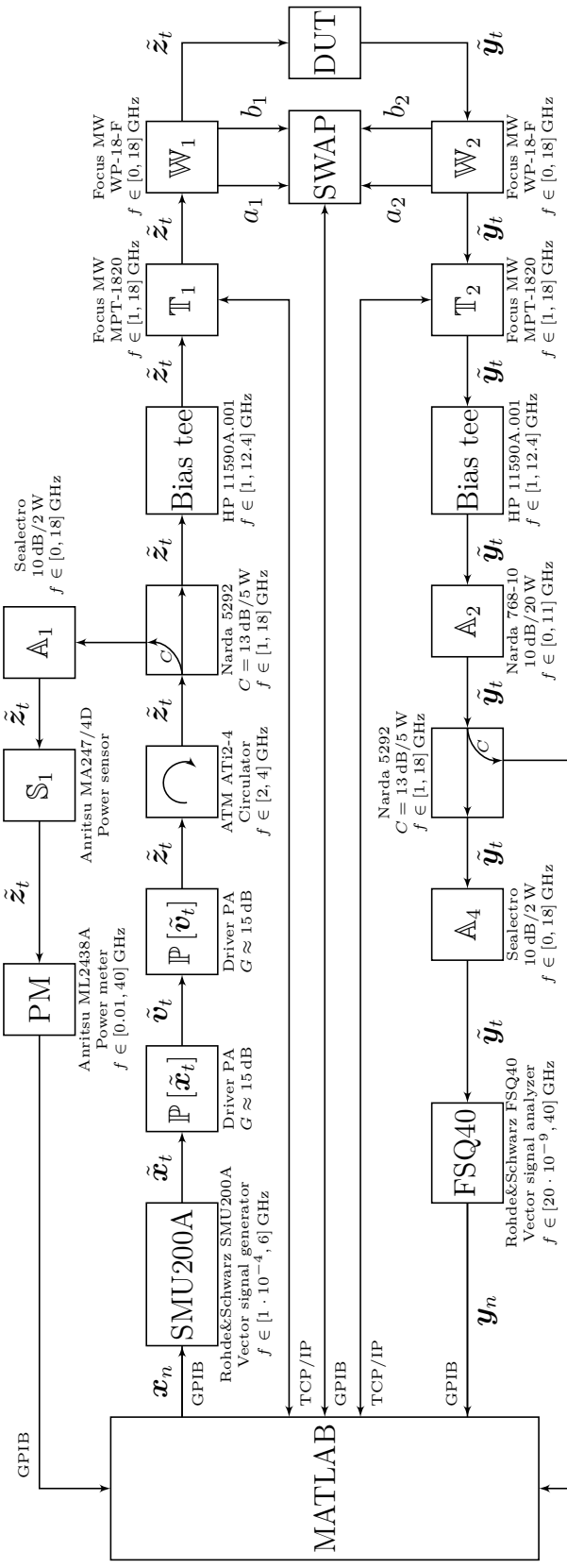


Figure E.3: Measurement setup for load pull measurements. It should be noted that there is only one power meter used in the setup, although two blocks in the illustration denotes a power meter. This is done in order to simplify the block schematic of the measurement setup.

Design Layout F

In this appendix an illustration of the layout of the microwave power amplifier used in the load/source pull setup is given. The components included in the illustration have the component values $R_1 = 180\,\Omega$, $R_2 = 56\,\Omega$, $C_1 = 1.20\,\text{pF}$, and $C_2 = 3.30\,\text{pF}$. The layout illustrated is valid for both designs with different dielectric constant. The difference between the two designs is the $50\,\Omega$ line width, hence, wherever there are differences in the design, these are explicitly pointed out. For symmetric structures, i.e. the two opposite sides of the series-connected parallel filter, all line widths are equal, even though the width is only specified at one side of the filter in the illustration. For lines with no width specified at all, the width used is the corresponding $50\,\Omega$ width. Circular shapes with stripes denote VIA holes.

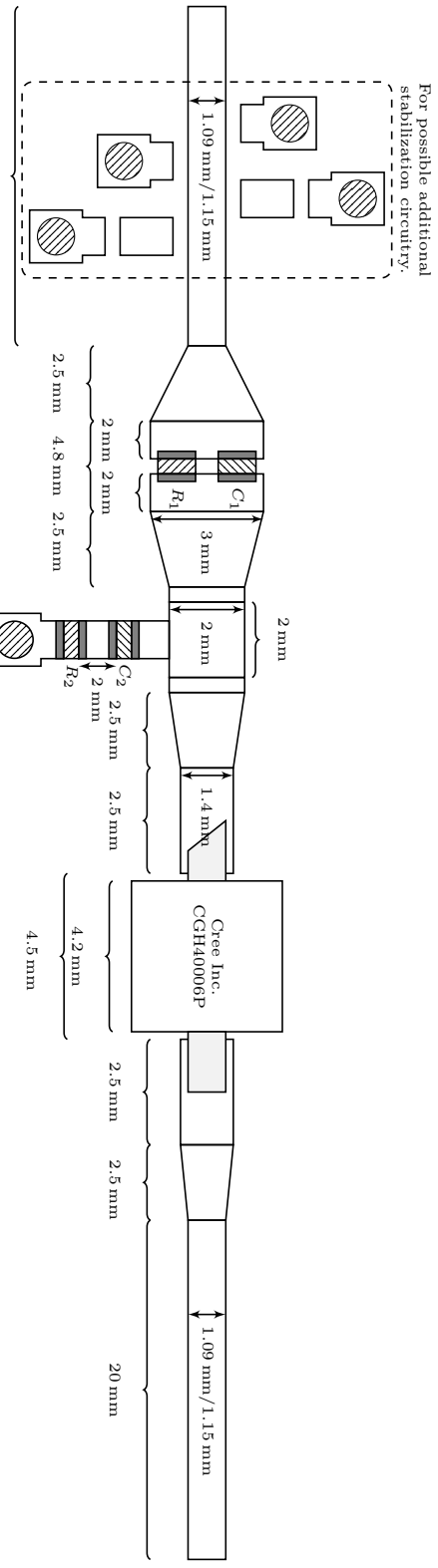


Figure F.1: PCB layout for microwave power amplifier design used in load/source-pull measurement setup. RF input on the left hand side, and RF output on the right hand side.

Additional Results Small- and Large Signal Measurements

G

In this appendix, additional results in terms of figures and tables from the small- and large signal measurements are given.

G.1 Small-Signal Measurements

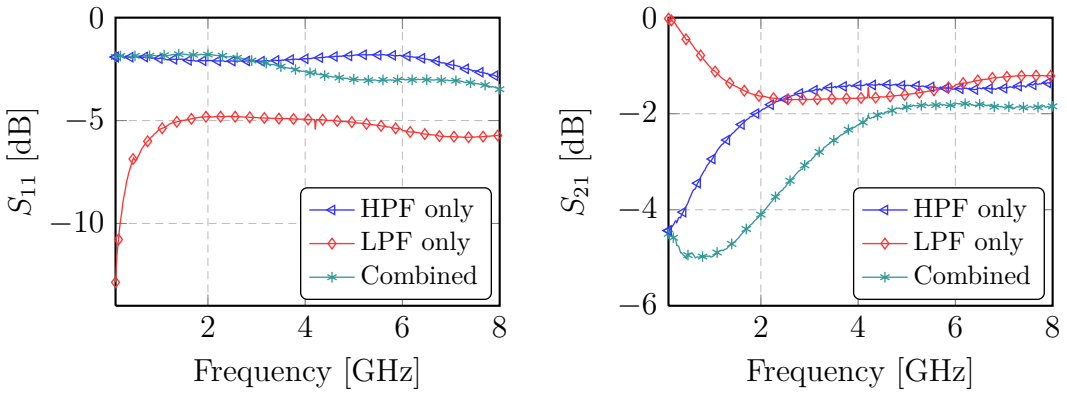
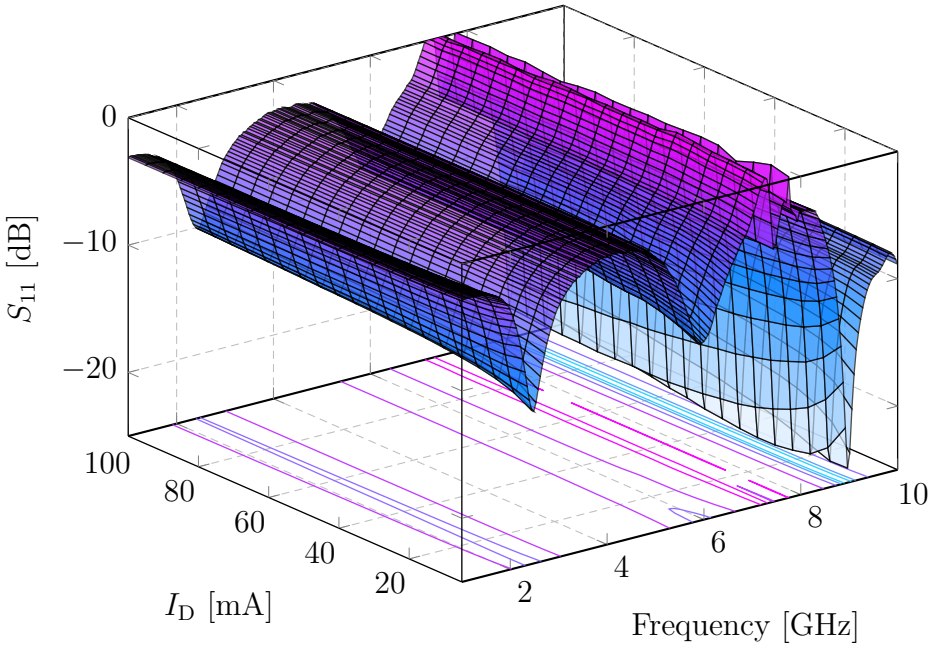
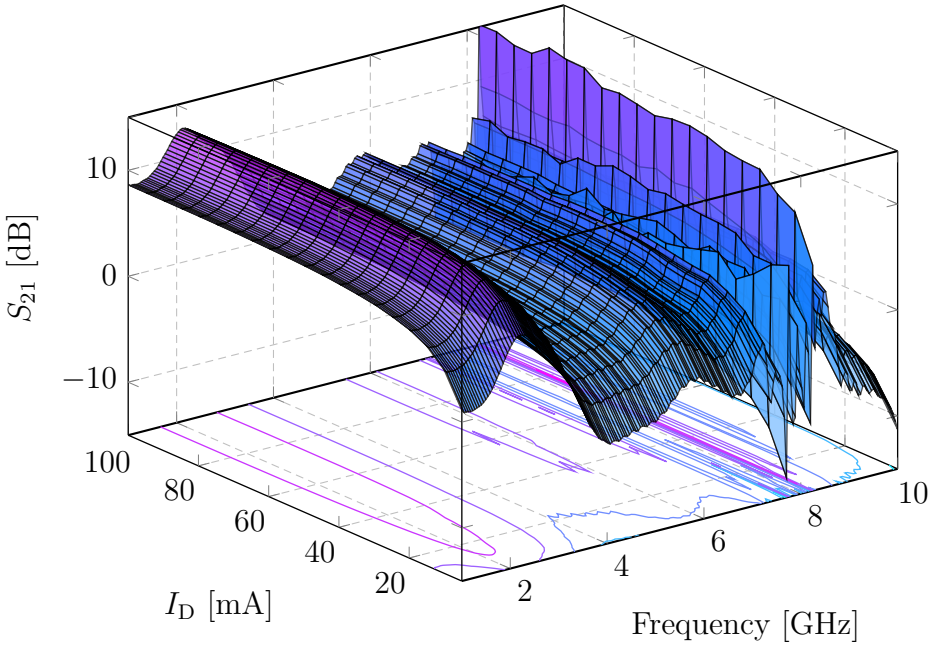


Figure G.1: Simulated forward mode S parameters for the two RC filters used in the stabilization circuit. In the plots, the forward mode S parameters for each of the filters, and for the cascaded connection of them, are plotted. In the figures, HPF is short for high–pass filter (parallel RC in series), while LPF is short for low–pass filter (series RC in shunt).

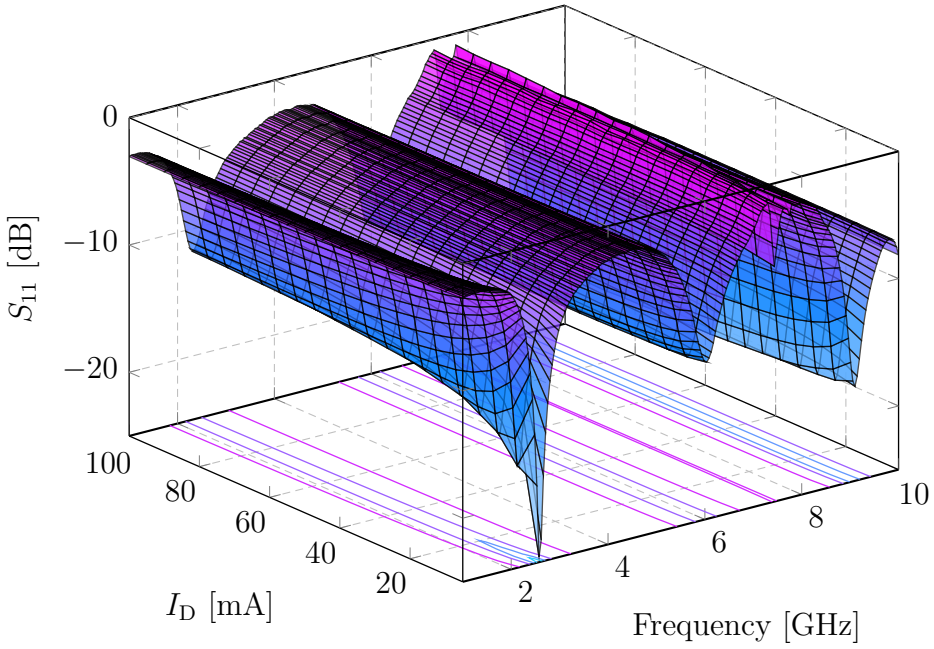


(a) S_{11} vs. I_D vs. Frequency.

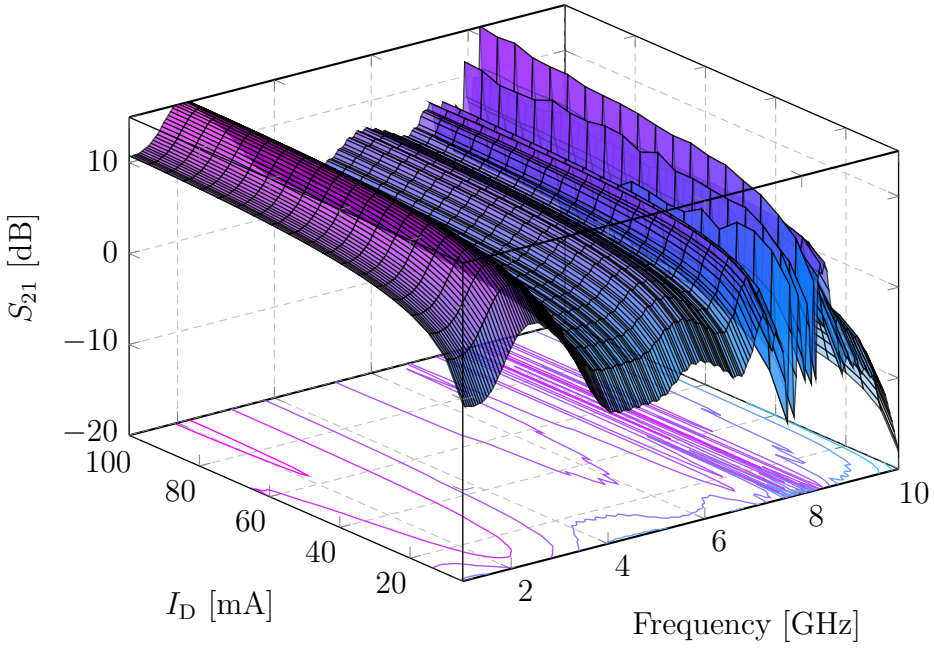


(b) S_{21} vs. I_D vs. Frequency.

Figure G.2: Measured S parameters for design with $\epsilon_r = 3.55$. In the measurements $I_D \in [5, 100]$ mA and $f \in [1, 10]$ GHz.



(a) S_{11} vs. I_D vs. Frequency.

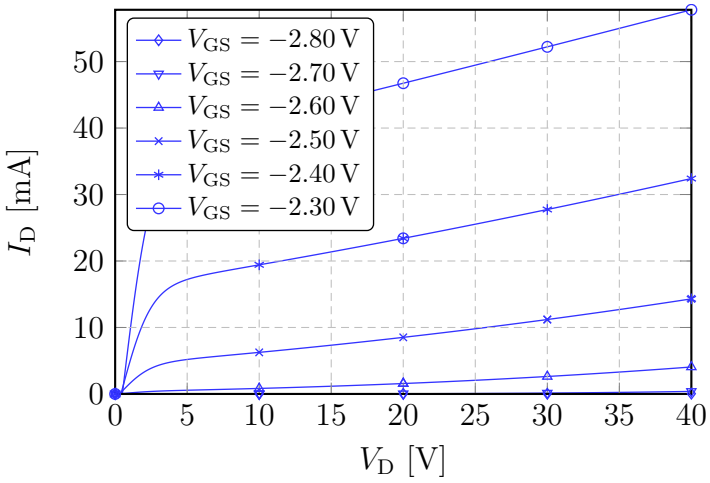


(b) S_{21} vs. I_D vs. Frequency.

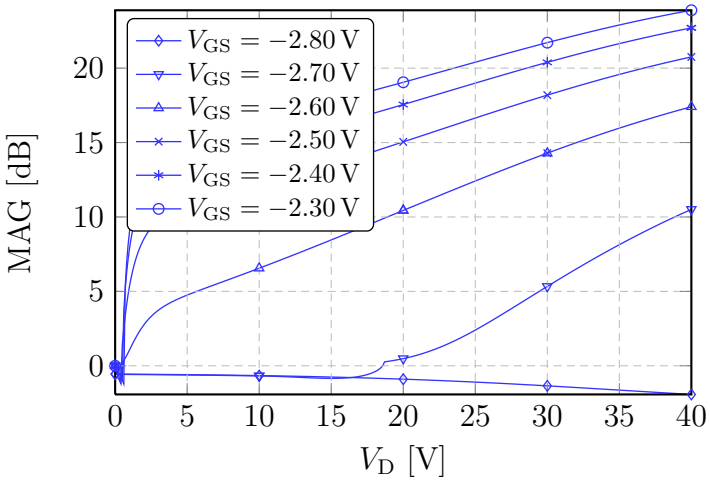
Figure G.3: Measured S parameters for design with $\epsilon_r = 3.30$. In the measurements $I_D \in [5, 100]$ mA and $f \in [1, 10]$ GHz.

CGH4006P DC/Bias Data H

This appendix presents simulated dc/bias data for the CGH4006P based on the nonlinear transistor model provided by Cree Inc. .



(a) IV curves for CGH4006P.



(b) MAG vs. biasing for CGH4006P.

Figure H.1: DC/Bias properties of CGH4006P.

Additional Results for Black-Box Modeling of Microwave Power Amplifiers

I

I.1 1 W GaAs pHEMT Class AB PA

I.1.1 Modeling Results for Memory Polynomial Model

Table I.1: Modeling results for the memory polynomial model with linearly increasing model order P and static kernel length of $M = 1$. The basis functions are the regular polynomials, and all polynomial orders are used. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 1 W PA.

M	P	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
1	1	1	1	−22.25	−33.59	−32.79
1	2	2	1.16×10^1	−27.85	−40.57	−40.54
1	3	3	1.01×10^2	−29.11	−46.31	−45.67
1	4	4	8.36×10^2	−29.22	−47.66	−46.24
1	5	5	7.15×10^3	−29.22	−47.66	−46.24
1	6	6	5.98×10^4	−29.24	−47.85	−46.47
1	7	7	5.18×10^5	−29.25	−47.93	−46.52
1	8	8	4.56×10^6	−29.25	−47.98	−46.55
1	9	9	4.03×10^7	−29.25	−47.99	−46.57
1	10	10	3.69×10^8	−29.25	−48.01	−46.57
1	11	11	3.42×10^9	−29.25	−48.01	−46.57
1	12	12	3.23×10^{10}	−29.25	−48.01	−46.57
1	13	13	3.07×10^{11}	−29.25	−48.01	−46.57

Table I.2: Modeling results for the memory polynomial model with linearly increasing model order P and a kernel length of $M = 2$. The basis functions are the regular polynomials, and all polynomial orders are used. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 1 W PA.

M	P	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
2	1	2	1.76×10^{1}	−23.15	−33.59	−32.79
2	2	4	1.01×10^{2}	−32.43	−41.24	−41.23
2	3	6	5.67×10^{2}	−38.32	−48.62	−49.20
2	4	8	3.57×10^{3}	−39.40	−51.55	−51.35
2	5	10	2.39×10^{4}	−39.42	−51.51	−51.44
2	6	12	1.70×10^{5}	−39.60	−52.21	−51.98
2	7	14	1.29×10^{6}	−39.65	−52.43	−52.26
2	8	16	1.04×10^{7}	−39.70	−52.59	−52.37
2	9	18	8.70×10^{7}	−39.71	−52.64	−52.46
2	10	20	7.62×10^{8}	−39.72	−52.66	−52.49
2	11	22	6.93×10^{9}	−39.72	−52.66	−52.50
2	12	24	6.47×10^{10}	−39.72	−52.66	−52.50
2	13	26	6.12×10^{11}	−39.72	−52.66	−52.50

I.1.2 Modeling Results for Spline Delay Envelope Model

Table I.3: Modeling results for the memory polynomial model with fixed model order of $P = 9$ and a linearly increasing kernel length. The basis functions are the regular polynomials, and all polynomial orders are used. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 1 W PA.

M	P	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
1	9	9	4.03×10^7	-29.25	-47.99	-46.57
2	9	18	8.70×10^7	-39.71	-52.64	-52.46
3	9	27	1.44×10^8	-39.82	-52.91	-52.77
4	9	36	2.19×10^8	-39.91	-53.25	-53.16
5	9	45	3.11×10^8	-39.94	-53.35	-53.29
6	9	54	4.23×10^8	-39.95	-53.41	-53.33
7	9	63	5.34×10^8	-39.96	-53.43	-53.34
8	9	72	6.41×10^8	-39.97	-53.45	-53.34
9	9	81	7.68×10^8	-39.98	-53.45	-53.35
10	9	90	9.23×10^8	-39.99	-53.45	-53.35
11	9	99	1.10×10^9	-39.99	-53.45	-53.36
12	9	108	1.26×10^9	-40.00	-53.45	-53.37
13	9	117	1.40×10^9	-40.01	-53.45	-53.36
14	9	126	1.53×10^9	-40.01	-53.45	-53.35
15	9	135	1.64×10^9	-40.02	-53.44	-53.37

Table I.4: Modeling results for the memory polynomial model with fixed model order of $P = 6$, fixed kernel length $M = 2$, and using all polynomial orders. The polynomial basis functions are varied, together with the memory polynomial model (regular/triangular). For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 1 W PA.

Type/basis	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
reg/$O_n^{(1)}$	12	2.20×10^4	−39.60	−52.21	−51.98
reg/$O_n^{(2)}$	12	1.01×10^3	−39.60	−52.21	−51.98
reg/H_n	12	3.70×10^3	−39.56	−52.05	−51.83
reg/$\tilde{H}_n$	12	3.40×10^6	−39.56	−52.03	−51.81
reg/L_n	12	1.01×10^3	−39.60	−52.21	−51.98
reg/T_n	12	2.56×10^6	−39.60	−52.21	−51.98
reg/U_n	12	2.73×10^6	−39.60	−52.21	−51.98
reg/B_n	12	1.70×10^5	−39.60	−52.21	−51.98
reg/$\tilde{L}_n$	12	1.64×10^6	−39.55	−52.01	−51.80
reg/P_n	12	1.70×10^5	−39.60	−52.21	−51.98
tri/$O_n^{(1)}$	10	8.94×10^3	−39.59	−52.18	−51.95
tri/$O_n^{(2)}$	10	4.03×10^3	−39.59	−52.18	−51.95
tri/H_n	10	1.56×10^4	−39.56	−52.03	−51.80
tri/$\tilde{H}_n$	10	1.59×10^6	−39.55	−52.01	−51.79
tri/L_n	10	4.03×10^3	−39.59	−52.18	−51.95
tri/T_n	10	1.02×10^6	−39.59	−52.18	−51.95
tri/U_n	10	1.15×10^6	−39.59	−52.18	−51.95
tri/B_n	10	6.97×10^4	−39.59	−52.18	−51.95
tri/$\tilde{L}_n$	10	8.08×10^5	−39.55	−52.00	−51.77
tri/P_n	10	6.97×10^4	−39.59	−52.18	−51.95

Table I.5: Modeling results for the memory polynomial model with fixed model order of $P = 6$, fixed kernel length $M = 2$, and using only even polynomial orders. The polynomial basis functions are varied, together with the memory polynomial model (regular/triangular). For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 1 W PA.

Type/basis	dim $(\boldsymbol{c})$	cond $(\underline{\boldsymbol{H}})$	nmse $(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	acepr $(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
reg/$O_n^{(1)}$	6	3.68×10^1	-36.65	-46.18	-46.24
reg/$O_n^{(2)}$	6	4.96×10^1	-38.05	-48.34	-48.45
reg/H_n	6	9.44×10^2	-38.98	-50.45	-50.61
reg/$\tilde{H}_n$	6	7.41×10^4	-38.64	-49.77	-50.06
reg/L_n	6	4.96×10^1	-38.05	-48.34	-48.45
reg/T_n	6	1.20×10^4	-37.43	-47.39	-47.83
reg/U_n	6	2.59×10^4	-37.43	-47.39	-47.83
reg/B_n	6	1.89×10^3	-37.43	-47.39	-47.83
reg/$\tilde{L}_n$	6	6.86×10^3	-37.47	-47.19	-47.70
reg/P_n	6	1.13×10^3	-37.47	-47.19	-47.70
tri/$O_n^{(1)}$	5	3.68×10^1	-36.65	-46.18	-46.24
tri/$O_n^{(2)}$	5	4.96×10^1	-38.05	-48.34	-48.45
tri/H_n	5	9.44×10^2	-38.98	-50.45	-50.61
tri/$\tilde{H}_n$	5	7.41×10^4	-38.64	-49.77	-50.06
tri/L_n	5	4.96×10^1	-38.05	-48.34	-48.45
tri/T_n	5	1.20×10^4	-37.43	-47.39	-47.83
tri/U_n	5	2.59×10^4	-37.43	-47.39	-47.83
tri/B_n	5	1.89×10^3	-37.43	-47.39	-47.83
tri/$\tilde{L}_n$	5	6.86×10^3	-37.47	-47.19	-47.70
tri/P_n	5	1.13×10^3	-37.47	-47.19	-47.70

Table I.6: Modeling results for the spline delay envelope model with a linearly increasing number of envelope segments, static kernel length, and a square segment matrix. The segment matrix is varied between being diagonal, upper triangular, and lower triangular. For the two columns with acepr values, the left and right column denotes the lower and upper adjacent channel, respectively. The modeled power amplifier is the measured 1 W PA.

Type/$[M\,Q\,L]$	$\dim(\boldsymbol{c})$	$\mathrm{cond}(\underline{\boldsymbol{H}})$	$\mathrm{nmse}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	$\mathrm{acepr}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ [dB]	
diag/[1 1 1]	4	8.36×10^2	−29.22	−47.66	−46.24
diag/[2 1 1]	5	9.64×10^2	−29.22	−47.67	−46.23
diag/[3 1 1]	6	6.89×10^3	−29.25	−47.91	−46.53
diag/[4 1 1]	7	1.95×10^4	−29.24	−47.83	−46.42
diag/[5 1 1]	8	5.48×10^4	−29.25	−48.00	−46.57
upper/[1 1 1]	4	8.36×10^2	−29.22	−47.66	−46.24
upper/[2 1 1]	5	9.64×10^2	−29.22	−47.67	−46.23
upper/[3 1 1]	6	6.89×10^3	−29.25	−47.91	−46.53
upper/[4 1 1]	7	1.95×10^4	−29.24	−47.83	−46.42
upper/[5 1 1]	8	5.48×10^4	−29.25	−48.00	−46.57
lower/[1 1 1]	4	8.36×10^2	−29.22	−47.66	−46.24
lower/[2 1 1]	5	9.64×10^2	−29.22	−47.67	−46.23
lower/[3 1 1]	6	6.89×10^3	−29.25	−47.91	−46.53
lower/[4 1 1]	7	1.95×10^4	−29.24	−47.83	−46.42
lower/[5 1 1]	8	5.48×10^4	−29.25	−48.00	−46.57

Example of MATLAB Implementation J

In this appendix an example of a MATLAB implementation for black–box behavioral modeling with a Memory Polynomial model is presented. The implementation is done with object–oriented MATLAB programming, so when calling the function, a MemoryPolynomialModel object will be created, and give the user access to all its member variables and member functions. In the following, the MATLAB code for the MemoryPolynomialModel class is given, together with some of its member functions and other functions it uses in the generation of the MemoryPolynomialModel object.[1]

All the system models presented in Chapter 4 are implemented in the same manner, and can be provided if desired.[2] The last code provided also shows an example of how to communicate with a measurement equipment (Anritsu ML2438A power meter for this case) thru a GPIB interface. MATLABs built–in functions are not presented in this chapter.

```
classdef MemoryPolynomialModel < handle
    % MP = MEMORYPOLYNOMIALMODEL(x,y,P,M,orders,type,basis,method)
    %
    % Implement a memory polynomial model (MP) for characterizing a
    % nonlinear system with a given nonlinear order (P) and memory depth
    % (M). The polynomial 'orders' used can either be 'all' or 'odd',
    % whereas the 'type' of the MP model can either be 'regular' or
    % 'triangular', which provides a decreasing nonlinear order for
    % increasing sample delays. The 'method' options allows the user to
    % specify if QR factorization ('qr') or a Moore–Penrose pseudo–inverse
    % factorization ('pinv') should be used in order to solve the
    % linear system of equations.
    %
    % Author: Dragan Mitrevski
    % Date: Sep, 2010
    % Contact: dragan.mitrevski@ieee.org

    %%
    properties (GetAccess='public', SetAccess='private')
        % Member variables for MemoryPolynomialModel object
        Order; % Nonlinear order
```

[1] Some functions are only used to check the inputs, and are thus of no interest with respect to how the actual model structure is implemented.

[2] Contact Associate Professor Morten Olavsbråten, the supervisor of the work.

```
        Memory; % Memory depth
        OrderTypes % Type or nonlinear orders (i.e. odd/even/all)
        Type; % Regular ('reg') og triangular ('tri') MP model
        Basis; % Type of basis function (i.e. Hermitian, Laguerre)
        NumOfCoeffs; % Number of coefficients to be estimated
        Coeffs; % Coefficient vector
        SignalMatrix; % Data signal matrix (regressor)
        ConditionNumber; % Condition number of data signal matrix
        NormalizedMSE; % Normalized MSE between desired and model output
        ModelInput; % Model input signal
        DesiredOutput; % Desired output signal
        ModelOutput; % Estimated model output signal
        MatrixFactorization; % Factorization for solving linear system
    end
    %%
    methods
        % Member functions
        function hObj = MemoryPolynomialModel(varargin)
            % Constructor for MemoryPolynomialModel object
            hObj.ModelInput = getArg(1,varargin,0);
            hObj.DesiredOutput = getArg(2,varargin,0);
            hObj.Order = getArg(3,varargin,7);
            hObj.Memory = getArg(4,varargin,5);
            hObj.OrderTypes = getArg(5,varargin,'all');
            hObj.Type = getArg(6,varargin,'regular');
            hObj.Basis = getArg(7,varargin,'pol');
            hObj.MatrixFactorization = getArg(8,varargin,'qr');
            NormData = check_norm(hObj.ModelInput,hObj.DesiredOutput);
            check_basis(NormData,hObj.Basis);
            ModSeries = basis_vs_ordertype(hObj.Basis,hObj.OrderTypes);
            hObj.NumOfCoeffs = calcCoeffs(hObj.Order, ...
                                          hObj.Memory, ...
                                          hObj.OrderTypes, ...
                                          hObj.Type);
            hObj.SignalMatrix = getSeries(hObj.ModelInput, ...
                                          hObj.Order, ...
                                          hObj.Memory, ...
                                          hObj.NumOfCoeffs, ...
                                          hObj.OrderTypes, ...
                                          hObj.Type, ...
                                          hObj.Basis, ...
                                          ModSeries);
            hObj.ConditionNumber = cond(hObj.SignalMatrix);
            hObj.Coeffs = estimateCoeffs(hObj.SignalMatrix, ...
                                         hObj.DesiredOutput, ...
                                         hObj.MatrixFactorization);
            hObj.ModelOutput = hObj.SignalMatrix*hObj.Coeffs;
            hObj.NormalizedMSE = nmse(hObj.DesiredOutput, ...
                                      hObj.ModelOutput);
        end

        function setSignalMatrix(hObj,SignalMatrix)
            % Set new signal matrix + update parameters that depend on it.
            hObj.SignalMatrix = SignalMatrix;
            hObj.ConditionNumber = cond(SignalMatrix);
            hObj.Coeffs = estimateCoeffs(hObj.SignalMatrix, ...
                                         hObj.DesiredOutput, ...
                                         hObj.MatrixFactorization);
            hObj.ModelOutput = hObj.SignalMatrix*hObj.Coeffs;
            hObj.NormalizedMSE = nmse(hObj.DesiredOutput, ...
                                      hObj.ModelOutput);
        end
```

```
        function setCoeffs(hObj,Coeffs)
           % Set new coefficient vector + update parameters that depend on
           % it.
            hObj.Coeffs = Coeffs;
            hObj.ModelOutput = hObj.SignalMatrix*hObj.Coeffs;
            hObj.NormalizedMSE = nmse(hObj.DesiredOutput, ...
                                      hObj.ModelOutput);
        end

        function setMethod(hObj,MatrixFactorization)
            % Set new method for solving the linear system + update
            % parameters that depend on it.
            hObj.MatrixFactorization = MatrixFactorization;
            hObj.Coeffs = estimateCoeffs(hObj.SignalMatrix, ...
                                         hObj.DesiredOutput, ...
                                         hObj.MatrixFactorization);
            hObj.ModelOutput = hObj.SignalMatrix*hObj.Coeffs;
            hObj.NormalizedMSE = nmse(hObj.DesiredOutput, ...
                                      hObj.ModelOutput);
        end
    end
end

function n = calcCoeffs(P,M,orders,type)
% Calculate the number of coefficients required for the memory polynomial
% model of choise (based on user-specified options)
%
% CALCCOEFFS calculates the number of coefficients required for the MP
% model of choice, based on the specified options e.g. only odd orders,
% even orders, triangular MP and so on.
%
% Inputs:
%   - P: nonlinear order
%   - M: memory depth
%   - orders: specify which orders that should be considered e.g only odd
%             orders ('odd'), only even orders ('even') or all ('all')
%   - type: specify which type of MP model to be used, regular or
%           triangular ('tri'), which reduce the nonlinear order for
%           increasing memory.
%
% Author: Dragan Mitrevski
% Date: Dec, 2010
% Contact: dragan.mitrevski@ieee.org

if strcmpi(type,'tri') % Triangular memory polynomial model
    % Odd orders only
    if strcmpi(orders,'odd')
        n = -1;
        for m = 0:M-1
            if P > m
                Ps = P-m;
            else
                Ps = 1;
            end
            for p = 1:2:Ps
                n = n+1;
            end
        end
    % Even orders only
    elseif strcmpi(orders,'even');
        n = -1;
        for m = 0:M-1
            if P > m
```

```
                Ps = P-m;
            else
                Ps = 1;
            end
            for p = 1:2:Ps
                n = n+1;
            end
        end
    % All orders
    else
        n = -1;
        for m = 0:M-1
            if P > m
                Ps = P-m;
            else
                Ps = 1;
            end
            for p = 1:Ps
                n = n+1;
            end
        end
    end
else % Regular memory polynomial model
    if strcmpi(orders,'odd') % Odd orders only
        if mod(P,2) == 1
            n = (P+1)/2*M;
        else
            n = P*M/2;
        end
    elseif strcmpi(orders,'even') % Even orders only
        if mod(P,2) == 1
            n = (P+1)/2*M;
        else
            n = P*M/2;
        end
    else % Even orders only
        n = P*M;
    end
end

end

function X = getSeries(x,P,M,coeffs,orders,type,basis,mod_bool)
% Get regression matrix for memory polynomial model with max nonliner order
% P and memory length/depth M.
%
% GETSERIES composes the regression matrix for a memory polynomial model
% (MP) of order P with memory depth M.
%
% Inputs:
%   - x: input signal, preferrably a row vector
%   - P: nonlinear order
%   - M: memory depth
%   - coeffs: number of coefficents required (only used for initalizing)
%   - orders: 'all' or 'odd' - which gives a regression matrix composed of
%             odd ordes only (1,3,5,...)
%   - type: regular MP model or triangular, which decrease the nonlinear
%           order for increasing memory lengths
%   - basis: specify which type of polynomial basis functions to be used.
%
% Outputs:
%   - X: regression matrix
%
```

```
% Author: Dragan Mitrevski
% Date: Sep, 2010
% Contact: dragan.mitrevski@ieee.org


Nx = length(x);
% Initialize matrix for computational speed
X = zeros(Nx,coeffs);

% Initialize counter for # columns in X
n = 0;

% Only odd oders used in the composition of the regression matrix
switch lower(orders)
    case 'odd' % Only odd orders used to compute regression matrix
        for m = 0:M-1
            % Delay signal with m samples
            x_mp = delay_signal(x,m);

            % Triangular memory polynomial model — reduce nonlinear order
            % with increasing memory
            if strcmpi(type,'tri')
                if P > m
                    Ps = P-m;
                else
                    Ps = 1;
                end
            else
                Ps = P;
            end

            % Construct regression matrix
            for p = 1:2:Ps
                c_vec = basis_function(p,basis);
                if mod_bool == 1 && p > 1
                    c_vec = c_vec(1:end-1);
                elseif mod_bool == 2 && p > 1
                    c_vec = c_vec(1:end-2);
                end
                n = n+1;
                X(:,n) = x_mp.*polyval(c_vec,abs(x_mp));
            end
        end
    case 'even' % Only even orders used to compute regression matrix
        for m = 0:M-1
            % Delay signal with m samples
            x_mp = delay_signal(x,m);

            % Triangular memory polynomial model — reduce nonlinear order
            % with increasing memory
            if strcmpi(type,'tri')
                if P > m
                    Ps = P-m;
                else
                    Ps = 1;
                end
            else
                Ps = P;
            end

            % Construnct regression matrix
            for p = 1:2:Ps
                c_vec = basis_function(p-1,basis);
```

```
                if mod_bool == 1 && p > 1
                    c_vec = c_vec(1:end-1);
                elseif mod_bool == 2 && p > 1
                    c_vec = c_vec(1:end-2);
                end
                n = n+1;
                X(:,n) = x_mp.*polyval(c_vec,abs(x_mp));
            end
        end
    case 'all' % All orders used to compute regression matrix
        for m = 0:M-1
            % Delay signal with m samples
            x_mp = delay_signal(x,m);

            % Triangular memory polynomial model - reduce nonlinear order
            % with increasing memory
            if strcmpi(type,'tri')
                if P > m
                    Ps = P-m;
                else
                    Ps = 1;
                end
            else
                Ps = P;
            end

            % Construct regression matrix
            for p = 0:Ps-1
                c_vec = basis_function(p,basis);
                if mod_bool == 1 && p > 1
                    c_vec = c_vec(1:end-1);
                elseif mod_bool == 2 && p > 1
                    c_vec = c_vec(1:end-2);
                end
                n = n+1;
                X(:,n) = x_mp.*polyval(c_vec,abs(x_mp));
            end
        end
end
end

function w = estimateCoeffs(A,d,factopt)
% Solve linear system Ax = b with the specified matrix factorization.
%
% ESTIMATECOEFFS(A,d,factopt) solves the system Ax = b with the specified
% matrix factorization method.
%
% Inputs:
% -A: input data matrix (M-by-N)
% -d: desired data vector (M-by-1)
% -factopt: matrix factorization option. As for now only pinv/none is
%           supported. If none is specified, MATLAB's \ operator is used,
%           and the system is solved with QR factorization with column
%           pivoting. If either pinv is specified, or if the system is
%           rank-deficient, Moore-Penrose's pseudo-inverse is used - as
%           this guarantees to obtain the least L2 norm.
%
% Outputs:
% -w: solution of Ax = b
%
% NB: the code makes uese of Bruno Luongs pseudoinverse function, freely
% available from:
% http://www.mathworks.com/matlabcentral/fileexchange/25453-pseudo-inverse
```

```
% The function can be replaced with MATLAB's internal pinv() function,
% which does the same.
%
% Author: Dragan Mitrevski
% Date: Sep, 2010
% Contact: dragan.mitrevski@ieee.org

% Get size of data matrix
[m n] = size(A);
% Check for rank-deficiency
rankA = rank(A);
% Get smallest dimension of A
minA = min(m,n);

% Check for rank-deficiency. If full-rank, QR factorization is used. If
% rank-deficient, pinv/pseudoinverse is used for solving the system.
if rankA == minA
    if ~strcmpi(factopt,'none')
        disp('Full-rank matrix - QR factorization with column pivoting');
    end
    w = A\d;
else
    disp('Matrix is rank deficient: rank(A) < min(m,n)');
    disp(['min(m,n): ',num2str(minA),', rank(A): ',num2str(rankA)]);
    disp('Solving system with pseudoinverse...');
    Ap = pseudoinverse(A);
    w = Ap*d;
end

function xd = delay_signal(x,d,delaytype)
% Delay a discrete-time signal with n samples.
%
% DELAY_SIGNAL(x,d,delaytype) delays the discrete-time input signal, x,
% with the specified amount of samples by zero padding either the beginning
% or the end of the input signal vector. How the vector is zero padded can
% be specified by the user.
%
% Inputs:
% - x: input signal (row vector)
% - d: length of delay in samples
% - delaytype: string specifying how the signal should be delayed, i.e. if
%              it should be zero-padded at the end of the vector or at the
%              beginning.
%
% Outputs:
% - xd: input signal delayed by d samples according to the specified delay
%       type
%
% Author: Dragan Mitrevski
% Date: Sep, 2010
% Contact: dragan.mitrevski@ieee.org

% Set default values if delaytype is missing
SetDefaultValue(3,'delaytype','pre2');

if d < 0
    error('Delay can''t be negative');
end

if strcmpi(delaytype,'post1')
    % post-delay #1 [x(1) ... x(N-d) 0 0 0]
    xd = [x(1:end-d); zeros(d,1)];
elseif strcmpi(delaytype,'post2')
```

```
    % post-delay #2 [x(1+d) ... x(N)..0 0 0]
    xd = [x(1+d:end); zeros(d,1)];
elseif strcmpi(delaytype,'pre1')
    % pre-delay #1 [0 0 0 x(1+d) ... x(N)]
    xd = [zeros(d,1); x(1+d:end)];
elseif strcmpi(delaytype,'pre2')
    % pre-delay #2 [0 0 0 x(1) ... x(N-d)]
    xd = [zeros(d,1); x(1:end-d)];
end

end

function p = hermitepoly(n,type)
% Compute coefficients for Hermitian polynomial of order n in a recursive
% manner.
% H_(n+1)(x) = xH_n(x)-nH_(n-1)(x) (probabilist)
% H_(n+1)(x) = 2xH_n(x)-2nH_(n-1)(x) (physicist)
%
% Inputs:
% -n: order of Hermitian polynomial
%
% Outputs:
% -p: (n+1) vector with coefficients for Hermitian polynomial of order n
%
% Author: Dragan Mitrevski
% Date: Jan, 2011
% Contact: dragan.mitrevski@ieee.org

if nargin < 1
    error('Order must be specified');
end

SetDefaultValue(2,'type','prob');

switch lower(type)
    case 'prob' % Hermitian polynomials for probabilists
        if n == 0
            p = 1; % Define H_0(x)
        elseif n == 1
            p = [1 0]'; % Define H_1(x)
        else % Recursivly find H_n(x) for n > 1
            p = cell(n,1); % Initialize cell-array
            p{1}  = 1; % Define H_0(x)
            p{2} = [1 0]'; % Define H_1(x)
            for m = 1:n
                p{3} = [p{2}; 0]-m*[zeros(2,1); p{1}];
                p{1} = p{2};
                p{2} = p{3};
            end
            p = p{3};
        end
    case 'phys' % Hermitian polynomials for physicists
        if n == 0
            p = 1; % Define H_0(x)
        elseif n == 1
            p = [2 0]'; % Define H_1(x)
        else % Recursivly find H_n(x) for n > 1
            p = cell(n,1); % Initialize cell-array
            p{1}  = 1; % Define H_0(x)
            p{2} = [2 0]'; % Define H_1(x)
            for m = 1:n
                p{3} = 2*[p{2}; 0]-2*m*[zeros(2,1); p{1}];
                p{1} = p{2};
```

```
                p{2} = p{3};
            end
            p = p{3};
        end
    otherwise
        error('Unknown type. Specify ''phys'' or ''prob''');
end
end

function [chApow chBpow] = measPowerML2384A(unit,emul,varargin)
% Communicate with Anritsu ML2438A power meter thru GPIB
%
% MEASPOWERML2384A communicates with the ML2438A power meter
% from Anritsu, and collects the measured power levels at the
% two power probes. The user can specify the unit of the
% returned data
%
% Inputs:
% -unit: specifies the unit ML2438A uses in the measurements
% -emul: specifies the emulation mode. The power meter supports
%        HP436A, HP437B, HP438A, and Anritsu ML4803 besides its
%        native mode.
% -varargin: variable input arguments, such as averaging options
%
% Outputs:
% -chApow: the measured power level at power probe A
% -chBpow: the measured power level at power probe B
%
% Author: Dragan Mitrevski
% Date: Feb, 2011
% Contact: dragan.mitrevski@ieee.org

BoardNumber = 0; % GPIB controller card number
DeviceNumber = 13; % Instrument GPIB address
powmeter = gpib('ni',BoardNumber,DeviceNumber); % Create object
powmeter.InputBufferSize = 512; % Set input buffer size
fopen(powmeter); % Open communication with instrument

fprintf(powmeter,'RST;*OPC');

% Check input arguments, and call nested function
if nargin == 2
    setEmulationMode(powmeter,emul); % Set emulation mode.
    setChannelConfig(powmeter,unit); % Set channel configuration
else
    setChannelConfig(powmeter,unit); % Set channel configuration
end

% Check variable input arguments, and call nested function
if nargin > 2
    for k = 1:numel(varargin)
        temp_str = inputname(k+2);
        if strcmpi(temp_str,'avg')
            avgoptions(powmeter,varargin{k});
        end
    end
end

% Measure power
fprintf(powmeter,'O 1;*OPC')
measChA = fscanf(powmeter);
chApow = str2num(measChA);
```

```
fprintf(powmeter,'0 2;*OPC')
measChB = fscanf(powmeter);
chBpow = str2num(measChB);

fclose(powmeter) % Close communication
delete(powmeter) % Delete object
end
%%
function avgoptions(powmeter,avg)
% AVGOPTIONS translates the specified averaging options to GPIB
% commands.
switch lower(avg.mode)
    case 'mov' % Moving average
        both = 'off';
        switch lower(avg.chan)
            case 'a' % Average on channel A
                chan = 'A';
            case 'b' % Average on channel B
                chan = 'B';
            case 'both' % Average on both channels
                both = 'on';
            otherwise % No averaging set, return to main function
                return
        end
        if strcmpi(both,'on') % Activate averaging on both channels
            fprintf(powmeter,'AVG A, MOV, %d;*OPC',avg.num);
            fprintf(powmeter,'AVG B, MOV, %d;*OPC',avg.num);
        else % Activate averaging on desired channel
            fprintf(powmeter,'AVG %s, MOV, %d;*OPC',chan,avg.num);
        end
    case 'rpt' % Repeat average
        both = 'off';
        switch lower(avg.chan)
            case 'a' % Average on channel A
                chan = 'A';
            case 'b' % Average on channel B
                chan = 'B';
            case 'both' % Average on both channels
                both = 'on';
            otherwise % No averaging set, return to main function
                return;
        end
        if strcmpi(both,'on') % Activate averaging on both channels
            fprintf(powmeter,'AVG A, RPT, %d;*OPC',avg.num);
            fprintf(powmeter,'AVG B, RPT, %d;*OPC',avg.num);
        else % Activate averaging on desired channel
            fprintf(powmeter,'AVG %s, RPT, %d;*OPC',chan,avg.num);
        end
    case 'auto' % Automatic average
        both = 'off';
        switch lower(avg.chan)
            case 'a' % Average on channel A
                chan = 'A';
            case 'b' % Average on channel B
                chan = 'B';
            case 'both' % Average on both channels
                both = 'on';
            otherwise % No averaging set, return to main function
                return;
        end
        if strcmpi(both,'on') % Activate averaging on both channels
            fprintf(powmeter,'AVG A, AUTO, %d;*OPC',avg.num);
            fprintf(powmeter,'AVG B, AUTO, %d;*OPC',avg.num);
```

```
        else % Activate averaging on desired channel
            fprintf(powmeter,'AVG %s, AUTO, %d;*OPC',chan,avg.num);
        end
    otherwise % No averaging, return to main function
        return;
end
end

%%
function setChannelConfig(powmeter,unit)
% SETCHANNELCONFIG specifies the desired settings at the specified channels
fprintf(powmeter,'CHCFG 1, A;*OPC'); % Set channel 1 to measure on probe A
fprintf(powmeter,'CHCFG 2, B;*OPC'); % Set channel 1 to measure on probe B

fprintf(powmeter,'CHRES 1, 3;*OPC'); % Set channel 1 resoluation to three
    digits
fprintf(powmeter,'CHRES 2, 3;*OPC'); % Set channel 2 resoluation to three
    digits

switch lower(unit) % Set measurement unit on both channels
    case 'w' % Set unit to Watt
        fprintf(powmeter,'CHUNIT 1 W;*OPC');
        fprintf(powmeter,'CHUNIT 2 W;*OPC');
    case 'dbm' % Set unit to dBM
        fprintf(powmeter,'CHUNIT 1 DBM;*OPC');
        fprintf(powmeter,'CHUNIT 2 DBM;*OPC');
    case 'dbuv' % Set unit to dBuV
        fprintf(powmeter,'CHUNIT 1 DBUV;*OPC');
        fprintf(powmeter,'CHUNIT 2 DBUV;*OPC');
    case 'dbmv' % Set unit to dBmV
        fprintf(powmeter,'CHUNIT 1 DBMV;*OPC');
        fprintf(powmeter,'CHUNIT 2 DBMV;*OPC');
    otherwise % Default is dBm
        disp('Default unit, dBm, set...');
        fprintf(powmeter,'CHUNIT 1 DBM;*OPC');
        fprintf(powmeter,'CHUNIT 2 DBM;*OPC');
end
end

%%
function setEmulationMode(powmeter,emul)
% SETEMULATIONMODE specifies the mode of operation for the power meter
% Default is its native mode.
switch lower(emul)
    case 'hp436a' % Emulate HP436A single channel power meter
        fprintf(powmeter,'EMUL %s;*OPC',upper(emul));
    case 'hp437b' % Emulate HP437B single channel power meter
        fprintf(powmeter,'EMUL %s;*OPC',upper(emul));
    case 'hp438a' % Emulate HP438A double channel power meter
        fprintf(powmeter,'EMUL %s;*OPC',upper(emul));
    case 'ml4803' % Emulate Anritsu ML4803 power meter
        fprintf(powmeter,'EMUL %s;*OPC',upper(emul));
    otherwise % Operate in default, native mode for Anritsu ML2438A
        fprintf(powmeter,'EMUL ML24XX;*OPC');
end
end
```