BEHAVIOURAL MODEL DERIVATION FROM SWEPT-POWER LOAD-PULL DATA

Reese Moore

Bachelor of Engineering
Electronic Engineering Major

Department of Electronic Engineering
Macquarie University

November 7, 2016

Supervisor: Professor Anthony Parker

ACKNOWLEDGMENTS

I would like to acknowledge Tony Fattorini and all the staff at MACOM in the North Sydney offices, for allowing me their time and resources in the completion of this thesis project. I would also like to acknowledge supervising academic, Professor Anthony Parker for his guidance and knowledge in the completion of this project.

STATEMENT OF CANDIDATE

I, Reese Moore, declare that this report, submitted as part of the requirement for the award of Bachelor of Engineering in the Department of Electronic Engineering, Macquarie University, is entirely my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualification or assessment an any academic institution.

Student's Name: Reese Moore

Student's Signature: RMoore

Date: 7/11/2016

ABSTRACT

Behavioural modelling is a method of describing an electronic circuit or device, without any knowledge of the internal workings of the device. This means that when placed in simulation software, the output signal of the behavioural model should be determined from the input signal. This thesis presents a method of producing a behavioural model from load pull simulations in AWR, and verifies the correctness of the behavioural model. This is done by applying the method to a UMS_GH25_TZ1S_8S150 transistor model and comparing output measurement results of the behavioural model and transistor model. It is shown that the behavioural model has no error in AM-AM or AM-PM simulations, however there are varying degrees of accuracy in IM3 based on load impedances applied to the model. Additionally, designing an amplifier circuit from behavioural models produced by the behavioural modelling process is briefly investigated in terms of fundamental and IM3 output power levels. It is shown for a two stage amplifier design a gain of 15.16dB is obtained and a three stage amplifier has a gain of 18.83dB. The results presented show that the potential of using the designed behavioural modelling process in amplifier design, however future research into the design process is needed before full implementation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electronic circuits are becoming increasingly more complex and the ability to analyse them is becoming more and more difficult. Even with computer simulation tools, such as SPICE, the amount of computing resources needed to simulate these circuits is becoming too great for the average computing system. For example a microprocessor can be built from millions of transistors and simulating the system on an average electronic engineers computer is simply not possible, due to the large amount of non-linear equations that need to be solved simultaneously [4]. Therefore a new method of simulating a large scale system, such as a microprocessor, is needed.

Behavioural models are currently being heavily investigated by engineers as a possible solution to the simulation speed problems described above. A behavioural model is an abstracted version of an electronic device or circuit where the output can be calculated by applying a mathematical function to the input. This means that none of the inner working of a device are needed to be simulated and hence can lead to a faster simulation speed [4] [7]. For example in the microprocessor, none of the transistors would need to be simulated and the output of the device can be found by applying a mathematical function to the input of the device.

Whilst behavioural models are showing promising signs of being of use in large system simulations, there are currently many problems in producing a behavioural model of a device or circuit. Due to the nonlinear nature of many electronic devices and circuits, there is no easy method for producing a behavioural model. This has led to many different methods being available to be applied to produce a behavioural model, with all methods having pros and cons. Another problem is that of the accuracy of the behavioural model [4]. Whilst faster simulations are better for designers, the loss of accuracy is a concern, and hence a balance is needed between the two. These two problems form the basis of this thesis.

1.1 Synopsis

MACOM Technology Solutions are industry partners in this Macquarie University thesis project. The project was initially proposed by Senior Principal Engineer, Tony Fattorini from MACOMs Sydney Design Centre. Ultimately this project may potentially be used by MACOM in future work, however using load pull information to produce a behavioural model is currently of limited knowledge to MACOM, and hence forms the basis of this thesis.

Producing a behavioural model of an electronic device requires a large amount of data, such that it is easy to generate a mathematical function that correctly describes the relationship between the input and output of a device. There are many ways to obtain this data, however for this thesis it is proposed that load pulling a device, under many different swept conditions should be able to produce the necessary data. To do this it is proposed that the use of National Instruments AWR software be used to load pull as the software already contains capabilities to do so. This load pull information should be able to be used to construct a behavioural model of an electronic device of circuit.

It should be noted that this thesis is not about creating a behavioural model of a single electronic circuit or device. This thesis is more concerned with producing a process that can take load pull information of any electronic device or circuit and produce a behavioural model for it. The validity of the produced behavioural model is also investigated in this thesis project. Finally one potential application of this process is shown in the form of designing a 2 and 3 stage amplifier, purely from behavioural models. For the purposes of example and validation, an already constructed transistor by MACOM, the UMS_GH25_TZ1S_8S150 will be used. It is assumed that this transistor has been already verified beforehand by MACOM and hence no checking of the actual transistor model will be performed in this thesis.

The main goals of this thesis can be summarised as:

1. Develop a process that can easily extract load pull data and produce a data file that can be imported into AWR for use in the Nonlinear File Based Behavioural Model (NL_F).

2. Compare the results of the Nonlinear File Based Behavioural Model to a transistor model in AWR in terms of AM-AM, AM-PM and IM3.

3. Use the Nonlinear File Based Behavioural Model to design a simple two stage and three stage amplifier system.

1.2 Scope

As mentioned this project is partnered with MACOM Technology Solutions and may be used in their future work and therefore needs to be limited in scope. For the possibility of MACOM using this project in future work all the tools and processes that are needed for the design and testing of the behavioural model need to be readily available to MACOM. Hence, it is therefore seen that this project only uses AWR software and MATLAB software as these are available for use by MACOM.

MATLAB was chosen as the scripting editor for this thesis due to previous familiarity with the program. Whilst AWR does have a scripting editor of its own, learning how it operates would have been too time consuming for this project, when a simple solution in MATLAB was available.

The physical transistor model used as an example case of the behavioural modelling process was provided by MACOM simply due to time and inexperience in designing transistor devices. Also this project is focused on the development of a process for behavioural models, not transistor design and hence a provided model is acceptable to be used. It should be noted that because of this acceptance of the physical model of the transistor via software, no real world measurements are made to compare with the behavioural model.

It is important to note that this thesis project does not propose any new polynomial based techniques for fitting a mathematical function to data points, and simply uses already usable software tools in AWR for these purposes and the building of behavioural models.

1.3 Outcomes

The intended outcomes of this project are to show that the developed behavioural modelling process can be used to accurately model an electronic device. This will be shown through a comparison between the physical model of the UMS_GH25_TZ1S_8S150 and the behavioural model. The main properties that will be looked at are AM-AM measurements, AM-PM measurements and IM3 measurements for a swept input power and a number of different load impedances. Following this the ability to plot advanced contour mapping of the behavioural model will be shown, to demonstrate how the model could potentially be used in the design process. In particular the Error Vector Magnitude (EVM) of the transistor will be plotted.

One potential application that MACOM has expressed interest in using behavioural models for, is in the design of an amplifier. A simple amplifier design is shown and the attributes of the amplifier are analysed in terms of the output power levels and the intermodulation output power levels.

1.4 Originality

Behavioural modelling for electronic circuits is not a new area of research, with many engineers conducting research into this area for decades [4]. The idea of using load pull to create a behavioural model is also not an entirely new concept either, however only with advances in computer technology and software has it been recently achievable. There are two main examples where load pull has been used to create a behavioural model that are relevant to this project. Liu et al. in [8] successfully showed that a load pull based behavioural model is producible for AM-AM and AM-PM measurements and Marbell et al. in [9] showed how a load pull based behavioural model can be used to design a power amplifier.

Whilst these papers are somewhat related to this thesis, this thesis is more focused on the development of a process that should be able to characterise any electronic device.

The two papers mentioned above only show that load pulling a device can produce a behavioural model and do not describe an automated process for carrying this out. The papers also do not go into detail about harmonic disturbances or other factors such as EVM.

The MATLAB scripts that were developed for this project were designed from scratch and do not take from other sources. The AWR tools used in the project, such as load pull and NL_F are available as part of AWR software packages and hence not designed specifically for this project. However the testbenchs and amplifier design used in this project were solely designed from scratch and used in this project to simply test the behavioural model.

Chapter 2

Background and Related Work

This chapter presents important background information for this thesis project as well as critically evaluating a number of related sources. The two most important aspects of this project are load pull and behavioural models. The information presented here is therefore directly related to these concepts and will be used in later chapters.

2.1 Load Pull Measurements

Load pulling a device is a method traditionally used for finding the optimum matching circuits of a device. This is done by taking a Device Under Test (DUT) and presenting the device with a wide range of impedances at either the input, output or both. Whilst this is done performance parameters such as output power magnitude, output power phase or intermodulation distortion of the device can be measured [10].

From these results an optimal impedance can be found by finding the appropriate performance parameter measurement and looking at the corresponding impedance when that measurement was taken. For example, suppose an amplifier were to be designed that only took a single tone input of 25dBm and all that was needed was the highest fundamental output power level. If this device were to be load pulled against a large range of impedances, a table could be constructed of output power levels and the corresponding impedance values. Finding the highest output power levels in the table indicate the optimum impedance values to design a matching circuit for.

Whilst the above mentioned example is one potential use for load pull, the process can also be enhanced by adding other swept parameters, aside from the impedance values. For example input power levels can typically also be swept alongside the impedance values, to gain a greater understanding of the device. This is particularly useful in the design process as it is unlikely that an amplifier would have an input power level set to a constant, such as in the example mentioned above. Adding this extra swept parameter however adds an added degree of complication, as it would be unlikely that all input power levels have the same optimal impedance value. Therefore a trade-off is typically needed to find a matching impedance that whilst not optimal, is the most appropriate for the input power levels.

Adding more swept parameters to a load pull measurement can also slow down the load pull process. Depending on the urgency that is required of the load pull measurement data, adding more swept parameters may be costly to a project. Hence it usually requires very strict discretion as to the parameters that are going to be swept. This means, for example, that whilst parameters such as frequency and bias voltages might be swept, if the end result of the measurement data does not require these to be swept, they are usually omitted [10].

2.1.1 Load Pull Measurement Techniques

Load pull measurements can be performed either in real life via sophisticated load pull machines or in simulation software. Mircowave101 in [10] give a reasonable explanation of how load pull is performed by a variety of different types of load pull techniques that can be employed in real world measurement situations. These include: vector-receiver load pull, active open loop load pull, hybrid active load pull and mixed-signal active load pull. However, as this thesis only uses simulated load pull none of these methods shall be discussed in detail.

This thesis uses AWR's simulation software to perform the necessary load pull measurements. AWR has an inbuilt load pull script that can be used to create an A/B waves data file that can be used in AWR to find the optimal impedance points. The script has options of supporting a variety of swept parameters depending on the user requirements.

2.1.2 Reflection Coefficient at the Load and Load Impedance Relation

Due to impedance mismatching of a device and load, there is some voltage reflections that occur at the source and the load. One way this is characterised is through S-parameters, partially $S_{1,1}$ and $S_{2,2}$ for the input and output loads respectively [11]. Whilst these are useful in design situations, this thesis is more interested in the reflection coefficients Γ_L which is related to $S_{2,2}$ via the equation

$$S_{2,2} = 20\log(\Gamma_L) \tag{2.1}$$

[6]. It should noted that the $S_{1,1}$ parameter can be derived from the same equation with Γ_L replaced by Γ_S. Figure 2.1 shows the relation between Γ and a circuit with source and load impedances.

The reason this thesis will look at the reflection coefficient over the S-parameters is because the reflection coefficient for the load can be found through the equation

$$\Gamma_L = \frac{Z_L - Z_0}{Z_L + Z_0} \tag{2.2}$$

where Z_L is the load impedance, either real or complex, and Z_0 is the characteristic impedance of the transmission line [2]. It is therefore much easier to equate the reflection coefficient of a load impedance.

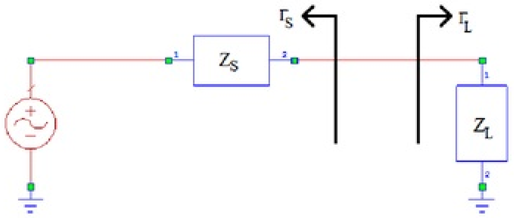


Figure 2.1: Relation between Γ_S and Γ_L and source and load impedances

It should be noted that for this thesis, the term "Gamma Points" refers to the set of Γ_L that were load pulled. The reason for this is because AWR uses the Γ_L terminology rather than the explicit load impedance, Z_L, at a single point on Smith chart. Therefore it is easier to use this terminology to describe the collective set of load impedance points.

2.1.3 Smith Chart Basics

A Smith chart is a special chart where all possible load or source impedances can be plotted inside a circular grid. Mathematically an impedance can be represented in the form $R + jX$ where R is the resistance and has to be greater than or equal to 0, and X is the reactance and ranges from $-\infty < X < \infty$ [10]. Therefore in a Smith chart, the x axis is the resistance and the y axis is the reactance. Starting at the leftmost point on the x axis is the 0 value and the rightmost point being ∞, with the exact centre of the x axis being equal to a resistance of 1. The y axis follows the outer ring of the circle with the top half being the positive j part, hence inductive and the bottom half being the negative j part, hence capacitive [10].

One important aspect of a Smith chart is that the chart is normalised to a specific characteristic impedance, Z_0. This means that to find a specific load impedance from a Smith chart, the value on the Smith chart must be multiplied by the characteristic impedance. For example if the characteristic impedance is equal to 50, the point (1, 0) on the chart is the load impedance of 50Ω and the point (0.35, -0.75) is the load impedance $17.5 - j37.5\Omega$ [10].

Whilst plotting impedances can be important, a more useful tool of a Smith chart is that of plotting contours. Contour plotting is typically used to display load pull measurement data visually, instead of in a table. The contours are generally interpolated data between points on a Smith chart, and hence is difficult to do by hand and only done via software. Figure 2.2 shows an example of two contour plots overlayed with each other. The pink contours relate to the power added efficiency (PAE) of a device, the blue contours relate to the fundamental power at the load and the green crosses indicate all the load impedances that were load pulled. The PAE contours are in steps of 1%, meaning there is a 1% difference in each contour ring, and the fundamental power contours are in steps of 1dBm. This example shows the maximum measurement for PLoad and PAE occur at two different impedance values, $17 + j21\Omega$ and $14.5 + j28.5\Omega$ respectively. Contour plotting can be therefore used to find an impedance point that is a compromise between the two maximums.

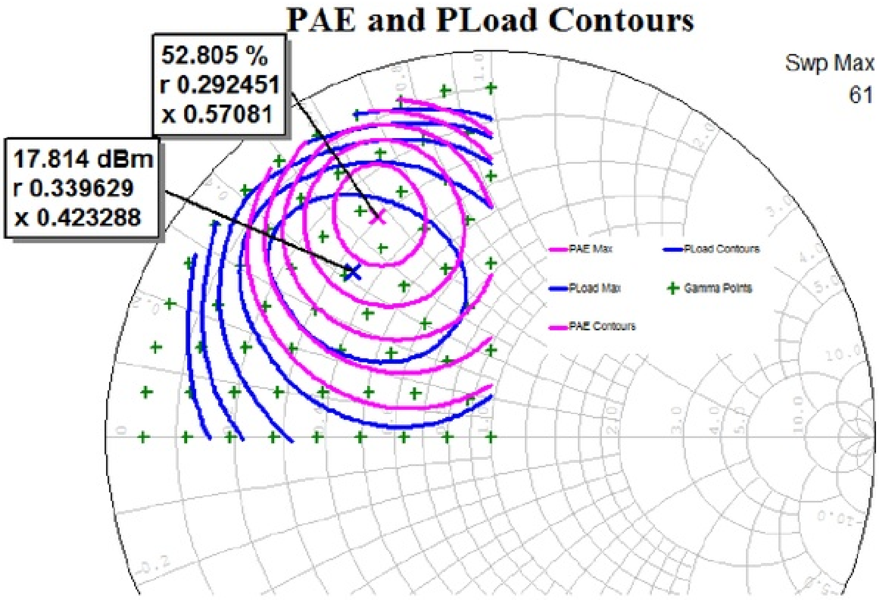


Figure 2.2: Example overlayed contour plot

2.2 Behavioural Models

2.2.1 What is a Behavioural Model?

In computer simulation packages for electronic devices there are typically three types of models that can be used: physical models, equivalent models or behavioural models. Each of these models has their advantages and disadvantages, especially when wanting to be used in the design process [7]. A physical model uses physics based equations and formulas to provide the most complete picture of a single electronic device. However this requires a large amount of computing resources and is very time consuming and hence is rarely ever used in a design of a large scale circuit as each device would require a large amount of time and resources to simulate. An equivalent model is an abstracted version of a physical model and is typically used in the design of small scale circuits [7]. An example of an equivalent model would be the small signal model of a MOSFET. Here the inner workings of a device are still characterised.

A behavioural model is an abstracted version of the equivalent model, and is typically in the form of mathematical expressions that describe an input-output relationship [7]. One of the simplest examples of a behavioural model would be of a single input, single output amplifier, as seen in figure 2.3. Here the amplifier has a constant gain of A and it can therefore be determined that for any input voltage, v_{in}, the output will be Av_{in}. Now whilst the amplifier may be constructed from a number of transistors, these are not simulated in the behavioural model, like they would be in a physical or equivalent model, hence leading to a faster simulation.

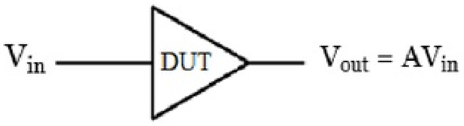


Figure 2.3: Example of a basic amplifier behavioural model

Whilst this faster simulation speed is a great advantage to designers, there is a trade-off for this speed mainly in the form of decreased accuracy. This decreased accuracy mainly comes from parasitic effects such as the frequency response of a device, or saturation effects from too high of an input [4]. For example, in the above mentioned behavioural model of the amplifier there is no limitations on the output equation. This means any input voltage, whether it be larger than the amplifier can handle or outside of its frequency range, is equally amplified. This is unrealistic and hence a behavioural model needs to try and model these parasitic effects whilst keeping a balance between speed and accuracy. This balance is currently an ongoing heavily researched topic for engineers as electronic circuits are getting larger in scale the need for faster and more accurate simulations are needed.

2.2.2 Behavioural Model Generation Process

Currently there is no agreed upon process as to how to best take an electronic device or circuit and produce a behavioural model of it. Ceperic and Boric in [4] attempt to generalise the process, which consists of four main steps.

1. Collecting a large amount of data from measurements on the device
2. Analysing the data in terms of reliability and accuracy
3. Developing a model from the data
4. Evaluating the derived model and revising if needed

These steps are very general, and as the authors admit, the third step of developing a model is the most ambiguous step as there a number of approximation methods that can be used to develop a behavioural model. Overall however this process does seem to be a fairly respectable approach to the behavioural modelling process, as other proposed process such as the one by Ghannouchi et al. in [5] tend to follow the same approach. Therefore this thesis will attempt to develop a system that follows this generalised process to build a behavioural model, such that it can be used by MACOM.

2.3 Approximation Methods

As mentioned one of the most ambiguous steps in developing a behavioural model is the development of the model from acquired data. This process is typically performed by an approximation method. The purpose of the approximation method is to develop the mathematical relationship between the input and the output that is used in the behavioural model. This is simple when the input and output data is linear, however most electronic devices are not linear and hence the process of creating an equation that describes the input-output relation is more difficult.

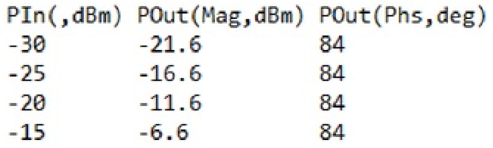
To add to the difficultly of the developing a model stage, is that there are a number of different approximation methods that can be employed. Ghannouchi et al. in [5] accurately detail a variety of different approximation techniques that can be used in the modelling stage, however as the authors explain, and re-enforced by Ceperic and Boric in [4], there is no single approximation method that is guaranteed to work for all data. This inconsistency can lead to a trial and error situation in picking an appropriate approximation method.

Outlined below are a couple of the most popular and emerging approximation methods. It should be noted that these approximation methods for fitting input and output data are for a single gamma point on a Smith chart. This thesis however, wants to attempt to build a working behavioural model for multiple gamma points on a Smith chart. Whilst this could be achieved by applying an appropriate approximation method at each individual gamma point, the amount of coding and automation needed for this is beyond the scope of this thesis and hence AWRs inbuilt behavioural modelling approximation method is used.

2.3.1 AWR Behavioural Models

AWR Design Environment is a powerful software package that encompasses many different software tools, such as Microwave Officce (MWO) and Visual System Simulator (VSS), which allow electronic engineers to perform a wide range of simulations and analysis of electronic components. Within the software there a number of behavioural modelling tools that are employable by a designer to accomplish their goals.

One of these tools is the File Based Nonlinear Behavioural Model (NL_F), which is an automated behavioural model. The NL_F is a 2 port system block implementable in VSS and operates off a text file of input and output data. Figure 2.4 shows an example of what a text file may look like. It can be seen that the headings for the columns are input power magnitude in dBm (Pin(,dBm)), output power magnitude in dBm (Pout(Mag,dBm)), output power phase in degrees (Pout(Phs,deg)). It should be noted that a more advanced behavioural model can be generated by including columns such as second and third order harmonic output power levels, output power levels at intermodulation frequencies or S parameter coefficient values [3].



```
PIn(,dBm) POut(Mag,dBm) POut(Phs,deg)
-30       -21.6         84
-25       -16.6         84
-20       -11.6         84
-15       -6.6          84
```

Figure 2.4: Example text file used by file based nonlinear behavioural model

From this text file seen in 2.4, the software knows that if the system block sees a -30dBm power level at the input, the output needs to be -21.6dBm with phase of 84°. One advantage of this model in VSS is its ability to calculate the output value of an input, even if the input is not explicitly listed. To do this there are three main methods that can be applied: a small signal polynomial, AM-AM/AM-PM and a large signal polynomial. These methods are used to fit a curve, and hence a mathematical expression, to the input and output realtion.

One particular feature of the NL_F model is the ability to break the text file up into blocks of information. This is done by introducing a "VAR VariableName = Value header above the column headers, where VariableName is a name and value is a numeric value [3]. The NL_F model in AWR then has a setting for that variable that can be set by the user, and the model will only look at the information related to that variable. Therefore, for this thesis, the variable setting can be used to break all the load pull information down into individual load impedance blocks. However one major downside to this is that only the specific values of the variable can be used, and if the user inputs a value for the variable that is not present in the data file, AWR simple rounds that value to the closest one available [3]. For example suppose a text file has block information under variables of values 1, 2 and 3. If the user inputted a value for the variable of 2.5, AWR would simply round it up to 3 and no interpolation between these data sets is performed.

Small Signal Polynomial

One of the methods AWR uses to fit a function to a set of data is to use a 5th order polynomial with an operating point. The operating point is generally determined to be the input power to the model, and hence will change if the input power is swept. The coefficients of the polynomial will therefore change as the operating point changes. The user manual for the NL_F in [3] gives little information into how these coefficients are calculated and this is reasonable considering AWR is not an open-source software package. The guide however does provide an example on what the output will be for a selected operating point from an example data set and the results do seem reasonable to the example data set, and therefore does somewhat show that the small signal polynomial is reliable.

AM-AM/AM-PM Model

The AM-AM/AM-PM Model utilises a look up table of dB values of input and output power or voltage, and simply interpolates any values between two points. The table is organsied by AWR software such that the input power is increasing from the minimum to maximum values, if the user had not input the text file in that order. From this any instantaneous input power level that falls between two points can be interpolated to find the corresponding output power level and phase. The user guide, [3], details the equations that are used to calculated the instantaneous output power and phase from an instantaneous input power level, $P_{InstInp,dB}$, falling between the points $P_{Inp,dB}[2]$ and $P_{Inp,dB}[3]$.

$$P_{InstOut,dB} = P_{Out,dB}[2] + \frac{P_{InstInp,dB} - P_{Inp,dB}[2]}{P_{Out,dB}[3] - P_{Out,dB}[2]} \tag{2.3}$$

$$\Delta\theta_{InstOut} = \Delta\theta_{Out}[2] + \frac{P_{InstInp,dB} - P_{Inp,dB}[2]}{\Delta\theta_{Out}[3] - \Delta\theta_{Out}[2]} \tag{2.4}$$

The model also has processes of calculating the output power and phase for when the input signal is lower or higher than any value in the text file [3].

Large Signal Polynomial

Much like the small signal polynomial, there is little information given about how the coefficients of the 5th order large signal polynomial are calculated. It should be heavily noted that the large signal polynomial is still in preliminary stages of development in AWR, and hence as the guide in [3] states should not be 100% trusted at this stage. This thesis however will use this method, alongside the before mentioned to provide reference to the correct working of the behavioural model.

2.3.2 Volterra Methods

Volterra methods are used to describe nonlinear effects a device, in particular the intermodulation effects. Sansen in [12] thoroughly describes how these methods are implemented in the description of intermodulation effects for a range of transistors. Whilst simplified to the transistor level, these methods are easily implementable to the system level. One of the ways that Volterra methods can be implemented is when a weakly nonlinear device has an input $u(t)$ and output $y(t)$. As Sansen [12] states, the output can be represented as a power series of the input. I.e.

$$y(t) = a_0 + a_1u(t) + a_2u(t)^2 + a_3u(t)^3 + \dots \tag{2.5}$$

where a_0 is the dc component and a_1, a_2, ... are constants. Following from this, if the input is a sinusoidal with an amplitude of A and angular frequency ω, in the form $A\cos(\omega t)$, the output becomes

$$y(t) = a_0 + a_1 A\cos(\omega t) + a_2 A\cos(\omega t)^2 + a_3 A\cos(\omega t)^3 + \dots \tag{2.6}$$

and with simplification using trigonometric identities

$$y(t) = (a_0 + \frac{1}{2}A^2) + (a_1 A + \frac{3}{4}a_3 A^3)cos(\omega t) + \frac{1}{2}a_2 A^2 cos(2\omega t) + \frac{1}{4}a_3 A^3 cos(3\omega t) + \dots \tag{2.7}$$

Using some mathematical analysis of the collected data, the constants for a_n can be calculated and therefore from this equation a behavioural model should be derived. However as Ceperic and Boris in [4] and Sansen [12] state, this method is limited to only weakly nonlinear devices and any device that exhibits a strongly nonlinear relationship between it's input and output, cannot be accurately described by this method.

2.3.3 Artificial Neural Networks

Artificial Neural Networks are a method of machine learning and as Ceperic and Boris state in [4], were first successfully shown to be implementable in developing a behavioural model of an electronic device in Litovski et al. in "MOS transistor modelling using neural networks" in 1998. Andrejevi and Litovski in [1] picked up on Litovski's original work and showed that neural networks can be used to implement a behavioural model that works in the time domain. Whilst somewhat dated by today's standards, it still serves as an example that artificial neural networks are a feasible option when behavioural modelling electronic circuits.

The downside to artificial neural networks is that they are a machine learning procedure. This means that they are not easily implementable in standard computing machines and usually require a fair bit of computing resources to correctly operate at an fast speed. Therefore these networks are far beyond the reach of this thesis and not necassarily implementable in MACOM's current capacity and hence not be used.

2.3.4 Other Methods

It is impractical to list all the potential approximation methods that are usable in behavioural modelling. Ghannouchi et al. in [5] gives a decent amount of in depth explanation of many different methods, including Volterra series and artificial neural networks. However one method that is not given much of an explanation about is support vector machines. Ceperic and Boric in [4] details how these learning processes work, but also give reason to believe that the future in behavioural modelling is in these machines. It is only with further research however that it can be found that Ceperic and Boric were the first to use support vector machines for behavioural modelling and therefore have a vested interest in this particular method.

One approximation method that is worth mentioning is the approximation method that is fully described in Turlington's work seen in [13]. This method is a polynomial that has a high degree of accuracy in plotting a set of data points to a mathematical function. However whilst this method is supported by Ghannouchi et al. in [5] as an appropriate way to develop a behavioural model, the method is far too complex for this project. The reason behind this is because that whilst it may be able to find an appropriate mathematical function between an input and output for 1 load impedance, this project is wanting to develop behavioural models for a large number of load impedances and hence the time required to do this is far too costly. Whilst this method is not appropriate for this project, a potential use for this method may be of use if appropriate coding could be developed to find the coefficients easily.

Chapter 3

AWR Tools for Load Pull and Behavioural Modelling

This chapter details how load pull measurements were conducted, how MATLAB scripts were developed to extract the necessary data from the load pull information and the designed testbenchs for the behavioural models in AWR. For this thesis there are two main situations that are investigated:

Single Tone Input: The input to an electronic device or circuit is a single tone sinusoidal waveform.

Two Tone Input: The input to an electronic device or circuit is a two tone waveform.

The single tone case is used for measuring the AM-AM and AM-PM measurements and the two tone case is used for measuring the IM3 and EVM of a device or circuit. Both of these cases follow a similar procedure and setup as outlined in this chapter.

3.1 AWR Load Pull Script

AWR software has an inbuilt load pull script that is used in this thesis to perform the necessary load pull measurements on a device or circuit. The script provides a necessary template where swept parameters and conditions can be set and the DUT can easily be interchanged. Examples of settable parameters include: swept single tone or swept two tone input signal, swept bias conditions and swept frequency conditions.

Once these parameters are set, the most important part that needs to be chosen is the gamma points that are to be load pulled at. The script provides an easy tool to distribute gamma points evenly over a Smith chart and therefore provides an easy way to load pull over the entire Smith chart without manually inputting Γ_L values. It should be noted that the load pull script can also be used for Γ_S values as well, however this ability is not used for this particular behavioural modelling instance.

Using the load pull script, a load pull measurement was made for the UMS_GH25_TZ1S_8S150 transistor model. The model was load pulled against the gamma points seen in figure 3.1, for both a single tone input and a two tone input under the set parameters of gate voltage $V_G = -2$V and drain voltage $V_D = 20V$, as seen in figure 3.2. The single tone input had a swept power range of -30dBm to 50dBm in steps of 5, at a frequency of 22GHz. The two tone input had a swept power range of -30dBm to 30dBm in steps of 10, at frequencies of 22GHz and 22.1GHz. It is important to note that for this load pull measurement the source impedance is set to 50Ω, which is not correctly matched to the transistor impedance.

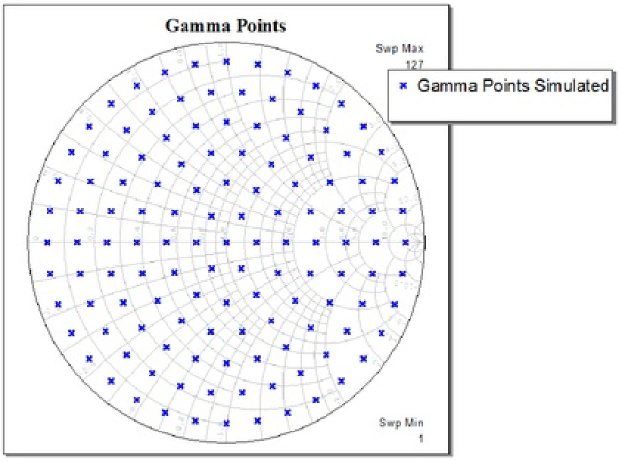


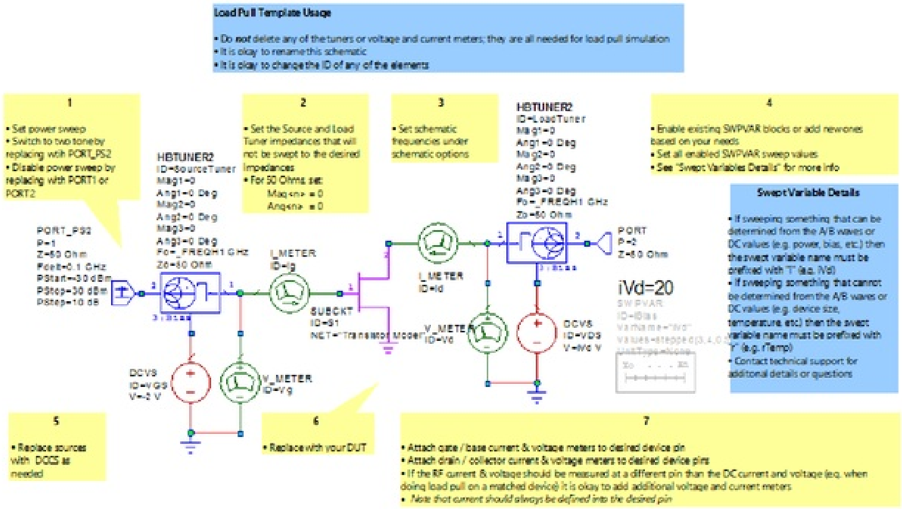
Figure 3.1: Gamma points measured during load pull

The output of the load pull measurements is a text based file of A/B waves that can be easily used by AWR to plot measurements, however cannot be used in the NL_F behavioural model. Therefore a method was needed to extract the AM-AM, AM-PM and IM3 information from this file and create a text file as seen in figure 2.4. To do this MATLAB scripts were needed to be developed to automate the process, as doing it by hand is impractical.

3.2 MATLAB Scripting

To overcome the problem of the load pull data not being in the correct format, MATLAB scripts were developed. The scripts were also needed to be developed simply due to the large amount of information that was created by AWRs load pull script not being able to be extracted by hand. For example, in the single tone case there are 127 different Γ_L values and at each of these points there are 17 different input power levels, meaning the data file for the NL_F would need to have 2159 rows which simply is not possible by hand.

For this thesis two main MATLAB scripts were developed: one for the single tone measurements and one for the two tone measurements. Both of these scripts work on the same principles, except the two tone script takes extra input variables and produces a text file with an additional column for the IM3 levels. Each of the scripts takes inputs



Load Pull Template Usage

- Do *not* delete any of the tuners or voltage and current meters; they are all needed for load pull simulation
- It is okay to rename this schematic
- It is okay to change the ID of any of the elements

1

- Set power sweep
- Switch to two tone by replacing with PORT_PS2
- Disable power sweep by replacing with PORT1 or PORT2

2

- Set the Source and Load Tuner impedances that will not be swept to the desired impedances
- For 50 Ohms set:
  Mag<n> = 0
  Ang<n> = 0

3

- Set schematic frequencies under schematic options

4

- Enable existing SWPVAR blocks or add new ones based on your needs
- Set all enabled SWPVAR sweep values
- See "Swept Variables Details" for more info

Swept Variable Details

- If sweeping something that can be determined from the A/B waves or DC values (e.g. power, bias, etc.) then the swept variable name must be prefixed with "i" (e.g. iVd)
- If sweeping something that cannot be determined from the A/B waves or DC values (e.g. device size, temperature, etc.) then the swept variable name must be prefixed with "r" (e.g. rTemp)
- Contact technical support for additonal details or questions

5

- Replace sources with DCCS as needed

6

- Replace with your DUT

7

- Attach gate / base current & voltage meters to desired device pin
- Attach drain / collector current & voltage meters to desired device pins
- If the RF current & voltage should be measured at a different pin than the DC current and voltage (e.g. when doing load pull on a matched device) it is okay to add additional voltage and current meters
- *Note that current should always be defined into the desired pin*

Figure 3.2: Load pull template used to perform single and two tone measurements

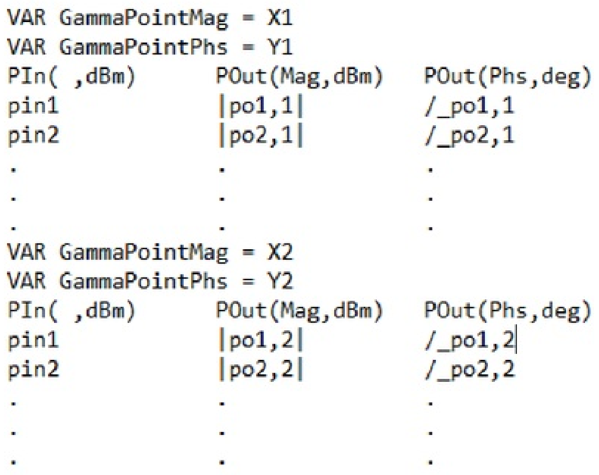
from tables generated by AWR as using the raw A/B waves data is simply impractical and exploiting AWRs ability to generate tables from A/B waves is a much easier method.

The desired output of each of the scripts is a text based file, with the desired format for the single tone script seen in figure 3.3. The two tone script will have a similar format, however an extra column will be needed for the IM3 level measurements. It can be seen that each gamma point is broken down into a separate block and defined by the magnitude and phase of the gamma point. These identifiers make every gamma point unique and can be used by the NL_F behavioural model to simulate a corresponding load impedance.

3.2.1 Single Tone Script

The Single Tone MATLAB script is designed to take various input tables, produced by AWR, and output a single text file in the format seen in figure 3.3. The input tables for this script are: one large table of fundamental harmonic (22GHz) power magnitude at the output of the transistor with input power levels being the columns and the gamma point index being the rows, one large table of fundamental harmonic (22GHz) power phase at the output of the transistor with input power levels being the columns and the gamma point index being the rows, a column vector of input power levels, a column vector of the magnitude of each gamma point and a column vector of the phase of each gamma point.

Firstly the script creates a matrix of the input power data and the output power data. The matrix is of size number of input power levels multiplied by number of gamma points

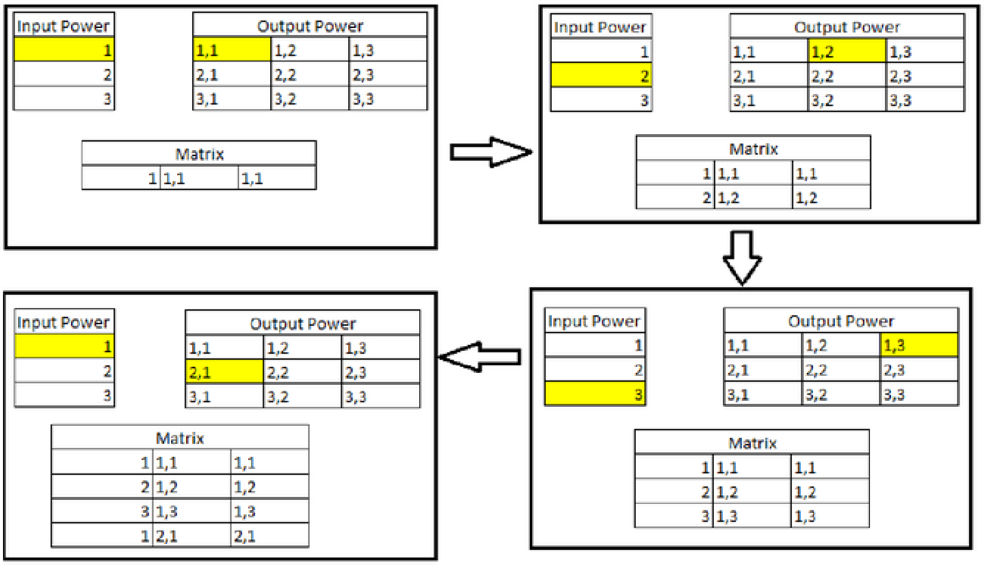


```
VAR GammaPointMag = X1
VAR GammaPointPhs = Y1
PIn( ,dBm)       POut(Mag,dBm)    POut(Phs,deg)
pin1             |po1,1|          /_po1,1
pin2             |po2,1|          /_po2,1
.                .                .
.                .                .
.                .                .
VAR GammaPointMag = X2
VAR GammaPointPhs = Y2
PIn( ,dBm)       POut(Mag,dBm)    POut(Phs,deg)
pin1             |po1,2|          /_po1,2|
pin2             |po2,2|          /_po2,2
.                .                .
.                .                .
.                .                .
```

Figure 3.3: Desired output format of text file from single tone MATLAB script

× 3. Therefore in this case it will be 2159 × 3. The script uses the properties that the dimensions of the output power magnitude and phase tables should be the same, as well as the number of columns in these tables will be equal to the number of rows in the input power vector. A simple example of the filling of the matrix is demonstrated through figure 3.4. It can be seen that the highlighted boxes represent the information to place in the matrix. It can be seen that while the input vector has it's data entered into the matrix the row information is placed into the matrix. Once the column vector is finished its loop it resets, whilst the output table simply moves down a row. This whole process carries out until all the values of the output table are completed. The following code performs this task.

```matlab
matrix=zeros(size(InputPower,1)*size(Mag,1),3);
CIP=1; %control input power row
COPR=1; %controls output power row
COPC=1; %controls output power col
multiplier=0;
index=1;

for n=1:size(matrix,1) %runs through rows of a matrix
    matrix(n,1)=InputPower(CIP,1);
    matrix(n,2)=OutputPowerMag(COPR,COPC);
    matrix(n,3)=OutputPowerPhs(COPR,COPC);
    index=index+1;
    if CIP<size(InputPower,1)
        CIP=CIP+1; %increase the input power to next row
        COPC=COPC+1; %move output col over 1
    else
        CIP=1;
        COPR=COPR+1;
        if index>size(InputPower,1)*size(Mag,1)
            index=1;
            multiplier=multiplier+1;
            COPR=1;
        end
        COPC=1+size(InputPower,1)*multiplier;
    end
end
```



Input Power
1
2
3

Output Power		
1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Matrix		
1	1,1	1,1

Input Power
1
2
3

Output Power		
1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Matrix		
1	1,1	1,1
2	1,2	1,2

Input Power
1
2
3

Output Power		
1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Matrix		
1	1,1	1,1
2	1,2	1,2
3	1,3	1,3

Input Power
1
2
3

Output Power		
1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

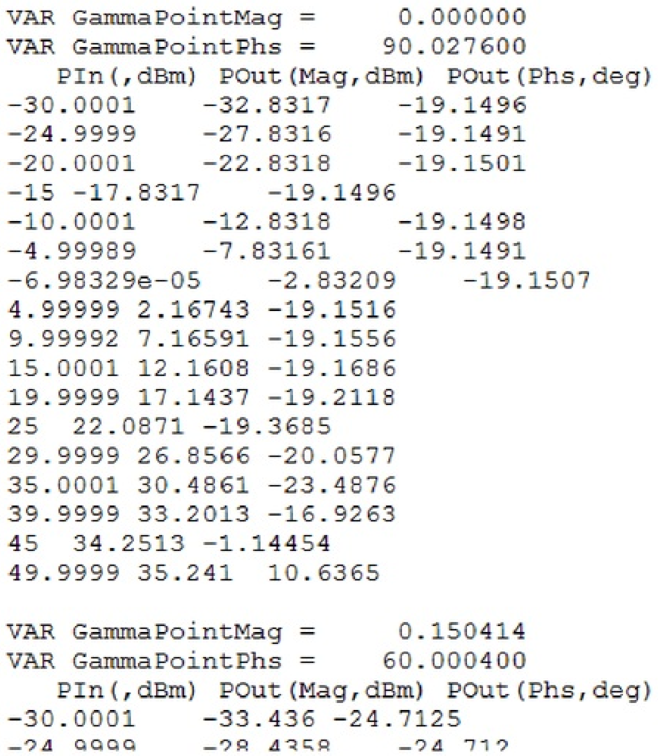
Matrix		
1	1,1	1,1
2	1,2	1,2
3	1,3	1,3
1	2,1	2,1

Figure 3.4: Simple example showing the filling of the matrix

The final part if this script is to create the text file in the necessary format. This is firstly done by printing the gamma point magnitude and phase information and then the headings for each of the columns. Following this, for the number of input power levels there are (in this case 17) that number of rows of the matrix are printed to the text file. This process then repeats, meaning gamma point information is then printed followed by headings for the remaining gamma points. The following code performs this task.

```
CGP=1; %control gamma point rows
outFile=fopen('ExtractedData.txt','w');
GammaPointsMag = 'VAR GammaPointMag = %12f';
GammaPointsPhs= 'VAR GammaPointPhs = %12f';
fprintf(outFile, GammaPointsMag, Mag(CGP,1));
fprintf(outFile, '\r\n');
fprintf(outFile, GammaPointsPhs, Phs(CGP,1));
fprintf(outFile, '\r\n');
fprintf(outFile,'%12s %12s %12s \r\n','PIn(,dBm)',
'POut(Mag,dBm)','POut(Phs,deg)');
f=1;
for k = 1:size(matrix,1)
    fprintf(outFile,'%g\t',matrix(k,:));
    fprintf(outFile,'\r\n');
    f=f+1;
    if f>size(InputPower,1)
        f=1;
        CGP=CGP+1; %move to next Gamma Point
        if CGP <= size(Mag,1)
            fprintf(outFile,'\r\n');
            fprintf(outFile, GammaPointsMag, Mag(CGP,1));
            fprintf(outFile, '\r\n');
            fprintf(outFile, GammaPointsPhs, Phs(CGP,1));
            fprintf(outFile, '\r\n');
            fprintf(outFile,'%12s %12s %12s \r\n','PIn(,dBm)',
                        'POut(Mag,dBm)','POut(Phs,deg)');
        end

    end
end
fclose(outFile);
```

It should be noted that this script has been designed to operate for any load pull information and not only specifically for this case. This is done such that MACOM can use the script for any future work that may require behavioural modelling via this method. A snippet of the final output text file can be seen in figure 3.5, and the entire code is provided as reference in Appendix A

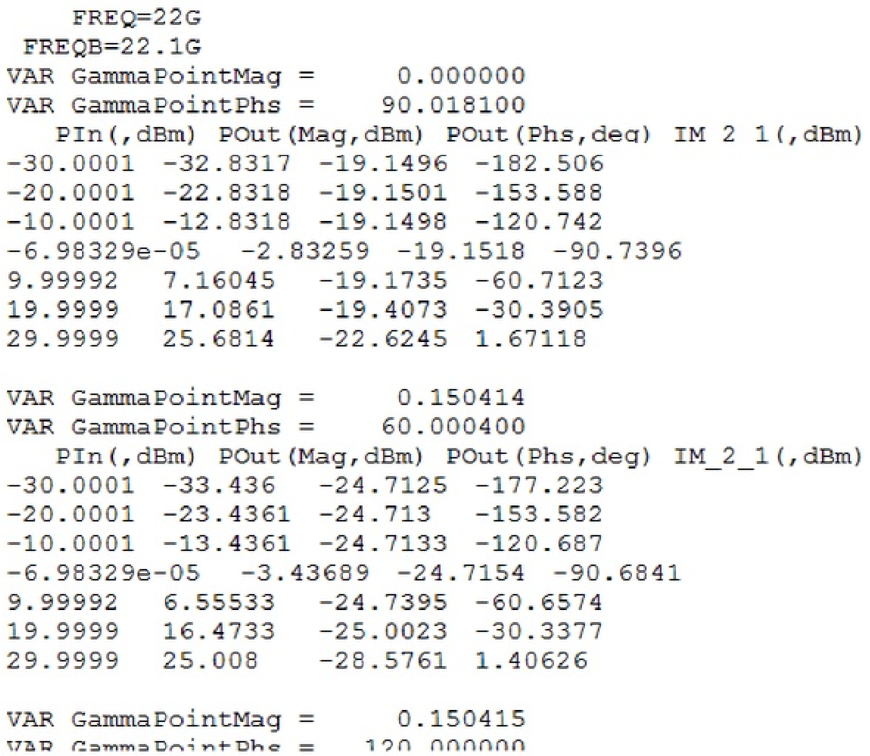


```
VAR GammaPointMag =      0.000000
VAR GammaPointPhs =     90.027600
   PIn(,dBm) POut(Mag,dBm) POut(Phs,deg)
-30.0001    -32.8317    -19.1496
-24.9999    -27.8316    -19.1491
-20.0001    -22.8318    -19.1501
-15 -17.8317    -19.1496
-10.0001    -12.8318    -19.1498
-4.99989    -7.83161    -19.1491
-6.98329e-05    -2.83209    -19.1507
4.99999 2.16743 -19.1516
9.99992 7.16591 -19.1556
15.0001 12.1608 -19.1686
19.9999 17.1437 -19.2118
25  22.0871 -19.3685
29.9999 26.8566 -20.0577
35.0001 30.4861 -23.4876
39.9999 33.2013 -16.9263
45  34.2513 -1.14454
49.9999 35.241  10.6365

VAR GammaPointMag =      0.150414
VAR GammaPointPhs =     60.000400
   PIn(,dBm) POut(Mag,dBm) POut(Phs,deg)
-30.0001    -33.436 -24.7125
-24.9999    -28.4358    -24.712
```

Figure 3.5: Snippet of the output file for the single tone script

3.2.2 Two Tone Script

The developed two tone script works in very much the same way as the single tone script, however takes one more input and has one more column in its output file. The extra input table that is needed for the two tone script is the output power in the third order harmonic. This table will have the same dimensions as the fundamental output power tables and hence can easily be added to the first part of the script as an extra column of the matrix. This then simplifies the problem as no drastically new script is needed to be developed for this particular case. However one important difference between the two output files is that the two tone script displays the frequencies of the two tones at the top of the text file. The entire code for the two tone script is provided in as reference in Appendix A, with a snippet of the output text file seen in figure 3.6.



```
      FREQ=22G
   FREQB=22.1G
  VAR GammaPointMag =      0.000000
  VAR GammaPointPhs =     90.018100
     PIn(,dBm) POut(Mag,dBm) POut(Phs,deq) IM 2 1(,dBm)
  -30.0001  -32.8317  -19.1496  -182.506
  -20.0001  -22.8318  -19.1501  -153.588
  -10.0001  -12.8318  -19.1498  -120.742
  -6.98329e-05   -2.83259  -19.1518  -90.7396
  9.99992   7.16045   -19.1735  -60.7123
  19.9999   17.0861   -19.4073  -30.3905
  29.9999   25.6814   -22.6245  1.67118

  VAR GammaPointMag =      0.150414
  VAR GammaPointPhs =     60.000400
     PIn(,dBm) POut(Mag,dBm) POut(Phs,deg) IM_2_1(,dBm)
  -30.0001  -33.436   -24.7125  -177.223
  -20.0001  -23.4361  -24.713   -153.582
  -10.0001  -13.4361  -24.7133  -120.687
  -6.98329e-05   -3.43689  -24.7154  -90.6841
  9.99992   6.55533   -24.7395  -60.6574
  19.9999   16.4733   -25.0023  -30.3377
  29.9999   25.008    -28.5761  1.40626

  VAR GammaPointMag =      0.150415
  VAR GammaPointPhs =    120.000000
```

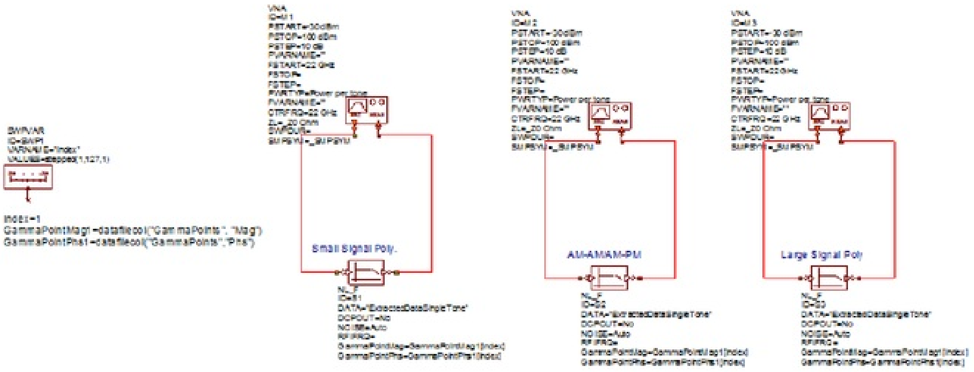
Figure 3.6: Snippet of the output file for the two tone script

3.3 AWR Behavioural Model Testbenchs

Following the successful implementation of the MATLAB scripts, the validity of the behavioural models had to be tested in AWR. To do this several testbenchs were constructed in VSS as well as MWO and several measurements were taken for the behavioural model, as well as the transistor model such that easy comparison can be performed.

3.3.1 Single Tone Testbench

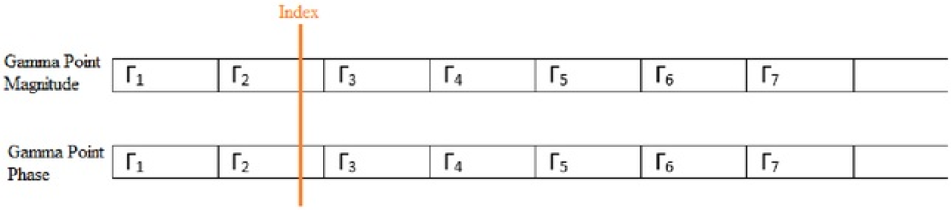
Figure 3.7 shows the testbench used to test the validity of the NL_F behavioural model, for all of the possible interpolation methods, for a single tone input. It can be seen that for each interpolation method the NL_F behavioural model is connected to a Vector Network Analyser (VNA), which provides the input signal, as well as measures the output signal. The VNA is set to 22GHz and set to sweep input power levels from -30dBm to 100dBm in steps of 10dB. 100dBm is well above the input power levels that were measured during the initial load pull, but will provide a good insight into how the behavioural model operates when there is no data entries in the text file for those levels.



```
SWPVAR
ID=SWP1
VARNAME="Index"
VALUES=stepped(1,127,1)

Index=1
GammaPointMag1=datafilecol("GammaPoints","Mag")
GammaPointPhs1=datafilecol("GammaPoints","Phs")

VNA
ID=M1
PSTART=-30dBm
PSTOP=100 dBm
PSTEP=10 dB
PVARNAME=""
FSTART=22 GHz
FSTOP=
FSTEP=
PWRTYPE=Power per tone
FVARNAME=""
CTRFRQ=22 GHz
ZL=_Z0 Ohm
SWPDUR=
SUPSYM=_SUPSYM

NL_F
ID=S1
DATA="ExtractedDataSingleTone"
DCPOUT=No
NOISE=Auto
RFFRQ=
GammaPointMag=GammaPointMag1[index]
GammaPointPhs=GammaPointPhs1[index]

VNA
ID=M2
PSTART=-30dBm
PSTOP=100 dBm
PSTEP=10 dB
PVARNAME=""
FSTART=22 GHz
FSTOP=
FSTEP=
PWRTYPE=Power per tone
FVARNAME=""
CTRFRQ=22 GHz
ZL=_Z0 Ohm
SWPDUR=
SUPSYM=_SUPSYM

NL_F
ID=S2
DATA="ExtractedDataSingleTone"
DCPOUT=No
NOISE=Auto
RFFRQ=
GammaPointMag=GammaPointMag1[index]
GammaPointPhs=GammaPointPhs1[index]

VNA
ID=M3
PSTART=-30 dBm
PSTOP=100 dBm
PSTEP=10 dB
PVARNAME=""
FSTART=22 GHz
FSTOP=
FSTEP=
PWRTYPE=Power per tone
FVARNAME=""
CTRFRQ=22 GHz
ZL=_Z0 Ohm
SWPDUR=
SUPSYM=_SUPSYM

NL_F
ID=S3
DATA="ExtractedDataSingleTone"
DCPOUT=No
NOISE=Auto
RFFRQ=
GammaPointMag=GammaPointMag1[index]
GammaPointPhs=GammaPointPhs1[index]
```

Figure 3.7: Testbench used to test the interpolation methods of the NL_F for a single tone input

It can also be seen that two vectors are created through the functions “datafilecol()”. These create two vectors, one of the simulated load pull gamma point magnitude and one of simulated load pull gamma point phase. These are important as the NL_F can only take variables that are present in the data file. Therefore by creating two vectors that have the gamma point data from the data file, at the same index, the gamma point that is wanted to be applied to the behavioural model can simply be controlled by an index variable through the use of the tuner tool in AWR. Figure 3.8 shows how this operates. It should be noted that AWR automatically gives an index to the gamma points when load pull is performed and hence the same indexing method is applied here, meaning that the gamma point with index 1 in the load pull measurement is the same as the gamma point with index 1 in these created vectors.

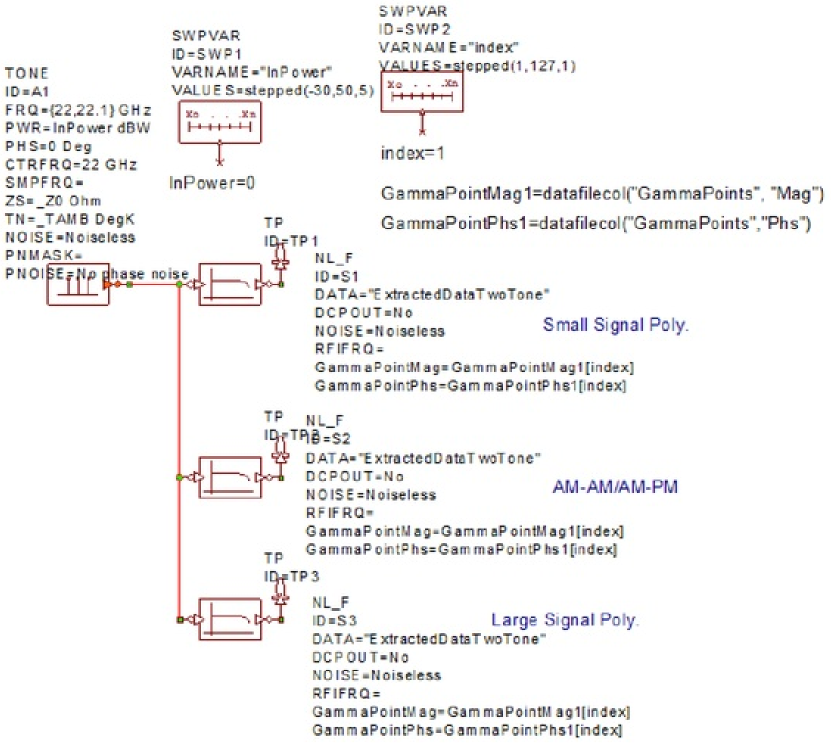


Index
Gamma Point Magnitude
Γ_1 Γ_2 Γ_3 Γ_4 Γ_5 Γ_6 Γ_7
Gamma Point Phase
Γ_1 Γ_2 Γ_3 Γ_4 Γ_5 Γ_6 Γ_7

Figure 3.8: Index keeping track of gamma point vector information

The testbench designed allows the easy measurement of both AM-AM and AM-PM measurements to be conducted. However one problem with behavioural models is that it is difficult to ensure that the model is working correctly for every single gamma point. This is because of the large number of gamma points that were measured in the load pull. However it can be assumed that if a number of gamma points, that are a fair distance apart on the Smith chart, are tested and show working results, all possible points of the behavioural model do operate correctly.

3.3.2 Two Tone Testbench



```
TONE
ID=A1
FRQ={22,22.1} GHz
PWR=InPower dBW
PHS=0 Deg
CTRFRQ=22 GHz
SMPFRQ=
ZS=_Z0 Ohm
TN=_TAMB DegK
NOISE=Noiseless
PNMASK=
PNOISE=No phase noise

SWPVAR
ID=SWP1
VARNAME="InPower"
VALUES=stepped(-30,50,5)
InPower=0

SWPVAR
ID=SWP2
VARNAME="index"
VALUES=stepped(1,127,1)
index=1

GammaPointMag1=datafilecol("GammaPoints", "Mag")
GammaPointPhs1=datafilecol("GammaPoints","Phs")

TP
ID=TP1
NL_F
ID=S1
DATA="ExtractedDataTwoTone"
DCPOUT=No
NOISE=Noiseless
RFIFRQ=
GammaPointMag=GammaPointMag1[index]
GammaPointPhs=GammaPointPhs1[index]

TP
ID=TP2
NL_F
ID=S2
DATA="ExtractedDataTwoTone"
DCPOUT=No
NOISE=Noiseless
RFIFRQ=
GammaPointMag=GammaPointMag1[index]
GammaPointPhs=GammaPointPhs1[index]

TP
ID=TP3
NL_F
ID=S3
DATA="ExtractedDataTwoTone"
DCPOUT=No
NOISE=Noiseless
RFIFRQ=
GammaPointMag=GammaPointMag1[index]
GammaPointPhs=GammaPointPhs1[index]
```

Figure 3.9: Testbench used to test the interpolation methods of the NL_F for a two tone input

Figure 3.9 shows the designed testbench for testing a two tone signal as in input to the behavioural model, in AWR. The testbench consists of a source generator that produces two signals at equal power levels, at two different frequencies. The frequencies here are 22GHz and 22.1GHz, with the swept power levels being from -30dBm to 50dBm in steps of 5dB. Like the single tone testbench, the input power levels are larger than the levels seen during the load pull measurement, to test the behavioural models ability to extrapolate data. Again all the interpolation methods are to be tested via this testbench.

This testbench contains the same vector method for controlling the gamma points that was seen in the single tone testbench. For this thesis, this testbench is used to measure the IM3 levels of the behavioural model, however with a more sophisticated behavioural model, other harmonic disturbances can be measured, such as IM5 and IM7.

3.3.3 Transistor Testbench

To make comparison between the measurements of the behavioural model easier, a simple testbench of the transistor model was developed in MWO. Figure 3.10 shows the designed testbench with the biasing setting of the transistor set to the same conditions that load pull was performed under. It can be seen that the input port applies a single tone signal, with swept power from -30dBm to 50dBm in steps of 5dB, at a frequency of 22GHz, and is used to take measurements of AM-AM and AM-PM. This maximum limit of the input power was chosen due to the time needed to simulate higher power levels being too long. This port was changed to a two tone signal port with frequencies of 22GHz and 22.1GHz, and swept input power levels of -30dBm to 30dBm for measurements of IM3. The load impedance at the input port is 50Ω in both cases to correctly simulate the load pull conditions. These testbenchs are just a simplified version of the load pull template and therefore should provide the same exact measurements that were seen in the load pull data files.

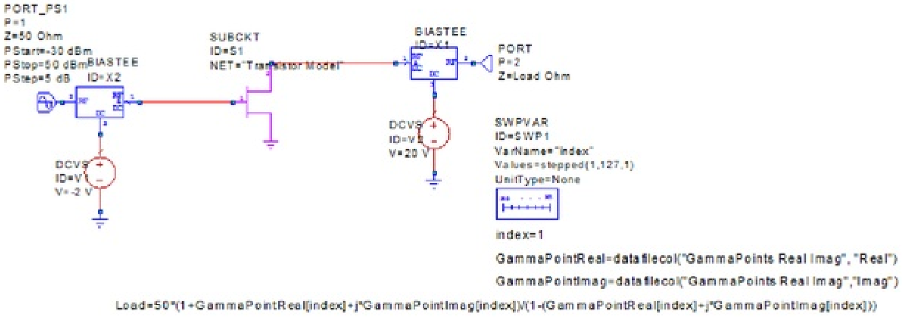


Figure 3.10: Testbench used to test the transitor model against the behavioural model

To control the load impedances seen at the output port of the transistor model, a similar method is implemented as the other testbenchs, however with slight variations. The first difference is that the vectors created in the transistor testbench are of normalised real and imaginary components of the gamma points. These values are then placed in the equation

$$Z_L = Z_0 \frac{1 + \Gamma_L}{1 - \Gamma_L} \tag{3.1}$$

which is simply a rearranged version of equation (2.2). The index variable is again used to control the gamma point selection, similar to the other testbenchs.

3.3.4 QAM/EVM Testbench

The final testbench that was designed for the behavioural model is used for measuring the EVM of the behavioural model. This testbench can be seen in figure 3.11, and it can be seen that a 16-QAM Gray Coded signal with average power of 25dBm is to be used as the test signal. This input power level has to be manually adjusted to a selected level by the user as automation in sweeping through a large number of power levels is simply too lengthy in time. The behavioural model interpolation method here is set to “Auto” as testing each interpolation method is too resource and time consuming.

EVM measurements can only be made in VSS and hence there is no comparison available to check the accuracy of the measurements. However one potential use for this testbench is in the ability to plot EVM on a Smith chart overlayed with other measurements such as IM3. For this thesis the EVM measurement of the EVM testbench will be plotted alongside the IM3 measurement from the Small Signal Polynomial Two Tone Testbench, to find the most optimal point that the behavioural model operates at, whilst considering that the source impedance, and hence Γ_S is not optimally matched.

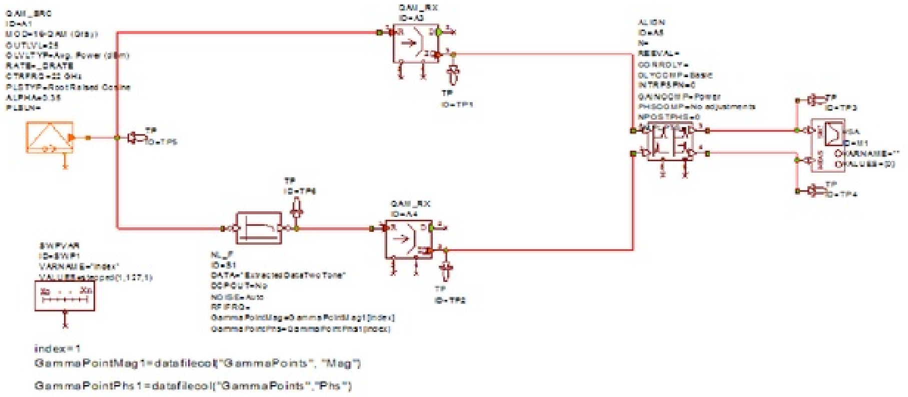


Figure 3.11: Testbench for EVM measurements of the behavioural model

Chapter 4

Using Behavioural Models for Amplifier Design

One potential application of this behavioural model method, particularly of interest by MACOM, is in the use of amplifier designing. Typically amplifier design is done at the transistor level in MWO which requires biasing circuits, transistors and passive elements and various tuning elements to find optimum matching circuits. However to make it simpler it is proposed that behavioural models be used in VSS as none of these overheads are needed to be constructed.

Figure 4.1 shows how an amplifier could potentially be designed with behavioural models. It can be seen that tuners are used to find the optimal Γ_S and Γ_L of each stage of the amplifier. This can be achieved by the behavioural model via the gamma point variables that control the data blocks. By selecting the optimal gamma point it will allow the behavioural model to simulate the optimal matching circuit conditions needed for maximum power transfer.

The main problem in this method however is selecting the optimum gamma point. Whilst the gamma point for each individual behavioural model for each stage will be the same, due to symmetry in the design, each stage will have a different optimal operating points. For example in the behavioural model described in previous chapters, there are 127 different gamma points and as each stage will have different optimal load impedances this means that there is 16,129 different gamma point combinations for a two stage amplifier and 2,048,383 different gamma point combinations for a 3 stage amplifier.

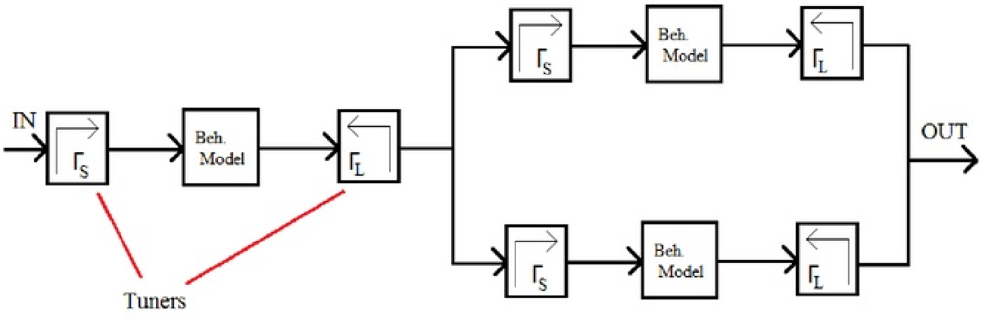
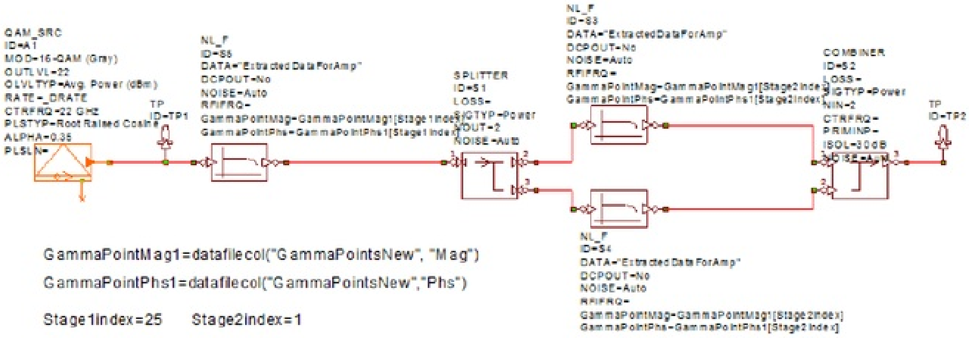


Figure 4.1: Example design for an amplifier based on behavioural models

4.1 Method

Figures 4.2 and 4.3 show a two stage amplifier design and three stage amplifier design respectively in VSS. The input signal to these amplifiers is a 16QAM Gray Coded signal with average power of 22dBm. Each of the designs takes the output of a single, first stage, behavioural model and splits it equally to then be passed into more behavioural models. At the end of the final stage in each design, the signals are combined together and measured. The key measurements for these particular models is the output power magnitude and output power IM3 levels.



```
QAM_SRC
ID=A1
MOD=16-QAM (Gray)
OUTLVL=22
OLVLTYP=Avg. Power (dBm)
RATE=_DRATE
CTRFRQ=22 GHz
PLSTYP=Root Raised Cosine
ALPHA=0.35
PLSLN=

TP
ID=TP1

NL_F
ID=S5
DATA="ExtractedDataForAmp"
DCPOUT=No
NOISE=Auto
RFIFRQ=
GammaPointMag=GammaPointMag1[Stage1index]
GammaPointPhs=GammaPointPhs1[Stage1index]

SPLITTER
ID=S1
LOSS=
SIGTYP=Power
NOUT=2
NOISE=Auto

NL_F
ID=S3
DATA="ExtractedDataForAmp"
DCPOUT=No
NOISE=Auto
RFIFRQ=
GammaPointMag=GammaPointMag1[Stage2index]
GammaPointPhs=GammaPointPhs1[Stage2index]

NL_F
ID=S4
DATA="ExtractedDataForAmp"
DCPOUT=No
NOISE=Auto
RFIFRQ=
GammaPointMag=GammaPointMag1[Stage2index]
GammaPointPhs=GammaPointPhs1[Stage2index]

COMBINER
ID=S2
LOSS=
SIGTYP=Power
NIN=2
CTRFRQ=
PRIMINP=
ISOL=30dB
NOISE=Auto

TP
ID=TP2

GammaPointMag1=datafilecol("GammaPointsNew", "Mag")
GammaPointPhs1=datafilecol("GammaPointsNew","Phs")
Stage1index=25     Stage2index=1
```

Figure 4.2: Designed two stage amplifier from behavioural models

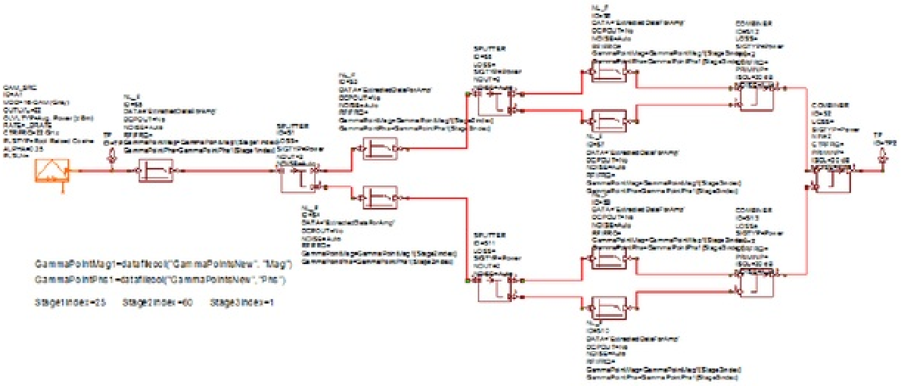

GammaPointMag1=datafilecol("GammaPointsNew", "Mag")
GammaPointPha1=datafilecol("GammaPointsNew", "Pha")
Stage1Index=25
Stage2Index=60
Stage3Index=1

Figure 4.3: Designed three stage amplifier from behavioural models

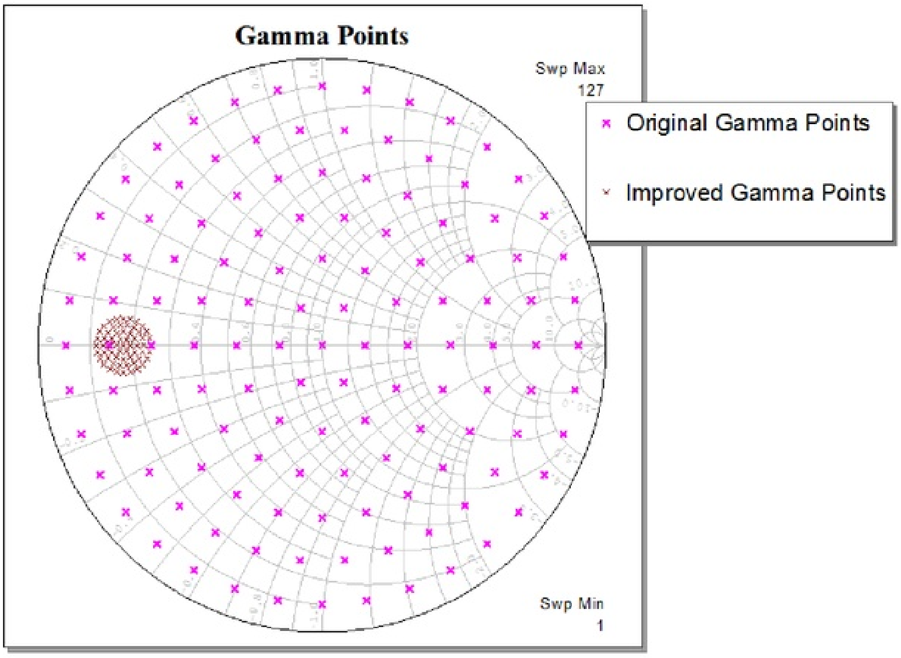


Figure 4.4: Original and improved gamma points

It is important to note that the data used for these behavioural models are somewhat different to the data used in previous chapters. This means that load pull was performed again, however this time the source impedance, and hence Γ_S is matched to the transistor model. This means that there is better power transfer, however does remove the Γ_S tuners in figure 4.1. This therefore meant that the only variable that is able to be tuned is the load gamma point. Another difference that was implemented in the load pull was the number of gamma points and area of Smith chart to sweep. The improved gamma point selection was made in accordance to the EVM results and by having a smaller number of gamma points, finding the optimal gamma point was therefore easier. Figure 4.4 shows the difference between the original load pulled gamma points and the improved load pulled gamma points.

Chapter 5

Simulation and Verification

This chapter will investigate the results of several simulations performed, as described in previous chapters. Due to the large number of gamma points that were load pulled, it is impractical to show the AM-AM, AM-PM and IM3 measurements for every single gamma point. Hence a selection of gamma points have been chosen to be analysed, with all the selected gamma points being relatively far away from each other and presenting vastly different impedances to the behavioural model. The index of the gamma point, load impedance (Z_L), magnitude of reflection coefficient ($|\Gamma_L|$) and phase of reflection coefficient ($\angle\Gamma_L$) of these points are summarised in table 5.1.

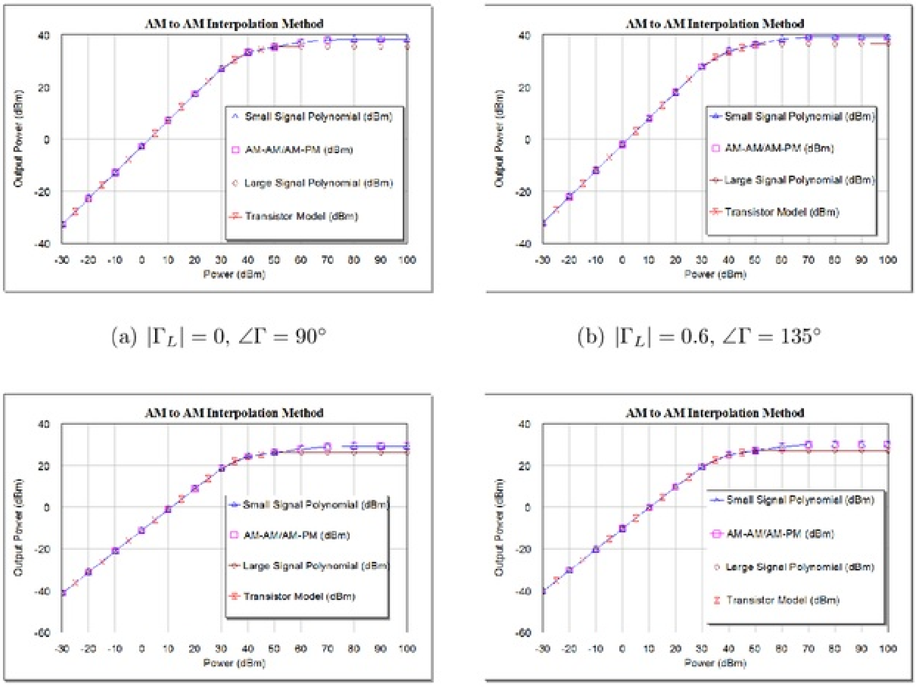
Index	Z_L (Ω)	$\lvert\Gamma_L\rvert$	$\angle\Gamma_L$ (°)
1	50	0	90
46	$14.42 + 19.23j$	0.6	135
117	$4.36 - 41.77j$	0.9	-100
62	$230.23 + 165.7j$	0.75	12

Table 5.1: Characteristics of gamma points chosen for behavioural model measurements

5.1 Behavioural Model Validation

5.1.1 AM-AM Results

Figure 5.1 shows the AM-AM results of the different interpolation methods available in AWR for the NL_F, as well as the AM-AM results of the transistor model for the gamma points specified in table 5.1. Collectively all three of the interpolation methods, correctly emulate the behaviour of the transistor model in all four gamma point cases. This can be seen particularly in the region of -30dBm to 50dBm of input power, where in all four graphs, all four traces overlap each other finely. For this region the behavioural model



(a) $|\Gamma_L| = 0$, $\angle\Gamma = 90°$

(b) $|\Gamma_L| = 0.6$, $\angle\Gamma = 135°$

(c) $|\Gamma_L| = 0.9$, $\angle\Gamma = -100°$

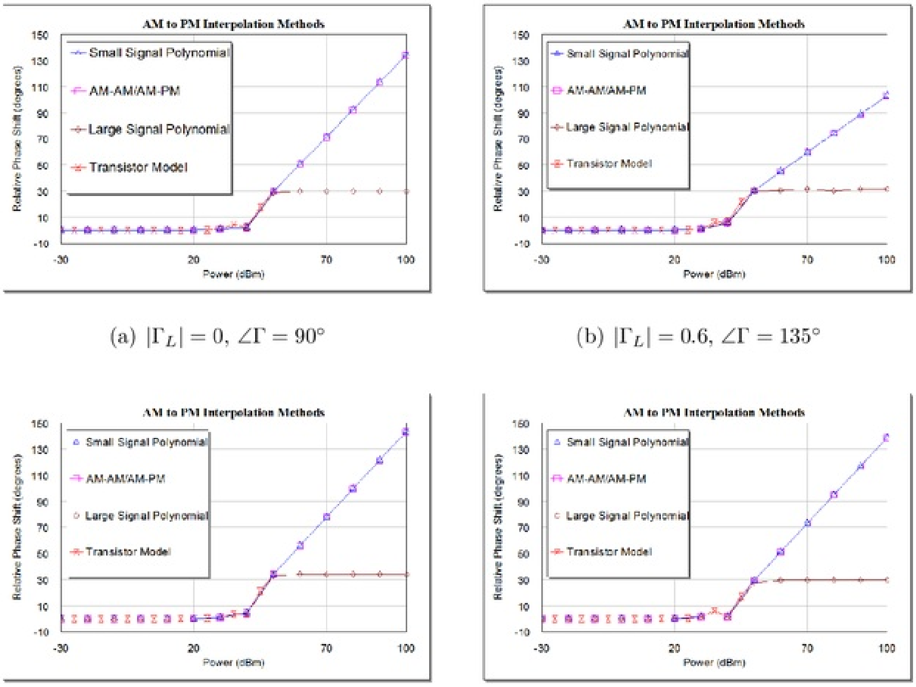
(d) $|\Gamma_L| = 0.75$, $\angle\Gamma = 12°$

Figure 5.1: AM-AM simulations of NL_F behavioural model and transistor model for a selection of gamma points

has information about output power, as this region was load pulled. Above this region however is where the interpolation methods start to deviate.

It is difficult to determine which of the three interpolation methods is the most accurate above 50dBm, as there is no transistor model trace to compare to due to limitations in simulating the transistor above 50dBm. It can be seen that in all four graphs that at 35dBm the transistor model starts to saturate and therefore gain should become compressed meaning that the behavioural model traces should flatten as more input power is applied. This is the case for all three interpolation methods, however the small signal polynomial and AM-AM/AM-PM interpolation methods are 3dB higher than the large signal polynomial, which stays at the same level as the final transistor measurement, and hence the final load pull measurement in the data file. This trend can be seen in all four graphs in figure 5.1, however does not provide an accurate proof as to what the transistor would do in this region. It is unlikely that the transistor would simply clip in the same way that the large signal polynomial does, and this can be seen by the gradient between measurements of the transistor model leading up to 50dBm reducing in size, however not being 0 at 50dBm. This gradient trend also suggests that a final 3dB rise, as seen in the small signal polynomial and AM-AM/AM-PM interpolation methods, is also unlikely.

5.1.2 AM-PM Results



(a) $|\Gamma_L| = 0$, $\angle\Gamma = 90°$

(b) $|\Gamma_L| = 0.6$, $\angle\Gamma = 135°$

(c) $|\Gamma_L| = 0.9$, $\angle\Gamma = -100°$

(d) $|\Gamma_L| = 0.75$, $\angle\Gamma = 12°$

Figure 5.2: AM-PM simulations of NL_F behavioural model and transistor model for a selection of gamma points

Figure 5.2 shows the AM-PM results of the different interpolation methods available in AWR for the NL_F, as well as the AM-PM results of the transistor model for the gamma points specified in table 5.1. It should be noted that due to VSS and MWO operating in different ways there was always a phase difference between the interpolation methods and the transistor model which was unnecessary as only the relative phase difference, i.e. the trend of the graphs, was needed to be analysed, and made it difficult to compare the interpolation methods to the transistor model. Hence all the measurements in this figure are referenced to a zero point, to allow easy comparison.

As mentioned, the transistor starts to saturate at 35dBm input power and it can be seen that in all 4 of the graphs the interpolation methods and transistor model are all equivalent up until this point. From the region of 35dBm to 50dBm the transistor model starts to deviate from the interpolation methods, and after 50dBm no transistor model is presented due to simulating limitations but the small signal and AM-AM/AM-PM interpolation methods stay the same and deviate from the large signal polynomial.

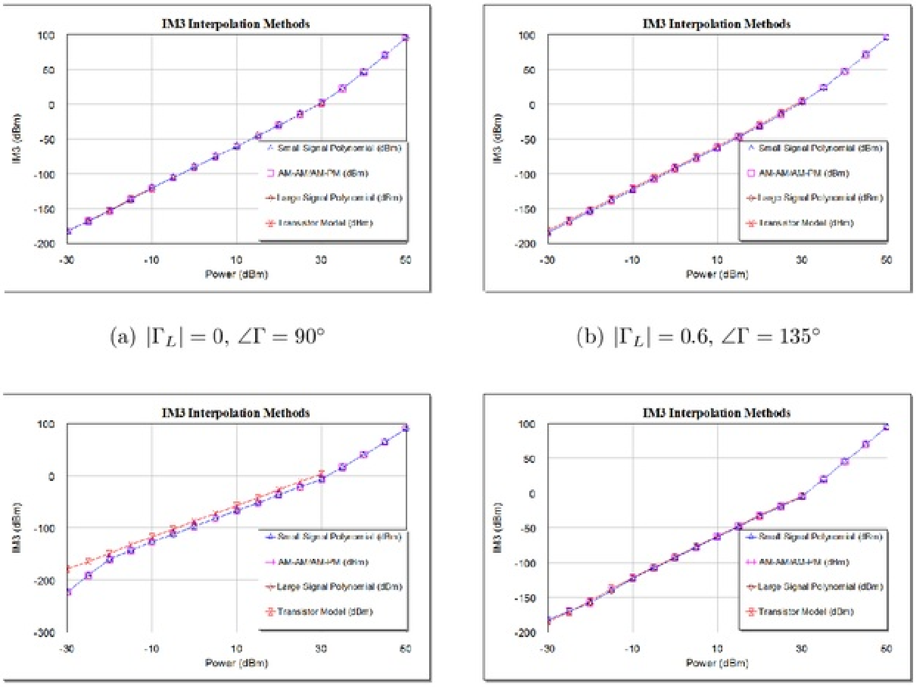
In the region of 35dBm to 50dBm, the transistor model starts to lose its alignment with the interpolation methods. This can be particularly seen in figures 5.2(b) and 5.2(d) at input power levels 35dBm and 45dBm. Whilst the input power levels were measured in the initial load pull and hence appear in the data file for the NL_F, they are not simulated in the single tone testbenchs and hence when plotting the AM-PM measurements this input power level is skipped. The consequences of this mean that AWR simply connects the points at 30dBm to 40dBm, and this can be seen all the graphs. To remedy this mistake, a simple solution would be to set the single tone testbenchs to steps of 5dB instead of 10dB, however the errors presented here are less than 3% and do not result in highly inaccurate results.

The region where it is likely that the most inaccuracy would occur between the behavioural model interpolation methods and the transistor model is in the region above 50dBm input power. It can be seen that the small signal polynomial and the AM-AM/AM-PM traces steadily increase, whereas the large signal polynomial clips and remains relatively flat. Due to the limitations in simulating the transistor above 50dBm, it is difficult to determine whether any of these methods show accurate results, however it is unlikely that any do show the correct results. The small signal polynomial and AM-AM/AM-PM results seem the most inaccurate as a steadily increasing phase shift is highly unlikely. Whilst the phase shift may increase, increasing in a linear fashion such as seen in all four graphs is highly unlikely. The large signal polynomial clips at the final data point in the load pull measurement final, similar to the AM-AM cases. Whilst plausible that this could occur, it is unlikely that the phase shift of a transistor would simply clip in this nature.

5.1.3 IM3 Results

Figure 5.3 shows the IM3 results of the different interpolation methods available in AWR for the NL_F, as well as the IM3 results of the transistor model for the gamma points specified in table 5.1. The results are obtained from the two tone testbench, with input frequencies of 22GHz and 22.1GHz, hence meaning the IM3 result is the power levels at 22.2GHz. One important feature of these graphs is that the three interpolation methods of the NL_F are all equal, with no large deviation between the large signal polynomial method compared to the other two methods.

It can be seen that figure 5.3(a) shows the most accurate results, with no error between the interpolation methods and the transistor model, figure 5.3(d) shows a slight error, figure 5.3(b) shows an increasing error, and figure 5.3(c) shows the worst error. Whilst not shown, through experimental analysis of other gamma points, the point corresponding



(a) $|\Gamma_L| = 0$, $\angle\Gamma = 90°$

(b) $|\Gamma_L| = 0.6$, $\angle\Gamma = 135°$

(c) $|\Gamma_L| = 0.9$, $\angle\Gamma = -100°$

(d) $|\Gamma_L| = 0.75$, $\angle\Gamma = 12°$

Figure 5.3: IM3 simulations of NL_F behavioural model and transistor model for a selection of gamma points

to figure 5.3(c), $|\Gamma_L| = 0.9$, $\angle\Gamma = -100°$, is the largest error seen for any measured gamma point. The causes of these error cannot attributed to the interpolation methods, or plotting in AWR as the measurements at each input power level corresponds to the same IM3 level seen in the data file. Whilst the exact cause of these errors is unknown, one potential reason for them is the load impedance applied to the transistor in the initial load pull or transistor testbench. This can be seen by the larger errors occurring at gamma points where the load impedance has a higher reactance than resistance. Therefore as $|\Gamma_L| = 0.9$, $\angle\Gamma = -100°$ has a load impedance of 4.36-41.77j the reactance of this point is nearly 10 times larger than the resistance, and could hence cause some problem in either the load pull measurement or transistor testbench.

Whilst figures 5.3(b) and 5.3(d) have interpolation methods that follow the shape of the transistor model IM3 levels, figure 5.3(c) does not. The cause of this difference is unknown. Whilst the error could potentially be attributed to the difference between reactance and resistance of the load impedance, as described above, it could also potentially be a glitch in the transistor testbench and in AWRs simulation. Whilst a glitch is unlikely it cannot be fully eliminated from probable causes due to when simulations were performed the large number of computing resources that were needed could have caused a simple glitch to occur.

One important aspect of the IM3 measurements that can be seen in all fours graphs of figure 5.3, is that the IM3 levels rise steadily after the final load pull measurement point of 35dBm. From previous results, such as AM-AM, it is known that the transistor enters saturation at 35dBm and hence it is expected like the AM-AM the IM3 output power levels should saturate as well. However it can be seen that this does not occur. The main reason for this is not occurring is that the initial two tone load pull was only conducted to 30dBm and hence not pushed into saturation, meaning the load pull data file has no information that the transistor has entered saturation. Due to the long amount of time needed to perform two tone simulations above 30dBm, it is impractical to perform the simulations for every gamma point on the Smith chart. However to investigate whether the IM3 levels would saturate, the point $|\Gamma_L| = 0$, $\angle\Gamma = 90°$, was chosen to be investigated, with the entire behavioural model process being carried out for the single point, this time with input power levels from -30dBm to 40dBm.

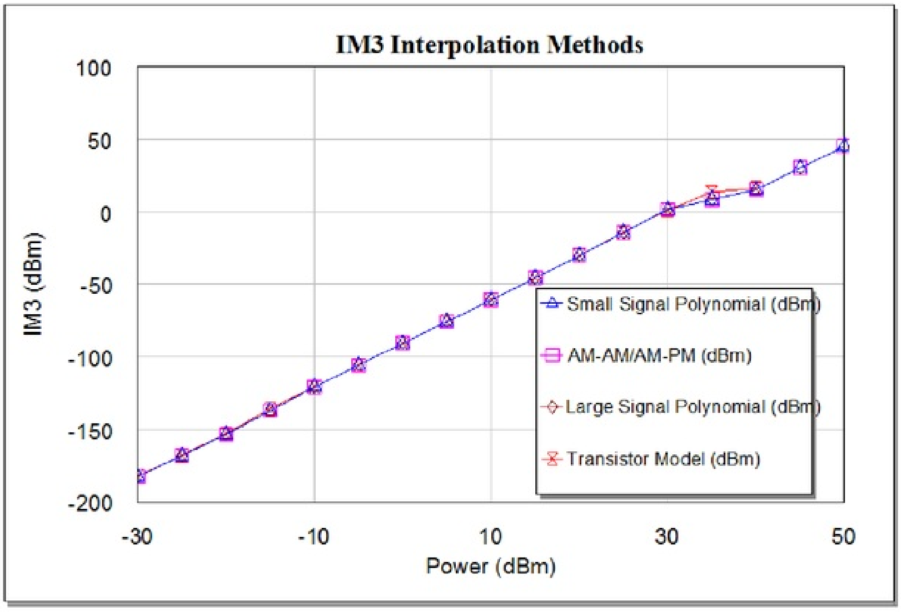


Figure 5.4: Improved transistor simulation with behavioural model simulation at $|\Gamma_L| = 0.6$, $\angle\Gamma = 135°$

The results of the IM3 levels for the gamma point $|\Gamma_L| = 0$, $\angle\Gamma = 90°$ at a more higher detailed initial load pull are shown in figure 5.4. Again like the initial IM3 measurements, for this particular gamma point there is no error between the transistor and the behavioural model. It can be seen however that this time the transistor model is pushed into saturation at 35dBm input power and starts to flatten out at 40dBm. This isnt exactly the same for the interpolation methods however, with the 35dBm input power of the interpolation methods being less than the transistor model. Despite this it can be seen that the gradient between the 35dBm and 40dBm of the interpolation methods is less than the gradient between 30dBm and 35dBm. This should indicate to the behavioural model that the model has, or at least is, entering saturation and should start to flatten. However like the previous IM3 results, it continues to rise linearly and does not take into consideration this small decrease in gradient. One potential method to fix this problem would be to rerun the load pull at even higher input power levels, however due to limitations in the computing system that the simulation was run on, this is not possible.

5.1.4 Contour Plotting

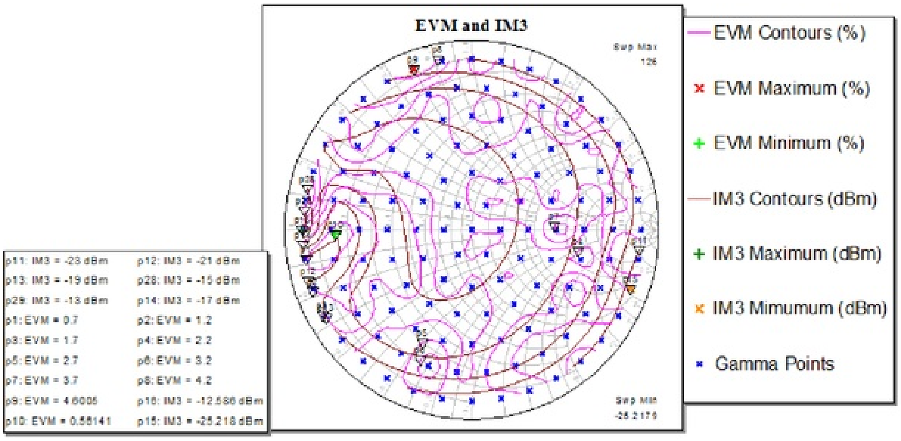


Figure 5.5: Overlayed contours of EVM and IM3

Figure 5.5 shows the contour plot of the EVM and IM3 of the behavioural model at input power of 25dBm. The EVM contours are in steps of 0.5% and the IM3 contours are in steps of 2dBm. There is no possible way to verify these results against the transistor model as the EVM measurements are only producible in VSS, where the transistor can only be simulated in MWO. However whilst verification is not possible, this method does show that the EVM can be plotted of a transistor, if a behavioural model is created.

From figure 5.5 it can be seen that the most optimal point in terms of EVM is at around 7Ω. The worst operating point in terms of EVM is at around 5+35jΩ. The most optimal point in terms of IM3 is around 2.5Ω, and the worst is around 78.5-260jΩ. This information can be used in design processes to select suitable operating points based on required needs, and different information can be used in other simulations depending on needs as well. For example, in the designing of amplifier circuits for this thesis, load pull was conducted around 7Ω due to the minimum EVM measurements at this point, as the input signal to the amplifier was to be a 16QAM signal at 22dBm, which is similar to this case.

5.2 Amplifier Design Results

As mentioned in previous chapters, the hardest part about constructing an amplifier design in the format described is the selection of gamma points for each stage. In this thesis a load pull was conducted again, in a smaller region, around 7Ω in accordance to the results seen by the EVM measurements of a single transistor. This meant that load impedance between two neighbouring points of the load pull was relatively small and hence severely eased the process of selecting the correct gamma points needed for optimising the amplifier as the small change caused little to no difference in the measurements. Whilst this is useful for this particular example, if a larger load pull was conducted, such as the original load pull of this thesis, the method of simply scrolling through all the points is impractical. The gamma points selected for each stage of the two stage amplifier for this thesis are characterised in table 5.2 and the gamma points selected for the three stage amplifier for this thesis are characterised in table 5.3.

Stage	Index	Z_L (Ω)	$\|\Gamma_L\|$	$\angle\Gamma_L$ (°)
1st	25	7.5+2j	0.738277	175.206
2nd	1	8.5	0.7	-180

Table 5.2: Characteristics of gamma points chosen for 2 Stage amplifier design

Stage	Index	Z_L (Ω)	$\|\Gamma_L\|$	$\angle\Gamma_L$ (°)
1st	25	7.5+2j	0.738277	175.206
2nd	60	12-1j	0.608744	-177.685
3rd	1	8.5	0.7	-180

Table 5.3: Characteristics of gamma points chosen for 3 Stage amplifier design

5.2.1 Cascaded Fundamental Power Levels

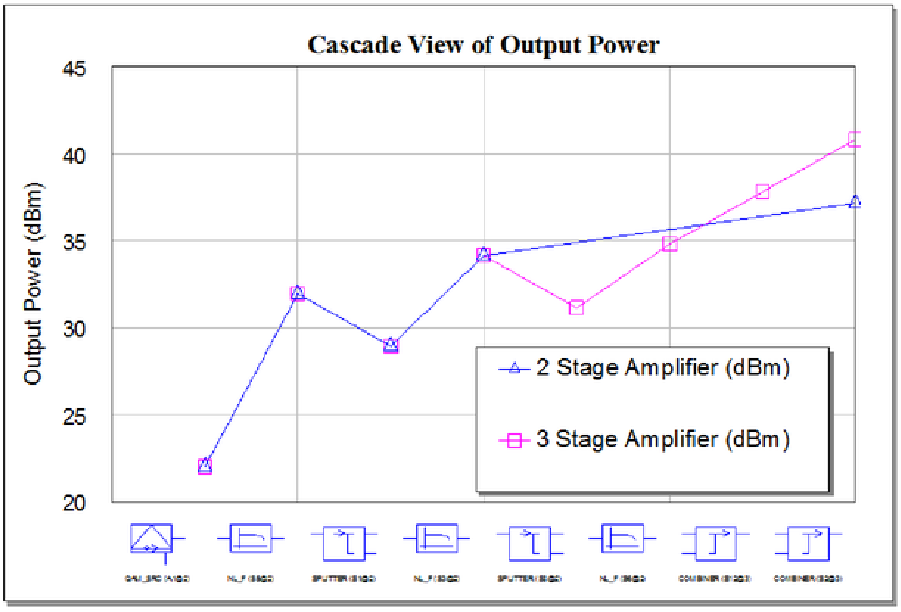


Figure 5.6: Cascaded view of fundamental output power of both the two stage and three stage transistor amplifiers

Figure 5.6 shows the fundamental output power levels at each stage of the amplifier designs. It can be seen that for the two stage amplifier case the signal rises from 22dBm to 37.16dBm, meaning the amplifier has a gain of 15.16dB. The three stage amplifier has an output of 40.83dBm, meaning that amplifier has a gain of 18.83dB. It can be seen that the three stage amplifier has the exact same output power levels at its second stage, compared to the two stage amplifiers. This suggests that the correct gamma points for the amplifiers have been determined.

5.2.2 Cascaded IM3 Levels

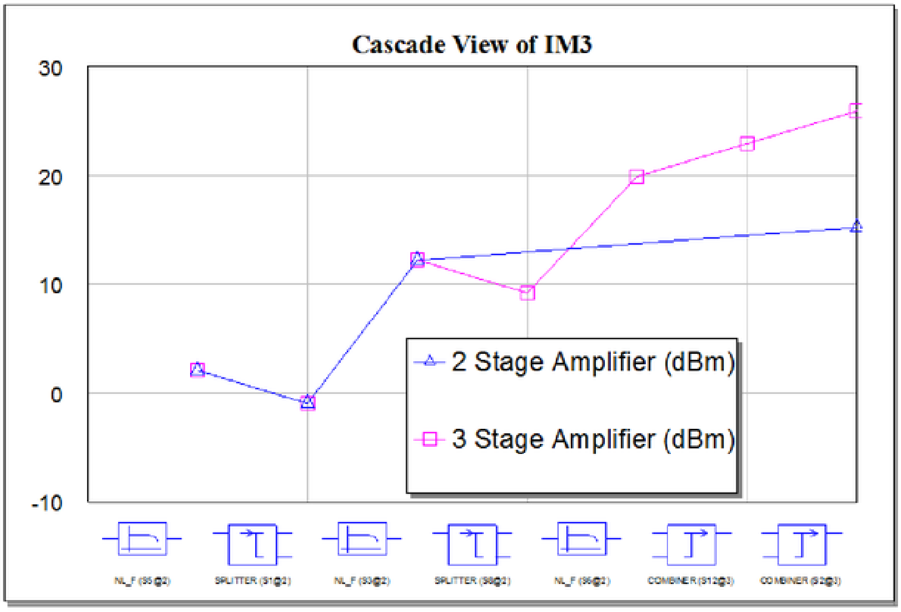


Figure 5.7: Cascaded view of IM3 for both the two stage and three stage transistor amplifiers

Figure 5.7 shows the third order harmonic output power levels at each stage of the amplifier designs. It can be seen that the two stage amplifier has an ultimate output IM3 level of 15.22dBm, whilst the three stage design has an IM3 of 25.93dBm at its final output. Again it can be seen that the second stage of the three stage amplifier is equal to the two stage amplifier suggesting the correct gamma points have been selected.

Chapter 6

Analysis of Simulation Results

This chapter analyses the results seen in chapter 5 and provides explanations as to the why errors and particular methods were used. The behavioural model validation is first discussed, followed by the amplifier designs.

6.1 Behavioural Model Validation

6.1.1 Results Analysis

Chapter 5.1 show varying degrees of accuracy for the NL_F behavioural model. For the AM-AM results there are no errors between the interpolation methods and the transistor model when there is available information in the data file, i.e. under 50dBm input power. Above 50dBm there is no information in the file and hence the measurements by the testbench rely solely on the interpolation methods. It is difficult to determine which method is the most accurate above 50dBm as the transistor model could not be simulated there, however as mentioned it is doubtful that any of the three methods gives fully accurate results. It appears that the large signal polynomial method gives a result that is the "worst case scenario" were it simply clips at the final measurement in the data file, whereas the small signal polynomial and AM-AM/AM-PM methods give a much more optimistic approximation by simply adding 3dB to this final value. It is therefore more plausible that the transistor model would actually fall in between these values.

The AM-PM results presented in chapter 5.1 show that the interpolation methods are very similar to the transistor model whilst not in saturation, however in saturation where no information is present in the data file, the interpolation methods vary greatly. After 50dBm, where the data in the load pull file terminates, the large signal polynomial clips at the final value in the file, whilst the small signal polynomial and AM-AM/AM-PM methods continue to rise linearly. All of these traces may be plausible results as AM-PM measurements are harder to predict in saturation than AM-AM. Despite this however, the methods presented in the charts do not seem 100% plausible as it would be unlikely for the AM-PM measurement to rise linearly without eventually clipping, or that the measurement would clip at the final value seen in the text file.

The IM3 measurements presented in chapter 5.1 show the largest errors of any measurements performed on the behavioural model. As mentioned the cause of these errors is unknown however there is a trend of larger errors occurring at gamma points with smaller resistance and larger reactance. Despite these errors between the transistor model and the interpolation methods, it can be seen that IM3 measurements for the behavioural model do not compress in the same way that the AM-AM measurements do. This is unexpected as IM3 measurements should compress. To ensure that the measurements were not a case of not supplying the behavioural model with enough information about the transistor, a second more intense load pull was conducted to drive the transistor into saturation. These results are seen in figure 5.4 and still it can be seen that the behavioural model does not saturate the results of IM3 at saturation input powers.

From these results it can be determined that the behavioural model, NL_F, does not operate correctly in areas of saturation where there is no information contained within the data file it operates off. Whilst the behavioural model may operate correctly in saturation when data is provided this was unable to be fully determined in this thesis due to limitations in the load pull simulations. However there is strong signs that this is the case as the AM-AM and AM-PM measurements do contain some data points where the transistor is in saturation. The transistor goes into saturation at around 35dBm input power and these two measurements are load pulled to 50dBm and hence it can be seen through the respective graphs that the interpolation methods do in fact closely follow the transistor model between 35dBm and 50dBm.

The EVM and IM3 measurements shown here are a simple example of a selection of measurements that can be used by a designer to determine the most efficient operating point of the behavioural model. Other parameters such as intermodulation distortion, that is the difference in output power levels between the fundamental and the nth order intermodulation frequency can also be measured of the behavioural model. These values can then be plotted on a Smith chart to see contours and determine the most optimum operating point based on these measurements.

6.1.2 Overall NL_F Analysis

One particular feature of the NL_F behavioural model that was discovered through experimental analysis, but not shown in this thesis is the effects of applying a different frequency two tone signal to the behavioural model. This was seen by applying a two tone signal at the frequencies 33GHz and 35GHz to the two tone behavioural model testbench and looking at the IM3 results. The results that were seen exactly match the results seen in figure 5.3, and hence it can be deducted that the behavioural model does not interpolate any frequency information whilst performing two tone measurements.

Despite the limitation of not being able to fully predict effects in saturation when no information is available, it does not fully rule out the use of this behavioural modelling process for applications where behavioural models are needed. For example, in the process of designing an amplifier circuit from behavioural models, having a transistor in saturation is not of use. Therefore, as long as the transistors used are kept below the saturation power levels, this method of behavioural modelling is of use.

Another limitation of the NL_F behavioural model that has been seen throughout this project is the rounding of variables to values that appear in the data file. In the investigations of AM-AM, AM-PM and IM3 the only gamma points that are able to be accurately measured are those that were present in the load pull. This means that any gamma point that is not measured in the initial load pull cannot be simulated by the behavioural model. This could potentially be an issue for future use of this method as whilst load pulling more gamma points will result in a more accurate behavioural model, it increases the time needed to perform a load pull measurement. Ideally it would more beneficial if AWR could interpolate data between the data blocks presented, however this would require a large amount of investigation, time, coding and resources to even to even test the viability of this process.

The behavioural model presented here only simulates the transistor under certain conditions. These conditions in particular are set frequencies and set bias voltages. Whilst adding swept frequencies and biases is possible through adding more variables to the data block headers, in the same way that gamma point information is applied, in doing so it adds another dimension of complexity to the initial load pull. This means that at every gamma point, every input power, every frequency and every bias voltage must be applied and this will increase the time needed for the load pull to run. Additionally the limitations of using variables in the data file is also present, in that the values for the variables must be within the data file otherwise they will be rounded. This could present problems in the accuracy of the behavioural model because to save on time higher steps may be added to swept parameters in the initial load pull. For example, a transistor could be load pull with many swept parameters and to save time large step sizes could be used. This could cause problems at the gate bias settings where the difference between cut-off and the transistor being active could be 0.1V. If the steps in the load pull are not less than 0.1V, it could potentially cause problems in the behavioural model. Therefore it is not necessarily more advantageous to add more swept parameters at the load pull in the hopes of obtaining a more advanced behavioural model.

Despite the limitations discussed in this section, the NL_F behavioural model does provide an easy method for engineers to create a behavioural model in their simulation software. If the NL_F model is to be used in combination to the designed behavioural modelling process, caution must be taken to not create severally invalid results. Ultimately this means that supplying the NL_F with as much information as possible will ensure the correct operation of the behavioural model.

6.2 Amplifier Designs

6.2.1 Amplifier Results Analysis

The EVM measurements presented in this thesis were used as a basis for designing an amplifier circuit. Having a large number of gamma points meant that a large portion of points would be undesirable, and hence the point where minimum EVM was the point chosen to perform a more detailed load pull. This had additional benefits of reducing the number of gamma points, as well as providing the potential to simply cycle through the gamma points and select the point that had the highest output gain at each stage. It should be noted that for these particular amplifier examples, the gamma points were chosen solely based on the point that would give the highest output power of the final stage of the amplifier. It would be unlikely that a real world use of this method would solely rely on one output parameter, when multiple would likely need to be considered. For example a maximum IM3 and minimum fundamental output power may be a requirement of the amplifier design and hence the gamma points selected would need to ensure these requirements are met.

The results presented in chapter 5.2 show a two stage amplifier with output power levels of 37.16dBm, with IM3 levels of 15.22dBm for a 16QAM 22dBm average power input signal. These results are somewhat acceptable, with the amplifier having a gain of 15.16dB which is a reasonable gain for a three stage amplifier. A similar reasoning can be applied to the three stage amplifier, as it has a fundamental output power level of 40.83dBm, IM3 of 25.93dBm and hence gain of 18.83dB, which is fair for a 3 stage amplifier.

Despite the relatively good gain of the fundamental power levels, one concern is the relative magnitude of the IM3 level being. For the two stage amplifier the IM3 level is 41% of the fundamental output power level, and for the three stage the IM3 level is 63.51% of the fundamental output power level. Whilst the two stage levels are reasonable, the three stage levels are particularly high. To fix this problem one potential solution would be to perform a load pull simulation again, this time however closer to the point where minimum IM3 levels occur, not minimum EVM occurs.

One important consideration that needs to be taken into account for these particular amplifier design examples is that the load impedances are relatively finely tuned. This can be particularly seen in the gamma point selection for the amplifier design being centered around 7Ω and the gamma points being very close to one another, meaning the change in load impedances is very small. This finely tuned gamma point selection is almost impracticable a real world environment as applying a fine impedance as seen in the gamma point selection is almost impossible. Therefore whilst it doesn't disprove the use of the amplifier designs, having such a fine load pull would ultimately affect the potential of applying the amplifiers in a real world situation.

6.2.2 Amplifier Design Process Analysis

Due to time restrictions the full extent of the amplifier design process was not completed. Some areas of investigation that were not looked at include: load pull closer to lower IM3 levels, applying a swept input power or the effects of adding a variable source impedance to the NL_F. Adding a swept input power level would be a crucial step in the design process of an amplifier as it would be unlikely that a single level input signal would constantly be applied for all situations. The changing source impedance would be useful in providing better customisation to the amplifier. This is because the behavioural model in stage one will not have the same optimal source impedance that the behavioural models in stage two have. However for this example, all behavioural models have the same source impedance.

Despite these features not being investigated due to time limitation, this method of amplifier design does show some promising potential. The main benefits of this design method is the relatively simple setup needed to construct the staged amplifier circuits, as well as the speed at which simulation occurs being very much faster than a full amplifier circuit simulation. Some downsides of this method however are: the selection of gamma point for each stage, and the eventual need to design a bias circuit in the circuit simulator if the amplifier was to be fully implemented.

Chapter 7

Conclusions and Future Work

This thesis aimed to present a method of deriving a behavioural model in AWR, based on simulated load pull measurements. Along with this, the NL_F behavioural model was chosen to be used in AWR, and therefore needed to be tested. To test both the designed process and the NL_F behavioural model, a provided UMS_GH25_TZ1S_8S150 transistor model was used. Finally, one potential application of this behavioural modelling process was briefly examined which is the potential for amplifier design through behavioural models.

The behavioural modelling process described in this thesis consists of three main components: the load pull measurement, the developed MATLAB scripts, and the developed testbenchs. All of these components work correctly in providing a functioning working component, that when combined can be used to create a behavioural model.

As this thesis describes, the NL_F behavioural model can provided varying degrees of accuracy. The AM-AM and AM-PM results show the most accuracy, whereas the IM3 results do show a margin of error for particular load impedances. However, overall the behavioural model does take the output of the designed modelling process and produce relatively good results.

The amplifiers designed in this thesis was a very basic and brief look into one potential application that the behavioural modelling process can be used for. The results of the amplifiers do show promising results in terms of fundamental output power, however more analysis is needed to determine whether this application can be of particular use.

7.1 Future Work

7.1.1 Further Behavioural Model Tests

To test the behavioural modelling process designed for this thesis, a supplied transistor model was used, however the designed method should be implementable for any electronic device. Therefore it is proposed that the entire process be re-conducted from the beginning, except this time with an already proven, working amplifier instead of the transistor.

By doing this the method and NL_F can be tested and compared to the results of the transistor to see if any errors or oversights are present in the results presented here.

Another application that is potential the most vital in successfully implementing if this process were to be used in the future, is the application of real life load pull data to the behavioural model. The work presented in this thesis only uses simulated load pull, however having a real set of load pull measurements potentially require new MATLAB scripts to generate the data file in the correct format. Also testing the real load pull measurements against the simulated measurements would be needed to see the accuracy of the NL_F.

Another potential future work recommendation for this project would be to somehow reduce the tedious nature of providing information to the MATLAB scripts. Currently there is a process of constructing tables in AWR and exporting and importing to MATLAB and then exporting and importing the generated data file into AWR. This process could potentially be removed if the MATLAB scripts were instead written in AWRs scripting editor. This was not investigated in the project due to unfamiliarity with the scripting editor and time constraints meaning a quick solution in MATLAB was a much more reasonable option.

7.1.2 Further Amplifier Design Tests

The results presented in this thesis for the application of the amplifier design process, are preliminary and a much more comprehensive analysis is needed to be investigated before MACOM could fully use the method in their design process. Whilst the results presented here are promising, there are factors that are not shown that need consideration including: stability analysis and designing of bias networks so the transistors can operate correctly. Also testing the behavioural model designed amplifier against a similarly constructed real world amplifier would be a necessary step if this method was to be used in the future.

Chapter 8

Abbreviations

DUT	Device Under Test
EVM	Error Vector Magnitude
NL_F	File Based Nonlinear Behavioural Model
MWO	Microwave Office
VNA	Vector Network Analyser
VSS	Visual System Simulator

Appendix A

MATLAB Code

A.1 Single Tone Script

```
%Author: Reese Moore
%Extraction of Load-Pull Data for a single tone meausurement is performed
%in this script
%It takes three input text files where one is a matrix of output power
%magntiude called OutputPowerMag
%The second is a matrix that takes the output power phase called
%OutputPowerPhs
%The third data file, called GammaPoints, consists of three column
%vectors: the gamma point magnitutde and phase named Mag and
%Phs respectively %and column vector of input power named InputPower.


%%%%%OutputFile--->ExtractedData.txt format%%%%%%
% VAR GammaPointMag = x1
% VAR GammaPointPhs = y1
% PIn(,dBm) POut(Mag,dBm)   POut(Phs,deg)
% PI1       |PO1|           ang(PO1)
% PI2       |PO2|           ang(PO2)
%
% VAR GammaPointMag = x2
% VAR GammaPointPhs = y2
% PIn(,dBm) POut(Mag,dBm)   POut(Phs,deg)
% PI1       |PO1|           ang(PO1)
%

% clear all
% uiimport('OutputPowerMag.txt')
% uiimport('GammaPoints.txt')
% uiimport('OutputPowerPhs.txt')
matrix=zeros(size(InputPower,1)*size(Mag,1),3);
CIP=1; %control input power row
COPR=1; %controls output power row
```

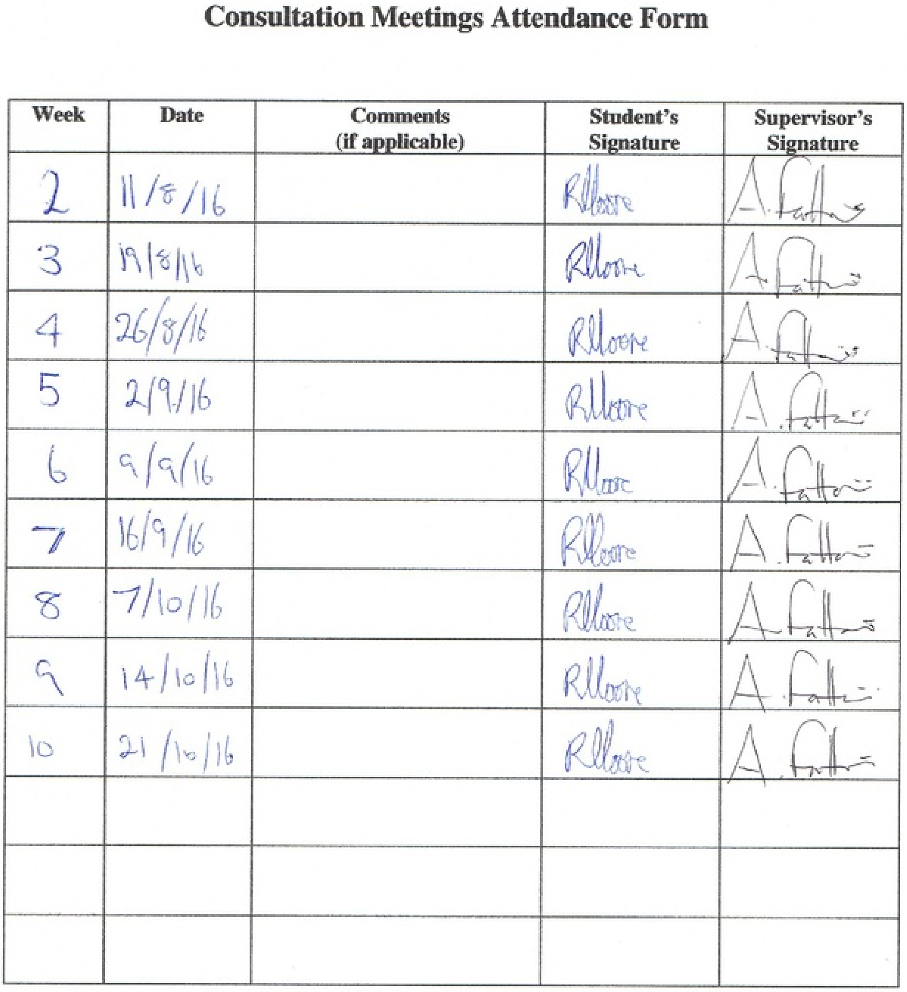
```
COPC=1; %controls output power col
multiplier=0;
index=1;

for n=1:size(matrix,1) %runs through rows of a matrix
    matrix(n,1)=InputPower(CIP,1);
    matrix(n,2)=OutputPowerMag(COPR,COPC);
    matrix(n,3)=OutputPowerPhs(COPR,COPC);
    index=index+1;
    if CIP<size(InputPower,1)
        CIP=CIP+1; %increase the input power to next row
        COPC=COPC+1; %move output col over 1
    else
        CIP=1;
        COPR=COPR+1;
        if index>size(InputPower,1)*size(Mag,1)
            index=1;
            multiplier=multiplier+1;
            COPR=1;
        end
        COPC=1+size(InputPower,1)*multiplier;
    end
end

CGP=1; %control gamma point rows
outFile=fopen('ExtractedData.txt','w');
GammaPointsMag = 'VAR GammaPointMag = %12f';
GammaPointsPhs= 'VAR GammaPointPhs = %12f';
fprintf(outFile, GammaPointsMag, Mag(CGP,1));
fprintf(outFile, '\r\n');
fprintf(outFile, GammaPointsPhs, Phs(CGP,1));
fprintf(outFile, '\r\n');
fprintf(outFile,'%12s %12s %12s \r\n','PIn(,dBm)',
'POut(Mag,dBm)','POut(Phs,deg)');
f=1;
for k = 1:size(matrix,1)
    fprintf(outFile,'%g\t',matrix(k,:));
    fprintf(outFile,'\r\n');
    f=f+1;
    if f>size(InputPower,1)
        f=1;
        CGP=CGP+1; %move to next Gamma Point
        if CGP <= size(Mag,1)
            fprintf(outFile,'\r\n');
            fprintf(outFile, GammaPointsMag, Mag(CGP,1));
            fprintf(outFile, '\r\n');
            fprintf(outFile, GammaPointsPhs, Phs(CGP,1));
            fprintf(outFile, '\r\n');
            fprintf(outFile,'%12s %12s %12s \r\n','PIn(,dBm)',
                        'POut(Mag,dBm)','POut(Phs,deg)');
        end

    end
end
fclose(outFile);
```

A.2 Two Tone Script

```
%Author: Reese Moore
%Extraction of Load-Pull Data for a 2 tone meausurement is performed in
%this script It takes four input text files where one is a matrix of
%output power magntiude called OutputPowerMag The second is a matrix
%that takes the output power phase called OutputPowerPhs The third
%data file, called GammaPoints, consists of three column vectors:
%the gamma point magnitutde and phase named Mag and Phs respectively,
%and column vector of input power named InputPower.
%The fourth data file, called ThirdOrderOutputPower, is a matrix of
%power located in the third order harmonic 2*f2-f1



% clear all
% uiimport('OutputPowerMag.txt')
% uiimport('GammaPoints.txt')
% uiimport('OutputPowerPhs.txt')
% uiimport('ThirdOrderOutputPower.txt')

matrix=zeros(size(InputPower,1)*size(Mag,1),4);
CIP=1; %control input power row
COPR=1; %controls output power row
COPC=1; %controls output power col
multiplier=0;
index=1;

for n=1:size(matrix,1) %runs through rows of a matrix
    matrix(n,1)=InputPower(CIP,1);
    matrix(n,2)=OutputPowerMag(COPR,COPC);
    matrix(n,3)=OutputPowerPhs(COPR,COPC);
    matrix(n,4)=ThirdOrderOutputPower(COPR,COPC);
    index=index+1;
    if CIP<size(InputPower,1)
        CIP=CIP+1; %increase the input power to next row
        COPC=COPC+1; %move output col over 1
    else
        CIP=1;
        COPR=COPR+1;
        if index>size(InputPower,1)*size(Mag,1)
            index=1;
            multiplier=multiplier+1;
            COPR=1;
        end
        COPC=1+size(InputPower,1)*multiplier;
    end
end

CGP=1; %control gamma point rows


outFile=fopen('ExtractedData.txt','w');
GammaPointsMag = 'VAR GammaPointMag = %12f';
GammaPointsPhs= 'VAR GammaPointPhs = %12f';
fprintf(outFile,'%12s \r\n','FREQ=22G');
fprintf(outFile,'%12s \r\n','FREQB=22.1G');
fprintf(outFile, GammaPointsMag, Mag(CGP,1));
fprintf(outFile, '\r\n');
fprintf(outFile, GammaPointsPhs, Phs(CGP,1));
fprintf(outFile, '\r\n');
fprintf(outFile,'%12s %12s %12s %12s \r\n','PIn(,dBm)',
'POut(Mag,dBm)','POut(Phs,deg)', 'IM_2_1(,dBm)');
f=1;
for k = 1:size(matrix,1)
    fprintf(outFile,'%g\t',matrix(k,:));
    fprintf(outFile,'\r\n');
    f=f+1;
    if f>size(InputPower,1)
        f=1;
        CGP=CGP+1; %move to next Gamma Point
        if CGP <= size(Mag,1)
        fprintf(outFile,'\r\n');
        fprintf(outFile, GammaPointsMag, Mag(CGP,1));
        fprintf(outFile, '\r\n');
        fprintf(outFile, GammaPointsPhs, Phs(CGP,1));
        fprintf(outFile, '\r\n');
        fprintf(outFile,'%12s %12s %12s %12s \r\n','PIn(,dBm)',

            'POut(Mag,dBm)','POut(Phs,deg)', 'IM_2_1(,dBm)');
        end

    end
end
fclose(outFile);
```

Appendix B

Consultation Meeting Attendance Form



Consultation Meetings Attendance Form

Week	Date	Comments (if applicable)	Student's Signature	Supervisor's Signature
2	11/8/16		RMoore	A. Fatt
3	19/8/16		RMoore	A. Fatt
4	26/8/16		RMoore	A. Fatt
5	2/9/16		RMoore	A. Fatt
6	9/9/16		RMoore	A. Fatt
7	16/9/16		RMoore	A. Fatt
8	7/10/16		RMoore	A. Fatt
9	14/10/16		RMoore	A. Fatt
10	21/10/16		RMoore	A. Fatt

Figure B.1: Consultation Meeting Attendance Form

Bibliography

[1] M. Andrejevic and V. Litovski, "Electronic circuits modeling using artificial neural networks," *JOURNAL OF AUTOMATIC CONTROL*, vol. 13, no. 1, pp. 31–37, 2003.

[2] AntennaTutorials, "Transmission Lines: Reflection Coefficient and VSWR," Online. [Online]. Available: http://www.antenna-theory.com/tutorial/txline/transmission3.php

[3] AWRCorp, "Nonlinear behavioral model (file-based): Nl_f," Online. [Online]. Available: https://awrcorp.com/download/faq/english/docs/VSS_System_Blocks/NL_F.htm

[4] V. Ceperic and A. Baric, "A brief overview of the black-box behavioural modelling of electronic circuits for transient simulations," in *MIPRO, 2012 Proceedings of the 35th International Convention*, 2012, pp. 72–77.

[5] F. M. Ghannouchi, O. Hammi, and M. Helaoui, *Behavioral Modeling and Predistortion of Wideband Wireless Transmitters.* John Wiley & Sons, 2015.

[6] LeLeivre, "Reflection Coefficient to Impedance Converter," Online. [Online]. Available: http://leleivre.com/rf_gammatoz.html

[7] J. Liu, "Practical behavioral modeling technique of power amplifiers based on loadpull measurements," Master's thesis, University of South Florida, 2005.

[8] J. Liu, L. P. Dunleavy, and H. Arslan, "Large-signal behavioral modeling of nonlinear amplifiers based on load-pull am-am and am-pm measurements," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 8, pp. 3191–3196, 2006.

[9] M. N. Marbell, M. Simcoe, R. Wilson, M. Chidurala, and S. Ward, "A load-pull-based behavioral model for doherty pa design," in *Wireless and Microwave Technology Conference (WAMICON), 2011 IEEE 12th Annual*, 2011, pp. 1–5.

[10] Microwaves101, "Load Pull for Power Devices," Online. [Online]. Available: http://www.microwaves101.com/encyclopedias/load-pull-for-power-devices

[11] ——, "Smith Chart Basics," Online. [Online]. Available: hhttp://www.microwaves101.com/encyclopedias/smith-chart-basics

[12] W. Sansen, “Distortion in elementary transistor circuits,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 3, pp. 315–325, 1999.

[13] T. Turlington, *Behavioral Modeling of Nonliear RF and Microwave Devices.* Artech House, 2000.