ABSTRACT

WALKER, AARON L. Behavioral Modeling and Characterization of Nonlinear Operation in RF and Microwave Systems. (Under the direction of Professor Michael B. Steer).

Modeling and characterization strategies were developed to capture the complex nonlinear behavior of both the components within an RF circuit and within an assembled system. Metrology techniques using common laboratory instrumentation were developed for the remote characterization of assembled RF devices and the result of this characterization is presented here for the first time. In addition, two novel intermodulation relative phase measurement systems were developed that have 90 dB of dynamic range which is a considerable improvement over existing systems. This enables the measurement of weaker nonlinearities and increases the accuracy of these measurements. A behavioral modeling architecture and extraction procedure were developed to take advantage of the metrology techniques to allow the separation of odd-ordered nonlinear effects and baseband upconversion effects in several amplifiers. The results of applying this modeling strategy has produced a general model capable of predicting both magnitude and phase asymmetries in nonlinear components for the first time. Concepts from the remote characterization, intermodulation phase measurement systems, and the modeling efforts were combined to demonstrate the extraction of two-port transmission parameters from one-port measurements in a archetypal circuit of an integrated bandpass filter in an RF front-end. It is shown how this information can be used to tune a filter.

Behavioral Modeling and Characterization of Nonlinear Operation in RF and Microwave Systems

by

Aaron L. Walker

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Doctor of Philosophy

Electrical Engineering

Raleigh

2005

Approved By:

Dr. Kevin G. Gard	Dr. W. Rhett Davis
Dr. Michael B. Steer Chair of Advisory Committee	Dr. Gianluca Lazzi

To Emily

Biography

Aaron L. Walker was born in Topeka, Kansas, USA in 1975. He received the B.S. degree in Nuclear Engineering from Kansas State University in 1997 and the M.S. degree in Electrical Engineering from North Carolina State University in 1999. From 1999 to 2002 he held positions as a design engineer with Atmel and as an ASIC design and verification engineer with Zaiq Technologies. Since 2002 he has been a research assistant in the Department of Electrical and Computer Engineering at NCSU. His research interests include nonlinear behavior and characterization in RF systems, microwave metrology, and behavioral modeling.

Acknowledgements

I would like to express my appreciation for the support and guidance provided by my advisor Dr. Michael Steer during my PhD research. His considerable knowledge of all things microwave aided greatly in the completion of my work. I am also grateful for the access to Dr. Steer's professional contacts that resulted in funding for my research by ARO and the NAVSEA EODTECHDIV. I would also like to thank the members of my committee Dr. Kevin Gard, Dr. Gianluca Lazzi, and Dr. Rhett Davis as well as Dr. Doug Barlage for their comments and discussions on my research.

Thanks also go to my graduate student colleagues Khaled, Won-hoon, Frank, Wael, and especially Sonali, Nikhil, Mark, and Jayesh for stimulating conversation and an exchange of knowledge that extended my own research and experience. I thank my family for instilling in me a hard-work ethic and the perseverance to accomplish whatever goals I choose. Finally, I would like to thank my wife Emily whose invaluable support made this work possible.

Contents

List of Figures **ix**

List of Tables **xii**

List of Symbols **xiii**

1 Introduction **1**
1.1 Motivations and Objectives of This Study 1
1.2 Behavioral Modeling of Nonlinear Systems 2
1.3 Metrology for Nonlinear Behavior 3
1.4 Overview of RF Manufacturing Test Challenges 4
1.5 Original Contributions 4
1.6 Thesis Overview 6
1.7 Published Works 7
1.7.1 Journals 7
1.7.2 Conferences 7
1.8 Unpublished Works 8

2 Review of Nonlinear Analysis, Behavioral Modeling, and Measurement Techniques **9**
2.1 Introduction 9
2.2 Description of Nonlinear Behavior 10
2.3 Review of Analysis Techniques 12
2.3.1 Small-signal Analysis 12
2.3.2 Large-signal Analysis 12
2.4 Review of Behavioral Modeling Techniques 16
2.4.1 Common Nonlinear Behavioral Models 17
2.4.2 RF Behavioral Modeling Techniques 21
2.5 Review of Measurement Techniques 26
2.5.1 Small-signal Measurements 27
2.5.2 Large-signal Measurements 27
2.5.3 Communications-Based Metrics 31

2.6 Summary 32

3 Nonlinear Metrology **33**
3.1 Review of Two-Tone Measurements 34
3.1.1 Stimulus Sources 34
3.1.2 Stimuli Combination Devices 35
3.1.3 Isolators 37
3.1.4 Stimulus Amplification 37
3.1.5 Passive Intermodulation 38
3.2 Wireless Two-Tone Measurement 39
3.2.1 Nonlinear Reflection 39
3.2.2 Wireless Characterization Apparatus 41
3.3 Multi-Channel Feed-Forward Cancellation System to Improve Dynamic Range 42
3.3.1 Feed-Forward Justification 44
3.3.2 Cancellation Mechanisms 45
3.3.3 Cancellation System Architecture 46
3.3.4 Limitations of the Cancellation System 50
3.3.5 Quantization Noise Limit on Cancellation 50
3.4 Intermodulation Relative Phase Measurement 55
3.4.1 Intermodulation Phase Measurement Background 57
3.4.2 Vector Intermodulation Analyzer for Measuring Relative Intermodulation Phase Versus Input Power 59
3.4.3 Vector Signal Analyzer-Based Vector InterModulation Analyzer . . . 63
3.5 Summary of Metrology System Development 69

4 The Multi-Slice Behavioral Model **71**
4.1 Behavioral Modeling Justification 72
4.2 Description of Multi-Slice Model 72
4.3 Features of the Multi-Slice Model 73
4.3.1 Modularity 73
4.3.2 Applicability to Multiple Simulation Types 74
4.3.3 Capture of Memory Effects 74
4.4 Multi-Slice Model Extraction 80
4.5 Summary 82

5 Remote Characterization of RF Devices **84**
5.1 Introduction 84
5.2 Two-Way Radio Characterization 85
5.3 AMPS Cellular Phone Characterization 88
5.4 Characterization of Various Electronic Products 88
5.5 Development of a Single-Slice Model for the Two-Way Radio Response . . . 92
5.5.1 Power Off Third-Order Model Extraction 94
5.5.2 Power Off Fifth-Order Model Extraction 98
5.5.3 Power On Model Extraction 98
5.5.4 Power On Third-Order Model Extraction 101

5.5.5 Power On Fifth-Order Model Extraction . . . 101
5.6 Conclusion . . . 101

6 A Low-Noise Amplifier Modeled with a Multi-Slice Behavioral Model 104
6.1 Introduction . . . 104
6.2 Amplifier Multi-Slice Behavioral Model Extraction . . . 105
6.2.1 Linear Networks . . . 105
6.2.2 Nonlinear Blocks . . . 107
6.3 Application of the Magnitude-Only Model . . . 109
6.4 Conclusion . . . 116

7 Multi-Slice Behavioral Models of Nonlinear Amplifiers with Memory Employing Vector Intermodulation Measurements 119
7.1 Introduction . . . 119
7.2 Development of the Multi-Slice Model Incorporating IMD Phase Information 120
7.2.1 Model Parameter Extraction . . . 120
7.3 Comparison of Power Amplifiers . . . 121
7.3.1 MESFET Amplifier . . . 121
7.3.2 MOSFET Instrumentation Amplifier . . . 123
7.3.3 Darlington HBT Amplifier . . . 124
7.4 Summary of Results . . . 126

8 In-Circuit Filter Tuning Using a Nonlinear Distortion Response from a One-Port Measurement 137
8.1 Introduction . . . 137
8.2 Overview of Filter Tuning in RF Systems . . . 138
8.3 Overview of Filter Tuning Using an Intermodulation Distortion Response . 139
8.4 Filter Transmission Extraction via Intermodulation Distortion . . . 140
8.5 Filter Tuning Example . . . 143
8.6 Conclusion . . . 146

9 Conclusions and Future Work 152
9.1 Summary of Research and Original Contributions . . . 152
9.2 Future Research . . . 154

A Intermodulation Phase Measurement System Calibration 155
A.1 Effect of Source Phase Shift on Measurements . . . 156

B Algorithm Details for the Multi-Tone Cancellation System 159
B.1 Algorithm Equations . . . 160
B.1.1 Miscellaneous Implementation Details . . . 164
B.1.2 Matlab Code for Computation of the Exponent Term Table . . . 164
B.2 C Code for Algorithm Execution on a Freescale 56F8323 DSP . . . 165
B.2.1 Main Function . . . 165
B.2.2 Macro Definitions . . . 175
B.2.3 Functions Header File . . . 176

B.2.4 Power Difference Exponent Search . . . 178
B.2.5 Multiplication Routine for Mixed Unsigned Fractional and Integer Values . . . 181
B.2.6 Multiplication Routine for an Unsigned Fractional and Signed Integer Values . . . 182
B.2.7 Peripheral Initialization Function . . . 182
B.2.8 Quadrature Magnitude Update Function . . . 184
B.2.9 Quadrant Search Function for Quadrature Phase Adjustment . . . 186
B.2.10 Quadrature Phase Update Function . . . 187
B.2.11 Interrupt Service Routines . . . 189

C Matlab Code for Magnitude-Only Multi-Slice Model Extraction 194
C.1 Main Script . . . 194
C.2 Correction of Measured IM3 Magnitude for Source Flatness . . . 200
C.3 Extraction Script for Linear Networks of First Slice . . . 201
C.4 Pre-Linear Network Extraction Function . . . 203
C.5 Post-Linear Network Extraction Function . . . 204
C.6 Third-Order Coefficient Extraction from Measured Two-Tone Response . . . 205
C.7 Modified Polynomial Fit Function . . . 206
C.8 Memoryless Polynomial Coefficient Extraction Function . . . 207
C.9 Baseband Contribution Extraction Function . . . 208
C.10 Model Output Generation Function . . . 209

D Matlab Code for Complex Multi-Slice Model Extraction 213
D.1 Scripts for GaAs MESFET Amplifier . . . 213
D.1.1 RF Source Correction Script . . . 213
D.1.2 Model Parameter Extraction and Output Generation Script . . . 221
D.2 Scripts for MOSFET Amplifier . . . 234
D.2.1 RF Source Correction Script . . . 234
D.2.2 Model Parameter Extraction and Output Generation Script . . . 242
D.3 Scripts for Darlington HBT Amplifier . . . 255
D.3.1 RF Source Correction Script . . . 255
D.3.2 Model Parameter Extraction and Output Generation Script for Manual Selection of Third and Fifth-Order Coefficients . . . 265
D.3.3 Model Parameter Extraction and Output Generation Script for Genetic Algorithm Fitting Procedure . . . 274
D.3.4 Model Parameter Extraction and Output Generation Script for Levenberg-Marquadt Fitting Procedure . . . 287

E Post-Processing Details for Raw Vector Measurements in the In-Circuit Filter Tuning Application 300
E.1 Matlab Code for Execution of Post-Processing . . . 301

Bibliography 311

List of Figures

2.1 Response of a nonlinear system to discrete tone stimulus. 11
2.2 Depiction of Volterra-series. 14
2.3 AM-AM, AM-PM model representation with the amplitude and phase modulation components considered separately (after [30]). 19
2.4 Diagrams of Wiener, Hammerstein, and Wiener-Hammerstein block model representations of nonlinear systems with memory. 20
2.5 PA model for a system with memory using the parallel Wiener model (after [39]). 23
2.6 PA model for a system with memory using a memory polynomial model with unit delay taps (after [9]). 24
2.7 PA model using a memory polynomial model with sparse delay taps (after [9]). 25
2.8 Simple general circuit for IMD asymmetry analysis, (after [43]). 26
2.9 Basic two-tone measurement apparatus. 29

3.1 Basic configuration for two-tone measurement apparatus employing independent signal sources. 34
3.2 High power two-tone measurement system with independent stimulus paths. 38
3.3 Typical RF receiver frontend with bandpass filter, LNA, image reject filter and RF mixer. 40
3.4 Diagram of RF frontend considering nonlinear reflection as generating a nonlinear response ($f()$) that travels through the frontend linear network and re-radiates. 41
3.5 Wireless two-tone characterization apparatus designed to minimize system generated IMD. 43
3.6 Cancellation contours depicting the amount of phase and amplitude errors allowable for a given cancellation level. 47
3.7 Schematic of active interference cancellation system. 49
3.8 Quantization noise limit on cancellation for pure phase and amplitude errors. 56
3.9 Measurement system block diagram incorporating phase-locked signal sources. 60
3.10 Test setup for measuring phase calibration data for RF sources. 61
3.11 Measured phase insertion of RF source and step attenuator corrected phase insertion at 400 MHz. 62

3.12 Corrected phase insertion of RF source vs. carrier frequency and output power demonstrating linear phase shift with output power and weak dependence on carrier frequency. 63
3.13 Block diagram of the VSA-based intermodulation phase measurement setup. 67
3.14 Block diagram of the hardware and software components of the quadrature phase measurement system. 68
3.15 Flowchart depicting nonlinear distortion phase measurement utilizing independent source and a vector signal analyzer. 70

4.1 Two-slice architecture of multi-slice model. 73
4.2 Phasor addition of first and second slice components depicting phase reference of baseband components producing asymmetries in IM3 phase and magnitude. Components of the IM3 response are (a) odd-ordered output from first slice, (b) upper (lower) baseband IM contribution from second slice, (c) lower (upper) baseband IM contribution from second slice, (d) total upper (lower) IM3 response from sum of slice outputs, (e) total lower (upper) IM3 response from sum of slice outputs. 81

5.1 Narrow band IM3L response of a powered two-way radio to RF illumination under swept frequency and power. 86
5.2 Narrow band IM3L response of an un-powered two-way radio to RF illumination under swept frequency and power. 87
5.3 Frequency response of IM3L in the handset transmit band for RF illumination of un-powered AMPS cellular phone. 89
5.4 Frequency response of IM3L in the handset receive band for RF illumination of un-powered AMPS cellular phone. 90
5.5 Frequency response of IM3L product from PCI video card under wide-band RF illumination. 91
5.6 Two-way radio normalized linear system estimation. 96
5.7 Third-order model extraction results for a two-way radio in the power off state 97
5.8 Estimate of the fifth-order intermodulation response for a two-way radio in the power off state based on a fifth-order model extraction 99
5.9 Estimate of the third-order intermodulation response for a two-way radio in the power off state based on a fifth-order model extraction 100
5.10 Estimate of the third-order intermodulation response for a two-way radio in the power on state based on a third-order extraction 102

6.1 A two slice nonlinear system behavioral model with passband modeling. . . 106
6.2 Extracted pre-filter and post-filter frequency responses. 111
6.3 Measured output power of (a) IM3H and (b) IM3L at −15 dBm input tone power. 112
6.4 Measured (+) and modelled (line) fundamental output power at (a) −1 dBm, (b)−10 dBm, and (c) −15 dBm input tone power. 113
6.5 IM3 modelled and measured frequency response at −7 dBm. 114
6.6 IM3 modelled and measured frequency response at −9 dBm. 115

6.7 IM3 modelled and measured response at 3 GHz, P_{out} vs. P_{in}. 116
6.8 IM3 modelled and measured response at 3 GHz, V_{out} vs. V_{in} 117

7.1 Measured magnitude and phase of the IM3 products for the Mini-Circuits 5 W amplifier. 123
7.2 Measured and estimated magnitude and phase of Mini-Circuits amplifier IM3 products using a single-slice model. 128
7.3 Measured and estimated magnitude and phase of Mini-Circuits amplifier IM3 products using a two-slice model. 129
7.4 Polar form of the fit to the Mini-Circuits amplifier IM3 data with the magnitude in dB, (magnitudes offset such that they are non-negative). 130
7.5 Measured AM-AM, AM-PM of the 10 W AR amplifier. 131
7.6 Measured magnitude and phase of the IM3 products for the AR amplifier. . 132
7.7 Measured and estimated magnitude and phase of AR amplifier IM3 products using a two-slice model. 133
7.8 Measured AM-AM, AM-PM of the Hittite Darlington amplifier. 134
7.9 Measured magnitude and phase of the IM3 products for the Hittite amplifier. 135
7.10 Measured and estimated magnitude and phase of Hittite amplifier IM3 products using various fitting techniques to extract the one-slice model, ($*$ - IM3L measured, $\times$ - IM3H measured, $\square$ - two-tone fit to a_3 and a_5, $+$ - nonlinear optimization fit to both data sets, $\diamond$ - GA fit to both data sets.) 136

8.1 Typical two-port frequency response of a bandpass filter with a high-side transmission zero. 140
8.2 Model of nonlinearly loaded filter. 141
8.3 VSA-based VIMA reflection measurement setup. 143
8.4 Schematic of nonlinearly loaded bandpass filter. 144
8.5 Desired and extracted S_{21} magnitude response for de-tuned filter. 146
8.6 Desired and extracted S_{21} phase response for de-tuned filter. 147
8.7 Desired and extracted S_{21} magnitude response for properly tuned filter. . . 148
8.8 Desired and extracted S_{21} phase response for properly tuned filter. 149
8.9 Comparison of extracted de-tuned filter S_{21} magnitude errors and the measured S_{11} response of the filter. 150
8.10 Simulated IM3L power output of the diode used as the nonlinear load for the filter under test. 151

E.1 Measurement system block diagram labelled with ports and interconnect designations for phase shift correction equation. 302

List of Tables

3.1 Dynamic range and phase uncertainty/error performance of published IMD phase measurement systems. 64

List of Symbols

ρ Waveform quality factor

ACPR Adjacent Channel Power Ratio

AM-AM Amplitude modulation of a DUT output resulting from the magnitude of the input signal

AM-PM Phase modulation of a DUT output resulting from the magnitude of the input signal

ANN Artificial Neural Network

ATE Automatic Test Equipment

AWGN Additive White Gaussian Noise

BER Bit Error Rate

CCDF Complementary Cumulative Distribution Function

DAC Digital to Analog Converter

DSP Digital Signal Processor

DUT Device Under Test

EVM Error Vector Magnitude

FFT Fast Fourier Transform

FIR Finite Impulse Response

HBT Hetero-junction Bipolar Transistor

IIP3 Input third-order intermodulation intercept point

IIR Infinite Impulse Response

IM Intermodulation

IM3 Third-order intermodulation distortion

IMD Intermodulation Distortion

IMR Intermodulation Ratio

LNA Low Noise Amplifier

LSB Least Significant Bit

MER Memory Effect Ratio

NVNA Nonlinear Vector Network Analyzer

PA Power amplifier

PIM Passive InterModulation

RFIC Radio Frequency Integrated Circuit

SFDR Spurious Free Dynamic Range

SNR Signal to Noise Ratio

VIMA Vector InterModulation Analyzer

VIOMAP Volterra Input Output Map

VNA Vector Network Analyzer

Chapter 1

Introduction

1.1 Motivations and Objectives of This Study

The inherent nonlinear behavior of semiconductor electronic devices is a major concern in the design of wireless products. Mitigation and management of nonlinear behavior is a focus of the microwave community as modulation formats with increasing information capacity require higher linearity in the transmit and receive circuitry to maintain signal fidelity. One of the difficulties in working with nonlinear behavior is the lack of full information about the distortion it causes using traditional characterization techniques. Therefore, one of the main objectives of this work was the development of metrology and modeling techniques for providing information about nonlinear behavior in the frequency-domain that is difficult or impossible to obtain with existing measurement techniques, particularly the measurement and modeling of nonlinear distortion product phase. The development of a behavioral modeling architecture and extraction procedure was associated with this work which led to a model capable of estimating distortion phase and magnitude asymmetries as a function of input signal level of an amplifier for the first time.

The goal of producing highly integrated or single-chip solutions for wireless analog front-ends represents another driving force in the RF and microwave industry today. Increased integration will enable smaller, more efficient communications products continuing their deployment in new markets, however this presents several new challenges including the

impact on manufacturing test. The challenges in manufacturing test will arise both in the need for more complete understanding of the nonlinear behavior of the Device Under Test (DUT) and the ability to characterize internal components in the RF front-end circuitry that lack external probe points. Thus, the other major objective of this work has been the development of a remote, non-contact characterization technique for testing an integrated communications product. The development of this novel process also enabled another key component of manufacturing metrology that of extracting the two-port transmission parameters of an integrated bandpass filter and using this information to tune the filter to the desired response.

1.2 Behavioral Modeling of Nonlinear Systems

In general, working with nonlinear behavior forces one to use tools that complicate the process of obtaining the desired solution. These include the analysis methods used to predict nonlinear operation. Ranging from simple "linearized" models to complex Volterra series analysis, these methods have limitations in either over-simplifying the problem at hand such that observed operation cannot be described, or by making the analysis so difficult that few designers care to employ the analysis technique regardless of its accuracy. Numerical models suffer from the same problems as the analysis types, either too simple or so complicated that not only is the extraction problem difficult, but it is very difficult to build intuition about the system from its model. Thus, a straight-forward, modular modeling approach based on capturing known phenomena within the system would provide intuition without overly complicating the process. A modular approach to the model construction problem would enable creation of a model with sufficient complexity to cover observed behavior in a directed manner. The behavioral modeling development in this work follows this approach.

Measurements-based behavioral models attempt to reproduce device performance under any operating condition by extracting parameters with a relatively small number of measurements and significantly increasing the simulation speed of the circuit under test. The speed of behavioral models allows difficult tasks in design and manufacturing test to be accomplished. In the design phase of an RF system, behavioral models allow the full simulation of the transmit and receive paths to investigate the performance of the

design at an early stage. This type of simulation would not be possible with traditional model representations due to the simulation time required. In the testing environment, behavioral model speed allows the rapid generation of accurate test vectors that are essential to reducing test times. To be effective in these two areas, the models used must reproduce actual device and system performance, thus the models must capture all significant behavior including complex nonlinear operation.

1.3 Metrology for Nonlinear Behavior

Increased reliance on measurement-based models has generated a need for more sophisticated metrology techniques in nonlinear characterization. Traditional techniques measure implied nonlinear behavior as in AM-AM, AM-PM measurements or a simple magnitude response for multi-tone intermodulation testing. These techniques worked well for predicting device performance for constant envelope communications signals such as FM, but the lack of correlation between these measurements and the performance of components exposed to the high peak-to-average ratio of digitally-modulated signals has demonstrated the need for more advanced measurement techniques.

One of the shortcomings in common nonlinear metrology is the lack of phase information about the nonlinear distortion. This masks the image of the phenomena at work within a system that results in deviation from the desired linear performance. Techniques to provide this information are a key component in advancing the art of RF and microwave design. Another need within the community is the development of a robust means for the measurement of weak nonlinearities as previously ignored phenomena begin to impact the linear operation of communications systems. The effect of passive intermodulation from interconnect components is chief among these phenomena and the measurement difficulties stemming from the need for the high dynamic range, 150 dB or more, to characterize this behavior is difficult to achieve. The history of increasing integration in electronics points to a third area for metrology improvement, that of testing highly integrated RF components. In particular, the extraction of the parameters for a single component in a chain of devices would be very useful during manufacturing test to aid yield testing and provide final tuning information in the case of filter components. The research presented here addresses these issues and provides a set of initial solutions to these challenges.

1.4 Overview of RF Manufacturing Test Challenges

The development of manufacturing metrology techniques should be focused on speed of the measurements and the implementation cost. Since time is proportional to cost, the speed of measurements in the laboratory needs to improve since current RF factory tests consume 60–80% of the total test time for RFICs [1]. Increasing measurement speed is not only about creating techniques that can rapidly measure device parameters, it can also include reducing the number of measurements necessary for a given device. Here a behavioral model-based approach to testing that utilizes a minimum number of measurements can decrease test time by comparing the specified model response and the same model response from the extracted parameters of the DUT [2].

The cost of test is a serious concern for all segments of the electronics industry including the microwave community. It is estimated by Akbay that the cost of test for RF and mixed-signal components approaches 40% of the total manufacturing cost [3]. In general the cost of test continues to increase such that the Semiconductor Industry Association (SIA) predicts by 2010 that it will cost more to test a transistor than to manufacture it [4]. The cost of attempts to utilize the test techniques of BIST and test pins from the analog and digital worlds is prohibitively high. RF BIST functionality can increase the die size by as much as 100% and is therefore discarded after the first release, [1]. In addition, the increased circuitry in a BIST solution will decrease the performance of the main circuit and additional pins for testing purposes increases the cost of the package and the test time since more RF pins need to be tested. A final consideration of the cost of RF metrology is the cost of the test equipment itself. In an ATE environment, RF equipment is very expensive, on the scale of two orders of magnitude more expensive than non-RF equipment [3]. The multi-million dollar cost of this test equipment translates to dollars-per-minute on the factory floor so techniques that utilize flexible instrumentation to perform multiple measurements is highly desirable.

1.5 Original Contributions

The issues of capturing complex nonlinear phenomena with behavioral models and enhancing nonlinear metrology for device characterization and manufacturing metrology were the motivating factors in the work presented here. With the goals of simplicity and

ease of implementation in mind, the research presented here sought to study and enhance traditional nonlinear system analysis and measurement practices used throughout the microwave industry. Considerable effort was placed on performing laboratory measurements to establish the behavior of the components within a modern wireless device. Chapter 5 of this work presents for the first time a study of the phenomenology of measuring characteristics of a communications device through remote means. Sections 3.4.2 and 3.4.3 present two novel measurement techniques for determining the relative phase of the intermodulation (IM) products arising from multi-tone stimulus. These techniques have a higher dynamic range than previously published works which results in lower phase uncertainty and the ability to characterize devices with weak nonlinear behavior.

Other significant work performed in the laboratory included development of high power stimulus and response apparatus for performing remote, non-contact RF system characterization and an automatic stimulus cancellation system suitable for high-power passive intermodulation (PIM) measurements or for active co-site interference suppression.

In parallel with the laboratory techniques, measurement-based model development sought to capture the complex nonlinear behavior of the systems under study. Methods to determine model architecture and parameter extraction strategies led to the development of the multi-slice behavioral model (Chapter 4). This model provides a convenient architecture for reproducing nonlinear behavior not possible with existing models. Several observed phenomena including intermodulation magnitude and phase asymmetry as a function of stimulus power can be captured easily with this model and its associated measurements, which is not possible with current models used by industry. In addition, the modeling and extraction technique enables separation of distortion into odd-ordered and baseband distortion contributions (Chapters 6 and 7). This allows characterization of the sources of nonlinear behavior that could enable design practices to mitigate the effect of these sources.

A final contribution of this work was the development of a metrology process for characterizing the two-port transmission parameters of an embedded filter using only one-port measurements. This enables in-circuit tuning of an embedded filter. This technique (Chapter 8) marks the first time that this type of characterization has been performed and represents a foundation for the development of this process into a potentially valuable tool in an RF communication product manufacturing environment.

The following lists these contributions in the order they appear:

- The development of two novel, high-dynamic range measurement apparati and techniques for measuring the phase of IM products (Sections 3.4.2 and 3.4.3).
- Development of the multi-slice behavioral model architecture and extraction procedures (Chapter 4).
- Demonstration of the ability to remotely characterize RF and microwave systems (Chapter 5).
- Application of the multi-slice model extraction procedure using intermodulation phase measurements to characterize multiple contributors to nonlinear behavior in multiple RF amplifiers (Chapters 6 and 7).
- Development of a metrology technique for two-port parameter extraction (both magnitude and phase) of an embedded filter and demonstration of the ability to tune a filter with this process (Chapter 8).

1.6 Thesis Overview

The second chapter of this thesis presents a literature review of the traditional and state-of-the art modeling and analysis techniques for nonlinear system as well as a review of the common measurement techniques employed for nonlinear characterization. Chapter 3 gives a detailed discussion of the pitfalls and performance requirements for nonlinear measurements and presents the operation and capabilities of the measurement systems developed for this research. Chapter 4 presents the theoretical development and extraction procedures for the multi-slice behavioral model. Chapter 5 presents the results of the phenomenological study of remotely characterizing RF and microwave systems, and Chapters 6 and 7 present the application of the multi-slice model architecture for the description of amplitude and phase asymmetries in multiple RF amplifiers. Chapter 8 explores the use of the IM phase measurement apparatus for parameter extraction used to tune an integrated filter for a circuit facsimile of an RF receiver. Finally, Chapter 9 contains a summary of the research performed and lists the significant results of this work.

1.7 Published Works

1.7.1 Journals

1. W. Jang, A. Walker, K. Gard, and M. B. Steer, "Capturing Asymmetrical Spectral Regrowth in RF Systems Using a Multi-Slice Behavioral Model and Enhanced Envelope Transient Analysis," *Int. Journal of RF and Microwave Computer-aided Engineering*, In Press.

1.7.2 Conferences

1. A. Walker, M. Steer, and K. Gard, "Simple, Broadband Relative Phase Measurement of Intermodulation Products," 65^{th} Automated RF Techniques Group Conf. Digest, June 2005.

2. A. Walker, M. Steer, K. Gard, K. Gharaibeh, "Multi-slice Behavioral Model of RF Systems and devices," *IEEE Radio and Wireless Conference (RAWCON)*, pp. 71–74, Sept. 2004.

3. A. Walker, P. M. Buff and M. B. Steer, "Remote detection of RF Systems via multitone sinusoidal excitation," Government Microcircuit Applications Conf., March 2004.

4. P. M. Buff, M. B. Steer, and A. Walker, "Experimental investigation of scanning electronic devices with microwave and millimeter wave signals," Government Microcircuit Applications Conf., March 2004.

5. J. H. Shaver, J. H. Shen, and A. L. Walker "A low-cost optical pachymeter for the diagnosis of glaucoma and ocular hypertension," Proc. 26^{th} Annual Int'l Conf. of the IEEE EMBS, pp. 2330–2333, Sept. 2004.

6. J. A. Janet, W. J. Wiseman, R.D. Michelli, A. L. Walker, S. M. Scoggins, "Using control networks for distributed robotic systems," 1999 *IEEE International Conference on Robotics and Automation*, pp. 1138–1143, May 1999.

7. J. A. Janet, W. J. Wiseman, R. D. Michelli, A. L. Walker, M. D. Wysochanski, R. Hamlin, "Applications of control networks in distributed robotic systems," 1998

IEEE International Conference on Systems, Man, and Cybernetics, pp. 3365–3370, Oct. 1998.

1.8 Unpublished Works

1. A. Walker, M. Steer, and K. G. Gard, "Capturing asymmetry in distortion of an RF system using a multi-slice behavioral model," *Microwave and Component Letters*, in review.

2. A. Walker, M. Steer, and K. G. Gard, "A vector intermodulation analyzer applied to behavioral modeling of nonlinear amplifiers with memory."

Chapter 2

Review of Nonlinear Analysis, Behavioral Modeling, and Measurement Techniques

2.1 Introduction

The difficulty in studying nonlinear, electronic systems arises from several sources. First among these is the challenging mathematics of nonlinear behavior of circuits and systems. Compared to the rigorous and intuitively obvious nature of linear systems theory, nonlinear analysis has not been reduced to a few simple techniques and algorithms for solving analytic expressions for system behavior. Several analysis techniques have been put forth including Volterra analysis, neural networks, and combined linear analysis with simplified nonlinear models. These analysis methods can be used on a wide range of nonlinear systems, however they have weaknesses in their ability to completely predict the operation of realistic devices and systems.

With the difficulty in determining an analytical solution for a nonlinear system, numerical simulations based on nonlinear differential equations are often employed to pre-

dict system performance. Once again this is not a trivial solution to the analysis of nonlinear behavior as the iterative numerical methods used for circuit analysis have their own limitations. These arise due to numerical noise and convergence of the discretized versions of the underlying differential equations. The simulation techniques used include traditional transient analysis, harmonic balance, and complex envelope, each with their areas of applicability. Although design by simulator can consume considerable compute resources, designer setup time, and interpretation time, with sufficient model and analysis accuracy, valuable results can be obtained that greatly aid the implementation of the desired system. The success of a simulation however, depends not only on the algorithms and numerical solutions used, but also on the validity of the model of the system or device being simulated. Almost all modern nonlinear models are based on measurements of an actual device to determine its behavior. Therefore conducting successful simulations relies on a foundation of measurement techniques and model parameter extraction.

This chapter will cover a brief description of nonlinear effects and the state-of-the-art in nonlinear system analysis, behavioral modeling, and measurement techniques. These concepts are all inter-related, so no technique employed in one can ignore the implications on the other tools used in the design of circuits and systems. In particular, this review will cover the current state-of-the-art in understanding and modeling techniques for capturing the frequency-domain complexity of nonlinear operation in RF and microwave circuits and systems.

2.2 Description of Nonlinear Behavior

Nonlinear behavior in analog and RF circuits results in system output not seen in linear systems. When viewed in the time domain, nonlinear effects may appear as clipping or distortion in the reproduction of an input signal. When analyzed in the frequency domain however, the impact of nonlinear effects becomes much more clear and quantifiable.

The most recognizable of the effects of nonlinear operation is the generation of additional frequency content at the system output not seen in the input stimulus. This effect for a two-tone stimulus and an arbitrary nonlinear DUT is shown in Fig. 2.1. This arises from higher order combinations of the input stimulus with itself and other signals within the circuit to generate frequency components at the intermodulation products and

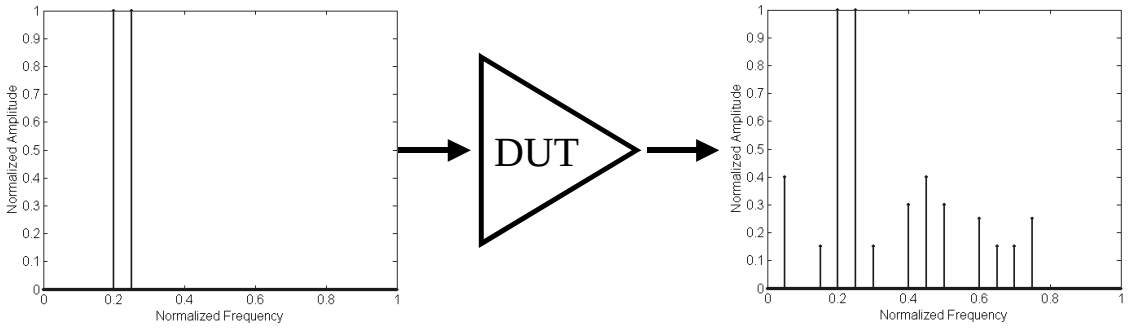


Figure 2.1: Response of a nonlinear system to discrete tone stimulus.

the harmonics of the input signals. This frequency content in the form of $\omega = m_1 \cdot \omega_1 \pm m_2 \cdot \omega_2 \pm \ldots m_N \cdot \omega_N$ for N input signals to the DUT, (where the m_i are integers), results in distortion of the original signals. The IM products that arise near the original input signal (separated by the difference frequency between the interacting stimulus ω_i and ω_j) are particularly troublesome as they cannot be filtered out. The harmonics and higher frequency IM products generated are of less interest in a narrowband system, except in the case of a spurious emissions or a multi-functional system where these spectral products may corrupt signals of interest in a different frequency band than the one from which they were created.

Saturation and threshold effects are also commonly observed in nonlinear devices. These effects limit the ability of the circuit to faithfully reproduce the system input, typically occurring in amplifier circuits where amplitude modulation at the input does not translate linearly to modulation at the output. These effects have become quite important as modern digital communications systems rely on modulation formats that vary both in amplitude and phase and therefore require high linearity in their transmit stages.

One of the least understood nonlinear effects is that of memory. In brief, all time-dependent effects within a nonlinear system are lumped together as memory effects. These take the form of wide-band memory effects such as those contributed by input and output linear networks in sequence with a nonlinearity [5], narrowband effects due to multiple nonlinear processes in sequence, feedback within a nonlinear circuit, and thermal processes

contributing to nonlinear behavior. Research efforts have recently been focused in this area as there are a several publications dealing with the subject of memory effects, [5–10]. Memory effects add complexity to the generation of IM products as they result in phase rotation of the multiple contributors to individual output spectral products that results in sizeable discrepancies in the predicted intermodulation performance through the classical nonlinear analyses.

2.3 Review of Analysis Techniques

2.3.1 Small-signal Analysis

Small-signal analysis is the simplest of the techniques for dealing with nonlinear circuits. The accepted definition of the small-signal region is the condition where the stimulus level does not result in a change in the bias conditions of the circuit. In this method, the nonlinear equations describing a system are solved for the quiescent or DC bias point. The equations are then linearized about this point under the assumption that the signal levels within the circuit are so small that they will not result in any deviation away from this operating point. With this approach, a closed-form solution to the circuit operation can be computed without iteration, thus making this method appropriate for hand-calculating a system response. The limitations of this approach are obvious as a linearized system cannot possibly predict nonlinear behavior and thus the utilization of this method is restricted to initial linear analysis to start a design, but does not offer a mechanism to predict actual circuit operation under large-signal stimulus.

2.3.2 Large-signal Analysis

To account for the nonlinear behavior of microwave devices and systems under excitation that do not hold to the linear small-signal assumption, a large-signal technique must be used. In large-signal analysis, the nonlinear equations that determine device operation are used either to solve for a solution at each operating point determined by the instantaneous signal levels within the circuit or are used to construct a model of the system that works over a wide operating range. The former method is used for numerical simulation of a system while the latter is used to predict system performance based on knowledge

about the device. The techniques of Volterra-series and power-series analysis are two of the most popular methods for performing nonlinear analysis.

Volterra Analysis

Volterra analysis is based on the concept of describing a nonlinear system through nonlinear transfer functions. These transfer functions are analogous to those in linear systems theory. Nonlinear transfer functions differ in that they are an extension of the single order linear system transfer functions to transfer functions of higher order. These transfer functions of higher order are multi-dimensional Fourier transforms of the Volterra kernels. The classical form of the Volterra-series approach is shown in Fig. 2.2, which shows the contributions to the system output via each of the kernels. The kernels themselves are n^{th} order impulse responses of the form $h_n(t-\nu_1, t-\nu_2, ..., t-\nu_n)$ that are used to find the response of the individual orders using an n^{th} order convolution integral with the input signal $x(\nu_i)$. A Volterra-series analysis defines the output as

$$y(t) = \sum_{n=1}^{\infty} y_n(t) \tag{2.1}$$

where $y_n(t)$ is given by the convolutional integral,

$$y_n(t) = \int_0^t \int_0^t \cdots \int_0^t h_n(t-\nu_1, t-\nu_2, ..., t-\nu_n) x(\nu_1) x(\nu_2) \cdots x(\nu_n) \partial\nu_1 \partial\nu_2 \cdots \partial\nu_n \ . \tag{2.2}$$

Here h_n is the kernel and $x(\nu_i)$ is the input to the system with an order-dependent delay ν_i. A useful feature of the Volterra method is that it automatically captures the memory effects of a nonlinear system, encapsulating these within the kernels regardless of the form of the input signal $x(\nu_i)$. This suggests that Volterra analysis can be used to fully describe any system under study. Unfortunately, the difficulties in practical application of this analysis technique precludes its widespread use.

The difficulty in employing this analysis is determining the kernels h_n for all but the lowest order terms. Analytical determination of Volterra kernels involves using nonlinear circuit primitives to arrive at a closed-form expression for the circuit equations. This approach becomes difficult as the analysis attempts to capture increasingly nonlinear operation. Boyd et. al. in [11], presented a measurement-based technique that was able to extract the first- and second-order kernels, but the range of applicability for nonlinear systems were those in which the nonlinear distortion products were below 40 dBc. In [12], the authors

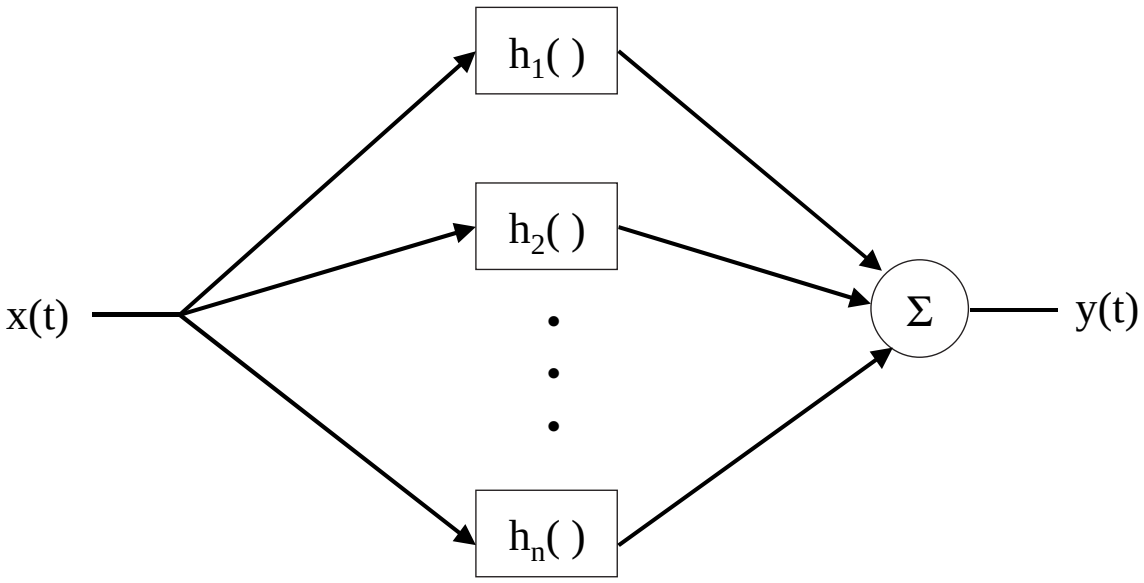


Figure 2.2: Depiction of Volterra-series.

presented the results of extracting up to the fifth-order kernel for a PHEMT amplifier. In practice, only kernels up to the third-order can easily be extracted which limits the use of Volterra analysis to weakly nonlinear systems [13]. While there is no rigid definition of weak nonlinear behavior, it is traditionally defined as systems that do not experience significant saturation, dead-zone, or hysteresis effects.

Power-series Analysis

The power series analysis is a simplification of the general Volterra-series approach. As shown in [14], the power series model assumes that the nonlinear behavior is memoryless, i.e. there cannot be any order-dependent time delay through the nonlinearity with this analysis method. While presenting a possible serious limitation to the use of this technique, the simplification in application and the intuition it extends to the user outweighs this limitation. Some compensation for the lack of memory effects within the nonlinear components of the system can be achieved by using some of the analyses that combine linear systems with memoryless nonlinearities.

The power-series approach uses a model with the output of the system given by,

$$y(t) = \sum_{n=1}^{N} y_n(t) = \sum_{n=1}^{N} a_n x^n(t) \ . \tag{2.3}$$

In the classical approach, the series coefficients a_n are real-valued and determined either through a Taylor series expansion of the underling nonlinear IV equations, [15], or by performing a fit to observed device performance either through experimentation or simulation. A power-series analysis predicts many of the characteristics of nonlinear systems including InterModulation Distortion (IMD), harmonic production and weak saturation effects, in addition to providing an intuitive description of nonlinear operation. With the power series analysis, output products of a particular order, (e.g. third-order intermodulation (IM3), second harmonic, etc.), can be associated with a particular order of the series expansion thus providing intuition about the source of distortion within the system. Although the use of power-series analysis is common and widespread in the RF and microwave community, the classical application of the analysis does fail to capture important nonlinear behavior including memory and saturation effects.

By only using real-valued coefficients, the power-series approach can produce vector sums of contributors to spectral content with phases of either zero or 180°. This limitation means that this method cannot track changes in phase of contributors as a function of input power or frequency, which are commonly observed in real devices and systems. When nonlinear memory effects occur, the use of real coefficients will not accurately predict the system performance, thus limiting the effectiveness of this technique. In addition, the power-series model is a non-physical model that cannot reproduce hard saturation and threshold effects. The series in (2.3) is an unbounded equation in the input signal $x(t)$. Even though the series must be terminated at some finite order N, the output is still unbounded without a limit on $x(t)$. This results in an inability of the power-series model to generalize about the behavior of the system at input power levels greater than those used for extraction of the series coefficients. Polynomials quickly diverge from the function they approximate outside of the range of inputs used to estimate the coefficients, so this requires that power-series analysis must compensate for the dynamics of the polynomial fit by providing supplementary information to ensure accuracy over the desired operating range of the system being analyzed.

A modification to the traditional power-series analysis to improve the phase accu-

racy of the analysis is to employ complex coefficients, [7], [16], and [17]. Since the phase of the output spectral products are vector sums of the individual contributors produced by the proper order, the use of complex coefficients allows the analysis to track phase changes as a function of input power as the contributors change magnitude with input power. Several authors have used this vector addition concept to explain memory effects in amplifiers that cannot be captured using real-valued coefficients, [8], [18], and [19]. In order to use a complex power series analysis it is necessary to measure both the phase and the magnitude of the system output. While this is a standard procedure in single-tone AM-AM, AM-PM testing, the same is not true for multi-tone analysis or complex modulation analysis since measurement of the phase of the new spectral products is quite difficult. Even with complex coefficients, power-series analysis cannot capture all possible memory effects since a complex coefficient can only produce individual order spectral components with a fixed phase. Thus, all output spectra produced by a particular order will have the same phase. For a two-tone analysis, this implies that the IM products for a particular order will have the same magnitudes and the same phase (ignoring an arbitrary phase constant associated with the initial phase of the stimulus), although these will not be limited to the 0° or 180° values imposed by a real-valued power series analysis.

2.4 Review of Behavioral Modeling Techniques

Behavioral modeling differs from that of physical modeling efforts in that the model architecture is only loosely tied, if at all, to the underlying construction of the device or system. While the behavioral modeling approach uses expert knowledge of the system to be modeled, in general, specific device features, such as nonlinear capacitances or transconductances in transistors, are not explicitly employed in a behavioral model. One of the driving forces behind behavioral modeling is that observability of internal processes is becoming more difficult as the physics at work within modern devices becomes harder to quantify. The complicated nature of a completely physical model requires detailed numerical simulation to predict device operation that becomes prohibitively expensive in time and compute cycles as the number of devices increases. The simulation of modern electronic designs, particularly wireless systems, would not be possible using physically accurate models. Thus, behavioral models that dramatically reduce the number of equations to solve during sim-

ulation allows analysis of complicated systems and circuits to aid the designer at an early stage.

Behavioral modeling is always based on measurements or device-level simulations of system operation. Hence measurement techniques for extracting the model parameters are very important and cannot be overlooked when designing a new model architecture. Since behavioral modeling is a technique for producing an approximation of real system operation, there are an infinite number of possible model formulations. The limitless flexibility in model design opens the door to designs that accurately reproduce system behavior, but do not increase the intuition of the designer that should be the goal of all good modeling approaches. This section will cover the commonly used model architectures and the state-of-the-art in the behavioral modeling community.

2.4.1 Common Nonlinear Behavioral Models

Measurement-based behavioral models have existed for some time in the RF and microwave community as well as other scientific disciplines, most notably in biology. Several approaches have been used to capture nonlinear operation both with and without memory effects. The power-series and Volterra-series analysis approaches in the previous section have also been used as the basis of many modeling efforts. Hybrid approaches using memoryless nonlinear functions and linear networks to capture memory effects have also been used extensively.

Volterra Series Modeling

The Volterra-series formality has been used by numerous groups to produce behavioral models, [11], [12], [14], and [20–28]. The difficulty with using the Volterra method for measurement-based models is at least as difficult as formulating the analytical representation of this model. While Chua and Boyd, [11], have written several papers on the procedure of measuring the Volterra kernels directly in the frequency domain, their efforts were limited to weak nonlinearities with only third-order and lower-order kernels considered. The complexity of measuring the kernels with the limited applicability to strong nonlinearities has limited this approach to very few real applications.

The work of Maas [15], [21, 22] and Pedro and Carvalho [26] with Volterra models

has focused on developing analyses to predict third-order IMD and magnitude asymmetries in nonlinear circuits, primarily power amplifiers. This work uses circuit-level models built with circuit primitives consisting of linear and nonlinear passive components and current sources. Maas developed the method of nonlinear currents which is a modification of Volterra series analysis in order to predict nonlinear behavior using these circuit models. The nonlinear currents method simplifies Volterra analysis by only requiring the computation of the frequency components of interest for each order of the Volterra kernels instead of the entire frequency response of the kernel. The computation of the kernels proceeds serially from first order to the highest order of interest where the output from each lower order comprise the current source(s) responsible for the next higher order. While this technique was used by both groups to predict intermodulation performance of real amplifier circuits [23] and [26], the use of a Volterra basis limits this technique to third-order models since computation of higher-order current sources becomes intractable. Thus it cannot handle large signal nonlinear behavior where the response can no longer be considered weakly nonlinear.

Another group utilizing a modification to the Volterra series formalism is that of Verbeyst and Bossche. Their development of the VIOMAP, (Volterra Input Output Map), also sought to produce a model based on a subset of the Volterra kernel's ability to cover any frequency. In this work the VIOMAP is extracted using standard active load-pull type measurements with the map consisting of kernels at specific frequency points of interest for the response of the DUT [27]. By restricting the measurements to a subset of the possible load impedances usually measured in a load-pull analysis, the VIOMAP results in faster DUT characterization. The extracted VIOMAP does a good job of predicting the gain of a weakly nonlinear system to varying load impedance and has been suggested as being the S-parameter equivalent for nonlinear devices. The VIOMAP is not restricted to load-pull prediction and has been used by this group to generate pre-distortion techniques that can improve the ACPR of an amplifier compared to pre-distortion computed from single-tone nonlinear characterization [28].

Power-series Modeling

The power-series modeling approach is the workhorse of the microwave community. This technique is commonly used to predict the linearity of nearly every component

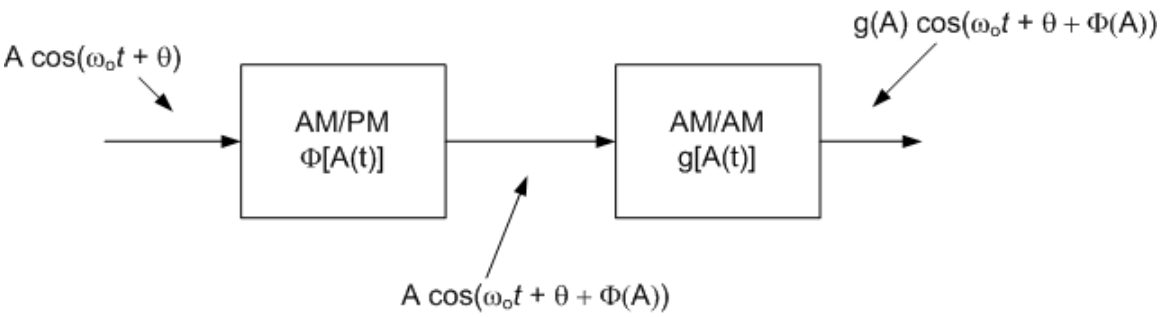


Figure 2.3: AM-AM, AM-PM model representation with the amplitude and phase modulation components considered separately (after [30]).

of a microwave circuit or system including power amplifiers, low-noise amplifiers, and mixers, [29]. The traditional usage of this model involves measuring the single-tone AM-AM behavior of a nonlinear DUT and then fitting a real-valued polynomial to the resulting output as a function of input voltage. This measurement is performed in the frequency domain so it remains to translate the RMS values measured to those useful for a time-domain, instantaneous model of the form in (2.3) that can be used in any type of simulation environment, [31]. Extension to the complex-valued coefficients covered previously uses both AM-AM and AM-PM measurements that are obtainable with modern microwave vector network analyzers (VNA). One form of an AM-AM, AM-PM model is shown in Fig. 2.3 where it is assumed that the amplitude modulation and phase modulation effects are independent. Here the phase component is computed first since the amplitude modulation effect modifies the amplitude of the input signal. Another implementation of this type of model uses the complex coefficients directly, computing the output of the model as

$$y(t) = \sum_{n=1}^{N} \mathrm{Re}(a_n x^n(t)) \ . \tag{2.4}$$

As with the analytical version, this method suffers from the inability to capture actual nonlinear behavior observed in real systems. Since the power-series or polynomial model produces fixed phase for each order of the series, the model cannot differentiate between upper and lower intermodulation products generated by the same order, thus this model cannot capture amplitude or phase asymmetry in intermodulation products. In general, this model cannot reproduce many of the features necessary to predict the operation

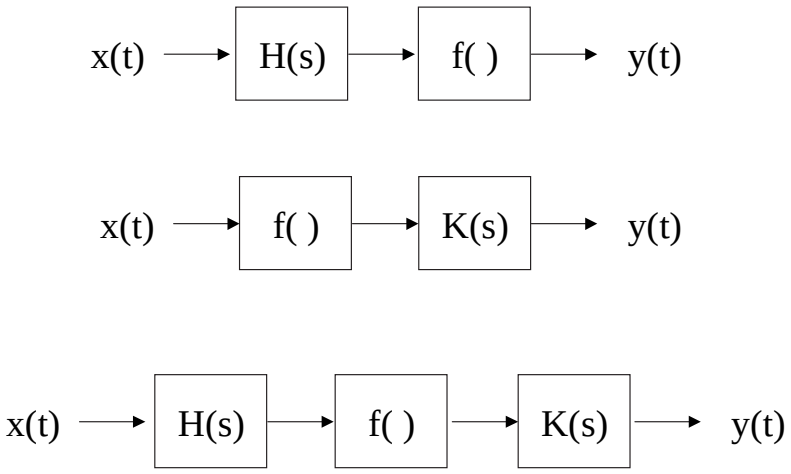


Figure 2.4: Diagrams of Wiener, Hammerstein, and Wiener-Hammerstein block model representations of nonlinear systems with memory.

of circuits with any type of memory effects. While this is expected since the power-series approach assumes a memoryless nonlinearity, the technique is valuable as an idealization of a nonlinear component and can be augmented with additional model constructs to increase modeling accuracy.

Hybrid Models

In the following, the definition of hybrid corresponds to models that use a memoryless polynomial or power-series representation for the nonlinearity combined with linear networks in a cascade network. These networks, also referred to as block models, are classified according to the order of the linear and nonlinear components as Wiener, Hammerstein, and Wiener-Hammerstein models, (Fig. 2.4). These models are widely used in the study of biological systems for simulating the complex behavior of the nervous system, vision, and numerous other processes [32–37]. They have also been applied to nonlinear circuits, [5, 7, 14]. The usefulness of the hybrid technique is that the analysis and parameter extraction for a single cascade model with a single nonlinear process is much simpler than that of the Volterra-series approach, while still allowing for the model to capture some memory effects.

Another advantage to the Wiener-Hammerstein architecture is that it represents a fundamental building block upon which more complicated models can be constructed.

Although a graphical description of the Volterra-series approach, e.g. Fig. 2.2, appears to be composed of multiple parallel block structures, in the use of hybrid structures a single nonlinear block represents the entire nonlinearity whether described by a Volterra-series or a polynomial. The field of system identification employs multiple variations of block architectures to create parallel linear-nonlinear-linear (PLNL), single-input multiple-output (SIMO), and multiple-input multiple-output (MIMO) structures [32], however this field deals with rather abstract applications of the modeling theory which do not readily translate to nonlinear circuits.

2.4.2 RF Behavioral Modeling Techniques

As the complexity of modern RF/microwave systems grows, the need for accurate models that allow simulation of the entire system has grown considerably. In addition to complexity, the push for better spectral efficiency, lower power consumption, and lower noise require that models must capture behavior that has long been ignored. Simulation at the system level also requires models to accurately predict the metrics of modern communications systems including Adjacent Channel Power Ration (ACPR), Bit Error Rate (BER), and Error Vector Magnitude (EVM). These requirements have resulted in significant developments in behavioral modeling of microwave systems and components by several groups.

At North Carolina State University, work by Steer, Gard, and Gharaibeh has focused on interpreting modeling approaches, developing new model architectures and using statistical techniques for model extraction. In [38] the authors detailed the need to use higher-order models to capture true ACPR behavior of a real amplifier and highlighted the general lack of correlation between traditional measurements of nonlinearity, such as intermodulation ratios, and ACPR. The primary reason for the lack of correlation stems from the difference in the peak-to-average ratios between the stimulus signals for measuring IMD and those of a digitally modulated signal, with the latter being much higher and thus resulting in different nonlinear behavior. Gard *et al.* in [31], developed a behavioral model using the statistical representation of the input signal, in this case a digitally modulated signal, to provide an accurate estimate of the ACPR as a function of signal power. The authors also used the results of the statistical analysis to predict the gain compression of a digitally modulated signal which differs significantly from that predicted by single-tone

AM-AM measurements. In [7] the authors further developed the statistical modeling work to predict distortion in a multi-channel WCDMA amplifier. A recent focus of this group has been the separation of correlated and uncorrelated in-band distortion for digitally modulated signals [10] and [40]. In [10], the authors presented an extension of the concept of distortion that results in gain expansion and compression of the desired output of an amplifier and that of uncorrelated distortion that contributed a noise component similar to AWGN. The result of this work showed that by measuring the uncorrelated distortion term using a feed-forward cancellation technique to separate the uncorrelated and correlated components, the measured uncorrelated distortion could be used to accurately model the measured SNR, EVM, and ρ of a CDMA signal.

Research at the Georgia Institute of Technology has focused on development of system-level behavioral models of power amplifiers for the improvement of pre-distortion linearization techniques. Ku *et al.* in [39], implemented a parallel Wiener model to capture the memory effects observed in a high-power amplifier as a function of tone spacing for two-tone stimulus (Fig. 2.5). A fitting procedure that added branches to the parallel model was applied that sought to minimize the error between the sum of the existing branches and the measured output of the DUT. No explanation however, was associated with the individual branches save for the first branch which represented a wholly memoryless component extracted from single-tone AM-AM, AM-PM measurements. The goal of this effort was to improve the model of the power amplifier such that a previously developed pre-distortion algorithm could be applied to an amplifier for which the memoryless model did not apply. In [9], the authors extended the fitting procedure by replacing the linear networks in the Wiener branches with sparse delay taps to capture the long-time constant memory effects while limiting the overall number of branches added to the model (Fig. 2.6). The sparse-tap model is extracted by solving for the optimum delay tap coefficients in matrix form from the error between the measured time-series data and the output of the model. The error formulation is shown graphically in Fig. 2.7. In this work, the model sought to predict the asymmetric IM magnitudes that arise due to memory effects. This approach resulted in good agreement with measured operation for two-tone stimulus and the authors used a Memory Effect Ratio (MER) to provide a metric for the influence of memory within a system. While these works showed good performance in modeling the memory effects in the amplifiers investigated, the fitting techniques used were heavily focused on numerical methods and as such the end result does not readily point to areas for improvement in the

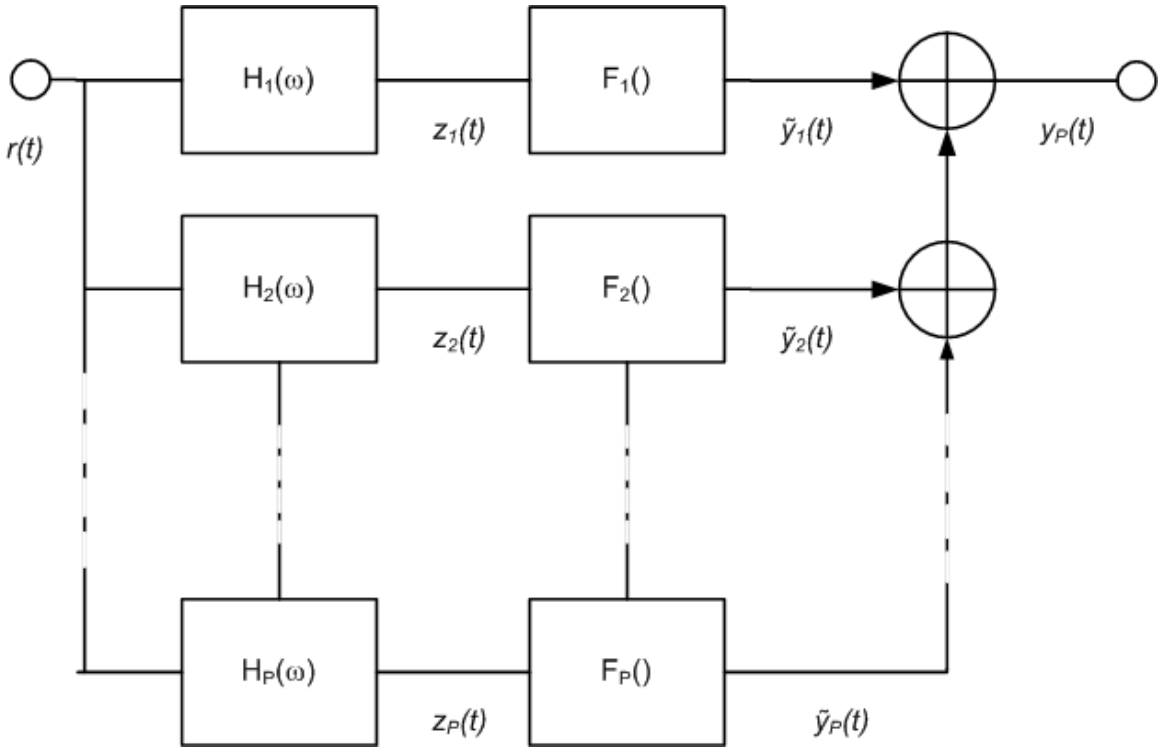


Figure 2.5: PA model for a system with memory using the parallel Wiener model (after [39]).

circuit design of the amplifiers.

Pedro and Carvalho at the Instituto de Telecomunicações in Aveiro, Portugal, have pursued equivalent circuit models (as in Fig. 2.8) of nonlinear devices to predict IM asymmetry and ACPR performance. In [41] and [42] the authors used behavioral equivalent circuits extracted from small-signal, harmonic and intermodulation distortion simulations to predict ACPR and study the effect of memory and asymmetry on this metric. Research presented in [43] sought to cover the study of intermodulation asymmetry in full, with the conclusion that this effect has numerous sources among them thermally induced nonlinear currents, upconversion of baseband currents in impedances at the input and output of active devices, and harmonic currents flowing through out-of-band impedances. Experimental investigations were used to determine the effect of bias point on the IM asymmetry, however the theoretical basis of this work focused on simple equivalent circuits that attributed asymmetries to so-called IM "sweet spots" at specific bias points. This work also stated that the asymmetric behavior resulted from imaginary components of the IM products, or in other

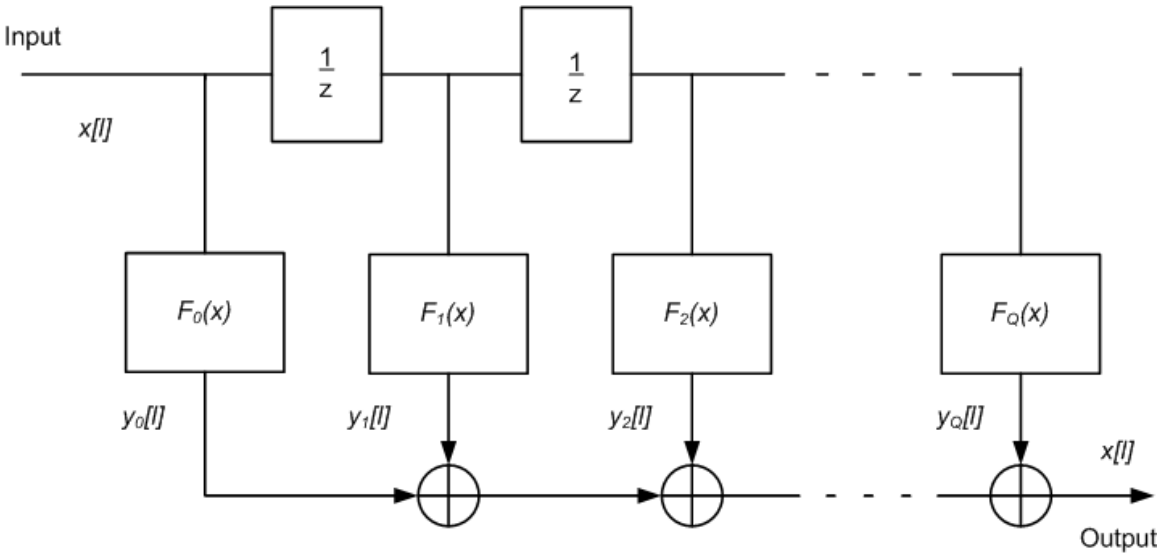


Figure 2.6: PA model for a system with memory using a memory polynomial model with unit delay taps (after [9]).

words the phase shift of the IM products, particularly those resulting from fundamental interaction with the baseband components due to even-order distortion. The description of parallel processes can accommodate multiple nonlinear effects such as odd-order transconductance distortion and the mixing of the input signal with even-order distortion (generated in the input nonlinear resistance) at the output transconductance of BJT devices.

The work of the above groups represents the state-of-the art in the behavioral modeling field, however attempts to create "*black-box*" models from measurements has existed for over 30 years. A recent review by Pedro and Maas [44], has attempted to put these efforts on a common theoretical footing. While not being completely comprehensive, the review does cover a large majority of the attempts by the microwave community to create models capable of reproducing the wide range of observed nonlinear behavior. This work produced significant background interpretation by classifying large numbers of modeling techniques into groups based on the rigorous mathematical foundation of the field of system identification. The authors showed how many of the polynomial-based models followed an FIR or IIR type filter structure of parallel branches with coefficients fit to nonlinear measurements. They also showed how the Volterra series is really just a form of a FIR filter where the nonlinearity arises from higher order versions of the input and the

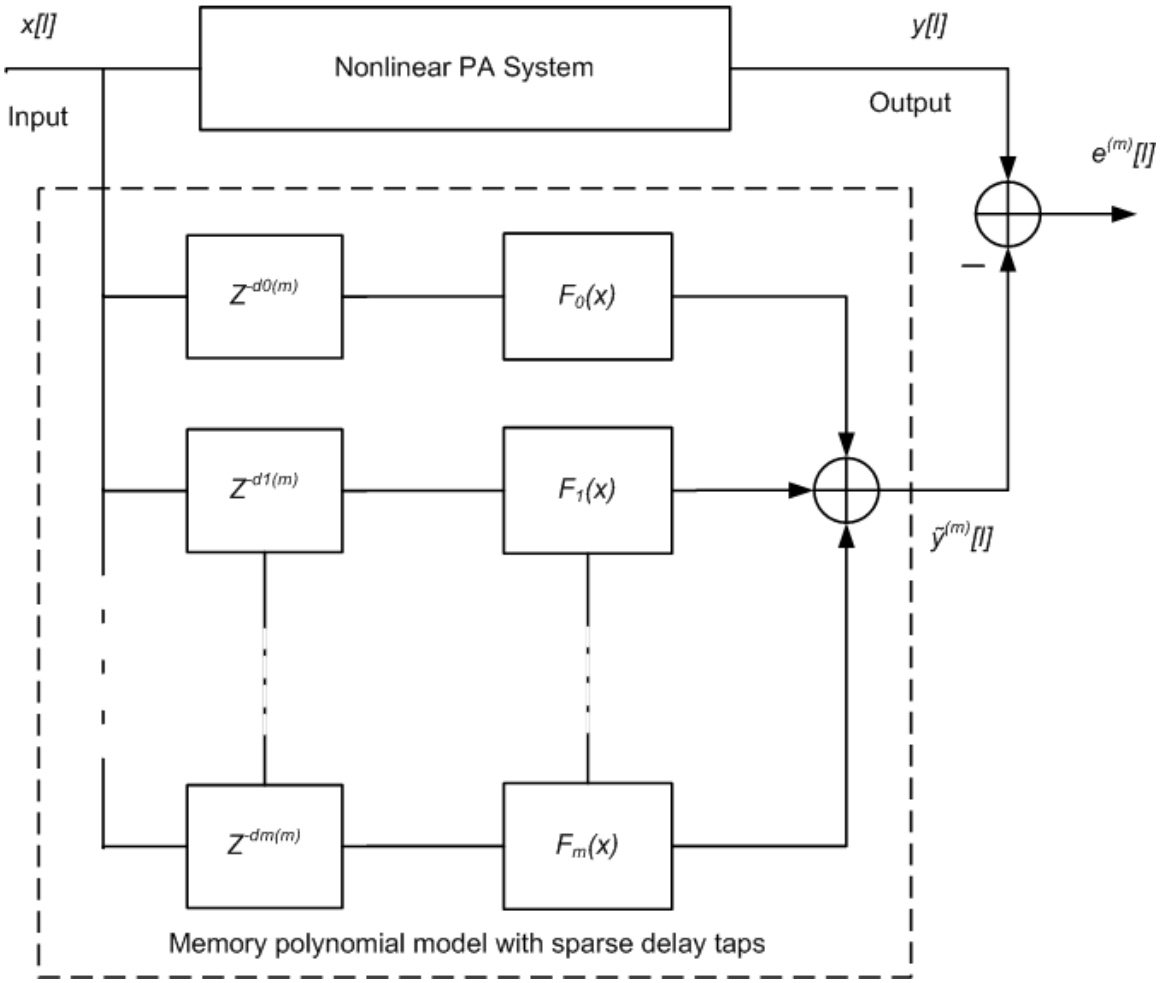


Figure 2.7: PA model using a memory polynomial model with sparse delay taps (after [9]).

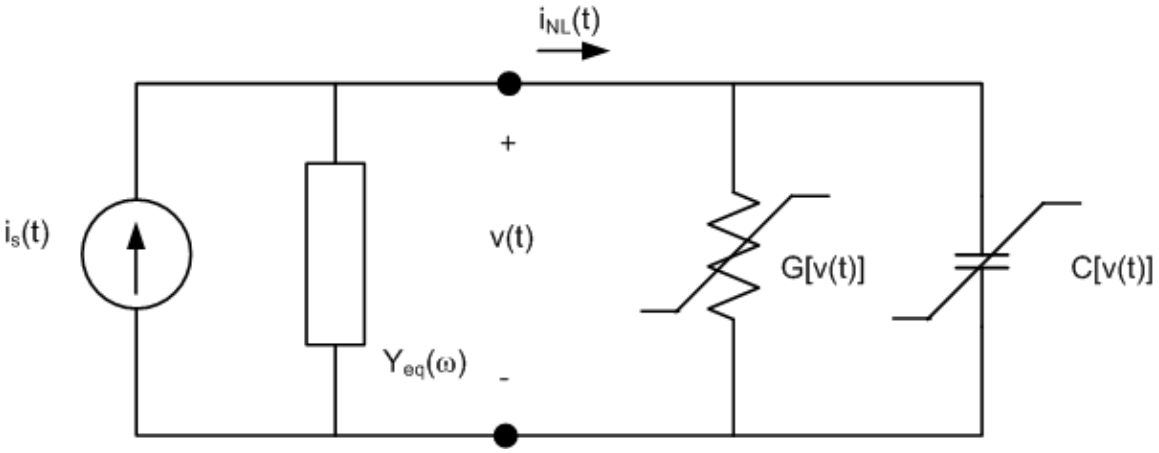


Figure 2.8: Simple general circuit for IMD asymmetry analysis, (after [43]).

Volterra kernels are given by the weighting of the filter coefficients. The use of Artificial Neural Networks (ANN) for model architectures was also shown to be similar to an FIR structure. However, the arbitrary nature of the fitting procedure for the synaptic weights in ANN models limits the intuition that can be gained from these types of models. However, ANN models do exhibit a better ability to generalize system output than polynomial-based models that rapidly deviate from observed behavior outside of the range of data used to extract the model.

2.5 Review of Measurement Techniques

Measurements on linear systems are quite straightforward and simple to perform. Microwave instruments such as the Vector Network Analyzer (VNA) are designed explicitly for making linear measurements on devices. The measurement of nonlinear characteristics is much more difficult than the linear case. Obstacles to reliable measurements arise from the prevalence of nonlinear behavior in electronic devices. This presents practical difficulties to performing good measurements with few good solutions. The widely used multi-tone testing of devices for measurement of intermodulation products is conceptually simple, but subtle phenomena must be guarded against to ensure that the measurement

system does not distort the measured nonlinear parameters. Metrics for quantifying the effects of nonlinear components under realistic stimulus signals have introduced new concepts in measurement, but these metrics are several layers of abstraction above the low-level phenomena making interpretation of the results difficult to decipher. Since time-domain instrumentation does not typically possess the dynamic range required to fully characterize RF/microwave components and systems, the measurements described here are conducted in the frequency-domain.

2.5.1 Small-signal Measurements

Compared to the time-domain methods used in analog and digital testing, the majority of measurements in the RF and microwave domain are done with frequency domain instruments. The most common instrument for measuring linear performance of multi-port devices in the frequency domain is the vector network analyzer. By relying on the linear assumption that no additional spectral content is produced by the device under test, this instrument can characterize all of the parameters necessary to define the network equations describing the current and voltage behavior at the DUT ports through the scattering or S-parameters. The linear nature of these measurements allow sophisticated error models to remove all of the systematic error sources within the measurement apparatus, both phase and magnitude errors due to impedance mismatch, dispersion, and loss. Many microwave components, (including cables, connectors, passive devices, and antennas), remain linear over a wide dynamic range of power levels so these measurements are sufficient for fully defining system operation. However, for devices with nonlinear behavior, the small-signal analysis techniques do not apply since these cannot capture the effects of this behavior.

2.5.2 Large-signal Measurements

When the stimulus signal(s) injected into a DUT push the device into its nonlinear regime, the measurements are classified as large-signal. In this region the operation of the device is dependent on the magnitude of the stimulus signals used to measure the device performance and so simple constant gain, loss, and phase shift parameters become functions of stimulus power, frequency and history. The following will summarize the common techniques in making large-signal measurements and the critical measurement concerns.

AM-AM, AM-PM Measurements

The simplest measurement procedure for quantifying nonlinear behavior are the AM-AM, AM-PM measurements. These measurements are taken by driving a DUT with a single-tone, constant frequency stimulus and varying the tone power. The AM-AM component, relates the effect of sweeping the DUT input amplitude to the amplitude of the DUT output. This is done strictly on a magnitude basis for a single tone, thus representing an RMS or peak magnitude. An AM-PM measurement tracks the change in phase of the DUT output compared to the input phase as a function of the input power. Both of these characterizations essentially measure the deviation of the device operation from the linear case and are not an explicit measure of the nonlinear behavior. In addition, this technique only characterizes the device operation at a single frequency, thus multiple measurements must be performed in order to capture the carrier frequency dependency of the nonlinear effects. Since the stimulus used in this measurement is a single frequency, it cannot capture any of the memory effects that correspond to baseband impedance or even-order distortion. Additional difficulty in relying on these measurements to describe device functionality comes from the poor correlation in single-tone performance to operation under realistic stimulus scenarios such as digital modulation [13] and [45].

Two-tone Measurements

The use of a stimulus comprised of multiple non-commensurate sinusoidal tones is the most widely used technique for measuring nonlinear behavior. As described previously, nonlinear operation results in the generation of additional spectral content and it is this content that is used to provide metrics on the linearity or lack thereof of a DUT. The well-known metrics of third-order Input Intermodulation Intercept Point (IIP3) and InterModulation Ratio (IMR) are used heavily in the microwave industry for assessing the performance of devices and systems. This measurement however, only quantifies the magnitude of the nonlinear spectral products and cannot provide any information on the phase of the nonlinear response. Even though the two-tone measurement technique is conceptually simple, there are multiple pitfalls that must be avoided or managed when making these measurements and these difficulties vary with power and frequency of the stimulus used.

The most basic two-tone measurement apparatus is shown in Fig. 2.9. In this setup, two RF signal generators are used to provide the stimulus, a microwave combiner

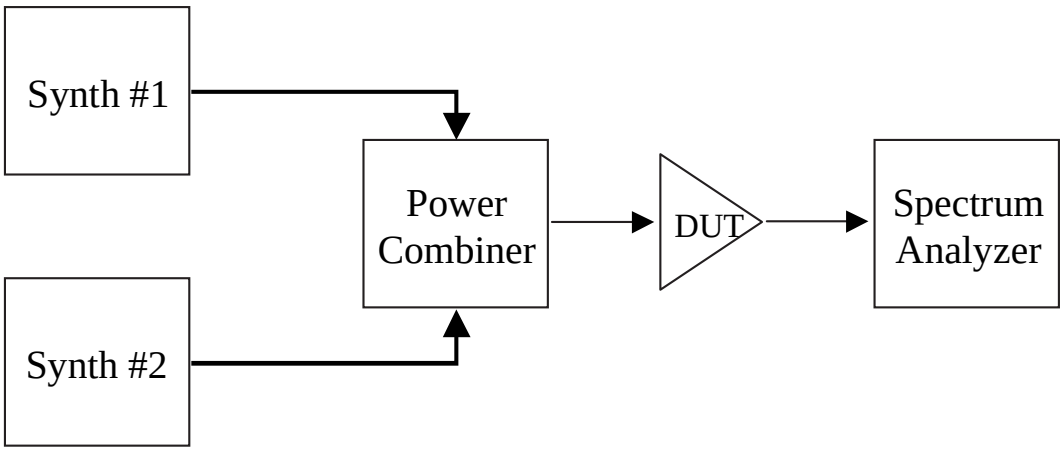


Figure 2.9: Basic two-tone measurement apparatus.

sums the signals for injection into the DUT and a spectrum analyzer is used to measure the DUT output power spectrum. In considering the potential sources of measurement corruption, the underlying theme is that nonlinear effects are numerous in origin and can occur anywhere within the device and the measurement apparatus [46].

The most important aspect of the measurement system in Fig. 2.9 is the isolation of the two signal generators. With improper levels of isolation between the generators, presence of the output of one generator at the output of the other will result in reverse intermodulation. The reverse intermodulation arises from nonlinear behavior at the source output when exposed to a tone different in frequency than the one it is producing. The interaction of the two frequencies at the output of the sources produces nonlinear products that propagate to the DUT input and will pass through the linear path of the DUT, appearing at the same frequencies as nonlinear products generated within the DUT. Eliminating this source of distortion can be achieved with a high-isolation power combiner architecture, such as a coupled-line hybrid, or by introducing isolators between the source outputs and combiner inputs. The use of these components results in a system with reduced bandwidth as isolators and hybrid combiners have no more than octave bandwidth. While wide-band attenuators could be used in place of isolators, this decreases the available stimulus power and wastes power.

Of almost equal importance to making good measurements is the receiver dynamic range. The front-end of the spectrum analyzer can also exhibit nonlinear behavior when exposed to large input signals. When conducting two-tone testing with high-levels of stimulus at the DUT output, the nonlinear operation of the spectrum analyzer receiver can contribute spectral products at the same frequency as the DUT. In the case of weakly nonlinear devices, this is of particular concern since simply attenuating the stimulus tones prior to the spectrum analyzer will also attenuate any weak nonlinear response thus limiting the ability to detect these signals. Since the nonlinear distortion products of interest are close in frequency to the stimulus tones, filtering cannot be used as this would also attenuate the nonlinear products as well.

One scenario which requires careful attention to dynamic range is that of measuring Passive InterModulation, (PIM). The study of PIM has been ongoing since its discovery in 1947 in terrestrial antennas, and in satellite communications antennas in the 1970's [48–51, 57]. Since then the phenomena has been observed on naval ships (the "rusty bolt" effect) [53], cellular base station tower components [54–56], and more recently mobile radio antennas, [59] and PCB traces [60,61]. A general description of the cause of PIM stems from the breakdown of the linear behavior of passive components such as cables, attenuators, and combiners usually at high input powers ($\sim$ 50 W). Since these components are only weakly nonlinear, the high-power stimulus elicits only a very small nonlinear response so detection of this response requires considerable dynamic range, in excess of 120 dB. The physical cause of PIM is hard to characterize due to the difficulty in measuring the behavior directly and in the wide and varied range of material properties that can cause PIM. The most readily explainable sources of PIM are hysteresis effects in ferrous materials and dissimilar metal contacts, both well-known to exhibit nonlinear I-V responses. In addition many other processes can contribute to PIM such as: poor contacts at mating surfaces, thermal effects from point contacts, vibration, tunnelling in thin dielectric films, and dielectric breakdown to name a few, see [62, 63]. About the best that can be done to mitigate or manage the generation of PIM results from good manufacturing practices in component design including using multi-layer plating of connectors with non-ferrous materials and using high-purity electro-plating and current path metals to eliminate dissimilar metal effects resulting from impurities [63].

In general, when conducting two-tone measurements, the existence of nonlinear behavior from a variety of sources must be anticipated. Nonlinear operation can occur

both within the DUT and the measurement system and can result from thermal effects, transmission and reflection signal paths, and signal coupling from antennas or signal traces. Since these sources will add vectorially with desired products, great care must be taken to ensure that these sources do not distort the desired measurement.

Nonlinear Vector Network Analyzer

A relatively new instrument for characterizing the nonlinear properties of active microwave devices is the Nonlinear Vector Network Analyzer (NVNA). This instrument has undergone several implementations since its introduction in 1992 by Van den Broeck and Verspecht [68–79]. The current version of this system digitizes downconverted frequency bands around the fundamental and its harmonics with an 8 MHz bandwidth. Unlike the traditional two-tone measurement system, this instrument can measure the phase of the resulting nonlinear response after extensive calibration to determine the phase of discrete frequency responses in the instrument. Until recently however, this system only measured the phase of the harmonics produced by a nonlinear DUT, but has been used in [79] to measure the phase of intermodulation products in two-tone excitation experiments. The primary limitation of using this instrument is its high cost, much more than that of a traditional network analyzer.

2.5.3 Communications-Based Metrics

It is well known that the discrete-tone measurements in the previous sections have little correlation with device performance under digitally-modulated excitation, [13], [45], and [46]. The lack of correlation stems from the differences in the characteristics of discrete tones and digitally modulated signals such as peak-to-average ratio, complementary cumulative distribution function (CCDF), and spectral content. Thus modern communications standards, GSM, CDMA2000, UMTS, EDGE, etc., define several metrics based on performance under a digitally-modulated test signal. The test signal is a modulated pseudo-random bit sequence with a given CCDF description of the ideal signal power statistics. One of the metrics used in transmitter testing is the amount of spectral leakage or regrowth allowed in adjacent channels known as ACPR. Wireless standards define frequency masks with adjacent channel power limits that the system under test must adhere to remain

compliant. ACPR testing provides a functional test for spectral-regrowth limited radios, i.e. where spectral regrowth limits the performance of the radio network. Performance limitations arise from the corruption of adjacent transmit channels by the output of other transmitters. Another measure of the distortion within a system is based on the deviation of the constellation points in the I/Q domain. EVM measures the average magnitude deviation of the system's constellation points with respect to the ideal constellation point locations, essentially quantifying the modulation quality in the presence of nonlinear distortion. Additional tests including BER and functional tests are used to quantify the device performance at a system level. While it is difficult to explicitly link distortion mechanisms to the system level metrics, these measurements allow designers to evaluate performance under realistic conditions and thus provide a useful tool for analysis.

2.6 Summary

Nonlinear operation of RF and microwave devices presents a considerable amount of complexity in system design. However, the microwave community has developed a large number of tools for dealing with this complexity in the areas of analysis, modeling, and metrology. Since the strictly analytical representations of nonlinear behavior fail to capture all observed phenomena (such as amplitude and phase asymmetries in IM generation resulting from multiple parallel nonlinear processes in a system), the trend in the research community has been towards developing measurements-based behavioral models. In the following, the novel measurement and modeling techniques for characterizing nonlinear behavior developed during this research are presented.

Chapter 3

Nonlinear Metrology

The work presented here focuses on providing modeling strategies and techniques for experimental characterization of RF and microwave circuits and systems. Thus, nonlinear measurement techniques and instrumentation usage plays a crucial role. In Section 3.1, a critical review of the measurement apparatus necessary for the "routine" two-tone characterization of intermodulation magnitude is presented. This review outlines the potential pitfalls of this measurement and the additional considerations for performing the measurement at high power levels (> 1 W). Extension of the wired two-tone test is outlined in Section 3.2 that covers the modifications introduced when performing a two-tone test in a non-contact or wireless manner. The wireless characterization approach exploits the nonlinear reflection phenomena (defined in Section 3.2.1) found in electronic devices.

This chapter also presents the development of an automated, two-channel, feed-forward cancellation system for increasing receiver dynamic range and two novel measurement systems for measuring the phase of intermodulation products arising from two-tone testing. The architecture, implementation details, automation algorithm description and the performance of the cancellation system is presented in Section 3.3. An analysis of the theoretical limits of cancellation possible for the developed system is also presented. Sections 3.4.2 and 3.4.3 present the development of two novel IM phase measurement systems. The former measures IM phase relative to a phase reference at a particular input stimulus power and the latter (based on a Vector Signal Analyzer) can provide an absolute phase as a function of both stimulus power and frequency.

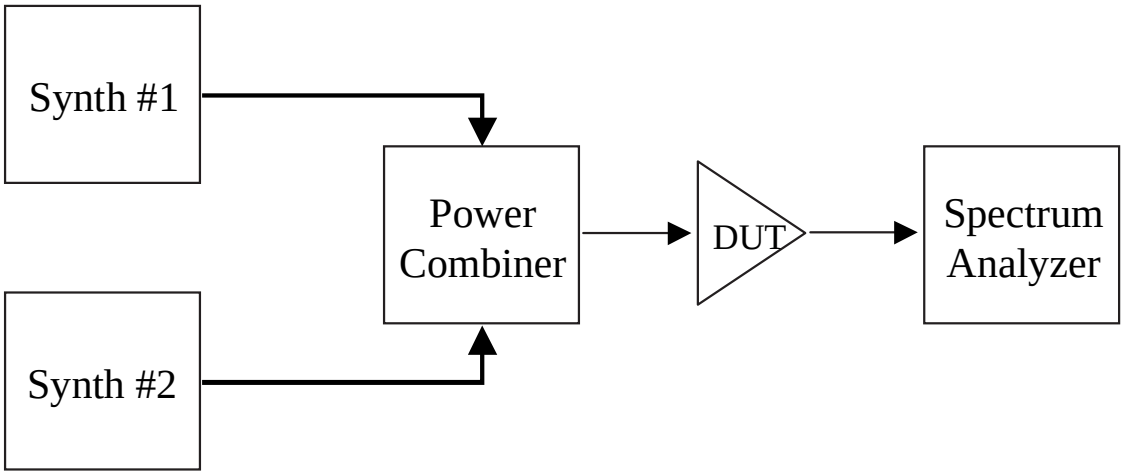


Figure 3.1: Basic configuration for two-tone measurement apparatus employing independent signal sources.

3.1 Review of Two-Tone Measurements

The traditional two-tone measurement of nonlinear behavior is well-known, widely performed, and presents multiple subtle pitfalls. A simplified block diagram of the instruments and components used in this measurement are shown in Fig. 3.1. In the following, the components that make up the measurement apparatus are reviewed and potential issues leading to nonlinear distortion generated by the components themselves are discussed. The mitigation of the system generated IM is particularly challenging when the measurement is performed at high power over a wide (greater than one octave) frequency range, therefore good measurement practices for these measurement conditions are also discussed.

3.1.1 Stimulus Sources

When using multiple signal generators to provide the stimulus signal, proper isolation techniques must be employed. The intermodulation products contributed by reverse-intermodulation between poorly isolated source contributes directly to intermodulation products that combine with those generated by the DUT. Reverse intermodulation differs from the traditional concept of intermodulation in that reverse intermodulation distortion

arises from one sinusoidal tone presented to the input of a nonlinear device and a tone of different frequency presented at the output. The resulting interaction in the nonlinear component generates intermodulation products similar to two-tones at the input of the device, in the case of source isolation, the nonlinear component is the output stage of the signal generator. As in an uncalibrated VNA, the effect of poor isolation can be seen in swept frequency measurements of intermodulation product power as ripple in the magnitude as the system IM and DUT IM combine alternately in-phase and out-of-phase. The use of isolators, attenuators, and high-isolation power combiners allows this source of IM products to be minimized.

While it may seem preferable to use a single modulated generator to produce the multi-tone stimulus, the linearity of the modulator in sources with this capability limits the use of these sources to tests of devices with pronounced nonlinearities. In current signal generators of this type, the two-tone InterModulation Ratio (IMR), (defined as the power ratio in dB of the desired sinusoidal signals to the IM products at that power level, has a maximum value of $\sim$ 75 dBc [67] which is insufficient for weak nonlinear measurements where the DUT IMR can be greater than 100 dBc at the maximum output power of the source (typically less than 20 dBm). Another issue with multi-tone signals from a single source is that laboratory synthesizers cannot generate high output power levels. Any amplification required must use an amplifier with high linearity or employ a large backoff to limit the amount of IM generated by the amplifier.

3.1.2 Stimuli Combination Devices

Devices for summing the stimulus signals from multiple signal sources include power combiners. Power combiners used for two-tone tests are overlooked as a limiting factor in the performance of a two-tone measurement system. Proper combiner architecture and termination are essential to minimizing the nonlinear response of this component. Both the Wilkinson combiner architecture and the coupled-line hybrid combiner will be considered here.

In the process of designing a broadband measurement setup that allows testing of broadband devices without configuration changes, the three-port Wilkinson combiner architecture appears to be a good choice. These combiners can be made broadband by employing multiple quarter-wave sections. A Wilkinson combiner provides equal impedance

seen at each port so long as each port is terminated with equal characteristic impedances. Because matching is achieved at all ports, this device is lossy and dissipates half of the input power of each of the combined tones in the internal resistor that terminates the fourth port. Analysis of the Wilkinson architecture proceeds by assuming the device acts as a combiner and that the input signals are at the same frequency [80]. In the analysis, the current that flows through the resistor between the two output branches is constant, and depends on the phase difference between the two input signals. Thus in this scenario, the combiner operates at steady-state with respect to the current through the internal termination resistor. When the two input signals are not of the same frequency however, the current through the termination resistor varies with the envelope of the combined signal. At high powers, this variation in current flow causes thermal memory effects that result in nonlinear distortion at the combiner output. In addition to this nonlinear process, the Wilkinson has limited isolation between the two input ports when used as a combiner. This requires the use of more components to properly isolate the stimulus sources.

A coupled-line hybrid does not employ the internal termination resistor of the Wilkinson combiner and therefore does not suffer from the nonlinear thermal effects. This combiner architecture also offers better isolation than the Wilkinson, increasing the rejection of system-generated IM from source reverse intermodulation. The tradeoff of using these combiners comes in that their bandwidth is usually limited with higher bandwidths possible but with reduced coupling factor preventing their use as an equal ratio combiner [80]. An even/odd-mode analysis of the a coupled line hybrid shows that for three of the ports terminated in a matched load and driven with a matched source, by design the hybrid will be matched at the input and have perfect isolation at the isolated port [80]. When using this hybrid as a combiner, two of the ports are driven so the "isolated" port sinks half of the input power of one of the sources. A high-quality terminator (low passive intermodulation) must be used on this port, capable of handling the power range of the stimulus without generating any nonlinear reflections. In a high-power application this means that the terminator must be capable of handling large amounts of power without generating any thermal or material-based nonlinear distortion. The thermal mode of distortion can be limited by a terminator structure having a low temperature coefficient or being composed of a long (100–300 m) terminated coaxial cable which presents a high-loss distributed load, while the material property source of distortion is minimized by appropriate plating of connectors with non-ferrous materials [57].

3.1.3 Isolators

Methods to improve the isolation between independent signal sources in a two-tone setup include using attenuators or isolators between the sources and the power combiner. The use of attenuators decreases the available power that can be supplied to the DUT so it is preferable to employ non-reciprocal isolators instead. The use of isolators must be regarded with caution as the very nature of the ferrite materials used in their construction exhibits nonlinear behavior. The nonlinearity arises from the magnetic hysteresis in the ferrite material that means that the electromagnetic fields in the device are a function of the history of the fields and the magnitude of the fields thereby violating a condition for linear operation. Because of this nonlinear behavior, ferrite isolators (or circulators configured as isolators) should never be used in a circuit path within the measurement system where both stimulus tones are present. These devices will contribute large levels of IM products and can easily overwhelm the response of even a strongly nonlinear active component. Isolators also have a strong nonlinear reflection similar to the reverse intermodulation performance of the signal generators. Thus it is imperative that when using isolators at the output of the generators, a high-isolation combiner must be employed to limit the exposure of the isolator output port to both stimulus tones in the DUT.

The impedance matching of the ferrite materials (within isolators) to that of a 50 Ω system impedance is quite difficult and limits the frequency range of isolators to an octave or less [104]. This constrains the ability to build a broadband measurement setup considerably. A benefit of this narrow passband comes from the elimination of bandpass filters to attenuate harmonics of the stimulus signal produced by the sources. Harmonics of each source can interact with even-order nonlinear processes and the fundamental of the other tone to produce odd-ordered distortion products that occur at the expected IM frequencies of odd-order distortion arising from the interaction of the two fundamental stimulus tones.

3.1.4 Stimulus Amplification

In high-power two-tone measurements it is necessary to use power amplifiers to increase the magnitude of the stimulus tones. Here high-power refers to measurements where the stimulus power is greater than the output of typical RF/microwave signal generators

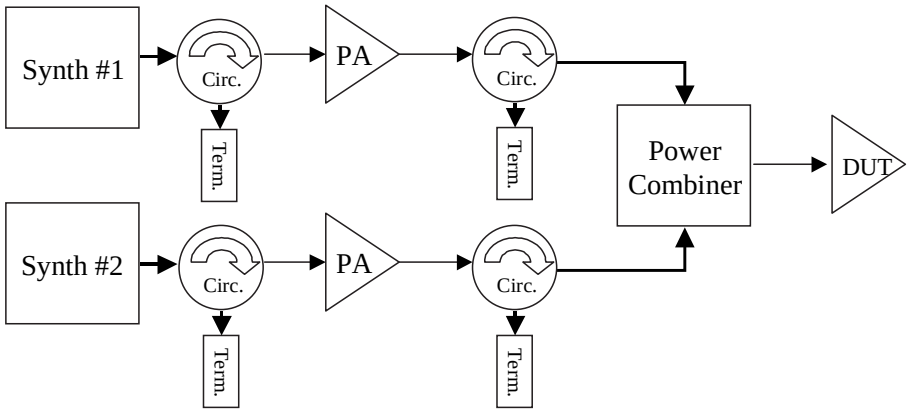


Figure 3.2: High power two-tone measurement system with independent stimulus paths.

that produce less than +20 dBm (100 mW) of output power. Amplifiers are also inherently nonlinear thus producing IM products when exposed to a two-tone input signal. For example, an Amplifier Research 10W1000C instrumentation grade, linear amplifier has an output third-order intercept (OIP3) specification of +50 dBm [105]. This means that if the amplifier is producing 10 W of power in each tone of a two-tone signal, the resulting IM products will only be 30 dB lower than the primary output (−30 dBc). This represents a large level of IM generation that would easily overwhelm the response of a weak nonlinearity under test. An example of a weak nonlinearity testing is that of PIM measurements where the IM products can be as low as −140 dBc at stimulus powers of 20 W for each tone. In terms of absolute power, this dBc ratio for the 20 W tone power (+43 dBm) would result in the power of the IM products being −97 dBm. Clearly the use of a single amplifier to increase the signal generator power would not be suitable for this type of measurement. Separate amplifiers in the stimulus signal paths as in Fig. 3.2 should be employed when greater DUT input powers are required. Again isolators need to be employed to limit reverse intermodulation at the output of the amplifiers.

3.1.5 Passive Intermodulation

High-power two-tone systems have another major area of concern not present in typical nonlinear characterizations. As discussed in Section 2.5.2 at high power levels, the linear assumption of the material properties of permittivity and permeability breakdown in

addition to a myriad of other effects that result from very high field densities in the passive components of the measurement apparatus. Under these conditions, the generally benign passive components in the system, (connectors, cables, attenuators, and combiners) begin to exhibit a nonlinear response to the stimulus signals passing through them. One of the more common sources of passive intermodulation in interconnect components is the effect of hysteresis in the connectors of components built from ferromagnetic materials such as steel. With proper silver plating of connectors, the intermodulation of low-PIM components can achieve intermodulation distortion (IMD) levels of less than -140 dBc at 20 W of power in each tone of a two-tone measurement [81].

3.2 Wireless Two-Tone Measurement

One of the novel contributions of this work is the development of methodology and techniques for performing wireless characterization of devices and systems. The ability to characterize and draw conclusions about the operation or construction of a target is more important for a system, such as a communications product, than a device, such as a transistor or single integrated circuit. By offering a point of entry into an assembled system, wireless characterization will allow additional testing of finished wireless communications and other RF products besides functional tests. As the complexity of the RF circuitry grows with multi-mode, multi-band products, the capability to perform manufacturing test or failure analysis through non-contact methods would likely decrease the time of testing which is the dominant cost in product manufacturing. While wireless characterization is not the end-all, it does offer another tool for the designer and the manufacturing line to determine the success of a product implementation. Understanding the abilities of the wireless method requires investigating the phenomenon of nonlinear reflection and the constraints on the measurement system setup.

3.2.1 Nonlinear Reflection

The receiver topology of most RF communications devices is similar to that shown in Fig. 3.3. The antenna provides a path for the incoming signal to propagate through a band-select filter that terminates in the LNA or the mixer if no LNA is present. Regardless

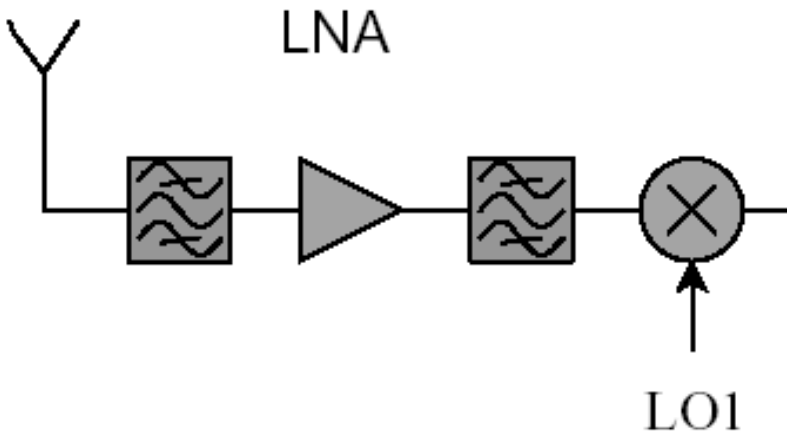


Figure 3.3: Typical RF receiver frontend with bandpass filter, LNA, image reject filter and RF mixer.

of the first active device in the receive circuit path, the front end of a communications device will appear as a linear system followed by a nonlinear component input port. One of the ways in which the nonlinear component will influence circuit operation is through a nonlinear input impedance. This nonlinear impedance will manifest itself as a mismatch between the linear feeding network and the nonlinear component. The mismatch will result in a reflection of a portion of the incoming signal back towards the antenna. Since the mismatch has a nonlinear dependence on the input signal amplitude the reflection will contain frequency content not present in the incoming signal. With sufficient input power, the reflected signal will radiate from the antenna after passing through the band-limiting filter. Thus contrary to the typical unilateral or feed-forward two-port characterization using two-tone stimulus, the wireless method provides a one-port view of the system where the nonlinear response that it generated is reflected back through the stimulus input path. A simplified diagram of the RF front-end considering nonlinear reflection and radiation under stimulus is shown in Fig. 3.4.

The production of additional spectral content provides a means whereby the response of the system to a remote stimulus can be measured. However, since the receive filter has a narrow bandwidth, only signals within the bandwidth of this filter will be able to radiate from the antenna. This assumes that the filter is reciprocal which is generally the case in communications devices. Thus under multi-tone sinusoidal excitation of the system, this stimulus creates intermodulation products near the input stimulus and thus

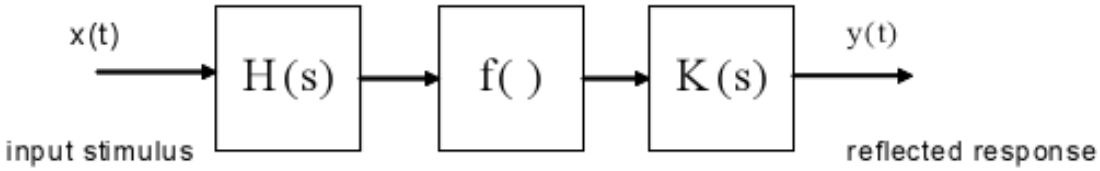


Figure 3.4: Diagram of RF frontend considering nonlinear reflection as generating a nonlinear response ($f()$) that travels through the frontend linear network and re-radiates.

within the passband of the filter. In the case of two sinusoids in the stimulus, the frequency separation should be small compared to the bandwidth of the input filter such that the intermodulation products of interest will fall within the passband and close to the primary input tones. With appropriate sweeping of the stimulus frequencies and power levels, the characteristics of both the nonlinear input of the LNA and the linear system represented by the antenna, band-select filter/duplexor, and transmit/receive switch can be determined.

Besides intended passband information, experiments have shown that significant out-of-band information can be extracted using wireless excitation that may provide additional information about the DUT. The combination of the in-band and out-of-band information could well be used to positively identify and discriminate between functional and non-functional devices. This could be used in the same manner as ACPR masks where proper device operation is indicated by a device response that falls within predetermined limits.

3.2.2 Wireless Characterization Apparatus

The apparatus for a wireless characterization through two-tone stimulus is similar to the wired case. The only difference is that antennas are used to provide the stimulus and receive the response from the DUT. Due to geometric loss, it is necessary to amplify the output of the stimulus sources, thus a setup similar to that in Fig. 3.2 should be used, employing separate amplifiers in each of the stimulus paths. The use of two antennas for providing the stimulus eliminates the need to combine the stimulus in the wired domain as the signals will combine at the receive antenna of the DUT. With separate antennas, the antennas must be directional to provide further isolation between the two stimulus

paths, limiting the possibility of reverse intermodulation as well as concentrating the electromagnetic energy around the system under test. By providing a spatially localized high field region with directional antennas, the amount of transmit power required to achieve a desired field density at the DUT is reduced when compared to a setup with low gain antennas. High-gain antennas also decrease the required broadcast power levels by maximizing the power available at the DUT for a given stimulus level. Directional antennas provide isolation for the receive antenna as well, thus limiting the introduction of high-power interferers in the receiver front-end. While a single antenna could be used after combination of the stimulus to provide both the transmit and receive functions by using a circulator, the introduction of a circulator has a deleterious effect. The circulator itself generates significant intermodulation products thereby corrupting the characterization measurements. A diagram of the wireless characterization apparatus for minimizing system produced IM is shown in Fig. 3.5.

With the instrumentation portion of the apparatus assembled, the environment in which the measurements will be taken must be considered. Ideally, the measurements would be performed within an anechoic chamber that would limit the impact of environmental effects of materials outside of the chamber. Measurements can be made without a chamber so long as they are performed at relatively low-power (< 5 W), and isolated from electronic devices as any active semiconducting element will emit a nonlinear response under RF illumination of sufficient power. Chamber-free measurements must therefore include a solid baseline to ensure that environmental effects do not corrupt the intended measurements. Again the use of directional antennas aids in limiting the exposure of the environment to the stimulus when conducting these measurements without a chamber.

3.3 Multi-Channel Feed-Forward Cancellation System to Improve Dynamic Range

The presence of the strong stimulus signal at the receiver input is a concern in both high-power wired and wireless multi-tone measurement systems. The stimulus in this case acts as interference and heavily impacts the dynamic range of the system. It is the Spurious Free Dynamic Range (SFDR) of the receiver that is important here as the

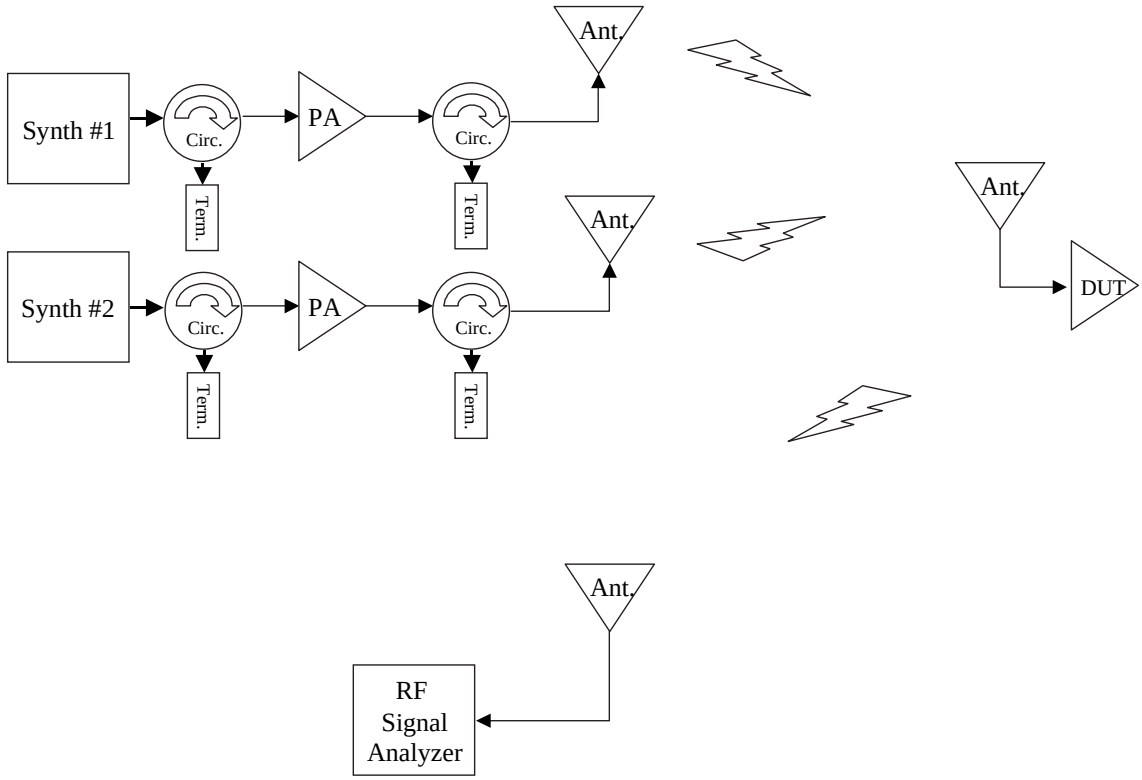


Figure 3.5: Wireless two-tone characterization apparatus designed to minimize system generated IMD.

stimulus will cause nonlinear operation of the receiver front-end and can overwhelm the desired nonlinear response of the device under test. Removal of the interfering tones would increase the dynamic range of the receiver allowing the measurement of weak nonlinear responses which would arise both in passive intermodulation and wireless characterization measurements. The problem of removing the interfering signals at the receiver input is simplified by the fact that the parameters of these signals are known, and could be classified as co-site interference. In the following, a multi-channel feed-forward cancellation system for improving the receiver dynamic range will be described.

3.3.1 Feed-Forward Justification

It would seem that with available modern signal processing capabilities, the removal of interference could proceed within the digital domain, thus eliminating the non-idealities of analog RF and microwave components. However, when the interference represents signal levels that will result in nonlinear operation of the receiver input circuitry itself, waiting to handle this in the digital domain is not an option as the input to the digital circuitry has already been distorted. The only option then is to reduce the level of the interference prior to the receiver input and this can only be done through attenuation, filtering, or feed-forward techniques.

Reducing the interference levels through attenuation is the simplest choice from an implementation standpoint, but this method reduces the dynamic range of the system by limiting the minimum nonlinear response that can be observed. The limitation arises from the effective increase of the receiver noise figure from any passive means used to attenuate the incoming signal. Filtering also seems reasonable, but given that the intermodulation products of interest lie near the interfering stimulus, filters to remove only the stimulus tones would need to have very steep skirts and it would be impossible to make these filters tunable to allow wide-band frequency stimulus sweeps. A feed-forward cancellation architecture is inherently broadband since there are no limitations on the generation of the cancellation signal and the components used in the feed-forward loop can be made very broadband. The wide bandwidth of the feed-forward technique enables considerable flexibility in the generation of the stimulus allowing wide frequency sweep ranges, arbitrary spacing of multiple tones, and sweeping the stimulus signals through both in-band and out-of-band frequency ranges for the system under investigation.

The use of feed-forward techniques for cancellation of undesired signals has been used in several areas including Frequency-Modulated Continuous Wave (FMCW) radar [82], RF amplifier linearization [83, 84], and phase noise metrology [106–110]. The similarity between these applications and the system presented here is greatest for the first and last of these areas since these too seek to suppress the main signal to increase the dynamic range of the receiver. The linearity application differs by using feed-forward cancellation to remove nonlinear distortion from a transmitter amplifier output to increase the effective power range over which the amplifier is linear. The cancellation performance of these applications has been reported as 40 dB over 80 MHz of bandwidth for the linearization application by Cavers and Smith [84] and 80 dB in the phase noise metrology application by Rubiola and Giordano [109]. The caveat on the results of the system in [109] is that this measurement required painstaking tuning of the manual phase and amplitude adjustment components, mounting the apparatus on a vibration free table, and averaging measurements over several minutes, thus this is not a practical solution for integration into a receiver system.

3.3.2 Cancellation Mechanisms

The cancellation of a continuous wave tone is a conceptually simple process of summing two signals of the same frequency and amplitude with a 180° phase difference between them. As expected, in practice this becomes more difficult. The first choice to make is whether to employ a separate signal source as the cancellation signal or use a portion of the original signal for cancellation. When using different sources it has been observed that instantaneous phase stability between sources with different frequency synthesizer architectures differs in time such that the sources are only frequency locked, not phase locked, [88]. This is the result of random phase variations in the independent fractional synthesizers employed in modern RF signal generators that are only required to have on average a particular time-varying phase relationship. This means that when using a separate source to provide a cancellation tone, in most cases the phase relationship will wander between the two tones requiring constant phase control to retain cancellation. Use of a feed-forward design removes this issue and is the ideal situation for the case where the interfering stimulus signal is available independently from the input at the receiver. In both cases the summing function can be performed with a passive power combiner.

3.3.3 Cancellation System Architecture

The architecture of the cancellation system developed here follows that of single antenna FMCW radar and a traditional feed-forward loop utilized in amplifier linearization [83] and [82]. The system differs from the amplifier linearization application since the cancellation occurs at the receiver input and it is the transmitted signal that is cancelled, not the nonlinear products in the transmitted signal. This system does not use a single loop for cancellation since the stimulus signals are generated independently allowing independent cancellation with two feed-forward loops.

The feed-forward loops perform the cancellation by minimizing the power at the output port of power combiners that carry the input signal to the receiver and the sampled outgoing stimulus tones. Since the stimulus tones are CW in nature, cancellation requires only simple amplitude scaling and phase shifting of the feed-forward component of the stimulus to track the stimulus tones at the receiver input. Theoretically and in the absence of noise, by using the same source for the cancellation as well as the stimulus, the cancellation could be perfect. Imperfections and dynamic range limitations in the components used to shift the amplitude and phase of the cancellation signal constrain the cancellation levels to less than ideal, although with manual adjustment the system developed here can achieve 60 dB of cancellation. This represents a maximum phase error of 0.06° of the 180° phase difference between the two signals assuming equal amplitudes. The estimation of cancellation performance uses the well-known equation for rejection where the difference in amplitude (in dB) of the power of the signal to be cancelled and the signal after cancellation is given as,

$$P_{1,\mathrm{dBm}} - P_{\mathrm{avg,sum,dBm}} = -10\log(\alpha^2 + 2\alpha\cos(\phi) + 1)\ . \tag{3.1}$$

The α term reflects the amplitude error in a multiplicative sense with $v_{\mathrm{cncl}} = v_{\mathrm{ref}}\alpha$, $(\alpha \geq 0)$, while ϕ represents the difference in the cancellation tone phase from 180° from the reference tone. A contour plot of (3.1) is shown in Fig. 3.6.

Hardware Design

A block diagram of the cancellation system designed by the author is shown in Fig. 3.7. The hardware design uses commercially available components mounted on evaluation boards and bench-top signal generators and a spectrum analyzer as the stimulus sources and

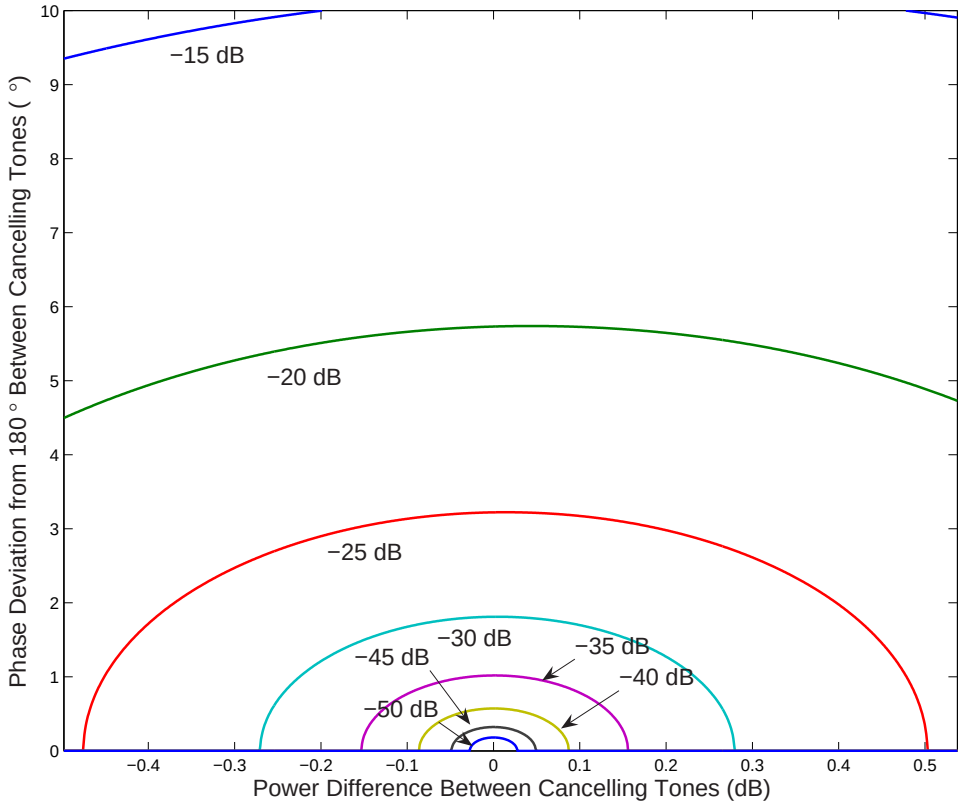


Figure 3.6: Cancellation contours depicting the amount of phase and amplitude errors allowable for a given cancellation level.

the receiver respectively [85], [86], and [87]. The system operates over the frequency band of 300–600 MHz, limited by the bandwidth of the hybrid couplers used. The phase shift and amplitude scaling functions are performed by broadband quadrature modulators where the feed-forward signal supplies the LO input and an internal quadrature splitter separates the signal into the In-phase (I) and Quadrature (Q) paths. Modulator inputs are driven by dual-channel Digital to Analog Converters (DACs), controlled by a hybrid DSP/microcontroller processor. The modulator provides both 360° of phase rotation with approximately 20 dB of attenuation control in its linear region. Feedback for the cancellation algorithm run by the microprocessor is provided by power detectors at several points in the reference and receive signal paths. The detectors produce an output voltage with a logarithmic response to their RF input. That is the response is linear in terms of decibels. The microprocessor digitizes these voltages with an on-chip, multi-channel ADC to guide the cancellation algorithm. Hybrid stripline couplers and combiners are used throughout the design for interconnect and to perform the RF signal summation.

Algorithm Execution

The design of the algorithm used to control the phase and magnitude adjustments was influenced by the desire to enable the system to adapt to changes in the interference spectrum. The goal of the algorithm is to minimize the power at the receiver input by suppression of large interferers. The dynamic nature of the co-site interference eliminated using a gradient descent method since the power minimization function is not well defined. Instead the algorithm uses a simple iterative procedure of first matching amplitude at the modulator outputs and then adjusting phase in a directed manner to achieve minimum power at the combiner output. The algorithm is simple enough to run on the 16-bit, 60 MHz fixed point processor, although the processor limits the execution speed of the algorithm such that the update rate to the modulators is well below the DAC limit of 20 MS/s. In practice an update rate of 2.7 kHz was achieved. A list of the equations and the details of the algorithm execution are given in Appendix B.

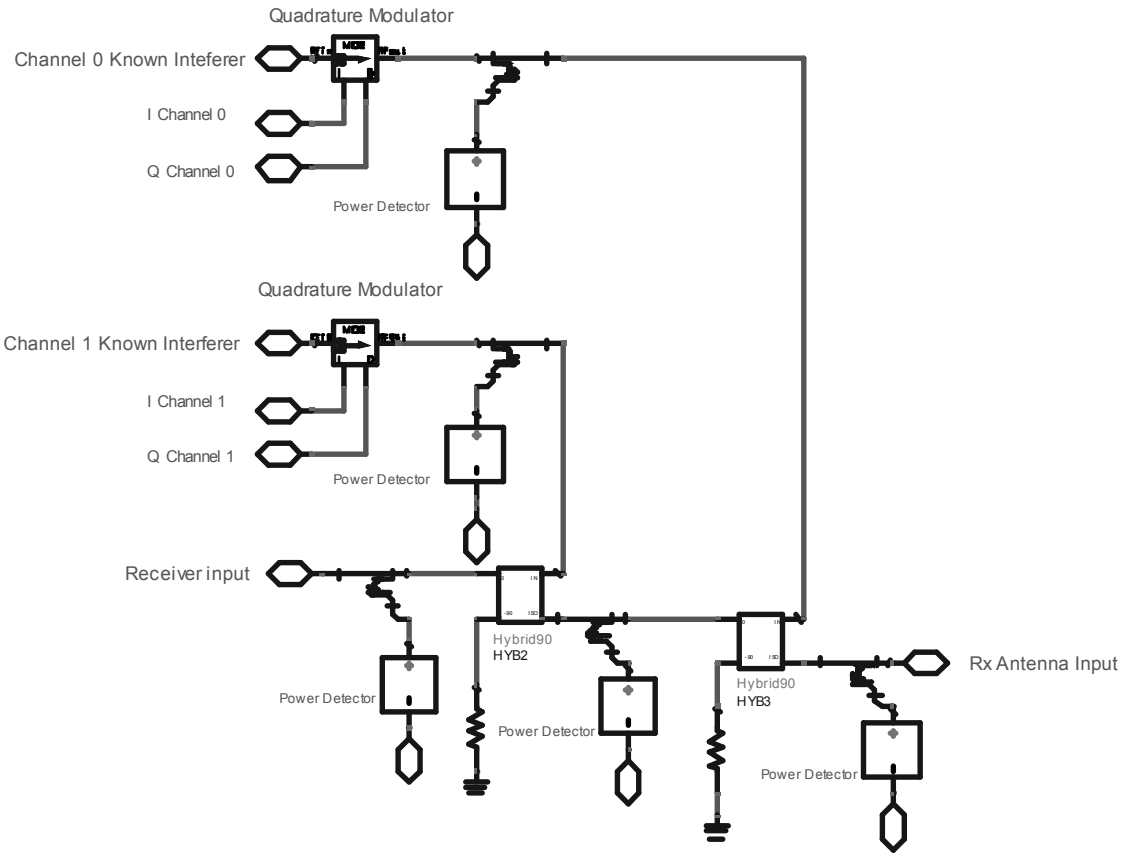


Figure 3.7: Schematic of active interference cancellation system.

3.3.4 Limitations of the Cancellation System

The primary limitations of the system described in the previous section consists of limits on operational bandwidth and the maximum cancellation level. While the cancellation system described here has an octave bandwidth limitation, design features to increase the bandwidth would ultimately be limited by the bandwidth of the interconnect components and the modulators and their matching networks. These components can be very broadband with multi-octave versions of hybrids, directional couplers and quadrature modulators quite common. The maximum cancellation level is limited by both the dynamic range of the modulators and the power detectors used in the feedback for the software control loop. The amplitude control of the quadrature modulators is limited by the linearity of the devices with available IC's having 20–40 dB of dynamic range. Phase control of the feed-forward component is limited by the ability to precisely define the amplitude of the I and Q components and decreases as the attenuation in the modulator is increased. The power detectors require a dynamic range on the order of the amount of cancellation required, thus for at least 60 dB of cancellation these components must have at least ~ 60 dB of dynamic range. Available power detectors have dynamic range in excess of 60 dB, but strict attention must be paid to the design of the interconnect and input power levels to the detectors for the power ranges used in the cancellation system to take advantage of this ability.

Secondary limitations of the system defined here are related to the response time of the individual components. The response time dictates the minimum period that can be used in the control loop operation. The critical path in this system is the settling time of the power detectors which limits the control loop rate to less than 40 kHz.

3.3.5 Quantization Noise Limit on Cancellation

Another limitation that must be investigated for this measurement system is the theoretical limits of cancellation based on the architecture of the system. Since this system uses DACs to drive the inputs of the quadrature modulator used for amplitude and phase adjustments, the effect of the quantization noise of the DACs was determined.

As discussed, the limitation on tone cancellation is given by the rejection equation where two sinusoids of the same frequency and nominally equal amplitude and phase are

combined with the cancellation power level given by

$$\Delta P = P_1 - P_{\text{sum,avg}} = -10\log(\alpha^2 + 2\alpha\cos(\phi) + 1)\ , \tag{3.2}$$

where P_1 is the power of the tone to be cancelled and $P_{\text{sum,avg}}$ is the average power of the sum of P_1 and the cancellation tone. The α term reflects the amplitude error in a multiplicative sense with $v_{\text{cncl}} = v_1\alpha$, $(\alpha \geq 0)$, while ϕ represents the phase difference between the cancellation tone and tone 1. The requirements for achieving a cancellation level of a particular value can be determined by solving for the phase and amplitude error separately as in

$$\epsilon_\phi = 180 - \frac{180}{\pi}\arccos\left(\frac{10^{(\frac{\Delta P}{10})} - 2}{2}\right)\ , \tag{3.3}$$

with $\alpha = 1$ and

$$\alpha = 1 \pm 10^{\frac{-\Delta P}{20}}\ , \tag{3.4}$$

with $\phi = \pi$.

In order to determine the limits on cancellation introduced by the DACs, it is necessary to determine the effect of quantization noise on the phase and amplitude error between the tone to cancel and the output of the modulator. Considering the voltage at the quadrature modulator output in terms of the in-phase and quadrature components, the effect of amplitude and phase noise on the cancellation voltage is given by

$$V_{I,\text{cncl}} = (V_{I,\text{ref}} + v_{n,I})\cos(\omega_{\text{LO}}t + \phi_{n,I}) \tag{3.5a}$$

$$V_{Q,\text{cncl}} = -(V_{Q,\text{ref}} + v_{n,Q})\sin(\omega_{\text{LO}}t + \phi_{n,Q})\ , \tag{3.5b}$$

where $V_{I/Q_{\text{ref}}}$ is the quadrature equivalent of the signal to be cancelled and $V^2_{\text{ref}} = V^2_{I,\text{ref}} + V^2_{Q,\text{ref}}$. The phase noise term can be grouped with the amplitude noise terms as

$$V_{I,\text{cncl}} = (V_I + v_{n,I} - \phi_{n,Q}(V_Q + v_{n,Q}))\cos(\omega_{\text{LO}}t) \tag{3.6a}$$

$$V_{Q,\text{cncl}} = -(V_Q + v_{n,Q} + \phi_{n,I}(V_I + v_{n,I}))\sin(\omega_{\text{LO}}t)\ , \tag{3.6b}$$

with an equivalent noise voltage for each path of

$$v_{n,I,eq} = v_{n,I} - \phi_{n,Q}(V_Q + v_{n,Q}) \simeq v_{n,I} - \phi_{n,Q}V_Q \tag{3.7a}$$

$$v_{n,Q,eq} = v_{n,Q} - \phi_{n,Q}(V_I + v_{n,I}) \simeq v_{n,Q} + \phi_{n,I}V_I\ . \tag{3.7b}$$

For the purposes of this analysis, the phase noise $\phi_{n,I/Q}$, is assumed to be much smaller than the control line noise voltage $v_{n_{I/Q}}$. Considering the effect of the noise voltage on just the phase performance with the assumption that the noise does not affect the modulator output amplitude begins by defining the phase error as

$$\epsilon_\phi = \phi_{\text{ref}} - \phi_{\text{cncl}} - \pi \ , \tag{3.8}$$

or

$$\epsilon_\phi + \pi = \arctan \frac{V_{Q,\text{ref}}}{V_{I,\text{ref}}} - \arctan \frac{V_{Q,\text{cncl}}}{V_{I,\text{cncl}}} \ . \tag{3.9}$$

Now taking the tangent of both sides of (3.9),

$$\tan\left(\epsilon_\phi + \pi\right) = \tan\left(\arctan \frac{V_{Q,\text{ref}}}{V_{I,\text{ref}}} - \arctan \frac{V_{Q,\text{cncl}}}{V_{I,\text{cncl}}}\right) , \tag{3.10}$$

where

$$\tan\left(\epsilon_\phi + \pi\right) = \frac{\tan\epsilon_\phi + \tan\pi}{1 - \tan\epsilon_\phi \tan\pi} = \tan\epsilon_\phi \simeq \epsilon_\phi \ , \tag{3.11}$$

since ϵ_ϕ is a small angle. With this assumption we can re-write the phase error in (3.10) as

$$\epsilon_\phi = \frac{\tan\left(\arctan\left(\frac{V_{Q,\text{ref}}}{V_{I,\text{ref}}}\right)\right) - \tan\left(\arctan\left(\frac{V_{Q,\text{cncl}}}{V_{I,\text{cncl}}}\right)\right)}{1 + \tan\left(\arctan\left(\frac{V_{Q,\text{ref}}}{V_{I,\text{ref}}}\right)\right)\tan\left(\arctan\left(\frac{V_{Q,\text{cncl}}}{V_{I,\text{cncl}}}\right)\right)} \tag{3.12a}$$

$$\epsilon_\phi = \frac{\frac{V_{Q,\text{ref}}}{V_{I,\text{ref}}} - \frac{V_{Q,\text{cncl}}}{V_{I,\text{cncl}}}}{1 + \frac{V_{Q,\text{ref}}}{V_{I,\text{ref}}}\frac{V_{Q,\text{cncl}}}{V_{I,\text{cncl}}}} \tag{3.12b}$$

$$\epsilon_\phi = \frac{V_{Q,\text{ref}}V_{I,\text{cncl}} - V_{I,\text{ref}}V_{Q,\text{cncl}}}{V_{I,\text{ref}}V_{I,\text{cncl}} + V_{Q,\text{ref}}V_{Q,\text{cncl}}} \ . \tag{3.12c}$$

In order to find the phase error in terms of the noise voltage it is necessary to expand (3.12c) using (3.7) and finding the expected value of both sides of (3.12c) yielding

$$\epsilon_\phi^2 = \frac{\left(V_{Q,\text{ref}}V_{I,\text{cncl}} - V_{I,\text{ref}}V_{Q,\text{cncl}}\right)^2}{\left(V_{I,\text{ref}}V_{I,\text{cncl}} + V_{Q,\text{ref}}V_{Q,\text{cncl}}\right)^2} \tag{3.13a}$$

$$\epsilon_\phi^2 = \frac{\left(V_{Q,\text{ref}}V_{I,\text{ref}} + V_{Q,\text{ref}}v_{n,I} - V_{I,\text{ref}}V_{Q,\text{ref}} - V_{I,\text{ref}}v_{n,Q}\right)^2}{\left(V_{I,\text{ref}}V_{I,\text{cncl}} + V_{Q,\text{ref}}V_{Q,\text{cncl}}\right)^2} \tag{3.13b}$$

$$\epsilon_\phi^2 = \frac{V_{Q,\text{ref}}^2 v_{n,I}^2 - 2V_{I,\text{ref}}V_{Q,\text{ref}}v_{n,I}v_{n,Q} + V_{I,\text{ref}}^2 v_{n,Q}^2}{V_{\text{ref}}^4 + 2V_{\text{ref}}^2 V_{I,\text{ref}}v_{n,I} + 2V_{\text{ref}}^2 V_{Q,\text{ref}}v_{n,Q} + V_{I,\text{ref}}^2 v_{n,I}^2 + 2V_{I,\text{ref}}V_{Q,\text{ref}}v_{n,I}v_{n,Q} + V_{Q,\text{ref}}^2 v_{n,Q}^2} \tag{3.13c}$$

$$\langle\epsilon_\phi^2\rangle = \frac{V_{Q,\text{ref}}^2\langle v_{n,I}^2\rangle + V_{I,\text{ref}}^2\langle v_{n,Q}^2\rangle}{V_{\text{ref}}^4 + V_{I,\text{ref}}^2\langle v_{n,I}^2\rangle + V_{Q,\text{ref}}^2\langle v_{n,Q}^2\rangle} \ . \tag{3.13d}$$

Here the expected value of the phase noise has been simplified by considering the noise voltages to be independent, zero-mean, random variables. The noise power values for the I and Q channels are considered to be equal since they are derived from DACs with an equal number of bits. This reduces (3.13) to

$$\langle \epsilon_\phi^2 \rangle = \frac{V_{\text{ref}}^2 \langle v_n^2 \rangle}{V_{\text{ref}}^4 + V_{\text{ref}}^2 \langle v_n^2 \rangle} = \frac{\langle v_n^2 \rangle}{V_{\text{ref}}^2 + \langle v_n^2 \rangle} \ . \tag{3.14}$$

Since the control voltage V_{ref}^2 is much larger than the quantization noise power, we have an expression for the expected squared phase-noise error as

$$\langle \epsilon_\phi^2 \rangle = \frac{\langle v_n^2 \rangle}{V_{\text{ref}}^2} \ . \tag{3.15}$$

A similar analysis can be performed to determine the effect of quantization noise on the amplitude matching error between the reference and cancellation signals. With the equivalent noise voltage representation of (3.7), the squared magnitude of the modulator output is

$$\begin{aligned} \mid V_{\text{cncl}} \mid^2 &= (V_{I,\text{ref}} + v_{n,I,eq})^2 + (V_{Q,\text{ref}} + v_{n,Q,eq})^2 \\ &= V_{I,\text{ref}}^2 + 2V_{I,\text{ref}} v_{n,I,eq} + v_{n,I,eq}^2 \\ &\quad + V_{Q,\text{ref}}^2 + 2V_{Q,\text{ref}} v_{n,Q,eq} + v_{n,Q,eq}^2 \ . \end{aligned} \tag{3.16}$$

The magnitude error is in terms of the mismatch constant α as $\alpha = V_{\text{cncl}}/V_{\text{ref}}$. The expected value of α squared is

$$\langle \alpha^2 \rangle = \frac{V_{\text{ref}}^2 + \langle v_{n,I}^2 \rangle + \langle v_{n,Q}^2 \rangle}{V_{\text{ref}}^2} \ . \tag{3.17}$$

Using $\langle v_{n,I}^2 \rangle = \langle v_{n,Q}^2 \rangle$ reduces this to

$$\langle \alpha^2 \rangle = 1 + \frac{2\langle v_n^2 \rangle}{V_{\text{ref}}^2} \ . \tag{3.18}$$

With the effect of the noise voltage power on the phase and amplitude error determined, the noise power limit on the cancellation performance can be computed. The form of the noise power follows the typical treatment of quantization noise in [111], i.e. a maximum error (e) of one-half a Least Significant Bit (LSB) or $q/2 = 1/(2^Q - 1)$, where Q is the number of bits in the DAC. Considering this noise to be uniformly distributed over the interval from $V_{\text{desired}} - \frac{q}{2}$ to $V_{\text{desired}} + \frac{q}{2}$ with zero mean, the mean normalized noise

power is

$$\begin{aligned}\sigma_e^2 &= \int_{\frac{-q}{2}}^{\frac{q}{2}} P(e)e^2 de \\ &= \frac{1}{q}\int_{\frac{-q}{2}}^{\frac{q}{2}} e^2 de \\ &= \frac{q^2}{12} \\ \sigma_e^2 &= \frac{1}{3}\left(\frac{1}{2^{2Q}-2^{Q+1}+1}\right) . \end{aligned} \tag{3.19}$$

Since the quantization noise power σ_e^2 was normalized for full-scale output $V_{\text{ref,max}}$, the previous noise voltage power is related by

$$\frac{\langle v_n^2\rangle}{\mid V_{\text{ref,max}} \mid^2} = \sigma_e^2 . \tag{3.20}$$

Now using (3.15) and (3.20) the limit on quantization noise on the phase error is

$$\begin{aligned}\langle \epsilon_\phi^2\rangle &= \sigma_e^2 \frac{\mid V_{\text{ref,max}} \mid^2}{V_{\text{ref}}^2} \\ \langle \epsilon_\phi\rangle &= \sigma_e \frac{V_{\text{ref,max}}}{V_{\text{ref}}} , \end{aligned} \tag{3.21}$$

where $V_{\text{norm}} = \frac{V_{\text{ref}}}{V_{\text{ref,max}}}$ is the normalized quadrature modulator output voltage . Now using (3.19) and (3.21) to find the phase error in terms of the normalized voltage output,

$$\langle \epsilon_\phi\rangle = \frac{1}{\sqrt{3}}\frac{1}{\sqrt{2^{2Q}-2^{Q+1}+1}} \cdot \frac{1}{V_{\text{norm}}} \qquad V_{\text{norm}}\epsilon[0,1] . \tag{3.22}$$

The limit on cancellation due to the phase noise can now be determined (with respect to the normalized voltage output) using (3.3) and (3.22) as

$$-\Delta P = 10\log\left(2+2\cos\left(\pi-\frac{\epsilon_\phi \pi}{180}\right)\right) . \tag{3.23}$$

Again a similar analysis relates the quantization noise in (3.19) to the amplitude limit on cancellation from (3.18) as

$$\langle \alpha^2\rangle = \frac{2}{3V_{\text{norm}}^2}\left(\frac{1}{2^{2Q}-2^{Q+1}+1}\right)+1 \tag{3.24}$$

and

$$\Delta P = -10\log\left(\langle \alpha^2\rangle - 1\right) . \tag{3.25}$$

Plotting the cancellation limits imposed by (3.23) and (3.25) in Fig. 3.8 for a DAC with $Q = 10$ bits (the number of bits used in the system described in the previous sections) shows that the amplitude error computations provide a more conservative estimate of the effect of cancellation performance of quantization noise (i.e. predicts worst cancellation possible for a given bit width). Inspection of (3.2) supports this result in that the magnitude error term *alpha* impacts the logarithmic cancellation performance quadratically whereas the impact of the phase error is compressed by the cosine function. This results from the magnitude errors having a direct effect on the modulator output voltage. The measured performance for the system described in the previous sections, the best cancellation achieved was 60 dB, at a normalized output of $V_{\text{norm}} \simeq 38\%$. This is better than predicted by the amplitude limit on cancellation by less than 8 dB. The assumption of the quantization noise as uniformly distributed about the desired control level overestimates the actual error for a single point as in this case such that the quantization error can theoretically be zero. Thus, measured performance could be better than the conservative estimate provided by (3.23) and (3.25).

3.4 Intermodulation Relative Phase Measurement

The development of two apparati for measuring the relative phase of the intermodulation products represents another novel component of this work. Sections 3.4.2 and 3.4.3 present the development of these systems. The former measures IM phase relative to a phase reference at a particular input stimulus power and the latter (based on a Vector Signal Analyzer) can provide an absolute phase as a function of both stimulus power and frequency. The first system could not determine the relationship between the IM phase and the stimulus frequency so the second system was developed to handle this case. In the past, measurement of the phase of frequency components arising from large-signal excitation of RF and microwave devices has been difficult to perform. Since nonlinear behavior results in frequency components at the device output not present in the stimulus, the biggest problem with measuring phase is establishing a phase reference for the new frequency content. Unlike small-signal single-tone measurements performed with a vector network analyzer, no convenient source of the new frequency content exists as a phase reference.

Omitting the phase of the additional frequency content limits the accuracy of system models that can be developed. The traditional techniques of using intermodulation

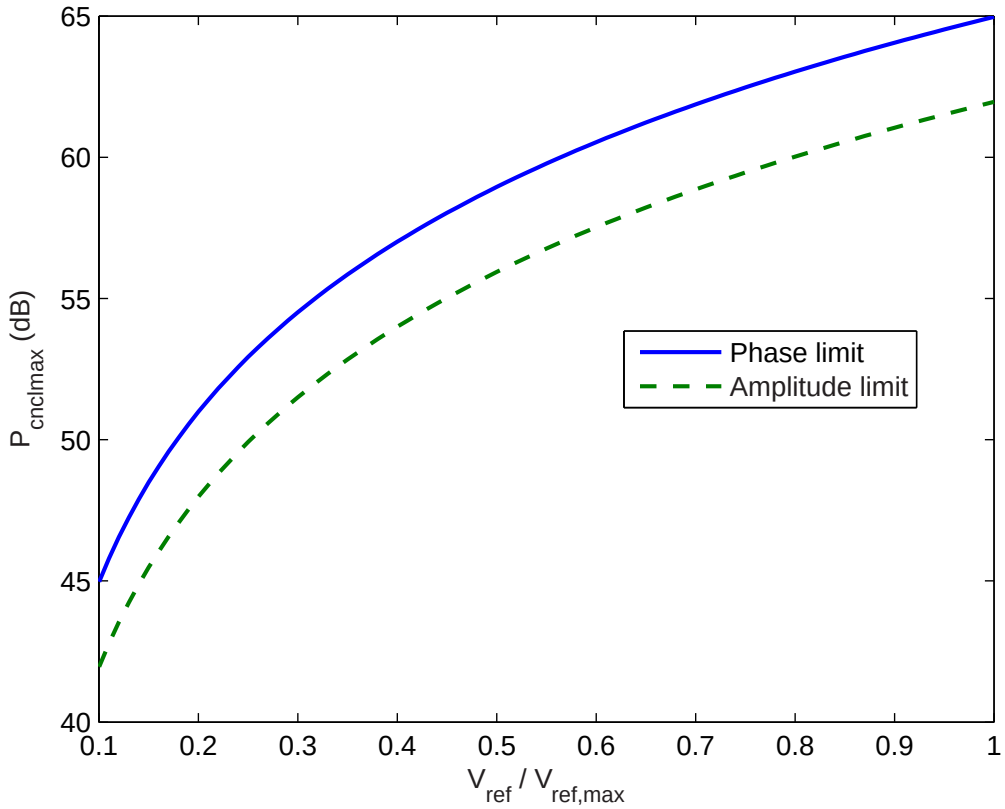


Figure 3.8: Quantization noise limit on cancellation for pure phase and amplitude errors.

intercept points and their inability to predict behavior with typical metrics such as ACPR and EVM are well known [13], [45]. Since the magnitude only approaches to system modeling do not consider the contributions of individual nonlinear processes through vector addition within the device or system, these methods may have difficulty tracking system performance. Measurement of the phase of all output spectral products of interest increases the level of knowledge about the DUT allowing designers to better understand the operation of the DUT in the large-signal regime.

The objectives of the measurement systems presented here are two-fold. First, by providing both phase and amplitude information for the frequency content produced by nonlinear behavior, the systems will allow development of models that can more closely track the operation of real devices and systems. Secondly, although several measurement instruments and setups exist to provide a measure of the relative phase of intermodulation products, these either introduce complexity through reference devices, or circuit schemes to produce the phase reference signal, or require expensive instrumentation in order to perform the measurement. The methods presented here are simple in both setup and operation as well as requiring only a small number of standard bench-top instruments already available in RF and microwave laboratories.

3.4.1 Intermodulation Phase Measurement Background

Though none are widely used, several methods have been developed to measure the relative phase of nonlinear products generated during large-signal excitation. The majority of these techniques can only measure the phase of harmonics of the fundamental signal and thus cannot be used to investigate in-band nonlinear behavior. The most mature of these methods is based on the NVNA described in Section 2.5.2.

The other phase measurement systems published in the literature have not undergone the extensive development of the NVNA. The setups used in [89] and [90] can only be used for wide-band characterization of devices since these only measure the phase of harmonic products. The system in [89] uses a straightforward measurement of the phase using a network analyzer with reference signals provided by a multi-harmonic signal generator. The system in [90] is much more complicated and uses frequency multipliers to supply the reference signals where phase is measured by a Microwave Transition Analyzer (MTA). The methods in [8, 91–94] are designed to measure the phase of the intermodulation products

directly. Those in [91] and [92] use reference nonlinearities as the source of the phase reference while those in [8], [93], and [94] use multiple mixers to generate the reference signals. The system in [93] uses five signal generators to produce the necessary reference signals and an accompanying large amount of interconnect and signal routing components to complete the measurement apparatus.

Reviewing these measurement setups reveals several important issues. First, the simplest of these methods rely on reference nonlinear devices to provide the reference signals, which implies that these devices have been characterized to a level sufficient for instrument grade calibration, an assumption that may be difficult in practice. Secondly, the more complicated schemes require extensive calibration due to their high number of components in the system or the use of multiple frequency mixers. The use of mixers in general has considerable implications as these devices themselves are highly nonlinear and thus care must be taken such that they do not introduce nonlinear behavior similar to that produced by the DUT, [94]. Most importantly, many of the systems require a large amount of instrumentation to implement, with many requiring a vector network analyzer (VNA), several employing multiple spectrum analyzers and sources, and others requiring MTAs. While these instruments are common in most microwave laboratories, utilization of so many instruments in complicated setups represents a high cost in both capital and time of use.

The goal of the measurement systems developed as part of this research is to provide simple, broadband measurement apparatus to determine the relative phase of the intermodulation products, specifically the IM3 products. Design decisions chose to avoid using reference nonlinearities in the system since the characterization of the reference introduces additional sources of error and the complexity of keeping the power level of the stimulus to the reference device constant during swept power measurements. Creating a simple system resulted in employing the minimum equipment necessary to perform the measurement, which reduces the implementation cost and calibration complexity. The methods presented here use similar concepts as the reference nonlinearity approaches. The relative phase is determined by cancelling spectral products from the DUT through adjustment of a reference signal phase and amplitude combined with the DUT output. The main difference in our methods lies in the use of a signal source with control over amplitude and carrier phase to provide the cancellation signal.

3.4.2 Vector Intermodulation Analyzer for Measuring Relative Intermodulation Phase Versus Input Power

In this section, the first Vector InterModulation Analyzer (VIMA) measurement system (developed for IM phase measurement) capable of measuring IM phase with respect to a phase reference is presented. The phase reference is established at a reference stimulus power level. The system consists of three identical, phase-locked RF sources, (Marconi Model 2024), hybrid combiners, isolators, and a spectrum analyzer. A block diagram of the setup is shown in Fig. 3.9. The RF sources provide both the stimulus to the DUT and the mechanism employed to measure the phase of any of the discrete tones in the DUT output. Measurement of phase is performed by cancelling the product of interest at the DUT output with Source 3 by varying its amplitude and phase with internal elements. The amount of phase shift for each power level in a sweep gives the phase shift of the DUT as a function of input power with respect to a chosen reference power level. The three sources share a common 10 MHz reference frequency with the spectrum analyzer. The common reference allows the synthesizers in each instrument to stay locked in frequency to other instruments at the same frequency, however this does not imply that the instruments are phase locked when frequencies are changed. In the design of this system, several synthesizers are used to provide the stimulus and cancellation tones. It was observed that when using different models of synthesizers, considerable phase drift occurred between the sources such that the phase measurement provided by cancellation was not repeatable. Thus, while the sources remained locked to the same frequency, having identical average time-varying phase, the instantaneous phase of the synthesizers did not track each other and thus led to variations in relative phase apparent in less than one minute of time. By using three identical sources with the same frequency reference, the phase drift issue was eliminated and the stability of phase between sources could be measured in hours. Calibration of the measurement setup requires removing the effect of the sources' phase shift with changes in output power and choosing a suitable reference for the relative phase measurement. The former requires characterization of the source output phase as a function of power and frequency, while the latter is based on the model for the phase of IM3 products at small-signal. In [91] the authors showed that the relative phase of the IM3 products was constant for the weakly nonlinear region of the input stimulus. This agrees with the behavior observed during single-tone AM-AM AM-PM analysis that shows nonlinear behavior is largely resistive, i.e.

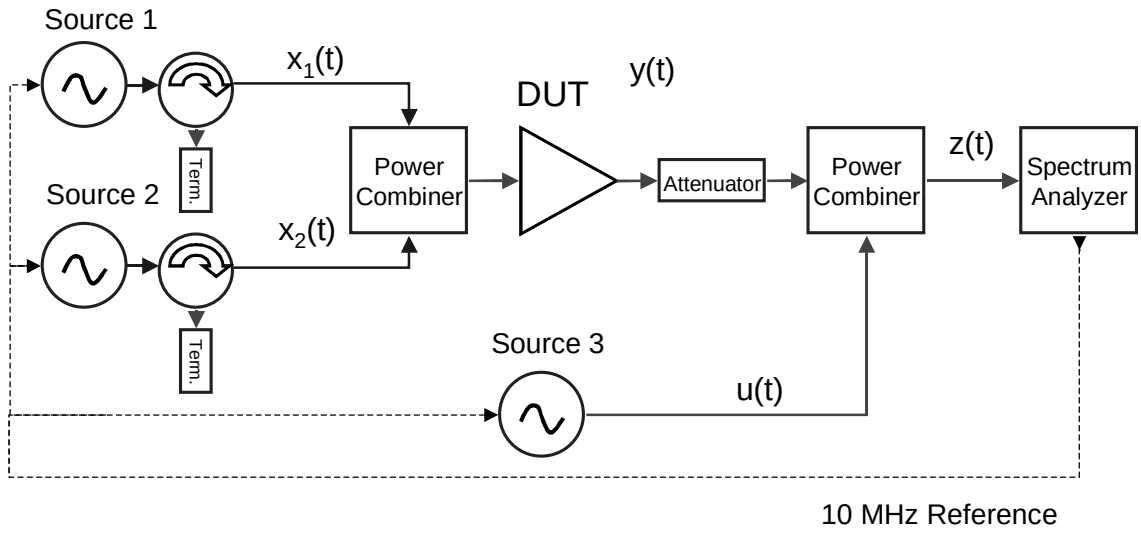


Figure 3.9: Measurement system block diagram incorporating phase-locked signal sources.

without phase shift, for small-signal inputs. Thus, for the measurement procedure here, we take the reference phase as that at a designated small-signal input level.

Source Operation

The sources used in this apparatus introduce phase shifts in the supplied RF stimulus and cancellation tones through two different mechanisms. The first results from switches in fixed attenuator settings in the source output path to accommodate the wide range in power settings available. Correction for this effect is straightforward as the attenuator switch points occur at deterministic output power levels. In addition, the attenuator lock feature of the sources removes this effect for the generators that supply the DUT stimulus as the power control range of this feature allows a large sweep in source output power (approximately 30 dB). Attenuator lock cannot be used on the source providing the IM3 cancellation tone as the range in power required for this source is much greater than 30 dB. The second mechanism for phase insertion by the sources arises from a linear shift in output phase with power change (power in dB).

Both of the phase-shift responses were measured using a network analyzer with an S-parameter test set, where one of the sources was used to supply the reference signal for

the test set and the other source drove Port 2 of the test set. In this setup, shown in Fig. 3.10, the relative phase between the reference source and the source at Port 2 is measured as the phase of S_{21}. Figure 3.11 displays the measured phase for a single frequency across a power range of -70 to $+13$ dBm. As shown, six distinct attenuator switch points occur over this power range. After removing the phase attributed to the attenuator steps, the linear nature of the phase shift with power is clearly evident. Figure 3.12 reveals that this effect is essentially constant across the frequency band with the phase shift normalized by the frequency of operation. The normalized phase shift (a time delay) with power shows that this response is fundamental to the design of the sources used as it is nearly constant across the frequency band $400 - 2400$ MHz.

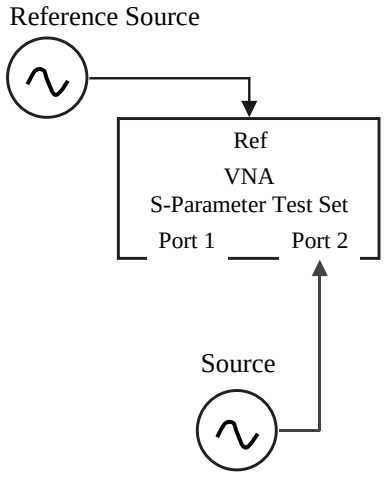


Figure 3.10: Test setup for measuring phase calibration data for RF sources.

Details of the measurement post-processing required to remove the effects of the phase injection by the sources is included in Appendix A. The end result of the calibration development reveals that the effect of the sources is to introduce an additive phase term to the phase shift contributed by the DUT, which can easily be quantified and removed.

The VIMA, Fig. 3.9, has a large dynamic range enabling the system to measure the intermodulation product phase and magnitude for weak nonlinearities or small signals. The amplitude uncertainty is determined by the dynamic range of the spectrum analyzer and corresponds to a system dynamic range of 90 dB and a maximum phase uncertainty of 2 degrees. These are compared to reported schemes for measuring the amplitude and

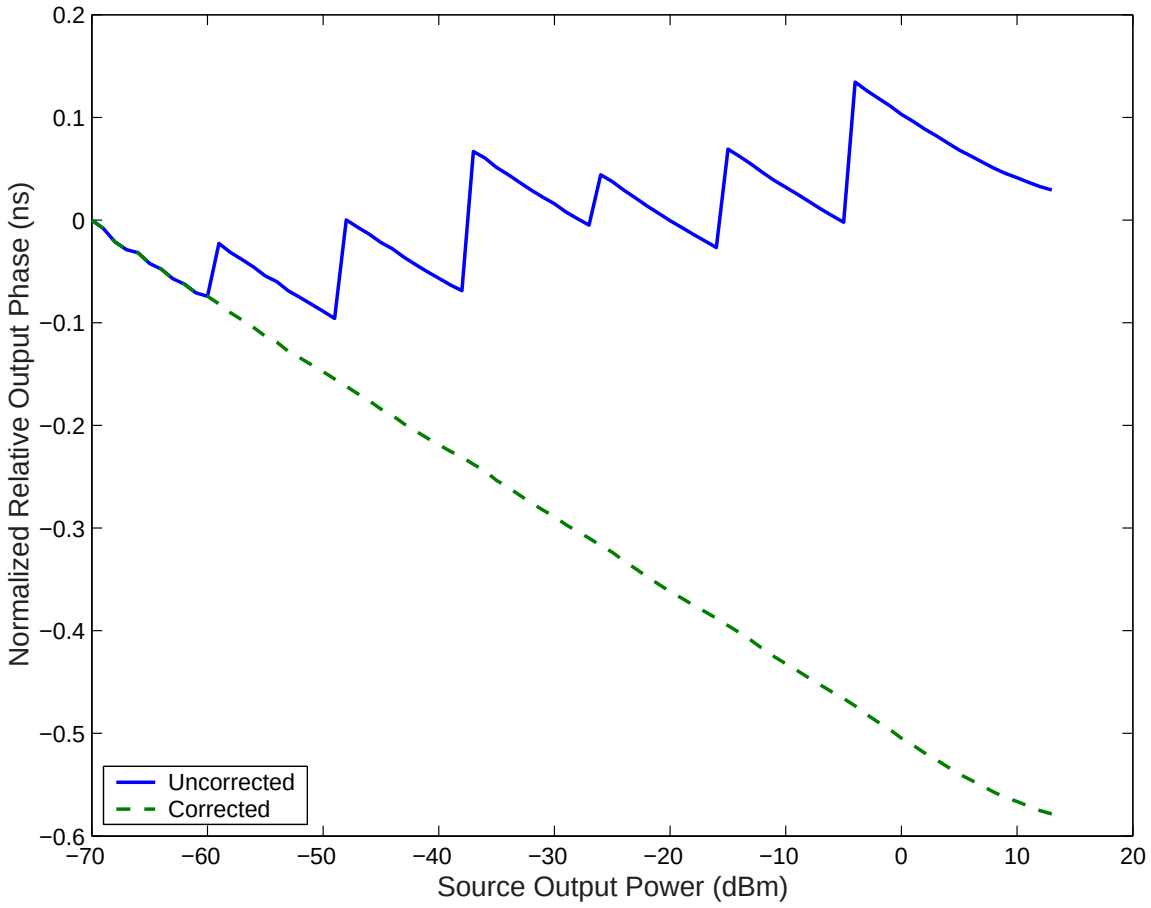


Figure 3.11: Measured phase insertion of RF source and step attenuator corrected phase insertion at 400 MHz.

phase distortion in Table 3.1. The dynamic range of this system was determined by both measurement and inspection of the spectrum analyzer specifications [114]. The dynamic range is important for the weakest nonlinear response of the DUT. At low input power levels, the distortion in the analyzer front-end can overwhelm the DUT response. In this system, the stimulus power at the internal mixer was -40 dBm while the weakest nonlinear device studied produced distortion products of -102 dBm at the mixer. During phase measurement, 30 dB of cancellation of the IM3 tones was achieved, thus the dynamic range of the system was at least 90 dB. The analyzer specification lists the dynamic range as ~ 105 dB for a -40 dBm power level which verifies the observed measurements.

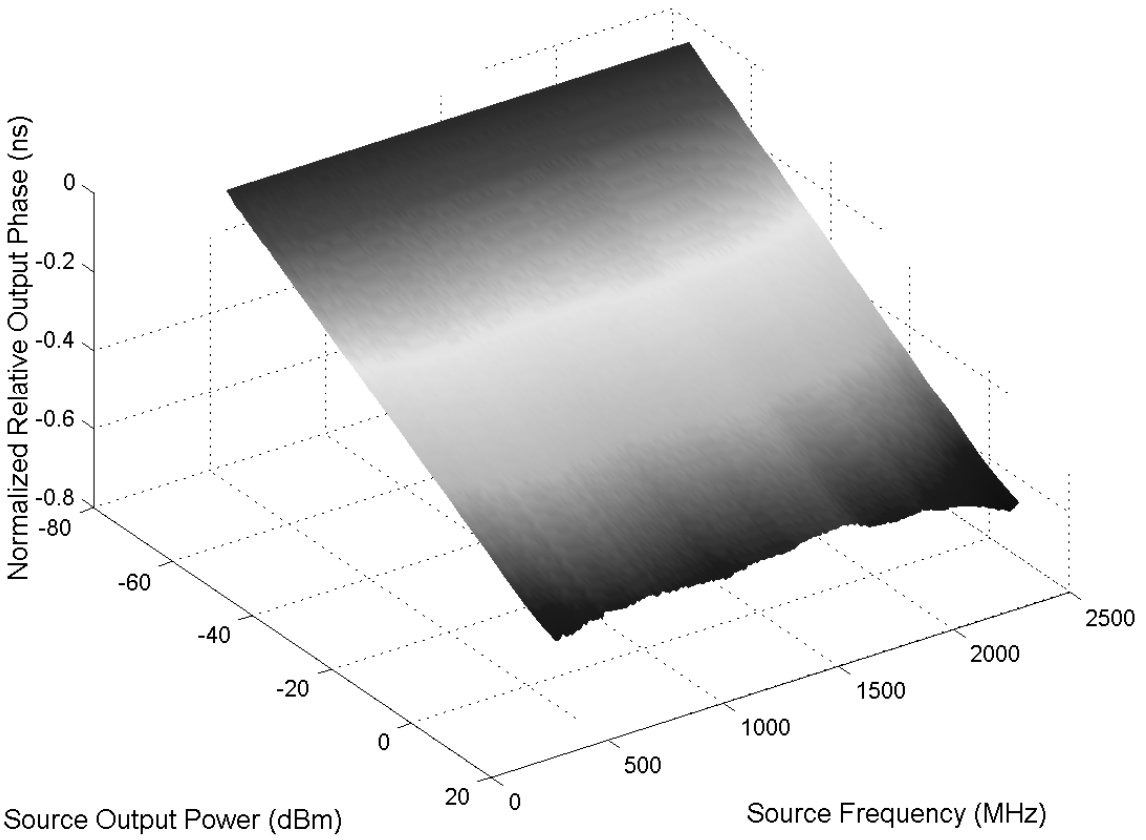


Figure 3.12: Corrected phase insertion of RF source vs. carrier frequency and output power demonstrating linear phase shift with output power and weak dependence on carrier frequency.

3.4.3 Vector Signal Analyzer-Based Vector InterModulation Analyzer

While the system described in Section 3.4.2 enabled intermodulation phase measurements without using a reference nonlinear device or a complicated instrument setup, a significant limitation existed. The system could only measure the IM phase as a function of the input stimulus power and could not be used to track the change in phase as a function of the stimulus frequency or frequency spacing. The behavior with respect to these parameters is important in order to determine the memory effects, both macro-level and micro-level, of the DUT. To measure the IM phase with respect to frequency requires tracking the phase of the sources used in the stimulus as they change randomly with frequency tuning. This function can be provided by a Vector Signal Analyzer (VSA) with appropriate calibration techniques.

Table 3.1: Dynamic range and phase uncertainty/error performance of published IMD phase measurement systems.

Group	Year	Dynamic Range (dB)	Phase error
Suematsu, et al. [91]	1997	35†	$< \pm 2^\circ$
Yang, et al. [92]	2001	unknown	$> \pm 1.8^\circ$
Maury NVNA [112], [113]	2003-4	60	$< 3^\circ$
Vuolevi, et al. [8]	2001	40†	$\pm 1.8^\circ$
Heymann, et al. [90]	2001	55	$\pm 10^\circ$
Crespo-Cadenas, et al. [115]	2005	60	$< 6^\circ$†
Pedro, et al. [116]	2005	60†	unknown
Walker, et al. [88]	2005	90	2° max., 0.5° avg.

† indicates estimated performance based on published results.

An emerging class of systems has been presented recently that takes advantage of advances in test and measurement equipment most notably the widespread availability of vector signal generators and analyzers. These instruments are capable of performing quadrature modulation and demodulation that allows extraction of phase information not possible with traditional spectrum analyzers. Pedro *et al.* demonstrated a system based on a statistical processing algorithm on the received signal for distortion phase determination [116]. This system relied on using a dual-channel synchronous receiver (in this case a Microwave Transition Analyzer (MTA)) to detect the stimulus input and the DUT response. Multiple experiments at a single frequency established a population of both the input stimulus and DUT response. The stimulus information was used to form an ideal phase reference and this was then cross-correlated with the DUT response to determine the IM phase. The major difficulty with this method was that the DUT response contained other frequency content which introduced uncertainty in the phase extraction. Crespo-Cadenas *et al.* used a VSG to generate a multi-tone stimulus and a VSA to demodulate the DUT response and perform a Fast Fourier Transform (FFT) on the time-domain baseband to extract the phase of the generated IM products [115]. The authors verified the extracted phase with a predistortion type technique where an out-of-phase version of the IM products were injected and cancellation of the IM products was observed. The cancellation levels were on the order of 20 dB which relates to $\sim 5^o$ of phase uncertainty. The techniques based on vector signal instruments is an inherent time-domain operation since these instruments digitize a signal with several MHz of bandwidth prior to computing the frequency response. This contrasts

the power detection over a narrow bandwidth Intermediate Frequency (IF) signal in tuned Local Oscillator (LO) spectrum analyzers. The wider bandwidth time-domain operation results in a reduction in the dynamic range possible to extract the phase as these instruments do not have the high dynamic range possible with strictly spectral domain instrumentation.

Development of a Vector Signal Analyzer-Based Vector InterModulation Analyzer

A VSA-based VIMA for phase reference measurement utilizing the same high-dynamic range tone cancellation technique for determining phase as the previous system is presented here. This VIMA is an extension of the previously presented system and has the ability to measure the effect of both stimulus frequency and power on the phase of a DUT's IM products.

The measurement system described here uses multiple independent RF sources and a single vector signal analyzer to determine the phase of third-order intermodulation (IM3) products arising from two-tone stimulus as a function of stimulus frequency. A pair of RF coaxial switches are used to direct the source outputs to the VSA for both the calibration and measurement portions of the measurement routine. A diagram of the system is shown in Fig. 3.13. The system described here takes advantage of the ability of any modulated signal to be represented in a quadrature format as

$$x(t) = I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t)\ , \tag{3.26}$$

where ω_c is the modulated carrier frequency and $I(t)$ and $Q(t)$ are the in-phase and quadrature modulation components. The system uses the inverse relation that allows VSG instruments to generate multi-tone signals with arbitrary initial phase by modulating a single carrier frequency.

In this case, the VSA is tuned to a fictional carrier frequency that lies between the stimulus tones at ω_c. Now the individual stimulus tones can be represented as

$$x(t) = A\cos((\omega_c + m\omega_\Delta)t + \phi)\ , \tag{3.27}$$

where ϕ is the initial random phase of the signal produced by an RF generator. The term ω_Δ is the offset frequency from the hypothetical carrier frequency which does not exist in a physical sense, but is used as the reference for a multi-tone system comprised of independent

frequency synthesizers. The coefficient m for the stimulus tones and IM3 products is ± 1 and ± 3 respectively.

For a single tone in the form of (3.27), a vector signal analyzer can extract the I and Q components of the RF signal in hardware to achieve,

$$x_I(t) = A\left[\cos(m\omega_\Delta t)\cos(\phi) - \sin(m\omega_\Delta t)\sin(\phi)\right] \tag{3.28a}$$

$$x_Q(t) = A\left[\cos(m\omega_\Delta t)\sin(\phi) + \sin(m\omega_\Delta t)\cos(\phi)\right] \ . \tag{3.28b}$$

Clearly, these terms are composed themselves of I and Q components modulating $m\omega_\Delta$ and thus a software quadrature demodulation can be performed on the sampled I and Q waveforms. This produces, after low pass sampling to remove the second harmonic term,

$$x_{I_I}(t) = \frac{A}{2}\cos(\phi) \tag{3.29a}$$

$$x_{Q_I}(t) = \frac{-A}{2}\sin(\phi) \tag{3.29b}$$

$$x_{I_Q}(t) = \frac{A}{2}\sin(\phi) \tag{3.29c}$$

$$x_{Q_Q}(t) = \frac{A}{2}\cos(\phi) \ . \tag{3.29d}$$

Thus it is clear that the initial arbitrary phase can be determined from any of the four equations. Since an inverse trigonometric function must be used to determine ϕ, a sine/cosine pair must be used to resolve the inverse function phase ambiguity. Of course in reality, the absolute phase angle will not be known since the initial phase of both the VSA hardware LO and the software LO used in the demodulation will vary with respect to the demodulated I and Q samples produced by the hardware demodulator. Additionally, any imbalance in the hardware I and Q channels will result in variations in the estimated generator phase. Besides the phase lock between the hardware LO of the VSA and the RF signal generators, the software LO must also be locked to the phase of the RF generator to maintain a stable phase estimate.

These variations result in random phase wander in the measurement which can be eliminated only through a low-pass filter with a very low frequency cutoff. Alternatively, since only the DC component of the software demodulated signals are needed for the measurement, a simple running average of the demodulated signal can yield the same result in considerably less time than that required for the filter output to reach steady state.

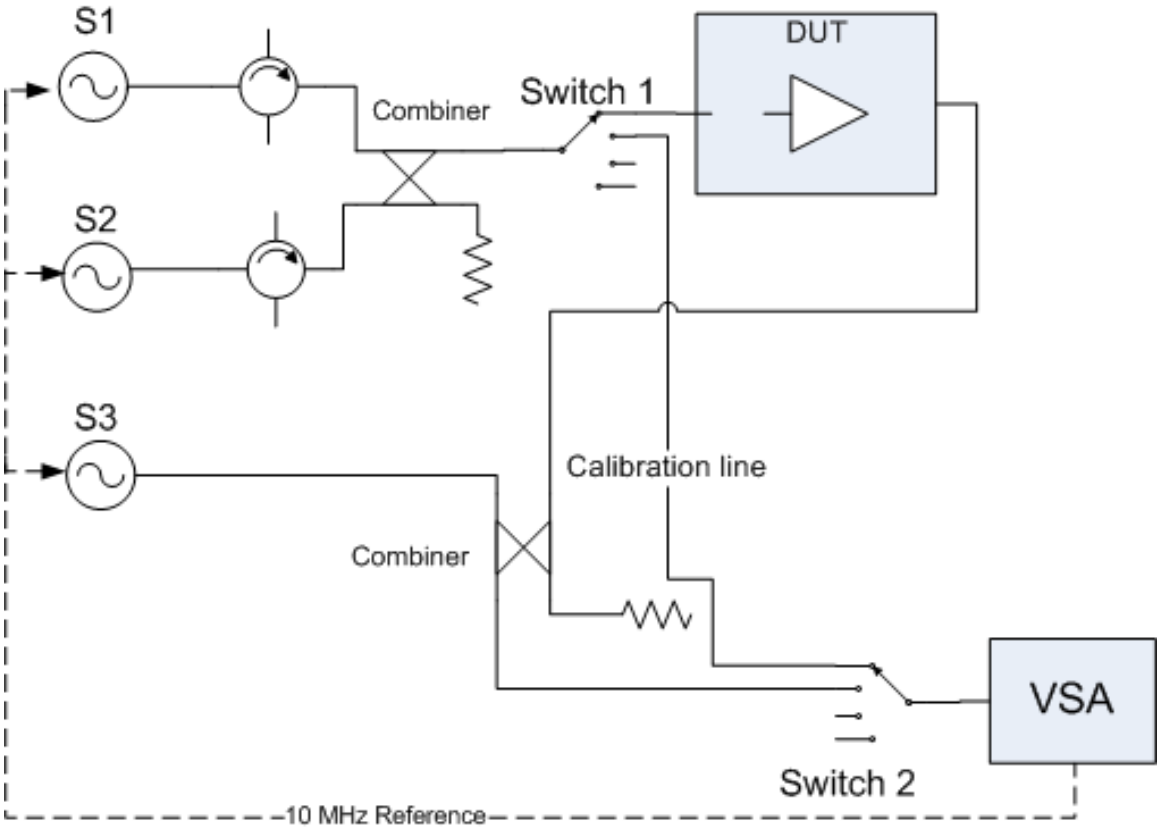


Figure 3.13: Block diagram of the VSA-based intermodulation phase measurement setup.

The system in Fig. 3.13 was implemented with the National Instruments PXI-5660 VSA and the same three independent phase-locked RF signal sources of the previous system. The software functions for demodulation and measurement automation were implemented with LabVIEW ®. The VSA allows direct access to the hardware I and Q signals that are necessary in order to process the data continuously in real-time for measurement of the arbitrary phase of the signal generators. A diagram of the partition of the hardware and software sections for executing the equations above is shown in Fig. 3.14. Measurement of the phase of the IM products proceeds by first making an estimate of the ideal phase of a memoryless nonlinearity exposed to the stimulus generators using the calibration path. The initial phase of Source 3 is then measured in a similar fashion with the stimulus generators disabled. The phase of this source is then adjusted so that it would cancel the response of an ideal nonlinearity exposed to the two-tone stimulus. At this point the DUT is switched

into the circuit and Source 3 is used to cancel the DUT IM product of interest. The phase of the DUT is the difference between the ideal estimate and the required phase shift necessary to achieve cancellation. The operation of the system proceeds as shown in the flow chart in Fig. 3.15 with the following main steps:

1. Initialize the instrumentation for the calibration phase.
2. Measure the arbitrary phase of the individual sources as above.
3. Compute and adjust the phase of the cancellation source for ideal nonlinearity.
4. Adjust phase and magnitude of cancellation source to achieve maximum cancellation.
5. Repeat for each frequency in the sweep.

By using cancellation of nominally equal magnitude tones, this technique can leverage the high-dynamic range of a magnitude-based spectrum analysis to measure the phase of the IM products as in [88] and [91]. Cancellation levels on the order of 40 dB are achievable resulting in a phase uncertainty of 0.5 degrees.

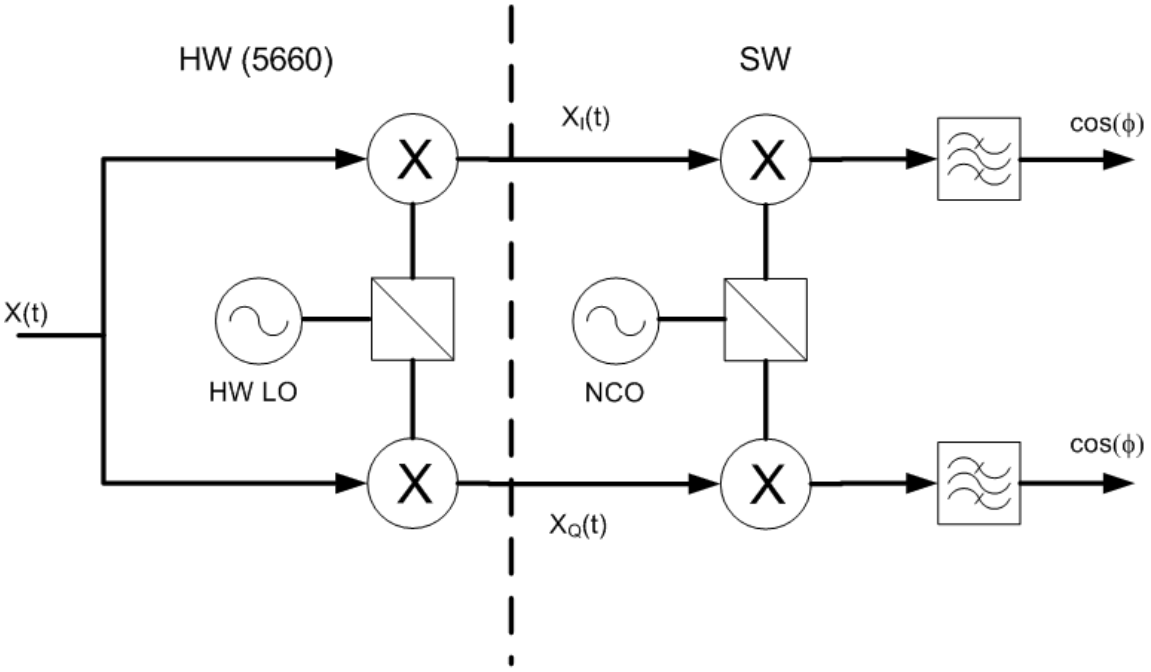


Figure 3.14: Block diagram of the hardware and software components of the quadrature phase measurement system.

3.5 Summary of Metrology System Development

The measurements focus of this work resulted in considerable effort being placed on understanding the existing metrology techniques and developing new techniques for expanding the knowledge about nonlinear behavior in RF and microwave systems. The AM-AM, AM-PM and two-tone magnitude measurements are the foundation of all nonlinear characterization techniques employed by the microwave community so the review presented in Section 3.1 provides an understanding of the practices on which all of the other techniques developed here are based. The extension of these traditional tests to the high-power two-tone case in Section 3.2, allowed the investigation of another of the novel contributions of this work, that of remote characterization of RF systems (Chapter 5). The other novel developments in this chapter were the two IM phase measurement systems. These VIMA systems represent a simpler implementation of this type of measurement than those in other works by requiring fewer components with the accompanying simplification in calibration and measurement execution. In addition, the use of high-dynamic range receivers gives these setups better dynamic range performance than previously published systems, an advantage when characterizing weak nonlinearities and for improving phase uncertainty. The two-tone cancellation system developed provides a path for further increases in the receiver dynamic range that will enable the study of nonlinear behavior currently inaccessible to existing instrumentation, (e.g. in-band PIM distortion).

Measurements on their own however, do not provide a complete solution towards developing an understanding of a system under investigation. Thus, the use of the relative phase versus power VIMA to characterize nonlinear amplifiers and the use of the VSA-based VIMA to extract two-port transmission parameters from one-port measurements of an integrated filter are presented in Chapters 7 and 8.

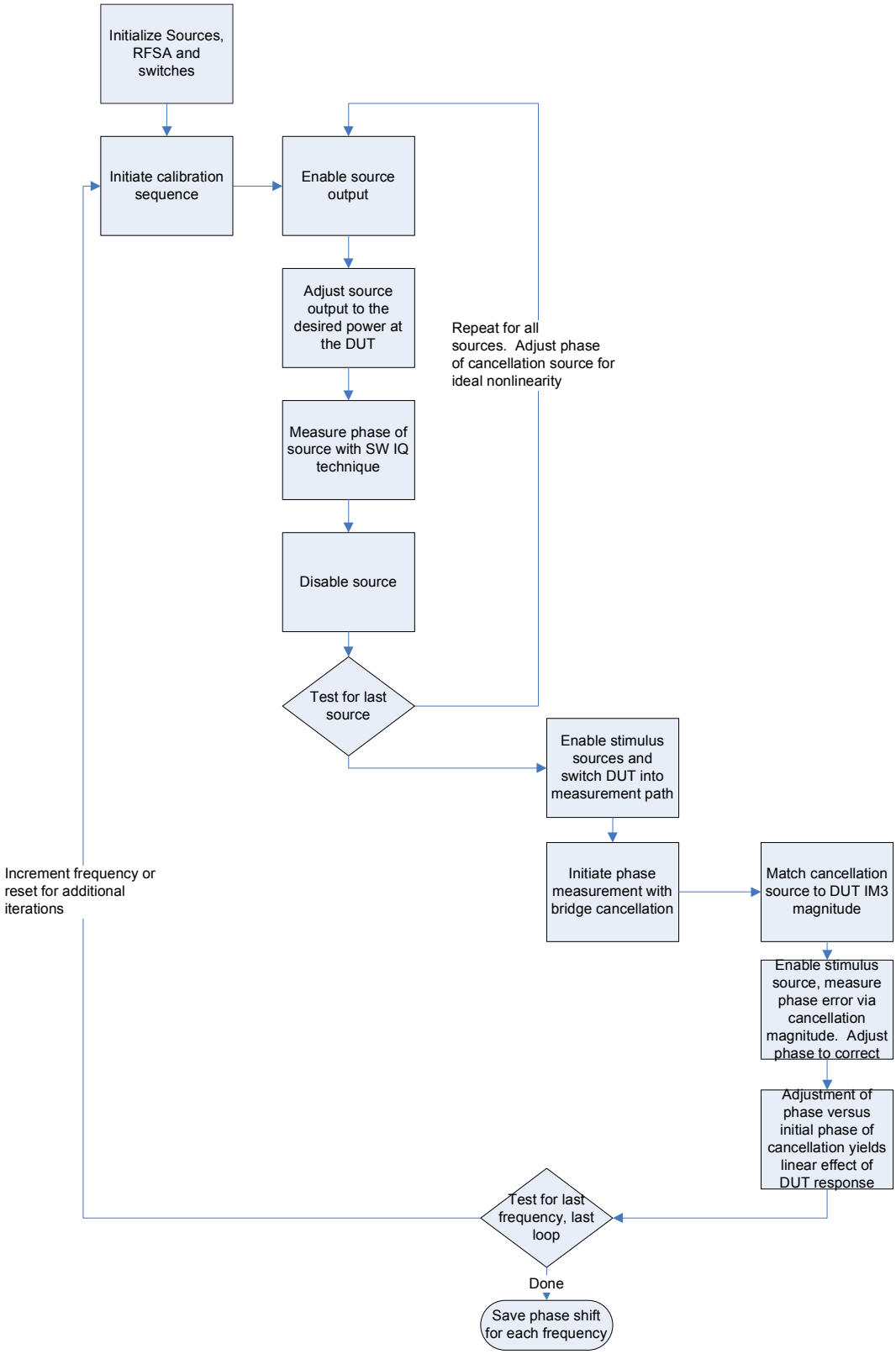


Figure 3.15: Flowchart depicting nonlinear distortion phase measurement utilizing independent source and a vector signal analyzer.

Chapter 4

The Multi-Slice Behavioral Model

Another novel contribution of this work is the development and use of a new modeling technique for capturing nonlinear device and system operation. The theory behind the multi-slice model will be presented here and the experimental validation of the model by accurately estimating the nonlinear behavior of multiple amplifiers will be presented in Chapters 6 and 7. The architecture and parameter extraction procedure of the multi-slice model enables the identification of multiple parallel nonlinear processes separately and captures both macro-level and micro-level memory effects. While multiple authors have published the concept of baseband contributions to the odd-ordered IM products, [6, 8, 9], the implementation of the multi-slice model marks the first time a model has been developed to account for the baseband component as a function of stimulus level. Incorporation of IM phase information from the VIMA systems allows model extraction that can track both amplitude and phase asymmetries in IMD. Since the model is implemented with general components representing filters, mixers, and equation-based nonlinearities, it can be used in any type of simulator technology including harmonic balance, transient, or complex envelope.

4.1 Behavioral Modeling Justification

The advantages of behavioral modeling include: reduced model complexity compared with physical models; capture of complex, non-observable operation within a device; and the ability of the approach to perform feature extraction that can provide intuition to the designer. Considering the infinite number of methods for creating a model that reproduces device behavior, we chose to let our knowledge of electronic circuits guide the modeling approach such that there remained a link between circuit constructs and the topology and components within the model. Several modeling techniques have been published that utilize highly non-physical model architectures that produce good agreement with measured data, but are so complex or abstract that they do not allow scaling to other systems or provide insight into the design under study [19] and [95].

One of the difficulties in system-level modeling is capturing the behavior of the internal processes that contribute to an observable external frequency response. In particular, the third-order intermodulation distortion products arising from multi-tone stimulus have been traced to several different phenomena. Since the contribution of these processes cannot be measured individually, test methods must be used to extract the individual components produced by each source. In a multi-slice model, we choose to represent each source of an output frequency band as a separate slice. In this manner, each nonlinear process with and without memory effects can be modelled separately with the total contribution in a particular frequency band being the phasor sum of the contributions of all slices.

4.2 Description of Multi-Slice Model

A general depiction of the multi-slice model is shown in Fig. 4.1. This modeling approach combines aspects of several different modeling techniques as well as concepts resulting from empirical results. The model can have multiple parallel slices of Wiener-Hammerstein cascades to capture the behavior of the multiple nonlinear processes at work within system or device. The slices are composed of linear networks that capture the memory effects of the nonlinearity within the slice.

The first slice captures single-nonlinearity effects present in the system under investigation which can include both even and odd-ordered effects. Additional slices capture nonlinearities in series such as even-order baseband upconversion, thermal effects, or other

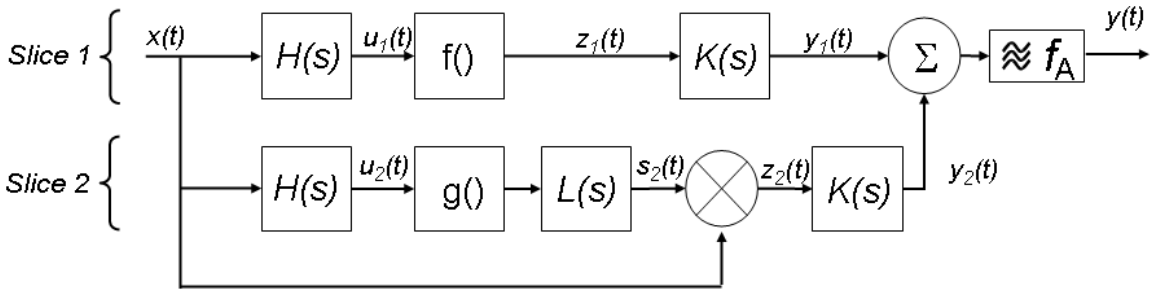


Figure 4.1: Two-slice architecture of multi-slice model.

memory effects. These slices contain ideal multipliers that allow mixing of the fundamental stimulus with the output of other nonlinear processes.

Physical support for this type of construction arises from the existence of multiple cascade nonlinearities within semiconductor transistors such as the cascade nonlinearity in a Hetero-junction Bipolar Transistor (HBT) where the voltage across the nonlinear base resistance determines the current through the nonlinear transconductance. The higher-order slices also represent a mechanism whereby nonlinear feedback through capacitive coupling from the output to the input could produce a cascade nonlinear effect. An output filter after the summation of the contributions of each slice serves to constrain the model output to the frequency band or bands of interest.

4.3 Features of the Multi-Slice Model

4.3.1 Modularity

The modular nature of the multi-slice model offers several unique attributes. First of all the model is scalable. Construction of a model capturing complex nonlinear behavior proceeds by adding the slices necessary to represent the different processes at work. Since the slices in the model operate independently of each other, the slices can be considered as a set of basis functions, the sum of which represents the complete model output. The flexibility

offered by the linear networks in each slice means that slices can be added to take into account different frequency bands of operation, thus both in-band (first zone or bandpass model) and out-of-band nonlinear products can be modeled with this architecture. This includes the higher order harmonics and intermodulation products that are normally not considered in narrow band systems. In addition to electrical sources of nonlinear operation, the slices are general enough to allow slice parameters to potentially capture other sources of nonlinear behavior such as thermal effects.

4.3.2 Applicability to Multiple Simulation Types

Another important attribute of this model approach is that the model elements can be equally defined in both the time and frequency domains. This allows the multi-slice model to be used in all types of numerical simulators including transient and harmonic balance. The elements may also be defined in baseband or low-pass equivalent for use in complex envelope simulations.

4.3.3 Capture of Memory Effects

The major benefit of the multi-slice model is its ability to describe multiple memory effects that are difficult or impossible to perform with most modeling techniques. While a Volterra-series-based model can reproduce nonlinear processes with memory, these are restricted to weakly nonlinear systems. The popular power-series approach cannot reproduce any memory effects, even using complex polynomials, except for those effects which result in the same phase shift for each nonlinear product produced by a given order of the series. In general, memory effects encompass all changes in the nonlinear behavior of a system as a function of frequency. The change in frequency may be either in the frequency of the stimulus for discrete tone experiments or in the separation frequency for multi-tone experiments. In the multi-slice model, memory effects are captured on both the macro level, (the input and output linear networks in each slice), and the micro level, (linear network between the nonlinearity and the ideal mixer in higher order slices).

Macro-Level Memory Effects

In the following, macro-level memory effects are defined as the impact on the DUT performance by the frequency response of the input and output linear networks around the nonlinearity. This means that the magnitude and phase response of these networks interact with the nonlinearity to shape the nonlinear response as a function of the frequency of the stimulus. For macro-level memory effects the two sinusoidal signals are considered to be at the same frequency since signals with a small frequency separation (a small percentage bandwidth compared to the linear network response) can be assumed to be affected equally by the pre- and post-nonlinearity networks. The linear networks have a physical basis since all real systems have matching networks and other linear networks at the input and output of the nonlinear components or circuits in the system. These elements do possess memory so a comprehensive modeling architecture must include these elements. The macro-level memory components do not explicitly attempt to capture memory effects within the nonlinear processes themselves (such as AM-PM conversion), rather this will be termed micro-level memory. The Wiener-Hammerstein structure used in the first slice of the multi-slice model retains the form of these input and output networks automatically. While these networks do not contribute to the micro-level memory effects, their interaction with the nonlinearity does result in considerable modification of the nonlinear system response than that predicted by a memoryless nonlinearity alone. An analysis of the response of a single slice demonstrates this interaction.

Analysis of First-Slice using Complex Coefficient

The first slice in the model is the dominant component represented by a Weiner-Hammerstein LNL block model [32, 37]. The nonlinearity in this slice is taken as a memoryless odd-ordered polynomial, although in general the nonlinearity could be represented as any nonlinear function. The linear networks $H(s)$ and $K(s)$ represent the real linear networks, such as matching networks, present in the system. These networks capture the macro-level memory effect which causes variations in the magnitude and phase response of the output as a function of the frequency of the input signal. The assumptions for the analysis of the slices consist of a quasi-static approach to the signals in the model and a treatment of the slices as independent such that they do not interact. The former assumption allows the signals in the model to be treated as steady state so that all linear networks

can be represented in the time domain as their frequency domain amplitude and phase response. The latter assumption makes the process of extracting the slices tractable since they are performed in sequence and not in an iterative manner.

Given a general stimulus composed of Q incommensurate sinusoids

$$x(t) = \frac{1}{2} \sum_{\substack{q=-Q \\ q\neq 0}}^{Q} A_q e^{j(\omega_q t + \phi_q)} \,, \tag{4.1}$$

with $A_q = A_{-q}$, $\omega_q = -\omega_{-q}$ and $\phi_q = -\phi_{-q}$, the output of the linear network $H(s)$ is

$$u_1(t) = \frac{1}{2} \sum_{q=-Q}^{Q} A_q e^{j(\omega_q t + \phi_q)} \, |H(\omega_q)| \, e^{j(\phi_{H(\omega_q)})} \,. \tag{4.2}$$

After passing through the complex polynomial block the output $z_1(t)$ for a given order n of the polynomial is given by

$$z_{1,n}(t) = \mathrm{Re}\left\{ \left\{ \frac{1}{2} \sum_{q=-Q}^{Q} A_q e^{j(\omega_q t + \phi_q)} \, |H(\omega_q)| \, e^{j(\phi_{H(\omega_q)})} \right\}^n |\, a_n| \, e^{j\phi_{a_n}} \right\} \,. \tag{4.3}$$

Here $z_1(t) = \sum_{n=0}^{N} z_{1,n}(t)$ and ϕ_{a_n} applied as a complex conjugate phase shift depending on the sign of the frequencies produced by raising the complex exponential representation to the power n.

At the output of the first slice we have

$$y_{1,n}(t) = \mathrm{Re}\left\{ |\, a_n \,||\, K(\omega)| \, e^{j(\phi_{a_n} + \phi_{K(\omega)})} \left\{ \frac{1}{2} \sum_{q=-Q}^{Q} A_q e^{j(\omega_q t + \phi_q)} \, |H(\omega_q)| \, e^{j(\phi_{H(\omega_q)})} \right\}^n \right\} \,, \tag{4.4}$$

where again the sign of $\phi_{K(\omega)}$ for a given expansion results in the production of a real overall result. The complete output of the first slice is given by $y_1(t) = \sum_{n=0} y_{1,n}(t)$. Expansion of (4.4) for a single order n yields

$$\begin{aligned} y_{1,n}(t) = \mathrm{Re}\left\{ |\, a_n \,||\, K(\omega)| \, e^{j(\phi_{a_n} + \phi_{K(\omega)})} \right\} \frac{1}{2^n} \sum_{q_1=-Q}^{Q} \cdots \sum_{q_n=-Q}^{Q} A_{q_1} \cdots A_{q_n} \, |H(\omega_{q_1})| \cdots |H(\omega_{q_n})| \\ \times e^{j(\Phi_{q_1} + \ldots + \Phi_{q_n} + \phi_{H(\omega_{q_1})} + \ldots + \phi_{H(\omega_{q_n})})} \,, \end{aligned} \tag{4.5}$$

where $\Phi_{q_i} = \omega_{q_i} t + \phi_{q_i}$, the frequency and initial phase of the individual tones. Using the concept of the frequency mixing vector $\vec{m}$ developed by Weiner and Spina [14], the output of (4.5) at a specific frequency can be determined. The mix vector has $2Q$ entries and enumerates the collection of positive and negative frequencies in the desired product as $\vec{m} = [m_{\omega_{-Q}}, m_{\omega_{-Q+1}}, \ldots, m_{\omega_1}, \ldots m_{\omega_Q}]$, with the constraint that $\sum_{q=-Q}^{Q} m_{\omega_q} = n$ for a given order. Using this convention the output for a given $\vec{m}$ is

$$\begin{aligned} y_{1,n,\vec{m}}(t) = \frac{|\,a_n\,|}{2^{n-1}}\,|\,K(\omega_{\vec{m}})\,|\,c_{\vec{m},n} A_{q_1} \cdots A_{q_n}\,|\,H(\omega_{\vec{m}})\,|^n \\ \times \cos(\omega_{\vec{m}} t + \phi_{\vec{m}} + \phi_{H(\omega_{\vec{m}})} + \phi_{K(\omega_{\vec{m}})} + \phi_{a_n})\,, \end{aligned} \tag{4.6}$$

with $\phi_{H(\omega_{\vec{m}})} = \sum_{\vec{m}} \phi_{H(\omega_m)}$. The coefficient $c_{\vec{m},n}$ is the multinomial expansion coefficient for the sum of permutations of $q_1 \ldots q_n$ that generates $\vec{m}$. The multinomial is defined by

$$c_{\vec{m},n} = \binom{n}{\vec{m}} = \frac{n!}{m_{\omega_{-Q}}! m_{\omega_{-Q+1}}! \cdots m_{\omega_{Q-1}}! m_{\omega_Q}!}\,. \tag{4.7}$$

For a low number of tones it is convenient to sum over all of the vectors $\vec{m}$ that produce a specific frequency output for a polynomial of order N as in [7]. The fundamental response of the first slice under a single tone stimulus is then given by

$$\begin{aligned} y_{1,\vec{m}}(t) = \sum_{n=1}^{N} \frac{|\,a_n\,|}{2^{n-1}} \binom{n}{\frac{n-1}{2}, \frac{n+1}{2}} A_{\omega_1}^n\,|\,H(\omega_{\vec{m}})\,|^n |\,K(\omega_{\vec{m}})\,| \\ \times \cos(\omega_{\vec{m}} t + \phi_{\omega_{\vec{m}}} + \phi_{H(\omega_{\vec{m}})} + \phi_{K(\omega_{\vec{m}})} + \phi_{a_n})\,, \end{aligned} \tag{4.8}$$

and the IM3 response under two tone stimulus is

$$\begin{aligned} y_{1,\vec{m}}(t) = \sum_{n=3}^{N} \sum_{l=0}^{\frac{n-3}{2}} \frac{|\,a_n\,|}{2^{n-1}} \binom{n}{\frac{n-3}{2} - l, l, l+1, \frac{n+1}{2} - l} A_{\omega_1}^{2l+1} A_{\omega_2}^{n-2l-1}\,|\,H(\omega_{\vec{m}})\,|^n |\,K(\omega_{\vec{m}})\,| \\ \times \cos(\omega_{\vec{m}} t + \phi_{\omega_{\vec{m}}} + \phi_{H(\omega_{\vec{m}})} + \phi_{K(\omega_{\vec{m}})} + \phi_{a_n})\,, \end{aligned} \tag{4.9}$$

The expression in (4.9) is identical for the upper and lower IM3 products (IM3H and IM3L respectively) with the difference being in the order of terms in the multinomial expansion which does not change the value. Thus, the well known result from (4.9) is that a complex polynomial representation of a memoryless nonlinearity cannot produce

IM3 products that have asymmetric amplitude or phase responses. Clearly the phase shift introduced by the polynomial coefficient is constant and shifts both IM3 products by the same amount. Here an asymmetric phase response is defined as a deviation in the amount of change in the phase since the $\phi_{\omega_{\vec{m}}}$ term will result in a constant phase difference between the IM3 terms with $\phi_{\omega_{\mathrm{IM3L}}} = 2\phi_{\omega_1} - \phi_{\omega_2}$ and $\phi_{\omega_{\mathrm{IM3H}}} = 2\phi_{\omega_2} - \phi_{\omega_1}$, but no phase difference that changes as a function of input power level.

Since a single complex polynomial cannot contribute to asymmetric behavior, an additional model component must be added to capture observed behavior in microwave devices. The multi-slice model seeks to employ slices that represent intuitive and realistic behavior, thus another slice representing baseband upconversion effects that contribute to asymmetry will be considered.

Analysis of Second Slice

The second slice shown in Fig. 4.1 is composed of identical linear networks for the macro-level memory effects as the first slice as well as an even-ordered complex polynomial, $(g(x) = k_0 + k_2x^2 + k_4x^4 + \ldots)$, representing the nonlinearity which generates the baseband terms. The linear network $L(s)$ consists of a real linear network with a lowpass response which limits the components that get upconverted via the ideal mixer to the first zone intermodulation products. The effect of baseband upconversion on the fundamental response for either single or multi-tone stimulus is negligible since the fundamental response is dominated by the linear term of the odd-ordered nonlinearity in the first slice. Physical support of a baseband contribution results from several processes including cascade nonlinear effects due to input and output nonlinearities in BJT devices and feedback of low-frequency distortion products through the DC bias networks of an amplifier discussed in [6] and [8]. We will show here how the second slice reproduces this behavior and how the baseband component of the nonlinear response potentially leads to amplitude and phase asymmetries of the IM3 products.

The response of the second slice is similar to that of the first with the expression

for $s_2(t)$ being equivalent to $y_1(t)$ or

$$s_{2,n}(t) = \mathrm{Re}\left\{ |\,k_n\,||\,L(\omega)\,|\; e^{j(\phi_{k_n}+\phi_{L(\omega)})} \right\} \frac{1}{2^n} \sum_{q_1=-Q}^{Q} \cdots \sum_{q_n=-Q}^{Q} A_{q_1}\cdots A_{q_n}\; |\,H(\omega_{q_1})\,|\cdots|\,H(\omega_{q_n})\,| \\ \times e^{j(\Phi_{q_1}+\ldots+\Phi_{q_n}+\phi_{H(\omega_{q_1})}+\ldots+\phi_{H(\omega_{q_n})})} \,. \tag{4.10}$$

Now with $L(s)$ having a lowpass response with a cutoff frequency far lower than the stimulus frequencies, all harmonics of the even-order nonlinearity will be rejected. Taking the output $s_2(t)$ at the baseband component at $\omega_{\mathrm{BBL}} = \omega_1 - \omega_2$, $(\omega_2 > \omega_1)$ yields

$$s_{2,\omega_{\mathrm{BBL}}}(t) = \sum_{n=2}^{N} \sum_{l=0}^{\frac{n-2}{2}} \frac{|\,k_n\,|}{2^{n-1}} \binom{n}{\frac{n}{2}-l,\, l,\, l+1,\, \frac{n-2}{2}-l} A_{\omega_1}^{2l+1} A_{\omega_2}^{n-2l-1}\; |\,H(\omega)\,|^n \\ \times \cos(\omega_{\mathrm{BBL}} t + \phi_{\mathrm{BBL}} + \phi_{L(\omega_{\mathrm{BBL}})} - \phi_{k_n}) \,, \tag{4.11a}$$

similarly for $\omega_{\mathrm{BBH}}\omega_2 - \omega_1$

$$s_{2,\omega_{\mathrm{BBH}}}(t) = \sum_{n=2}^{N} \sum_{l=0}^{\frac{n-2}{2}} \frac{|\,k_n\,|}{2^{n-1}} \binom{n}{l,\, \frac{n}{2}-l,\, \frac{n-2}{2}-l,\, l+1} A_{\omega_2}^{2l+1} A_{\omega_1}^{n-2l-1}\; |\,H(\omega)\,|^n \\ \times \cos(\omega_{\mathrm{BBH}} t + \phi_{\mathrm{BBH}} + \phi_{L(\omega_{\mathrm{BBH}})} + \phi_{k_n}) \,, \tag{4.11b}$$

assuming that $\phi_{H(\omega_1)} \simeq -\phi_{H(\omega_{-2})}$ and vice-versa. As shown by (4.11), the difference between the component at the positive and negative baseband frequencies is the phase terms ϕ_L and ϕ_{k_n}. Assuming $L(s)$ is a real network, (such as the DC bias network), forces $\phi_{L(\omega_1-\omega_2)} = -\phi_{L(\omega_2-\omega_1)}$, thus combined with ϕ_{k_n} these two terms result in a complex conjugate phase response between the positive and negative baseband frequencies that contribute to the first zone nonlinear response.

Now the output of the ideal mixer at the upper and lower IM3 products is

$$z_{2,\omega_{\mathrm{IM3H}}}(t) = \sum_{n=2}^{N} \sum_{l=0}^{\frac{n-2}{2}} \frac{|\,k_n\,|}{2^{n-1}} \binom{n}{l,\, \frac{n}{2}-l,\, \frac{n-2}{2}-l,\, l+1} A_{\omega_2}^{2l+2} A_{\omega_1}^{n-2l-1}\; |\,H(\omega)\,|^n \\ \times \cos(\omega_{\mathrm{IM3H}} t + 2\phi_2 - \phi_1 + \phi_{L(\omega_{\mathrm{BBH}})} + \phi_{k_n}) \,. \tag{4.12a}$$

and

$$z_{2,\omega_{\mathrm{IM3L}}}(t) = \sum_{n=2}^{N} \sum_{l=0}^{\frac{n-2}{2}} \frac{|\,k_n\,|}{2^{n-1}} \binom{n}{\frac{n}{2}-l, l, l+1, \frac{n-2}{2}-l} A_{\omega_1}^{2l+2} A_{\omega_2}^{n-2l-1} \,|\,H(\omega)\,|^n \times \cos(\omega_{\mathrm{IM3L}} t + 2\phi_1 - \phi_2 + \phi_{L(\omega_{\mathrm{BBL}})} - \phi_{k_n}) \,. \tag{4.12b}$$

At this point, adding the effect of the output network $K(s)$ is trivial. We have ignored the DC and baseband components that contribute to the linear response of the system at the fundamental stimulus frequencies. The linear term from the first slice dominates this response so this introduces negligible error.

The actual phase reference about which the baseband components are complex conjugates is evident from the results of measurement. The phasor plot in Fig. 4.2 shows how the symmetric magnitudes of the upper and lower baseband terms can create an IM3 responses that has both amplitude and phase asymmetry. A special case of the combination between the odd-order and baseband slice is when these share a common phase reference. In this situation, only phase asymmetries arise, and the IM3 terms rotate in a complex conjugate manner as a function of the ϕ_{k_n}'s which change the overall phase of the IM3 products as the higher order terms dominate at higher input power levels.

4.4 Multi-Slice Model Extraction

Extraction of the multi-slice model uses a combination of single and two-tone measurements. The procedure consists of a type of Gramm-Schmidt fitting geared towards extracting the most dominant contributors to the nonlinear behavior in sequence. This requires the assumption that the contributors are independent, which will be shown not to negatively impact the response of the model. A brief description of the extraction procedure is included here. Application of the multi-slice model to measured amplifier responses is presented in Chapters 6 and 7.

For the two-slice, first-zone model presented above, the measurements necessary consist of:

- Single-tone AM-AM, AM-PM swept over the frequency band of interest.

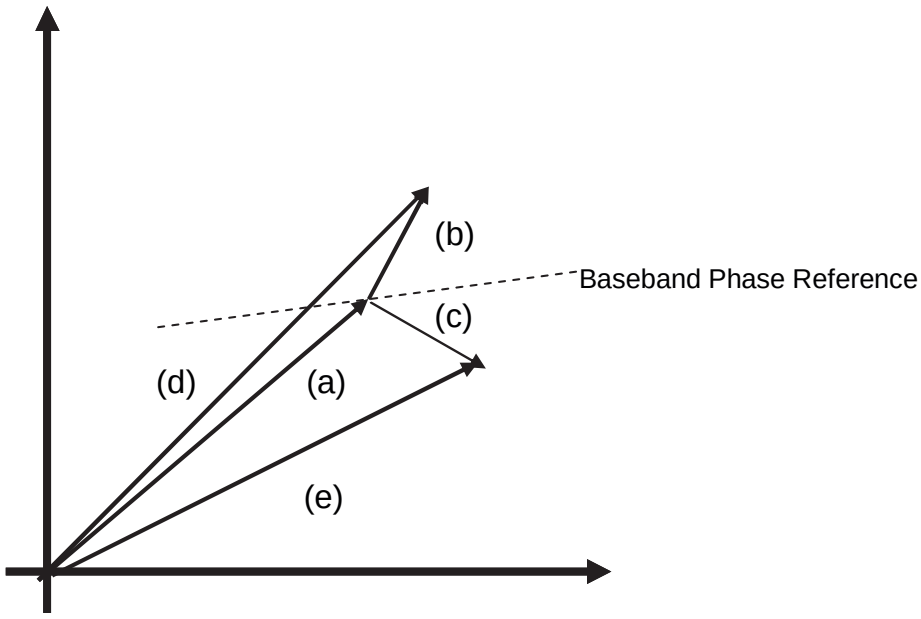


Figure 4.2: Phasor addition of first and second slice components depicting phase reference of baseband components producing asymmetries in IM3 phase and magnitude. Components of the IM3 response are (a) odd-ordered output from first slice, (b) upper (lower) baseband IM contribution from second slice, (c) lower (upper) baseband IM contribution from second slice, (d) total upper (lower) IM3 response from sum of slice outputs, (e) total lower (upper) IM3 response from sum of slice outputs.

- Two-tone swept-tone, swept-power IM product magnitude and phase measurements for a fixed tone separation.
- Two-tone swept-separation, swept-power IM product magnitude and phase measurements.

The extraction procedure begins by extraction of the most dominant slice, that being the strictly odd-ordered response captured in the first slice followed by determination of parameters for the upconverted baseband term captured in the second slice. The frequency response of the linear networks in the first slice are the first parameters extracted. The measurements used are the small-signal of the first two listed above. At small-signal the response of the nonlinearity can be assumed to be low order, third only, and the linear

response is extracted using a ratio between single tone and two-tone response of the DUT. The nonlinearity of the first slice is then extracted using single-tone data and the extracted filter responses. This can be done at a single frequency since it is assumed that the nonlinearity is memoryless. The nonlinearity can be a polynomial or another nonlinear function, with the order of the polynomial being determined by minimization in the least squares sense between the measured data (for the IM response) and the response of the first slice.

Extraction of the baseband component slice assumes that the macro-level memory effects determined in the first slice also shapes the second slice contribution. Thus these elements are the same as in the first slice. The data that is used to extract the second slice is the difference between the measured data and the predicted response from the first slice. The linear network $L(s)$ is extracted using the swept-frequency separation data and the frequency response of the macro-level memory blocks $H(s)$ and $K(s)$ at a single frequency. Here the assumption is that the micro-level memory captured by $L(s)$ is much narrower in bandwidth (on the order of several MHz) than the macro level response that is on the order of several hundreds of MHz. The response of $L(s)$ should be extracted at a power level of significant baseband contribution as this filter is responsible for shaping that response as a function of the baseband frequency. The nonlinearity $g()$ could then be extracted for a single separation frequency as a function of power.

4.5 Summary

The development of a multi-slice behavioral model including its extraction procedure provides a modeling architecture for capturing the complex nonlinear response of RF and microwave systems. The modular nature of the model enables the separation of the multiple nonlinear processes at work and allows extension of the model to capture the response of a DUT in non-first spectral zones. By using a block-oriented architecture, the model allows the use of physically descriptive components in an attempt to relate the model components to known circuit blocks within the DUT. This enables the user to derive intuition from the model that can aid in the optimization of a design or troubleshooting in a test environment. Considerable flexibility in the block implementation allows the model to be implemented in multiple simulator technologies.

The extraction procedure employs a straightforward serial process for determining

the slice parameters proceeding from the strongest contributor to the weakest for a particular frequency zone. Extraction of the model components requires interpretation by the user to properly represent the system under test. This technique differs from purely numerical fitting procedures that are only guided by error minimization techniques. This interaction between the expert knowledge of the user and the architecture of the model further reinforces the intuition gained by employing this modeling technique.

Chapter 5

Remote Characterization of RF Devices

5.1 Introduction

Prior to the research presented here, very little investigation of using an active wireless method for characterizing RF and microwave systems has been published in the literature. While some work has been done with the response of antennas loaded with nonlinear elements, this work focused more on the electromagnetic solution of the nonlinear equations involved than on the characterization of the nonlinear elements themselves [146–153]. Therefore, the work presented here represents the first use of an active probing technique for RF and microwave characterization.

The primary application of a non-contact probing technique would likely be a manufacturing test after assembly of a wireless communication product. Currently, devices and subsystems are characterized individually prior to assembly, after which the task of individual component characterization becomes more difficult if not impossible. As technology progresses and allows higher levels of integration, characterization of systems will become more difficult as the number of access points in a system decreases and the task of isolating out-of-specification components becomes more critical. Thus a non-contact manufacturing

test would present a mechanism to test components in an RF front-end without performing a full functional test.

To investigate the feasibility of the wireless characterization technique, several commercial products were illuminated with the test apparatus described in Section 3.2. The goal of this experimentation was to determine if a nonlinear reflection response (defined in Section 3.2.1) could be generated and detected to allow characterization of RF devices. These devices included a two-way radio, an AMPS cellular phone, and several miscellaneous pieces of electronic equipment. A single-slice model of a two-way radio was developed (Section 5.5) to demonstrate the efficacy of extracting model parameters from the measurements of the wireless probing technique.

5.2 Two-Way Radio Characterization

The first RF system considered for the feasibility study was a consumer-oriented two-way radio. This system was chosen due to its ready availability and low frequency of operation (460–470 MHz). The measurements were taken with the apparatus described previously in Section 3.2. The stimulus used to induce a response from this device was a fixed frequency separation two-tone CW sinusoidal illumination. The frequency separation was kept at 100 kHz as the stimulus was swept over a broadband frequency range covering 400–700 MHz which includes the operating bandwidth of the device.

Analysis of the response suggested that the radio had a Tchebychev filter response in its band of operation. This response was investigated further and additional narrowband frequency sweeps were run with the radio in the power on and off states over several different transmit power settings, (Figs. 5.1 and 5.2). As shown, a noticeable response occurred in both states. In general the response was similar for the two different states except that the power on response was larger in magnitude. This should not be interpreted as amplification of the nonlinear reflection since the hypothesis concerning the source of this response does not allow for the response to pass through an active amplifier. Rather the increase in the magnitude is likely due to a change in the physical phenomena responsible for the nonlinear input impedance of the first active device in the receive path, i.e. a properly biased semiconductor junction for the power on case compared to a floating junction in the power off situation.

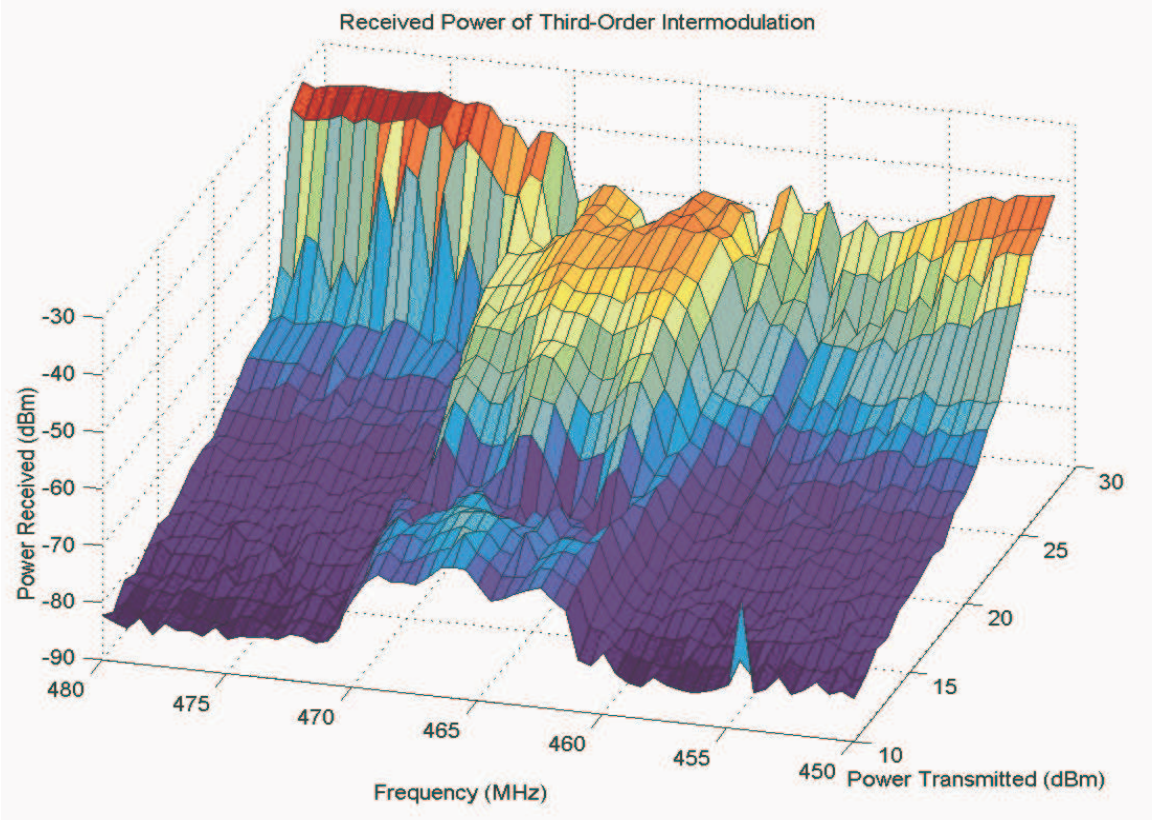


Figure 5.1: Narrow band IM3L response of a powered two-way radio to RF illumination under swept frequency and power.

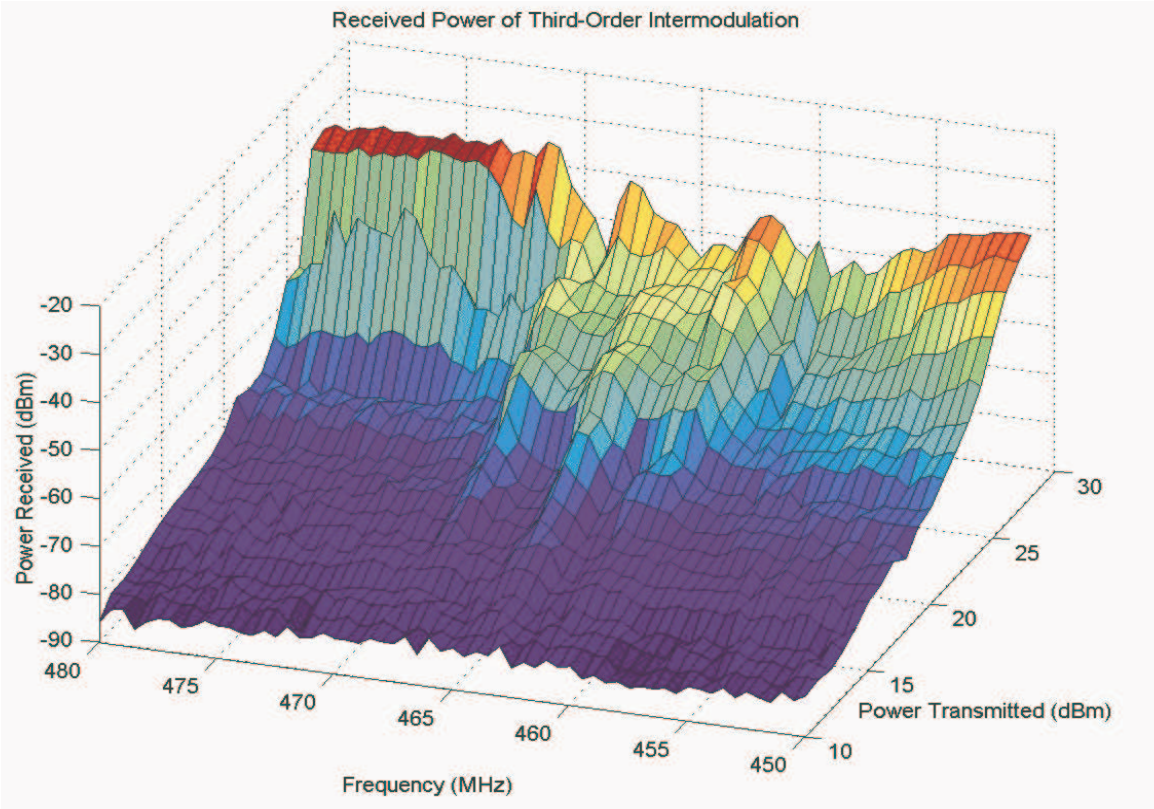


Figure 5.2: Narrow band IM3L response of an un-powered two-way radio to RF illumination under swept frequency and power.

5.3 AMPS Cellular Phone Characterization

An AMPS cellular phone was selected as the next target for the phenomenological study of the wireless characterization technique. The AMPS standard defines a transmit and receive band, which for the handset tested ranged from 824–849 MHz and 869–894 MHz respectively. This device was only tested in the power off state, but a response in the transmit and receive bands was observed, (Figs. 5.3 and 5.4). The receive band generated a considerably stronger response than the transmit band. This is likely due to the output of the transmit PA appearing more linear than the receiver due to resistive output matching and biasing. With the use of the wireless characterization technique on this phone, the reliability of this measurement to provide a measurable response for any type of RF system was established. Further testing on non-RF electronic devices remained to determine if characteristic responses could be measured for devices without an intentional antenna.

5.4 Characterization of Various Electronic Products

Since the core of the idea behind the wireless characterization of RF devices is that of nonlinear reflection resulting from the interaction between the semiconductor components with a physical connection to the antenna, the same phenomena should result when non-wireless products are illuminated. Experiments on two different pieces of electronics were conducted to prove this hypothesis.

The first item illuminated was a PCI video card from a PC computer. The card was tested in the same manner as all of the other components with a very broadband sweep of the stimulus frequency. This device did indeed emit a nonlinear response to the stimulus as shown in Fig. 5.5. In contrast to the RF systems tested, the response of the PCI card was much more broadband as could be expected since the response was purely a function of the interaction of the onboard digital circuitry which are very broadband in nature. In addition, the card only responded to the stimulus when physically oriented in a particular direction to the polarization of the transmit and receive antennas in the measurement apparatus. This was due to the nature of the "antenna" on the card which was provided by a long circuit trace. The second item illuminated was an unpowered laptop computer. This item was probed with the same frequency range and power levels as the PCI card, but no discernible response was observed. The failure of this technique to extract a meaningful

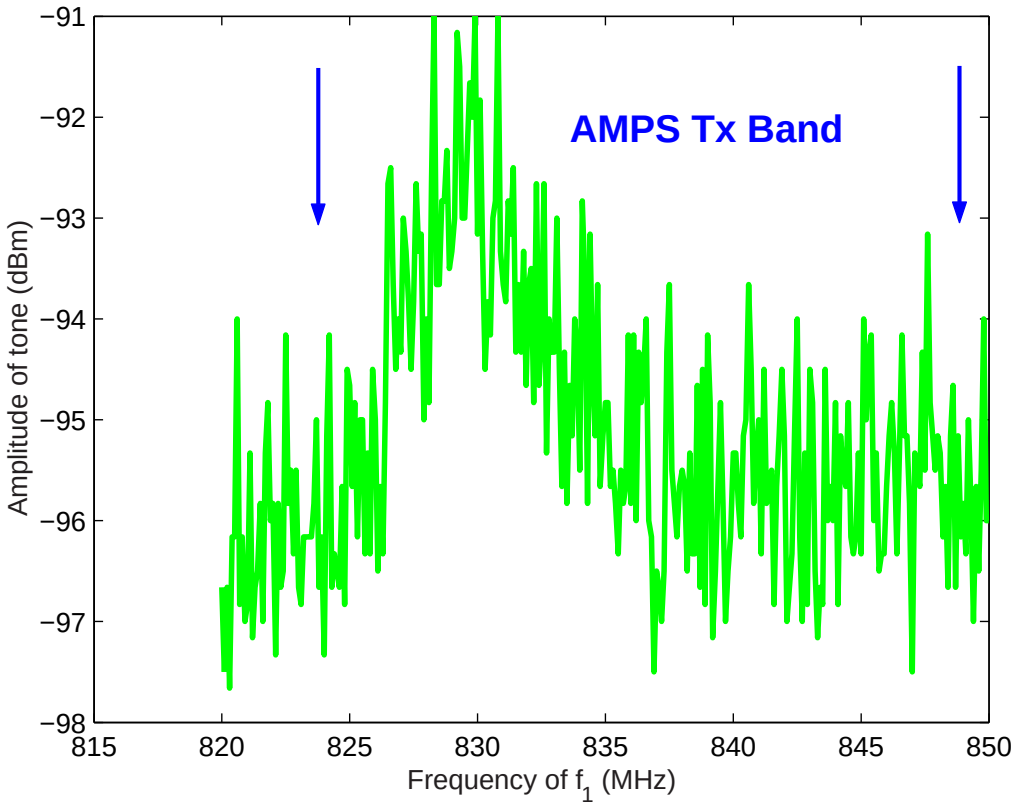


Figure 5.3: Frequency response of IM3L in the handset transmit band for RF illumination of un-powered AMPS cellular phone.

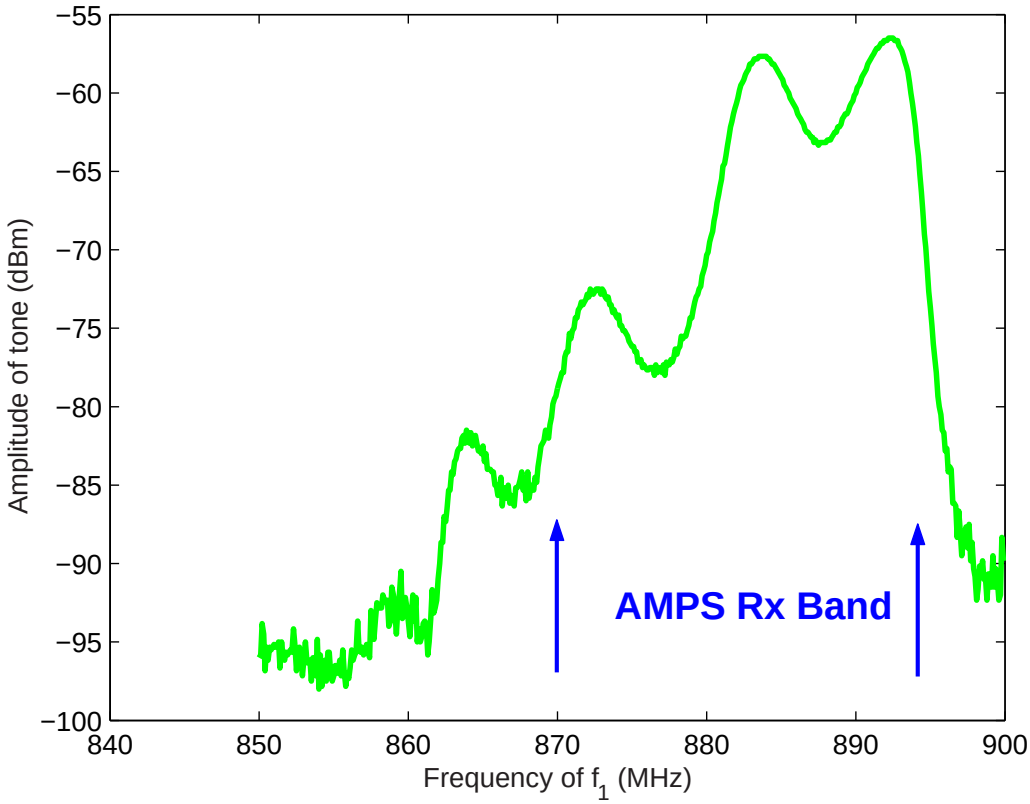


Figure 5.4: Frequency response of IM3L in the handset receive band for RF illumination of un-powered AMPS cellular phone.

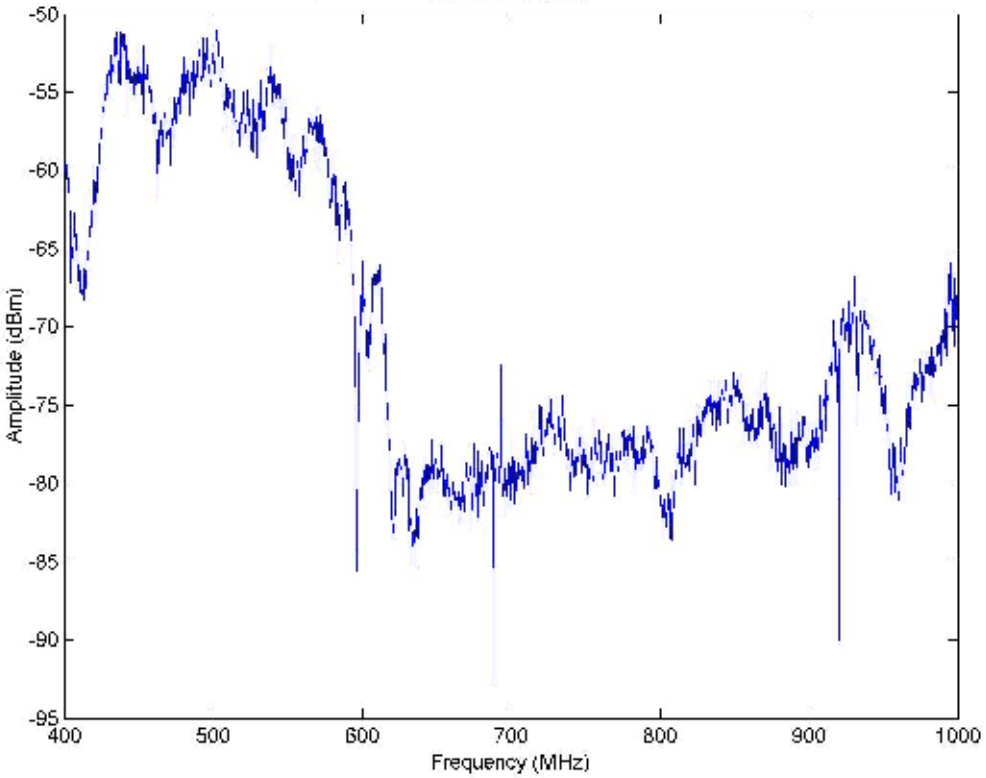


Figure 5.5: Frequency response of IM3L product from PCI video card under wide-band RF illumination.

characterization from this device likely was the result of insufficient stimulus power to elicit a response, or no unintentional antenna within the circuitry over the frequencies used. Since the focus of this research effort is for the characterization of RF and microwave devices with an inherent RF port for wireless characterization signal, no additional work was performed on general electronic equipment.

5.5 Development of a Single-Slice Model for the Two-Way Radio Response

With the phenomenology of nonlinear reflection and subsequent unique response from a wide variety of devices thoroughly investigated, the next step in development of the wireless characterization method involved creating a model to predict the system response to the non-contact stimulus. The data collected during the analysis of the two-way radios was selected for the development of a single slice model of the form in Fig. 3.4. Since the wireless characterization method used a multi-tone stimulus, there was no single tone information to base the model on as in AM-AM, AM-PM models. Therefore, the model developed for this device used only measured third-order and fifth-order IMD data, (only the lower IMD products were captured).

The complete response of the block-cascade system with real coefficients was given in (4.4) and can be re-written as

$$y(t) = \sum_{n=1}^{N} \frac{a_n}{2^n} \Big(\sum_{q_1=-Q}^{Q} \cdots \sum_{q_n=-Q}^{Q} A_n(q_1, \ldots, q_n) K(\omega_{q_1} + \ldots + \omega_{q_n}) \exp^{j(\omega_{q_1} + \ldots + \omega_{q_n})t} \Big) \ , \quad (5.1)$$

where a_n is the n^{th} order power series coefficient and A_n is a function of the input sinusoid amplitudes, A_q, and the response of the linear system $H(s)$ to the inputs

$$A_n(q_1, q_2, \ldots, q_n) = A_{q_1} A_{q_2} \ldots A_{q_n} H(j\omega_{q_1}) H(j\omega_{q_2}) \ldots H(j\omega_{q_n}) \ . \quad (5.2)$$

The total number of distinct sinusoids in the input is Q and the q_i terms are the particular frequency components of a particular summation in (5.1). It is clear that even for a small number of input sinusoids Q and a moderate order nonlinearity, N, the total number of terms in the solution of the system can become quite high. Many of these terms are not unique however, as they occur at the same frequency as other terms. Using the frequency

mix concept, introduced by Weiner and Spina in [14] and described in Chapter 4, simplifies computation of the spectral output of the model.

Now separating each order of the response in (5.1) results in

$$y(t) = \sum_{n=1}^{N} y_n(t) \ . \tag{5.3}$$

Combining this with the frequency mix vector, (5.1) can be rewritten as a sum over all possible frequency mixes and order of the nonlinearity as

$$y(t) = \sum_{n=1}^{N} \sum_{\vec{m}} y_n(t;\vec{m}) \ , \tag{5.4}$$

where $y_n(t;\vec{m})$ is the combination of all terms from (5.1) that have the same frequency mix vector $\vec{m}$ and thus produce a response at the same frequency. This output term is given by

$$y_n(t;\vec{m}) = \frac{c_{\vec{m},n}}{2^n} a_n A_n(\vec{m}) K(\omega_{-Q} m_{-Q} + \ldots + \omega_Q m_Q) \exp^{j(\omega_{-Q} m_{-Q} + \ldots + \omega_Q m_Q)t} \ , \tag{5.5}$$

with $c_{\vec{m},n}$ being the multinomial coefficient defined in (4.7) that specifies the number of terms in (5.1) that produce the same frequency mix vector for a given order n. Now since (5.5) only considers positive frequency mixes, and the negative frequency mixes must be considered for completeness, the total response for the n^{th} order component of a particular frequency mix is

$$y_n(t;\vec{m}) = \frac{c_{\vec{m},n}}{2^{n-1}} a_n \mid A_n(\vec{m}) \mid\mid K(\omega_{-Q} m_{-Q} + \ldots + \omega_Q m_Q) \mid \cos\left(\omega_{-Q} m_{-Q} + \ldots + \omega_Q m_Q\right)t \ . \tag{5.6}$$

Since the development of the response assumed the system was at steady state, the frequency domain response can be written as

$$Y_n(\omega_{\vec{m}}) = \frac{c_{\vec{m},n}}{2^{n-1}} a_n \mid A_n(\vec{m}) \mid\mid K(\omega_{-Q} m_{-Q} + \ldots + \omega_Q m_Q) \mid \ . \tag{5.7}$$

For the purposes of the models generated here, the relevant intermodulation products resulting from a two tone excitation were $2\omega_1 - \omega_2$ and $3\omega_1 - 2\omega_2$. Each of these frequencies has a unique frequency mix vector designated as $\vec{m}_1$ and $\vec{m}_2$ respectively, (with $\omega_2 > \omega_1$). For the third-order model only the first intermodulation product was considered, while for the fifth-order model both of these products were considered. Using (5.7) a third-order system has a response at $2\omega_1 - \omega_2$ given by

$$Y_3(\omega_{\vec{m}_2}) = \frac{3}{4}a_3 |\ A_{\omega_1}\ |^2 |\ A_{\omega_2}\ ||\ H(\omega_1)\ |^2 |\ H(\omega_2)\ |\ |\ K(2\omega_1 - \omega_2)\ |\ . \tag{5.8}$$

For a fifth-order system the response at $\vec{m}_2$ is

$$Y_5(\omega_{\vec{m}_2}) = \frac{5}{8}a_5 |\ A_{\omega_1}\ |^3 |\ A_{\omega_2}\ |^2 |\ H(\omega_1)\ |^3 |\ H(\omega_2)\ |^2\ |\ K(3\omega_1 - 2\omega_2)\ |\ . \tag{5.9}$$

For the response of a fifth-order system at $\vec{m}_1$, the frequency mix vectors of the third-order component and the fifth-order component of (5.1) that produce this frequency content must be considered. The third-order term is identical to (5.8) and the fifth-order component is represented by two frequency mix vectors $\vec{m}_a$ and $\vec{m}_b$ that produce components at $2\omega_1 - \omega_2$, given by

$$Y_5(\omega_{\vec{m}_a}) = \frac{15}{8}a_5 |\ A_{\omega_1}\ |^2 |\ A_{\omega_2}\ |^3 |\ H(\omega_1)\ |^2 |\ H(\omega_2)\ |^3\ |\ K(2\omega_1 - \omega_2)\ | \tag{5.10}$$

and

$$Y_5(\omega_{\vec{m}_b}) = \frac{5}{4}a_5 |\ A_{\omega_1}\ |^4 |\ A_{\omega_2}\ ||\ H(\omega_1)\ |^4 |\ H(\omega_2)\ |\ |\ K(2\omega_1 - \omega_2)\ |\ . \tag{5.11}$$

For a nonlinear model truncated to the fifth order, the total response at $2\omega_1 - \omega_2$ is

$$Y(2\omega_1 - \omega_2) = Y_3(\omega_{\vec{m}_1}) + Y_5(\omega_{\vec{m}_a}) + Y_5(\omega_{\vec{m}_b})\ . \tag{5.12}$$

5.5.1 Power Off Third-Order Model Extraction

The first attempt at modeling the measured system response assumed a third-order power series nonlinearity. The apparent Tchebychev response in the measured data formed the basis of the linear portion of the system response. The guiding equation of this model was (5.8), with the realization that the post-linear block of Fig. 3.4 is identical to the pre-linear block. Restating (5.8) with this information yields

$$Y_3(\omega_{\vec{m}_1}) = \frac{3}{4}a_3 |\ A_{\omega_1}\ |^2 |\ A_{\omega_2}\ ||\ H(\omega_1)\ |^2 |\ H(\omega_2)\ |\ |\ H(2\omega_1 - \omega_2)\ |\ . \tag{5.13}$$

Examination of (5.13) shows that if the linear system response and the magnitude of the input sinusoids were known, then extracting the power series coefficient from the measured $Y_3(\omega_{\vec{m}_1})$ is straightforward. However, part of the modeling process involves determining the parameters of the linear response, in this case the center frequency, bandwidth, and ripple of a Tchebychev filter. At this point a simplifying assumption is made regarding the

linear system response at the three different frequencies of (5.13). With a small frequency gap between the primary tones, it is assumed that response of the linear system is nearly constant across the frequencies ω_1, ω_2, and $2\omega_1 - \omega_2$, and can be approximated as $H(j\omega_1)$. This allows (5.13) to be rewritten as

$$Y_3(\omega_{\vec{m}_1}) = \frac{3}{4}a_3 \mid A_{\omega_1} \mid^2 \mid A_{\omega_2} \mid\mid H(\omega_1) \mid^4 \tag{5.14}$$

At this point it becomes clear that the estimation of the linear system response is of critical importance since any error in this portion of the response will be compounded by the exponent of that term. Comparatively, error in the estimate of the power series coefficient will not have as great an impact on the accuracy of the model.

The first step in extracting the parameters of the model began with normalizing the measured system response such that the peak value occurred at 0 dB. Where normalization consisted of taking the root of the measured response by the order of the nonlinearity plus one to account for the same linear network at the input and output of the linear-nonlinear-linear block. This was done at four different input powers, providing multiple data points since the linear system model should be the same for any input power. An estimate of the linear system parameters were then iterated upon until a sufficient match was obtained. Results of this step are shown in Fig. 5.6. The estimated values of the parameters for the linear system where: $f_0 = 464.5$ MHz, $BW = 8.6$ MHz, and a ripple of 2.25 dB. These parameters provide good agreement with the input powers used to estimate of 21, 22, and 23 dBm, although a rather poor fit of the 20 dBm data.

The next step in the modeling process was to extract the power series coefficient a_3. Since the model used was limited to third-order, this was done by solving (5.14) for a_3 and choosing a value from the 21 dBm data set. A more sophisticated extraction routine would utilize the actual slope of the IMD magnitude as a function of the input power to properly determine the order of the nonlinearity. The value obtained was $a_3 = 6.7 \times 10^{-3}$ V^{-2}. For this model, a_3 represents an aggregate nonlinear coefficient comprised of the actual power series coefficient, the transmit and receive efficiencies of the DUT antenna, antenna efficiency of the receiver antenna, and the geometrical attenuation of the signal as it travels from the DUT to the spectrum analyzer receive antenna. These terms are all linear in nature and they combine in a multiplicative term that can be unwrapped to provide more detail on the actual physics of the DUT response. A comparison of the estimated and measured response at $2\omega_1 - \omega_2$ is shown in Fig. 5.7. This figure reveals that the extracted

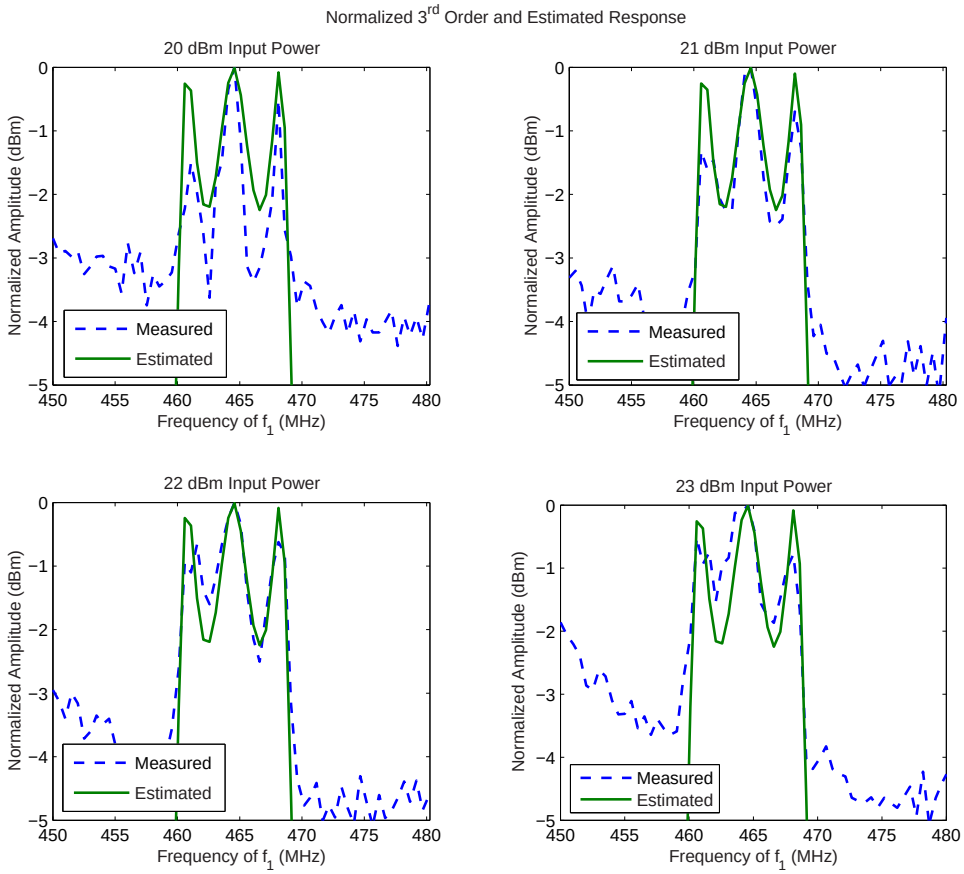


Figure 5.6: Two-way radio normalized linear system estimation.

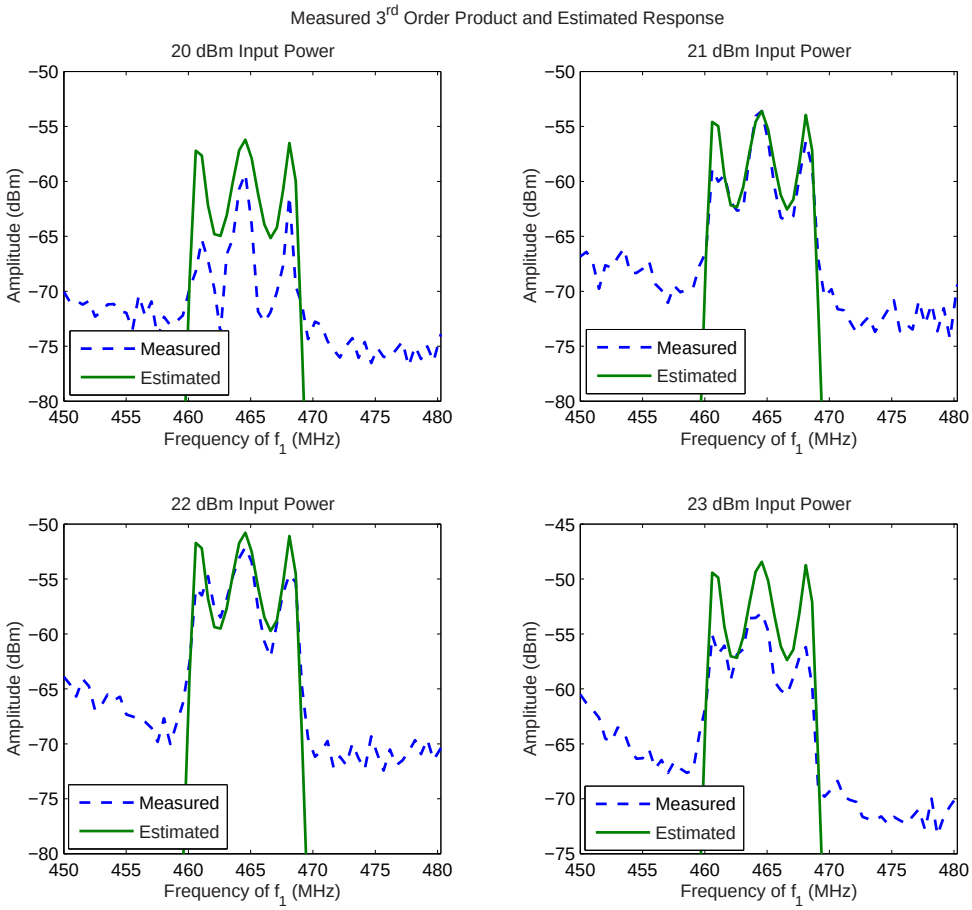


Figure 5.7: Third-order model extraction results for a two-way radio in the power off state

model fits well with the power at which the power series coefficient was obtained, but less well with the other input powers. This indicates that the nonlinear response of the radio is not limited to third-order and has contributions at $2\omega_1 - \omega_2$ from higher order components. With this information, a fifth-order model was extracted using both the third-order and fifth-order IMD product measurements.

5.5.2 Power Off Fifth-Order Model Extraction

Due to the deviation of the third-order model from the measured data, a higher order model of the nonlinearity was sought. The same basic steps as the previous section were taken except that the linear system and fifth-order power series coefficient were extracted by utilizing (5.9) in the form of,

$$Y_5(\omega_{\vec{m}_2}) = \frac{5}{8} a_5 |\, A_{\omega_1} \,|^3 |\, A_{\omega_2} \,|^2 |\, H(\omega_1) \,|^6 \,. \tag{5.15}$$

For this model, the center frequency and bandwidth of the filter remained the same, but the estimated ripple that provided a best fit was 1 dB, (see Fig. 5.8). The fit however suffered due to the poor quality of the data used for the fifth order extraction. The system response at this frequency was very near the noise floor of the spectrum analyzer for the settings in this experiment. Once the fifth-order model had been extracted, the third-order component of the model could be obtained by using (5.8), (5.10), (5.11), and (5.12). Once the power series coefficient for this component was computed, the total estimated response at $2\omega_1 - \omega_2$ was compared against the measured response, Fig. (5.9). This figure shows that fifth-order model provides a very poor fit of the measured data except for the input power that was used to extract the power series coefficients, 21 dBm. This further reinforces the conclusion that the nonlinear coefficients should be extracted using techniques similar to the single-tone techniques which fit a polynomial at a particular frequency to the nonlinear magnitude transfer function.

5.5.3 Power On Model Extraction

Formulation of a model for the power on state of the two-way radio followed the same process as the power off state. A third order model was developed, then a fifth-order model was attempted. The results of the fifth-order model however, illuminated the need

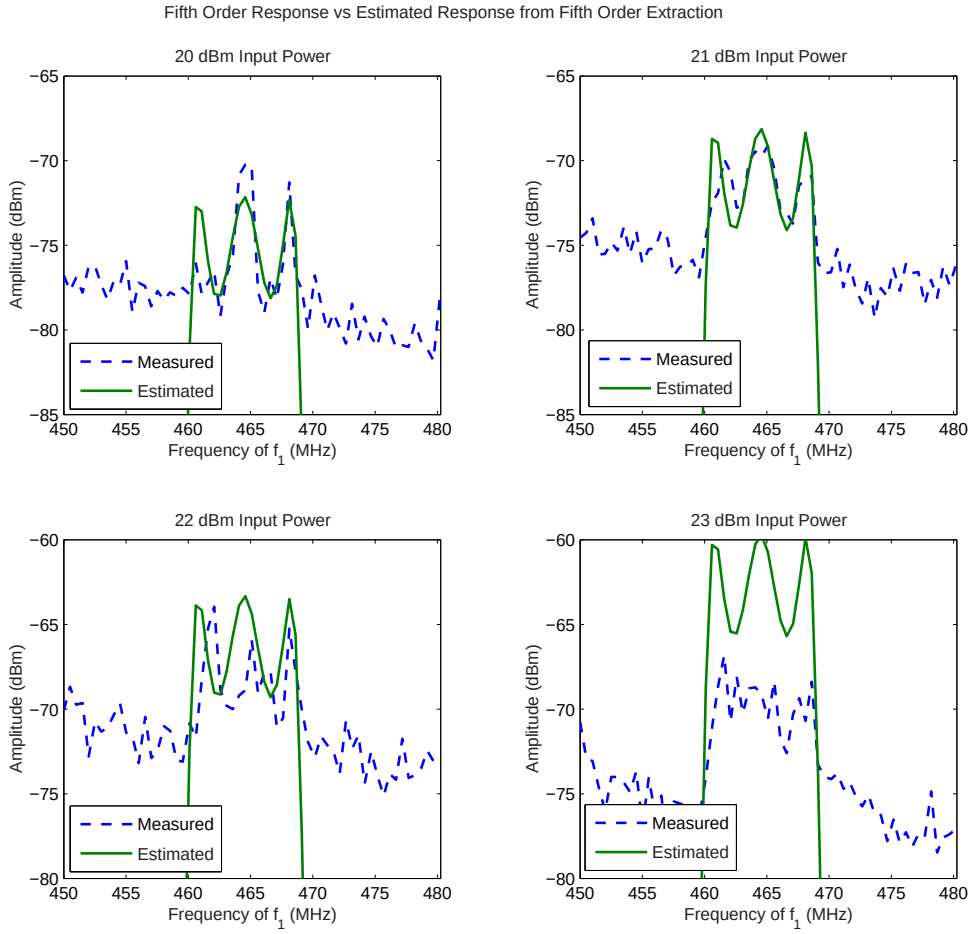


Figure 5.8: Estimate of the fifth-order intermodulation response for a two-way radio in the power off state based on a fifth-order model extraction

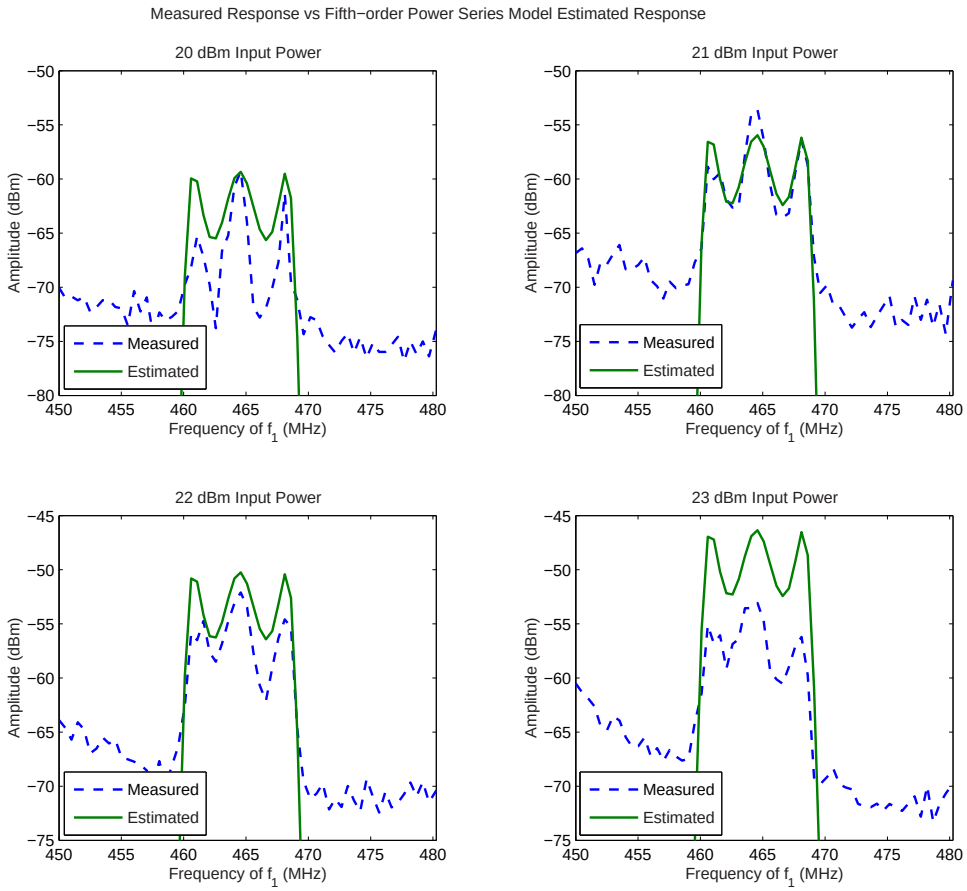


Figure 5.9: Estimate of the third-order intermodulation response for a two-way radio in the power off state based on a fifth-order model extraction

for a more sophisticated model since the parameters identified in this paper were insufficient to capture the system behavior.

5.5.4 Power On Third-Order Model Extraction

The first pass estimate of the parameters for this model were to use those found in the power off model. This model turned out to be a very poor match in both linear response and nonlinear response. Iterations upon the linear response resulted in a filter response with the same bandwidth and ripple parameters as in the power off state, but the center frequency was shifted higher to 466.1 MHz. Intuitively, this suggests that the powered circuit that creates the nonlinear reflection underwent a change in the input reactance which changed the filter transfer function. The results of this model are compared with the measured data in Fig. 5.10.

5.5.5 Power On Fifth-Order Model Extraction

The results of the fifth-order model extraction highlighted the limitations of this preliminary modeling approach. As in the power off case, the first step was to estimate the linear system parameters and the fifth-order power series coefficient at $3\omega_1 - 2\omega_2$. The coefficient extracted at a single power produced a model that largely overestimated the value of this frequency term at other input power levels. Reviewing the data taken, the power at this frequency appeared to be relatively constant over a 3 dB increase of input power. This outcome contradicts that predicted by (5.9) since this equation in dB indicates that for every 1 dB increment in input power, the output power should increase by 5 dB. Clearly, the nonlinear response does not follow the expected 3:1 or 5:1 slope for the third-order or fifth-order IMD products (respectively), predicted by the classical polynomial model.

5.6 Conclusion

The successful reproduction of the device response with a model based on measurements using the wireless characterization technique demonstrated the usefulness of this method. However, discrepancies in the prediction of device response over a wide power range pointed towards the need for a more thorough modeling strategy. These arose from

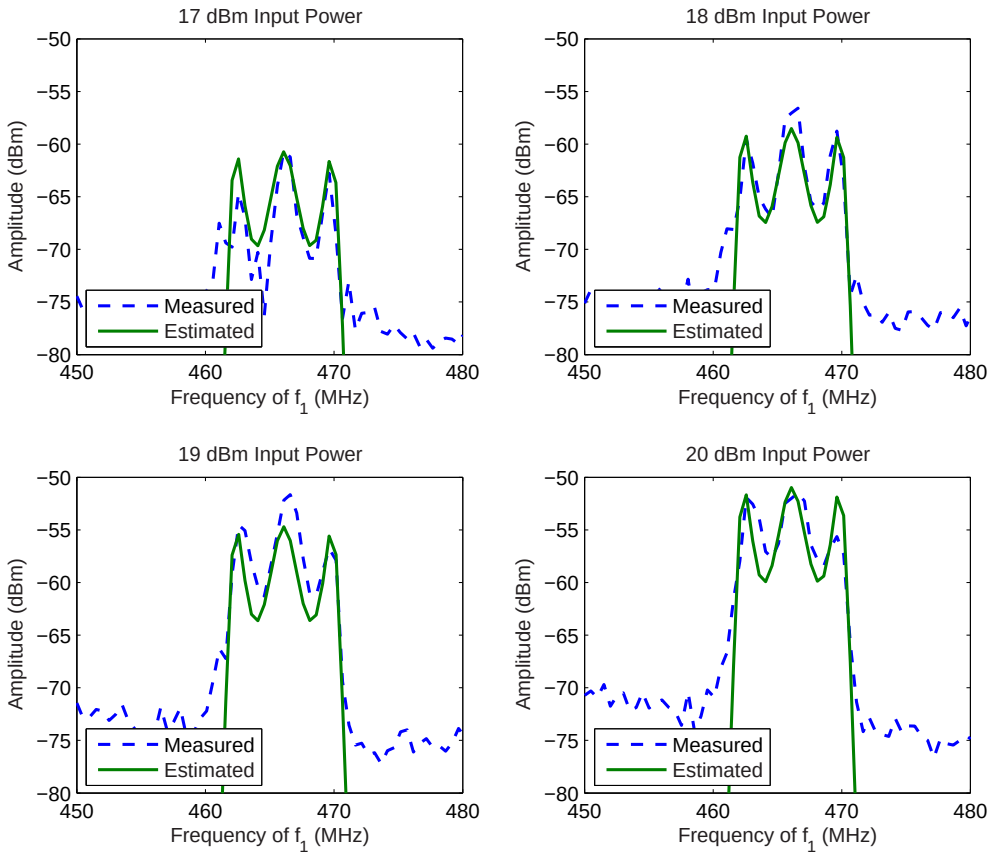


Figure 5.10: Estimate of the third-order intermodulation response for a two-way radio in the power on state based on a third-order extraction

a lack of single-tone measurements for extraction of the nonlinear polynomial coefficients. The stimulus/response mechanism was shown to reliably elicit a unique response from several types of RF devices and even some non-RF devices. With additional analysis this response can be used both to identify and develop models of the device under test using straightforward extraction procedures based on the classical polynomial model combined with the Wiener-Hammerstein model architecture. In addition to the ability to distinguish between different types of devices, the results of developing the model of the two-way radio demonstrated that the wireless characterization technique can distinguish between the power states of a device. This could be used to determine whether the bias point of the RF front-end component was properly set after system assembly, yielding additional characterization information besides that of the steady-state frequency response of the input linear network composed of the antenna and the band-select filter.

Chapter 6

A Low-Noise Amplifier Modeled with a Multi-Slice Behavioral Model

6.1 Introduction

Although the initial investigation of the wireless characterization system in Chapter 5 yielded promising results, the difficulties in achieving consistent measurement without the use of an anechoic chamber limited the ability of using this technique to develop more sophisticated models. All of the measurements presented in Chapter 5 were performed in a laboratory environment, not inside of a reflection and interference free anechoic chamber. Reflections from the walls of the room and nonlinear responses from various electronic devices within the lab contributed to the environmental noise level that limited the measurement consistency. Towards providing a more consistent measurement platform for model development, the wired measurement platform described in Section 3.1 was employed. Since the nonlinear reflection phenomenon is similar to that of the more traditional unilateral transmission modeling employed in the microwave industry, development of the multi-slice

model architecture utilized transmission measurements for the source of the characterization data. Using this approach, a low-noise amplifier (LNA) operating in the PCS band (1900 MHz), that exhibited significant IMD amplitude asymmetry, served as the test device for a multi-slice model based only on frequency-domain magnitude measurements.

6.2 Amplifier Multi-Slice Behavioral Model Extraction

Since the main purpose of the multi-slice model is to capture nonlinear behavior that cannot be predicted by classical polynomial modeling, validation of this modeling approach began by characterizing an amplifier with a high-level of observed IMD magnitude asymmetry. As stated previously, polynomial modeling, even with complex coefficients, cannot capture asymmetries between upper and lower IMD products resulting from odd-ordered nonlinearities. In addition to modeling the amplitude asymmetry, this model also sought to capture the wideband macro-memory effects introduced by the linear networks preceding and following the nonlinear device, which are composed of the input and output matching networks of the LNA. The micro-memory effects that result in the IMD asymmetry are well-known to result from baseband upconversion, thus they are a function of the frequency separation in the two-tone stimulus. For this model, data was taken for only one separation so the linear network $L(s)$ in Fig. 6.1 was not specifically extracted. Computation of the model parameters consisted of using AM-AM and two-tone measurements to first extract the linear network frequency responses and then extract the nonlinearity coefficients of memoryless polynomials.

The measurements needed for the extraction of this model included the following:

- IM3 product magnitudes for frequency sweep of constant separation two tone stimulus across the band of interest and sweep of stimulus power.
- Single-tone AM-AM measurement using VNA at frequency points corresponding to those in the two-tone sweep.

6.2.1 Linear Networks

Since the nonlinear functions are assumed to be memoryless, the linear network frequency responses must be extracted prior to estimating the nonlinear function parame-

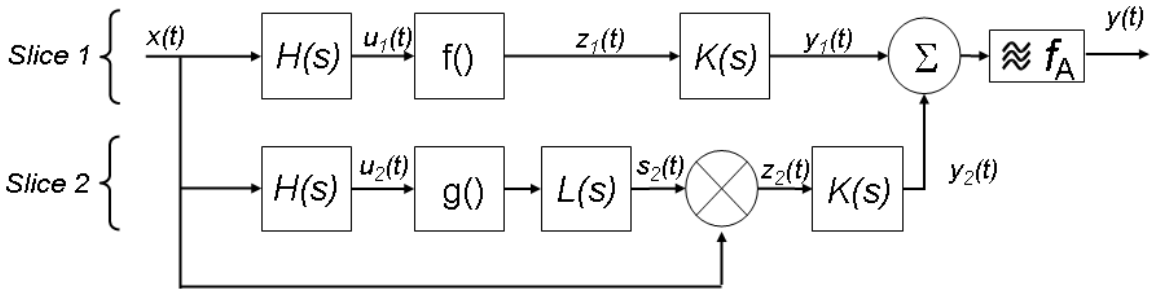


Figure 6.1: A two slice nonlinear system behavioral model with passband modeling.

ters. The basic assumption in this modeling effort is that all of the broadband memory effects can be captured by the linear networks. The extraction technique used to determine these networks attempts to associate them with actual input and output linear networks present in the system such as the matching and bias circuits. Since these networks are taken as real networks in the system, the linear networks for both slices of the model are identical. This is in contrast to other models where the networks are extracted in a self-consistent manner to ensure that the model tracks the frequency response of the measured system.

The extraction procedure is performed from measurements taken at small-signal input powers where there are no IM3 asymmetries and the effect of nonlinear terms above third order are insignificant. With no asymmetries, only the first slice of the model is used for the extraction since it captures the fundamental output response and the IM3 response at small-signal. The two-tone stimulus used in the measurements have 100 kHz separation so the theoretical model of the IM3 components considers that the input stimulus and the IM3 products are all at the same frequency. Under these conditions, the steady-state model output for the magnitude under single tone stimulus is,

$$v_{f_1,\text{out}} = \hat{a}_1 v_{\text{in}} |H(\omega)||K(\omega)| \ , \tag{6.1}$$

and the magnitude of the IM3 products generated during two-tone stimulus are given by

$$v_{\text{IM3,out}} = \frac{3}{4}\hat{a}_3 v_{\text{in},\omega_1}^2 v_{\text{in},\omega_2} |H(\omega)|^3 |K(\omega)|. \tag{6.2}$$

Here the terms $\hat{a}_i$ are the real polynomial coefficients for the nonlinear functions, which are

unknown at this time. By considering the fundamental and the IM3 products as occurring at the same frequency, a simple ratio of the predicted responses allows computation of $H(\omega)$ in the frequency domain:

$$\frac{v_{\mathrm{IM3,out}}}{v_{f_1,\mathrm{out}}} = \frac{\frac{3}{4}\hat{a}_3 v_{\mathrm{in},\omega_1}^2 v_{\mathrm{in},\omega_2}|H(\omega)|^3|K(\omega)|}{\hat{a}_1 v_{\mathrm{in},\omega_1}|H(\omega)||K(\omega)|}. \tag{6.3}$$

With input tones in the two-tone case having equal amplitudes this reduces to

$$\frac{v_{\mathrm{IM3,out}}}{v_{\mathrm{f_1,out}}} = \frac{\frac{3}{4}\hat{a}_3}{\hat{a}_1} v_{\mathrm{in},\omega_1}^2 |H(\omega)|^2. \tag{6.4}$$

Now since $H(\omega)$ is the only frequency dependent term we can normalize (6.4) along the frequency axis with respect to the maximum value of the ratio such that $H(\omega)$ has unity gain at one point in its frequency response.

With the leading linear network frequency response captured, the trailing network frequency response can now be obtained. Starting with the expression for the fundamental output tone, (6.1), the leading linear network response is divided out yielding

$$\frac{v_{f_1,\mathrm{out}}}{|H(\omega)|} = \hat{a}_1 v_{\mathrm{in},\omega} \mid K(\omega) \mid . \tag{6.5}$$

Again we can take the trailing linear system as having a lossless response at its maximum magnitude since this only introduces an unimportant constant that shifts the magnitude of the linear network response. The linear gain coefficient of the nonlinearity will incorporate the linear network losses.

6.2.2 Nonlinear Blocks

Once the linear network responses have been extracted the nonlinear functions can be estimated. The extraction method for the two slices in this model differ since they capture different system behavior.

The first slice nonlinearity produces the broadband fundamental output for all power levels and the system IM3 response under small-signal conditions. The nonlinearity is modelled by an odd-order polynomial by using a modified fit to the AM-AM data. The modification uses information from the two-tone small-signal response to produce a better fit to the IM3 response at small input powers. The coefficients are real since using AM-PM data for complex coefficient extraction would require knowing the phase of the IM3 products

in the model development which was not measured during this effort. Output from this slice at the IM3 tones is given by (4.9), expressed as

$$y_{\mathrm{IM3},1} = \sum_{n=3}^{N} \sum_{l=0}^{\frac{n-3}{2}} a_{n,l} v_{\mathrm{in}}^{n} |H(\omega)|^{n} |K(\omega)| \tag{6.6}$$

where N is the order of the polynomial and $a_{n,l}$ is the modified memory-less polynomial coefficient $\hat{a}_n$,

$$a_{n,l} = \frac{\hat{a}_n}{2^{n-1}} \binom{n}{\frac{n-3}{2} - l, l+1, l, \frac{n+1}{2} - l}. \tag{6.7}$$

The extraction of the coefficients $\hat{a}_n$ requires de-embedding the filter response and the multinomial coefficient from the fit to the single tone data since the fit of the polynomial is to the following equation from Chapter 4

$$y(t) = \mathrm{Re}\{\sum_{n=1}^{N} a_n x^n\} \, . \tag{6.8}$$

which takes x as the magnitude of the input stimulus in the voltage domain. The fit is performed at the frequency at which the AM-AM data was taken, chosen such that the maximum nonlinear response occurs, which provides a better fit to the nonlinearity.

The second slice nonlinearity produces the IM3 asymmetry seen in measured data with increasing input power. As shown in Section 4.3.3, the odd-order polynomial in the first slice cannot produce this effect since each of the IM3 tones is generated by the same coefficients. Physical understanding of the IM3 asymmetry suggests that this phenomena is a result of differences in the baseband impedance at positive and negative envelope frequencies, [18]. It has been established in [91], that the IM3 components and the fundamental have the same phase for the weakly nonlinear region of a system. This suggests that the phase reference for the baseband contribution to the IM3 products is the same as that predicted by the output of the first slice. The additional phase shift added by the impedance at baseband that generates the IM3 asymmetry can to first order be attributed to an expansion term for one of the tones and a compression term for the other. Since the actual phase shift is not known, this is taken as if the baseband contribution to the IM3 tones has an in-phase and an 180° component. As to which IM3 component contains the expansion and which the compression, the second-slice model considers the following baseband contributions to IM3.

For a second-order nonlinearity with real coefficients, the baseband components generated at $\omega_1 - \omega_2$ and $\omega_2 - \omega_1$ from (4.11) are

$$v_{bb,L} = \frac{1}{2}\hat{a}_2 v_{\text{in}}^2 |H(\omega)|^2 |L(\omega_1 - \omega_2)| \cos((\omega_1 - \omega_2)t - \phi_L) \ , \tag{6.9}$$

and

$$v_{bb,H} = \frac{1}{2}\hat{a}_2 v_{\text{in}}^2 |H(\omega)|^2 |L(\omega_2 - \omega_1)| \cos((\omega_2 - \omega_1)t + \phi_L) \ . \tag{6.10}$$

Here the linear network $L(s)$ is taken as a real network having an even amplitude response and an odd-phase response thus the effect of this component is to contribute a phase shift to the upper and lower baseband products in a conjugate manner.

Now the baseband components in (6.9) and (6.10) are translated to the IM3 tone frequencies by mixing with the fundamentals as shown in Fig. 6.1. These are given at steady state by

$$v_{\text{IM3L},bb} = \frac{1}{2}\hat{a}_2 v_{\text{in}}^3 |H(\omega)|^2 |L(\omega_1 - \omega_2)||K(\omega)| \cos((\omega_1 - \omega_2)t - \phi_L) \ , \tag{6.11}$$

and

$$v_{\text{IM3H},bb} = \frac{1}{2}\hat{a}_2 v_{\text{in}}^3 |H(\omega)|^2 |L(\omega_2 - \omega_1)||K(\omega)| \cos((\omega_2 - \omega_1)t + \phi_L) \ . \tag{6.12}$$

As (6.11) and (6.12) show, the baseband components upconverted to the IM3 tones have opposite phase shifts associated with respect to each other. A similar derivation can be applied to higher order terms in an even order polynomial representation of the second-slice nonlinearity. Since only a single frequency separation was measured and only the amplitude information was available, the actual phase introduced by $L(s)$ could not be determined. This limited the choice of the phase to zero and 180°. The even-order polynomial coefficients that determined the magnitude of the baseband component were extracted by fitting half of the measured difference in amplitude between the upper and lower IM3 products. This difference was then added and subtracted from the first slice output which predicts a magnitude response between the two IM3 products.

6.3 Application of the Magnitude-Only Model

The model extraction techniques presented above were applied to measurements taken on the LNA portion of the RFMD 2486 LNA+mixer IC. This device is intended for PCS applications and has an operational bandwidth from 1.5–2.5 GHz. Data was taken

over 2–4 GHz, which allowed modeling both in-band and out-of-band performance. Power levels of the stimulus were swept from −15 to −1 dBm. The two-tone measurements were taken with tones of equal power and constant frequency separation (100 kHz) as the tones were swept across the frequency band. The model extraction and implementation were performed in Matlab ® using frequency-domain expressions for the steady-state response. This device exhibited IM3 asymmetry over much of the frequency and power range of the stimulus. In the high-frequency, low-power region, the asymmetry was < 1 dB, while in the low-frequency, high-power region the asymmetry had a maximum value of 8 dB. The dependence of the asymmetry on the frequency of the stimulus tones will be explained in the following.

Extraction of the multi-slice model linear network frequency responses were computed from the −15 dBm input power point, (where there was little IM3 asymmetry), using AM-AM and IM3 measurements as in (6.4) and (6.5). As shown in Fig. 6.2, the pre-filter response has 10 dB of attenuation over the octave frequency band, while the post-filter response is flatter over this range. The large variation in the magnitude response at higher frequencies is a function of the IM3 measurements at small-signal levels. The IM3 distortion at these frequencies was near the noise floor of the spectrum analyzer used in the experiment and the collected data shows the expected variation in amplitude. As a consequence of this variation, the fit of the measured data at higher frequencies is degraded since the pre-filter response is raised to the power of the nonlinearity in each of the slices.

Without the high frequency variation, it can be seen that the majority of the frequency response of the system is due to the pre-filter. This is a reasonable result since the effect of the attenuation in the pre-filter is magnified by the nonlinearity so that the attenuation in the IM3 products is significantly higher, Fig. 6.3.

The nonlinear coefficients for the first and second slice were extracted at the 2 GHz frequency point as the amplifier had the greatest nonlinear response at this frequency. As the plots of the extracted linear networks shows, the frequency response is essentially monotonically decreasing from 2 GHz and thus the amplitude of the signals reaching the memoryless nonlinearity in the device are decreasing with frequency, thus limiting the nonlinear response. By extracting the polynomial coefficients at the frequency of minimum attenuation, a better estimate of the nonlinear response is computed which allows better fitting over the entire power and frequency range of the characterization.

The ability of the multi-slice model to track the measured fundamental output

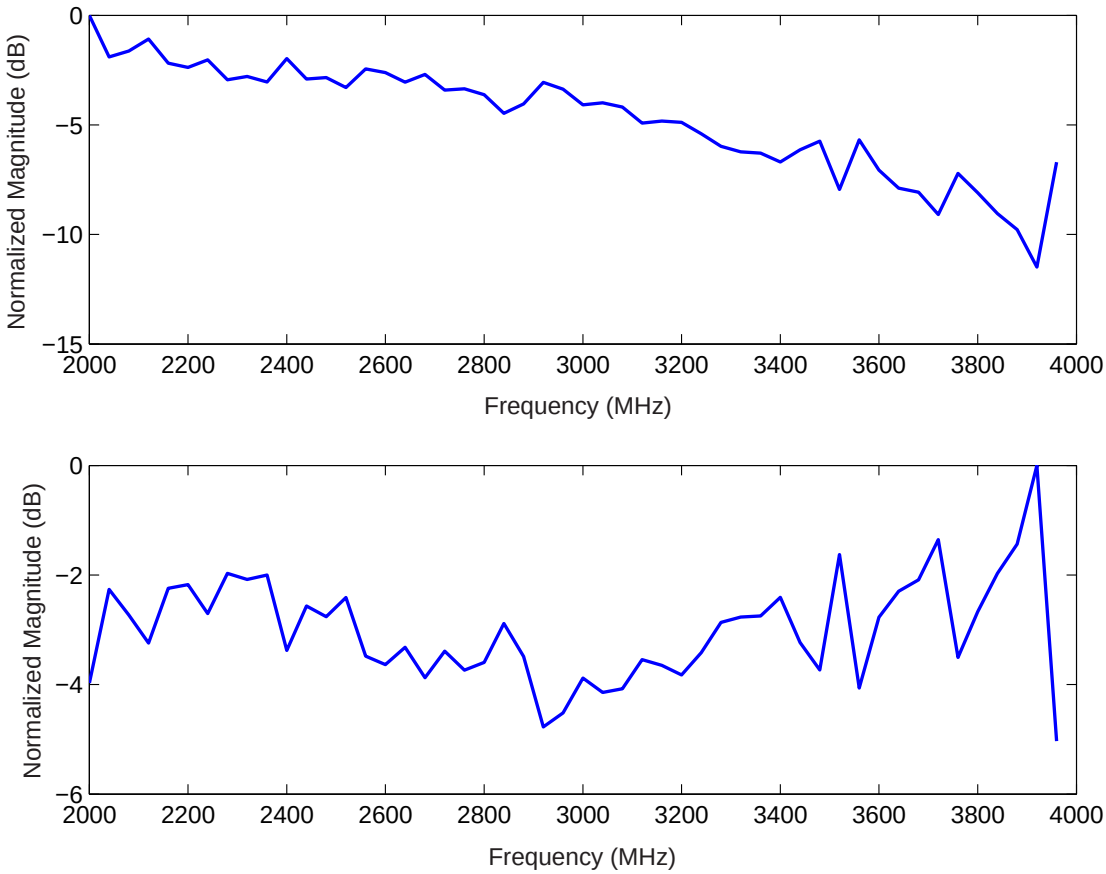


Figure 6.2: Extracted pre-filter and post-filter frequency responses.

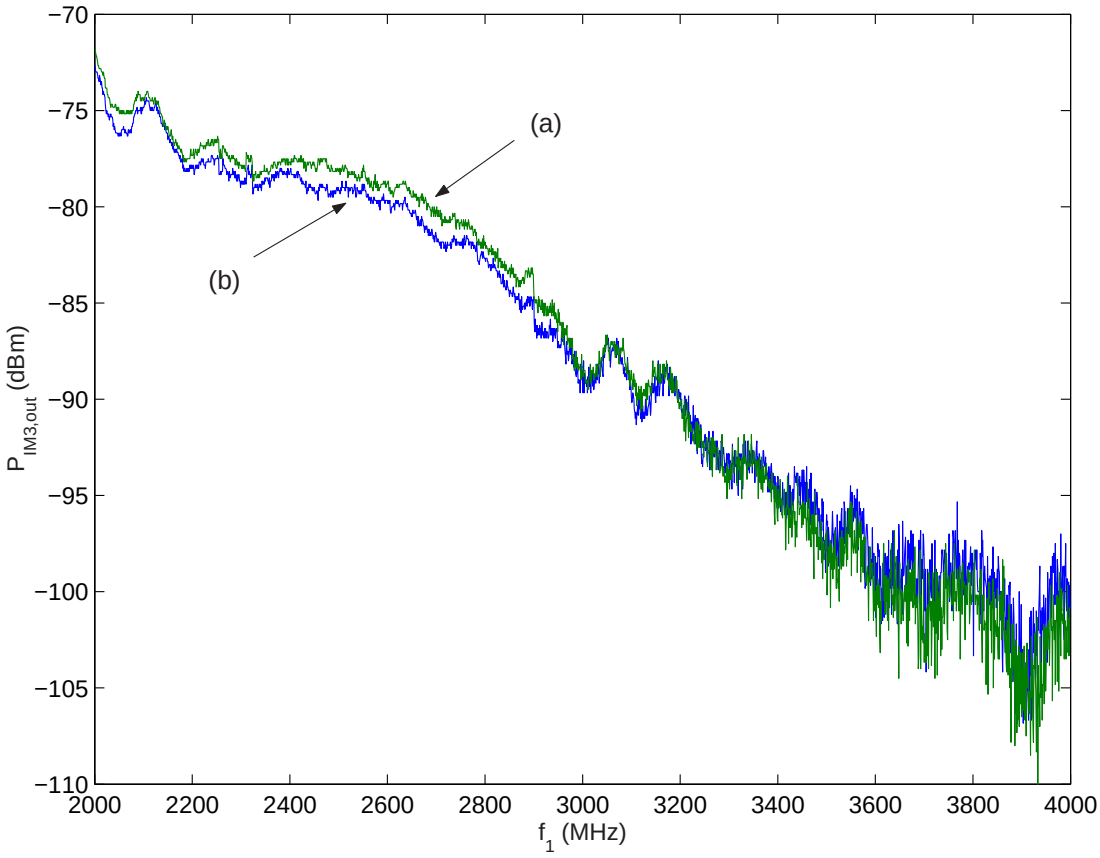


Figure 6.3: Measured output power of (a) IM3H and (b) IM3L at −15 dBm input tone power.

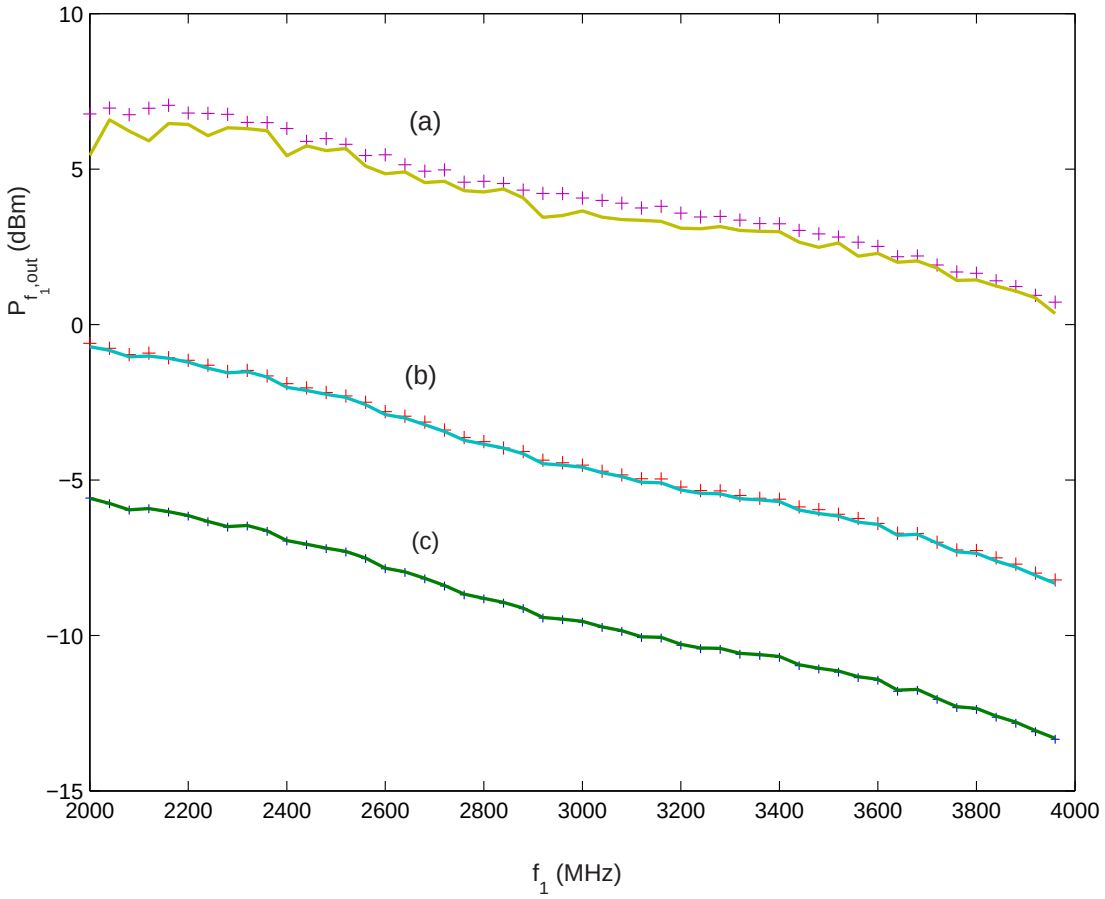


Figure 6.4: Measured (+) and modelled (line) fundamental output power at (a) −1 dBm, (b)−10 dBm, and (c) −15 dBm input tone power.

is shown in Fig. 6.4. Here the model output at three different input powers shows the excellent agreement with the measured data across the entire frequency span except at the highest input power where the polynomial fit to the measured data was less accurate. The model also predicts the IM3 asymmetry seen at higher input powers as shown in Figs. 6.5 and 6.6. Here the deviations between the model and the measured response reside mainly at the higher frequencies where measurement noise impacted the linear system extraction.

Plotting the IM3 measured and modelled data versus input power and input voltage shows that the model predicts the asymmetry well up to an input power of −6 dBm with large differences above this level (Figs. 6.7 and 6.8). The deviations result from the inability of the polynomial model in the first slice to accurately reproduce saturation effects in the IM3 products. This is a common difficulty with using polynomial modeling where

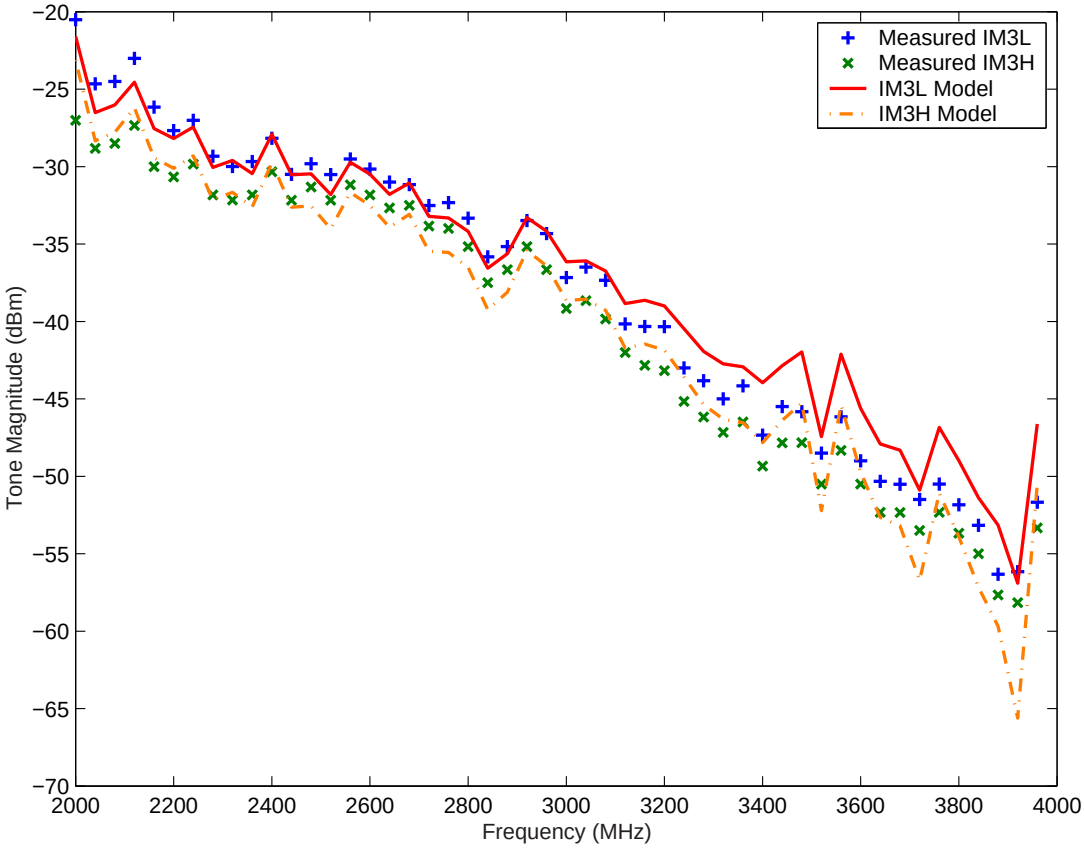


Figure 6.5: IM3 modelled and measured frequency response at −7 dBm.

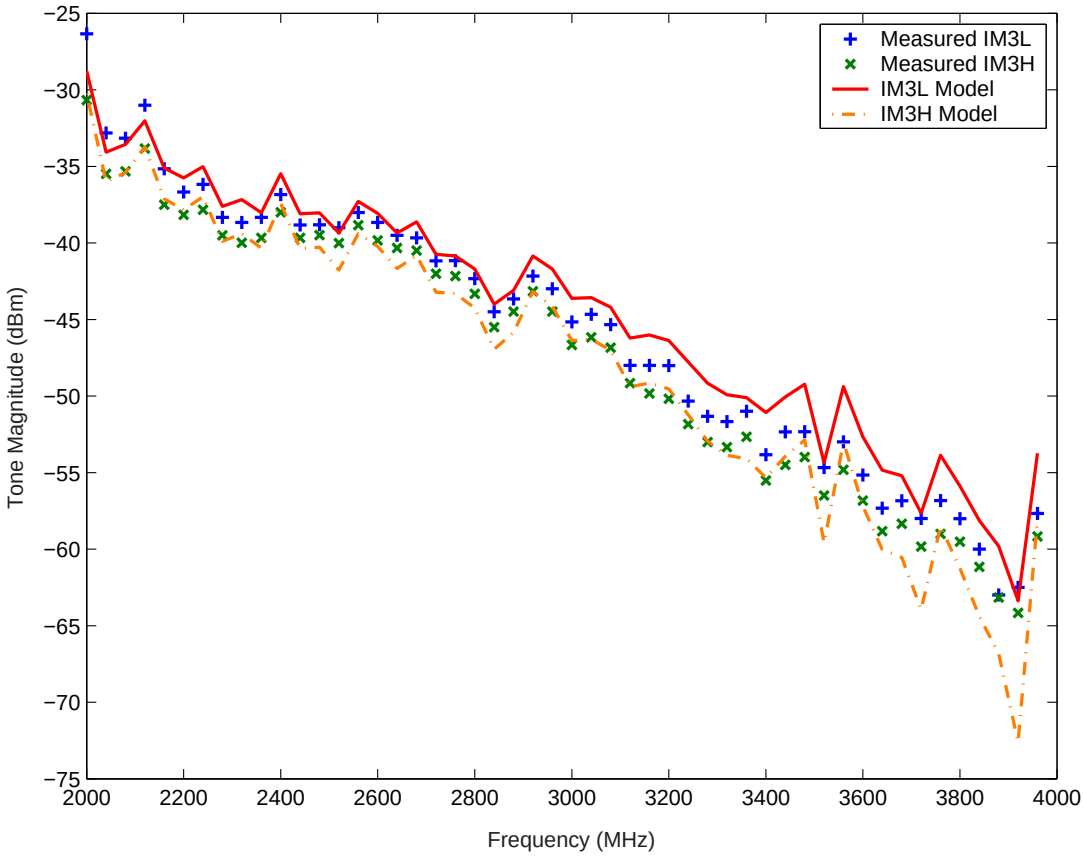


Figure 6.6: IM3 modelled and measured frequency response at −9 dBm.

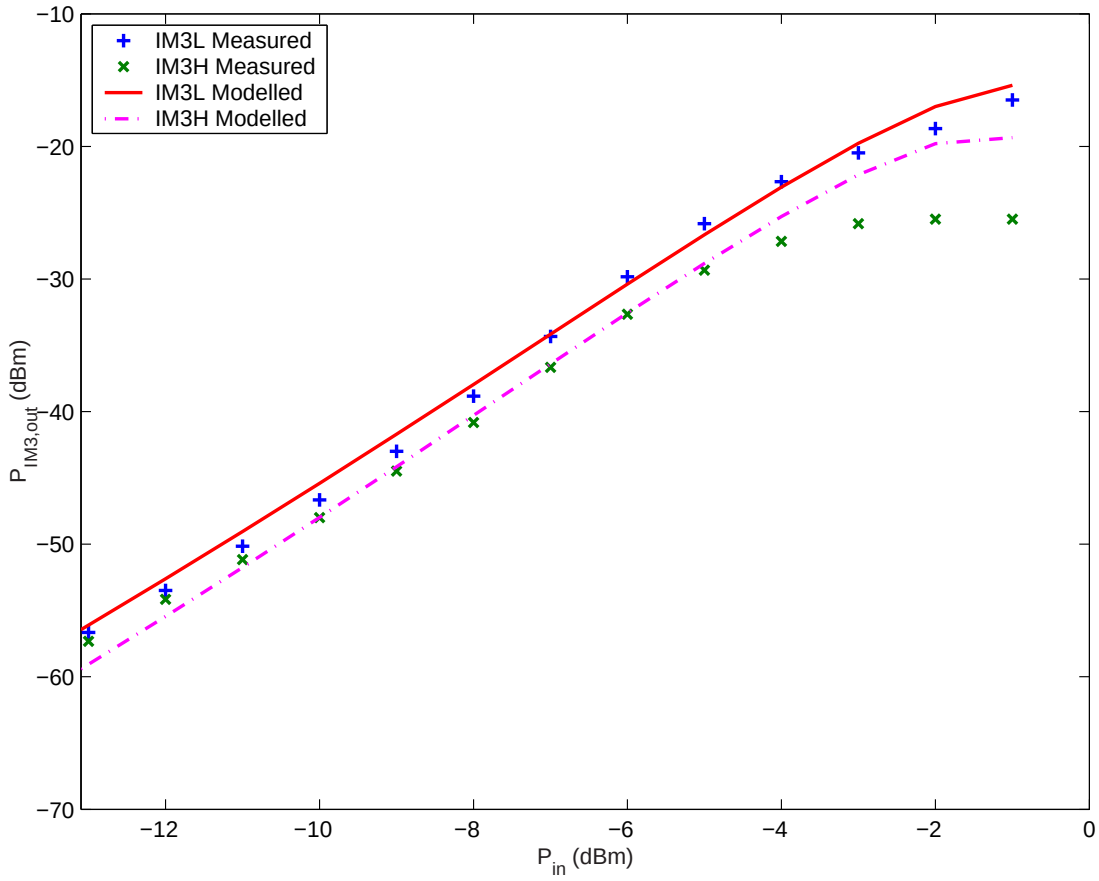


Figure 6.7: IM3 modelled and measured response at 3 GHz, P_{out} vs. P_{in}.

the single-tone fit fails to generalize well to complex multi-tone behavior.

6.4 Conclusion

The results of extracting the multi-slice model using only amplitude information were quite good over a range of stimulus power levels. This model did predict the fundamental output and IM3 asymmetries of the amplifier over a broad frequency range, demonstrating the ability to estimate both macro and micro-level memory effects. The assumptions used considering the phase of the baseband contribution to the IM3 products obviously did not fully capture this effect since the model did not track the asymmetry over

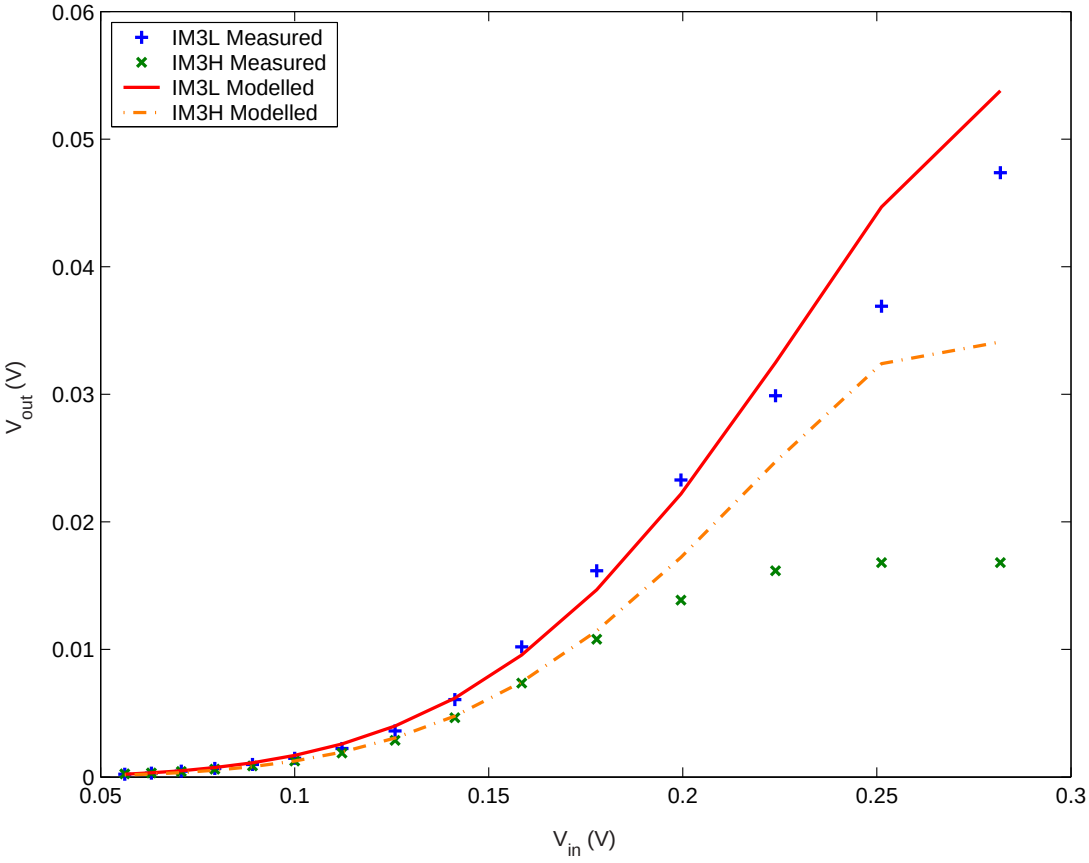


Figure 6.8: IM3 modelled and measured response at 3 GHz, V_{out} vs. V_{in}

the entire power range of the stimulus. However, the ability of the multi-slice architecture to reproduce in part the IM3 asymmetry phenomena that a traditional AM-AM, AM-PM model cannot predict validated the model concept and presented areas to focus on to refine the modeling and measurement procedures. Employing a measurement technique to determine the phase of the intermodulation products appeared as the most critical piece of the multi-slice model evolution.

Chapter 7

Multi-Slice Behavioral Models of Nonlinear Amplifiers with Memory Employing Vector Intermodulation Measurements

7.1 Introduction

Initial success with the magnitude-only version of the multi-slice model architecture in predicting IM3 asymmetries encouraged the development of the modeling technique to incorporate IMD phase information. With the implementation of the VIMA apparatus described in Section 3.4.2, data could now be taken to more accurately extract the parameters of the nonlinear blocks within the model. By incorporating both phase and magnitude information, the multi-slice model revealed previously unseen behavior related to the sources of intermodulation distortion in multiple amplifiers. The measurements and modeling of these amplifiers determined for the first time the effect on the magnitude and phase of the

IM products as a function of stimulus level due to baseband contribution effects.

7.2 Development of the Multi-Slice Model Incorporating IMD Phase Information

The architecture of the multi-slice model used to incorporate IMD phase information does not change from that of Fig. 6.1. Differences from the magnitude-base measurement model of Chapter 6 occur in the use of complex coefficients in the polynomials used to approximate the nonlinear blocks in each branch of the model as defined in Chapter 4. The use of complex coefficients allows the branches to create phase shifts in the output spectral products as a function of input power. This occurs for both the single-tone fundamental output and the IMD products during two-tone stimulus.

7.2.1 Model Parameter Extraction

In this subsection we describe the procedure for extracting the components of the two-slice model of Fig. 6.1 for the amplifiers studied here. The process for estimating the frequency response of the macro-level memory blocks $H(s)$ and $K(s)$ was shown in Section 6.2.1, and a similar series of steps can be used to determine the frequency response of $L(s)$ as a function of stimulus tone separation. The major task to be described here is extraction of the two sets of complex polynomial coefficients, a_i and k_i from (4.9) and (4.12).

The approach to the sequence of parameter estimation for the various slices is performed in order of the contribution to the nonlinear response. The odd-ordered polynomial in the first slice dominates this response so this slice is determined first. Extraction of the coefficients of the memoryless polynomial $f()$ follows the traditional AM-AM, AM-PM approach of fitting the single-tone response with the optimum fit in the least-squares sense. With this slice determined, the next step is to fit the difference between the first slice estimate of the IM3 products and the measured two-tone data with the even-order baseband polynomial. The difference is given by,

$$V_{\text{IM3,diff}} = V_{\text{IM3,meas}} - V_{\text{IM3,est}} \ . \tag{7.1}$$

There are actually two differences, one for each of the IM3 products. The value that is fitted

is the average difference between the measured IM3 products and the estimated first-slice component with one of the IM3 differences applied as a complex conjugate for the effect of the linear network $L(s)$. The effect of the ideal multiplication with the input stimulus is also accounted for by normalizing the difference in (7.1). Once the coefficients are generated, the baseband contribution to IM3 are computed with (4.11) and then summed with the first slice output to yield the total model response.

Error functions based on the difference between the estimate and the measured complex IM3 response are used to determine the polynomial order used in the fitting process. The order is adjusted to achieve the minimum error for each slice with respect to the IM3 tones even in the case of the first-slice which only uses single-tone information to compute the slice parameters.

7.3 Comparison of Power Amplifiers

Multi-slice behavioral models were developed from single-tone and IM3 vector measurements using the extraction procedure for three different power amplifiers. The amplifiers consisted of a 5 W multi-stage GaAs/Silicon MESFET amplifier (Mini-circuits ZHL-5W-1), a high-linearity 10 W instrumentation MOSFET amplifier (Amplifier Research 1000W10), and a SiGe HBT Darlington driver amplifier mounted on an evaluation board (Hittite HMC479ST89). The data collection process consisted of using the relative phase versus input power VIMA in Fig. 3.9 to measure the single-tone AM-AM, AM-PM response and the amplitude and phase of the IM3 products during two-tone testing. For this work the response of the amplifiers were measured at $f_1 = 450$ MHz and a frequency separation of 10 kHz. The input power levels were swept from the small-signal region of each amplifier to the saturation point, or the maximum level recommended by the manufacturer. The power at which the phase reference point was chosen depended on the amplifier under test and corresponded to the small-signal input region of the stimulus.

7.3.1 MESFET Amplifier

Measurements from a two-tone test of the GaAs MESFET amplifier are shown in Fig. 7.1, where the amplitude and phase measurement of the IM3 products are plotted.

The amplifier does not exhibit amplitude asymmetry of the upper and lower IM3 but there is phase asymmetry. The asymmetry becomes apparent as the input power level increases above the small-signal regime where gain compression begins to occur. The first step in the model extraction procedure was to determine the odd-ordered polynomial coefficients from the AM-AM, AM-PM data. The result of this step is shown in Fig. 7.2 for a complex polynomial of order 15. As expected, the resulting fit cannot track the asymmetry in the IM3 phase, however, the IM3 magnitude fit also fails to track the measured response. The error in the single-slice magnitude fit occurs at the same input power level as appearance of the asymmetric behavior in the phase of the upper and lower IM3 products. This is strong evidence that the upconversion of baseband effects from an even-order nonlinearity are becoming important. The fit of the odd-ordered model reinforces this hypothesis since the phase component of the fit roughly splits the difference in phase between the two IM3 products. This corresponds to the case in Fig. 4.2 where the baseband component has the same phase reference as the strictly odd-ordered components.

With the first slice of the multi-slice model extracted, the next step in the extraction procedure was to determine the even-order coefficients, (eighth-order in this case), for the nonlinearity that produces the baseband terms. The result of this step and the summation in a phasor sense of the second slice with the first slice is shown in Fig. 7.3. The resulting fit of the two-slice model was < 0.5 dB for the magnitude of the IM3 products and the phase estimation was $< 3^\circ$ for the majority of the input power sweep except for the small-signal region. In this region there exists a slight slope to the phase as a function of input power. This cannot be predicted by a fixed complex coefficient in the polynomial representation of the nonlinearities in the system since for this level of stimulus, the third-order coefficient is dominant, and thus the estimated phase will be constant. This may indicate the existence of another nonlinear contributor that loses importance as the input signal rises above the small-signal levels.

Given the phase information of the IM3 data, it is possible to plot the measured and model output on a polar plot, shown in Fig. 7.4, where the amplitude of the data is in dB.

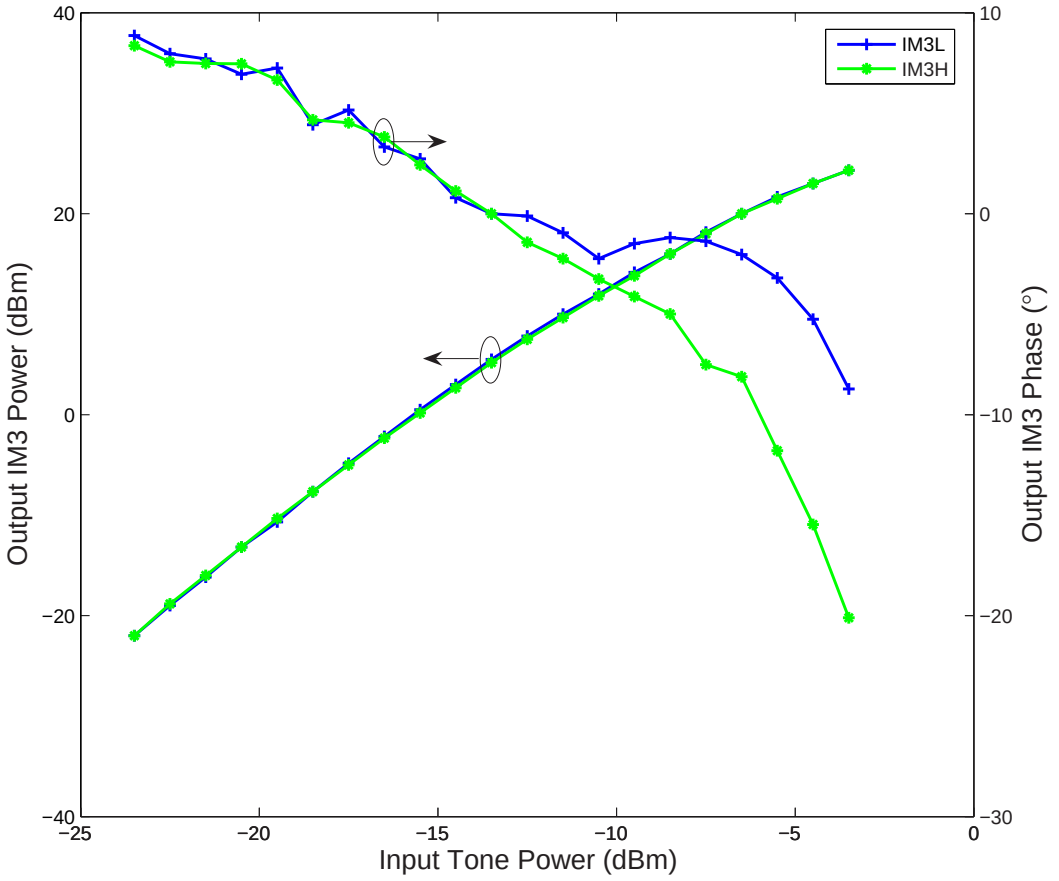


Figure 7.1: Measured magnitude and phase of the IM3 products for the Mini-Circuits 5 W amplifier.

7.3.2 MOSFET Instrumentation Amplifier

The instrumentation MOSFET amplifier was expected to have a very weak nonlinear response since its purpose is to provide laboratory-grade amplification without distortion. The single-tone data, Fig. 7.5, does not indicate any onset of compression for an input power of up to −3 dBm, although the phase of the output tone does begin to exhibit some AM-PM distortion as the input power is increased. Further evidence of the weak nonlinear response is displayed by the two-tone IM3 measurements (Fig. 7.6). Here the IM3 tones are strictly 3:1 in slope with the asymmetry consisting of a systematic type error. For this data, the IM3 phase data proves more informative about the source of the nonlinearities

contributing to the IM3 products. The phase response of the upper and lower IM3 products clearly has a complex conjugate relationship over the entire power range suggesting that the baseband upconversion term generates the entire IM3 response.

This hypothesis was reinforced by the relatively low order of the odd-ordered nonlinearity required to minimize the error functions presented in Section 7.2.1, that of $N_{s1} = 5$ and $N_{s2} = 8$, for the first and second slices respectively. Applying the fitting procedure for the two-slice model yielded a very good fit to the IM3 data, Fig. 7.7. Here the error in the IM3 amplitude was < 0.7 dB and the maximum error for the IM3 phase was 2.17° with an average phase error of 0.57° for IM3L and 0.59° for IM3H. This phase error lies below the uncertainty of the phase measurement itself, thus the error in the model is limited by the measurement procedure itself.

7.3.3 Darlington HBT Amplifier

The Darlington HBT amplifier investigated exhibited the most complex nonlinear response of all of the amplifiers. While the AM-AM, AM-PM measurements showed a typical response of constant phase insertion for small-signal input levels with a change in phase as the gain became compressed and neared the saturation power level, Fig. 7.8, the response of both the magnitude and phase of the IM3 products deviated considerably from the response of the other amplifiers investigated, Fig. 7.9. As shown, the IM3 magnitude response to an increasing input power transitions at a low level from a 3:1 slope to a 5:1 slope. This is not unexpected, however over the input power range where this slope remains constant (~ 10 dB), the IM3 phase varies by $\sim 30°$. This behavior cannot be explained by a complex polynomial representation of the polynomial since for a dominant order of the polynomial, the phase should approach a constant value.

The behavior of the IM3 phase of this amplifier led to difficulties in applying the fitting procedure of the multi-slice model. Due to the inability of the complex polynomial to track the considerable IM3 phase transitions ($\sim 40°$ over both the small-signal and large-signal regions), a reasonable fit to the data could not be achieved by the first slice and thus the second slice could not be extracted properly. In addition to the analytical approach to the polynomial coefficient solution using a Vandermonde approach, three additional numerical methods were employed to determine if an alternative technique could achieve a better fit.

The first technique involved using information from the two-tone measured data to improve the performance of the single-tone extracted slice to track both data sets. Since this amplifier experienced only fifth-order and higher effects as the amplifier approached saturation, the single-tone data could not properly extract the third-order component of the nonlinear response. Therefore, to improve the single-tone fit, the two-tone measured data was used to supply the third- and fifth-order coefficients. These two components of the single-tone response were then subtracted from the measured data and the traditional Vandermonde approach to the polynomial fit problem was applied.

The difficulty in applying this approach was that the phase reference for the a_3 and a_5 coefficients was unknown, but could not be arbitrary with respect to the measured single-tone phase information. Published works using polynomial fitting assumes a 180° phase shift between adjacent odd-order coefficients, however this is not technically correct since a strict change in sign of the coefficients will result in a sharp dip in the amplitude of the sum of these components at their intercept point. When using complex coefficients with independent phases, in general the relationship between adjacent coefficients is close to 180° of separation, but the phase difference can be much less when a smooth transition occurs in the change of phase of the IMD products. In order to assign the phase of these two coefficients, it was necessary to take into account the effect their phase would have on the higher order components of the polynomial response. The result of using this approach to improve the IM3 fit is shown in Fig. 7.10. While this approach produced a better fit to the IM3 magnitude over the entire input power range except for the highest levels, the estimated phase response only tracked the shape of the measured response but with a severe shift in the actual value of the response.

The next technique used was a Genetic Algorithm (GA) approach. The error function that guides the evolution in the GA process was composed of both the single-tone and the two-tone measured data sets. This was done to allow the GA to have complete information about the problem space over which the solution was sought. Optimizing this approach for the complex polynomial coefficients and generating multiple configurations for the initial population set resulted in an improvement in the fit of the phase of the two-tone data, but a rather poor fit of the IM3 magnitude with an error > 20 dB at the upper range of the power sweep, Fig. 7.10.

The final technique used was a nonlinear optimization process using the Levenberg-Marquardt search method. Again both measured data sets were supplied to the routine

to search for the appropriate solution over the problem space. Since this method uses the error function to guide the estimation of the next iteration's solution in contrast to the GA's random wander approach, this method converged much faster on the perceived solution. This approach did produce the best result in terms of minimum squared error of the IM3 fit, however there was still considerable error in the IM3 phase estimation and overestimation of the IM3 magnitude by as much as 6 dB, (Fig. 7.10).

After applying these numerical methods to the problem of extracting the polynomial coefficients of the memoryless nonlinearity assumed in the multi-slice model, the conclusion from the rather poor fit suggests that the unilateral polynomial model is inadequate in describing the observed behavior. Knowledge of the circuit implementation in this amplifier points to the need to incorporate a feedback term in the model. This remains an open area for investigation in order to improve the model of this particular amplifier, and is likely necessary for all Darlington-based amplifiers.

7.4 Summary of Results

Despite the marked difference in the IM3 response of the Darlington amplifier, all of the amplifiers shared a common attribute. The shared behavior involved the variation of the IM3 phase at small-signal levels which contradicts the constant phase assumption intuitively expected from single-tone measurements. For the nonlinear products though, this response likely results from small-signal nonlinear effects with a dependence on the power of the stimulus, in particular the effect of changing bias voltage on nonlinear parasitic capacitors in the transistors. This implies that these nonlinear elements dominate the IMD phase response at small-signal. In contrast, the linear term dominates the fundamental small-signal magnitude and phase response in single-tone measurements. Thus, these small-signal effects are not significant and cannot be detected at low input power levels resulting in constant phase as a function of input power. Therefore, in order to determine the effect of all of the nonlinear contributors in the system it is necessary to measure and characterize the IMD phase response since their effects cannot be observed in AM-AM, AM-PM tests or by IMD magnitude measurement alone. This has significance since the performance of pre-distortion power amplifier linearizers and estimation of nonlinear distortion depend strongly on the phase of the distortion.

Another aspect of this work that differs from other published IMD phase measurement approaches is the assumption about the source of the phase shift as a function of input power or in other words, the source of the memory effects in the system. Here we have shown the memory effects for a given frequency separation as a function of input power. Given the measured phase shift responses, only elements within the system with a dependence on the input stimulus level can generate the observed behavior. Other works in this area have focused on the reactive components of the bias network of an amplifier for the contribution of memory effects manifesting IMD phase and or magnitude asymmetries [6, 8, 9].

These effects result from sweeping the frequency separation of the stimulus tones which was not done in this work. The effect of sweeping the stimulus tones results in changes in the impedance of the bias network at the even-order baseband signals generated by the nonlinear device. The changing impedance results in higher voltages of the baseband signals that then interact with the fundamental signals in the nonlinearity to create odd-ordered intermodulation products. The work in this area has shown the correlation between the impedance of the bias network and the asymmetries in IM magnitude. However, none of the published works have presented a model or extraction procedure for this effect. While bias network effects may contribute to the memory effects in the amplifiers investigated in this chapter, they are not the source of the behavior reported here.

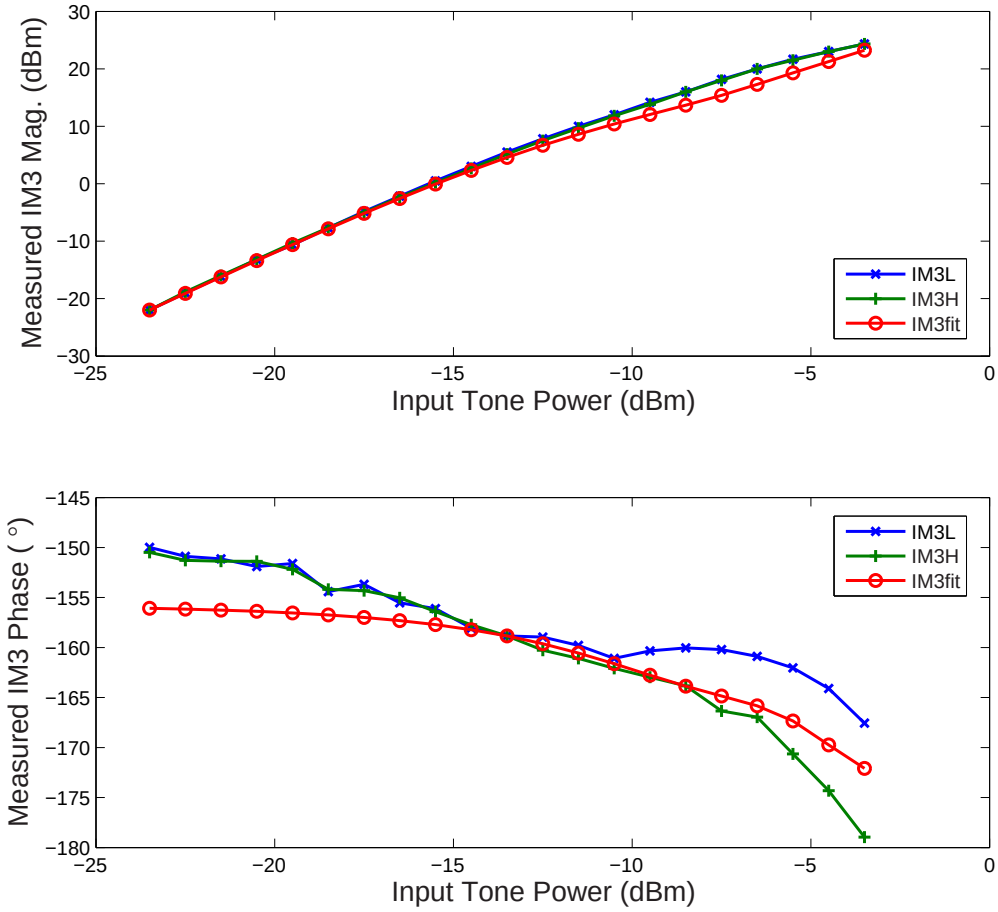


Figure 7.2: Measured and estimated magnitude and phase of Mini-Circuits amplifier IM3 products using a single-slice model.

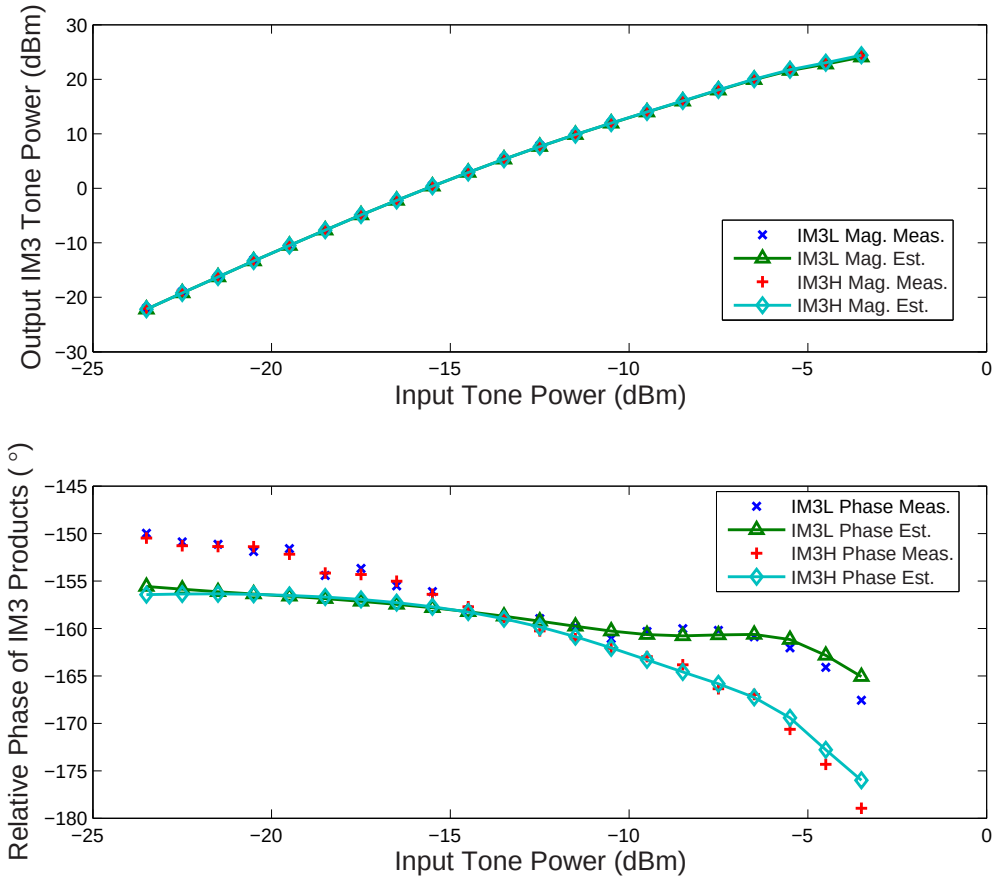


Figure 7.3: Measured and estimated magnitude and phase of Mini-Circuits amplifier IM3 products using a two-slice model.

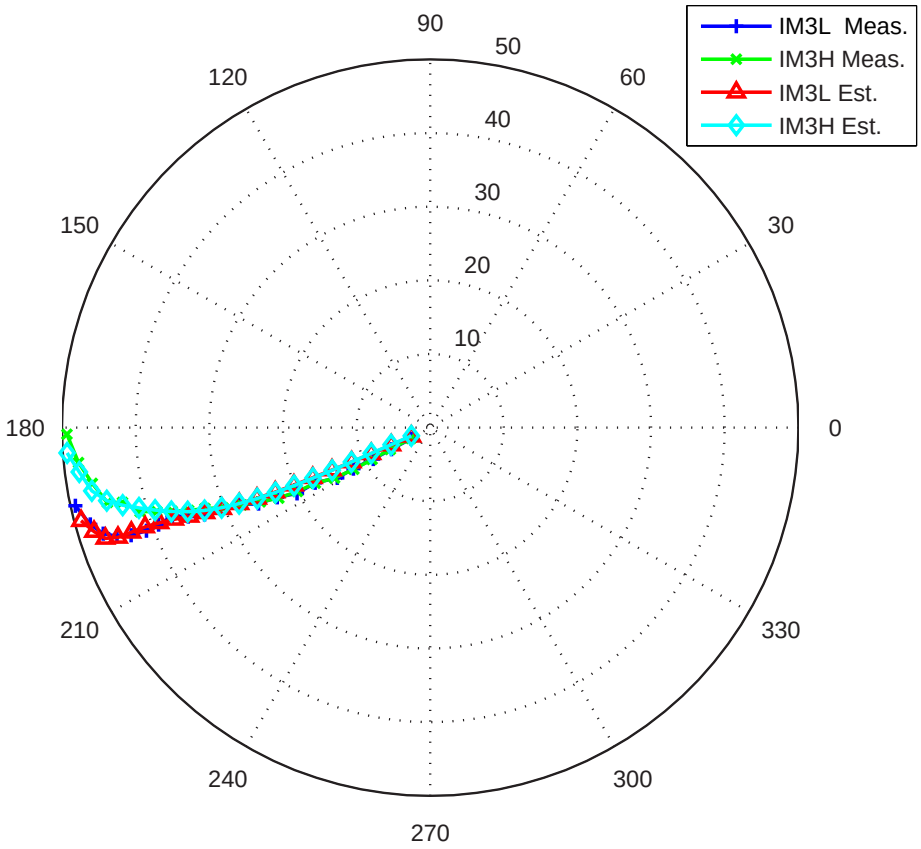


Figure 7.4: Polar form of the fit to the Mini-Circuits amplifier IM3 data with the magnitude in dB, (magnitudes offset such that they are non-negative).

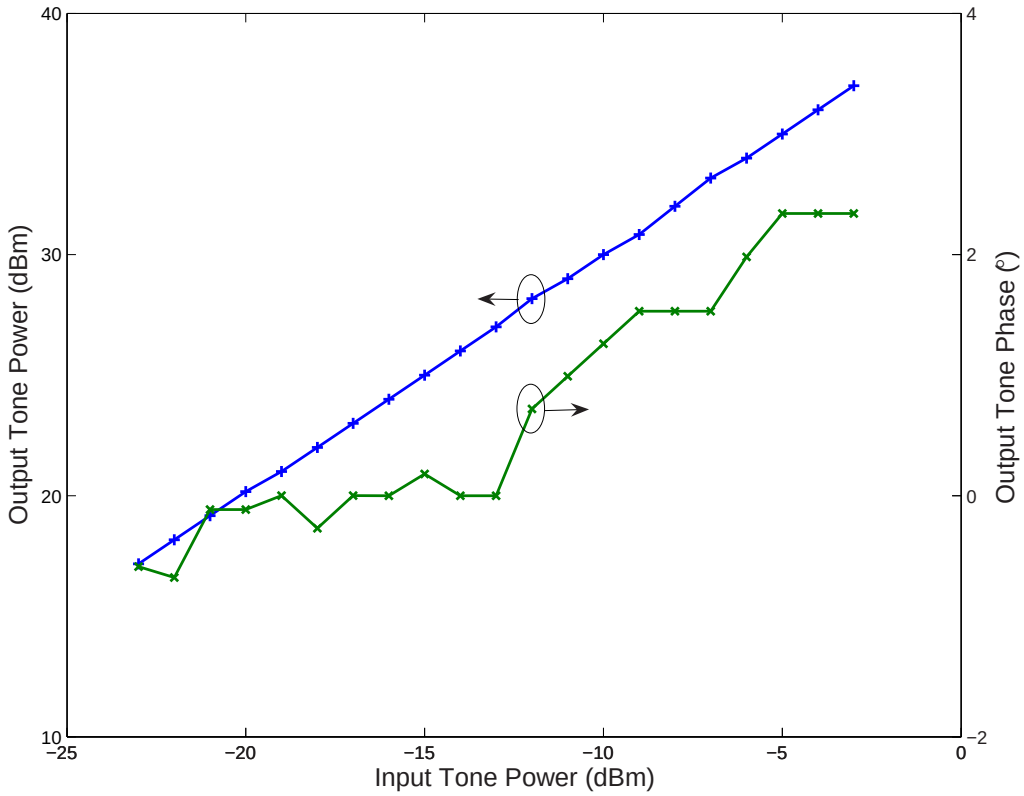


Figure 7.5: Measured AM-AM, AM-PM of the 10 W AR amplifier.

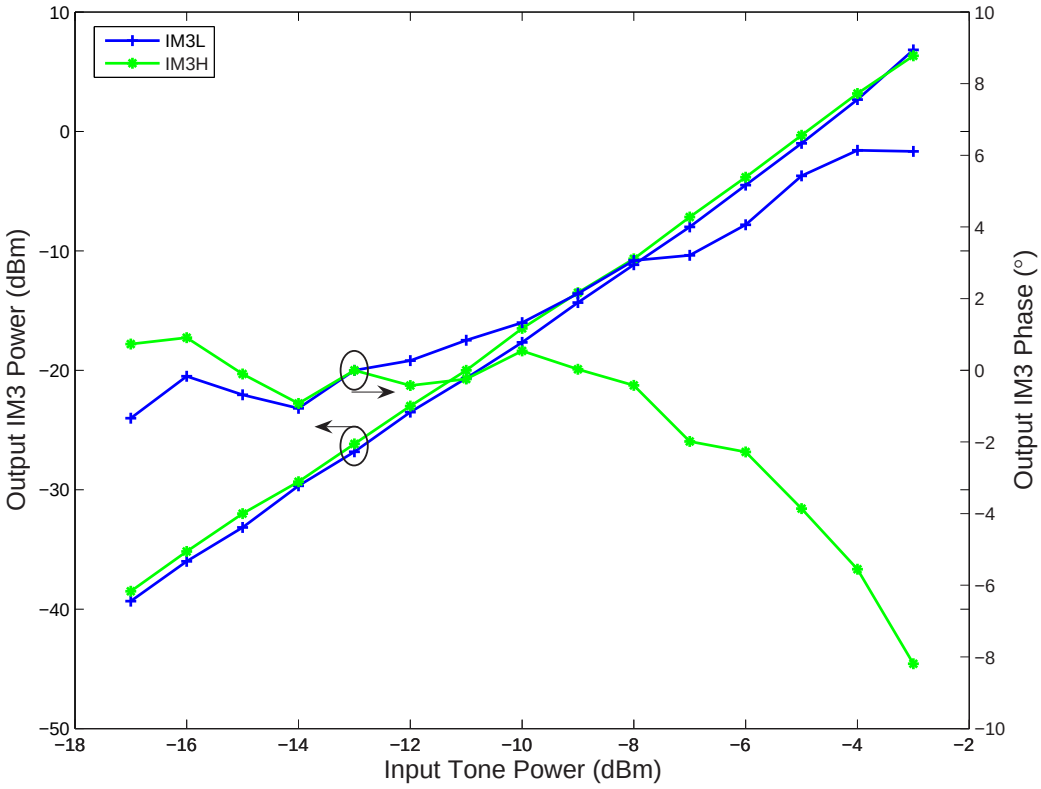


Figure 7.6: Measured magnitude and phase of the IM3 products for the AR amplifier.

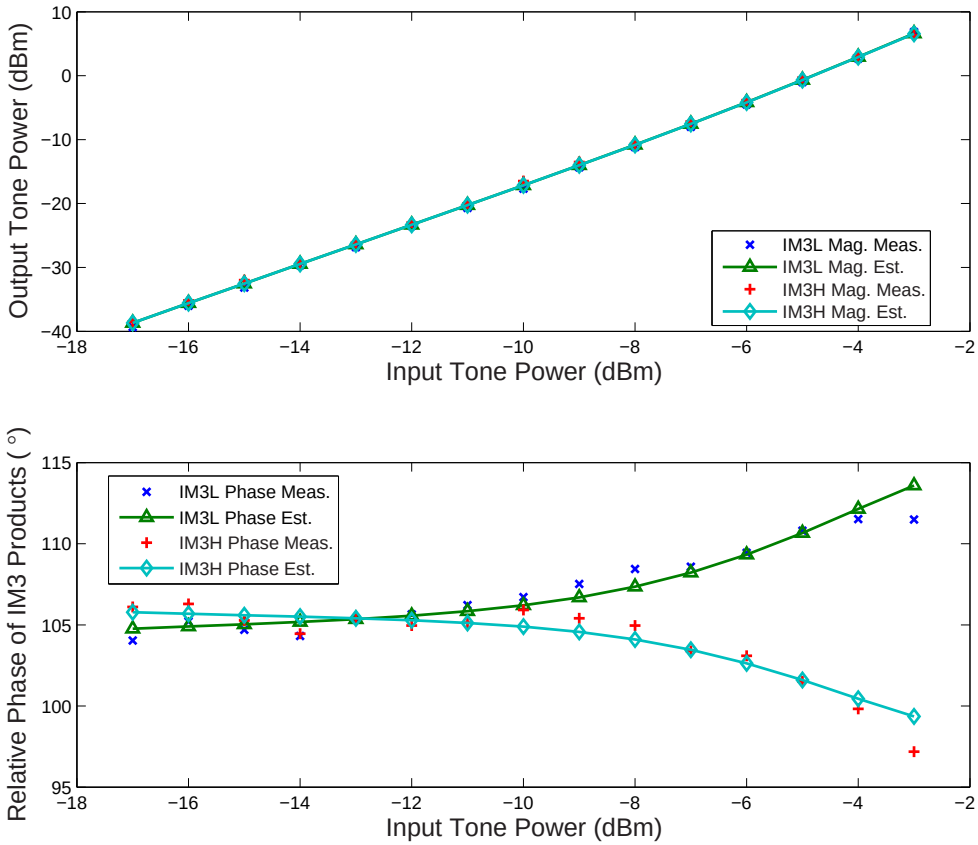


Figure 7.7: Measured and estimated magnitude and phase of AR amplifier IM3 products using a two-slice model.

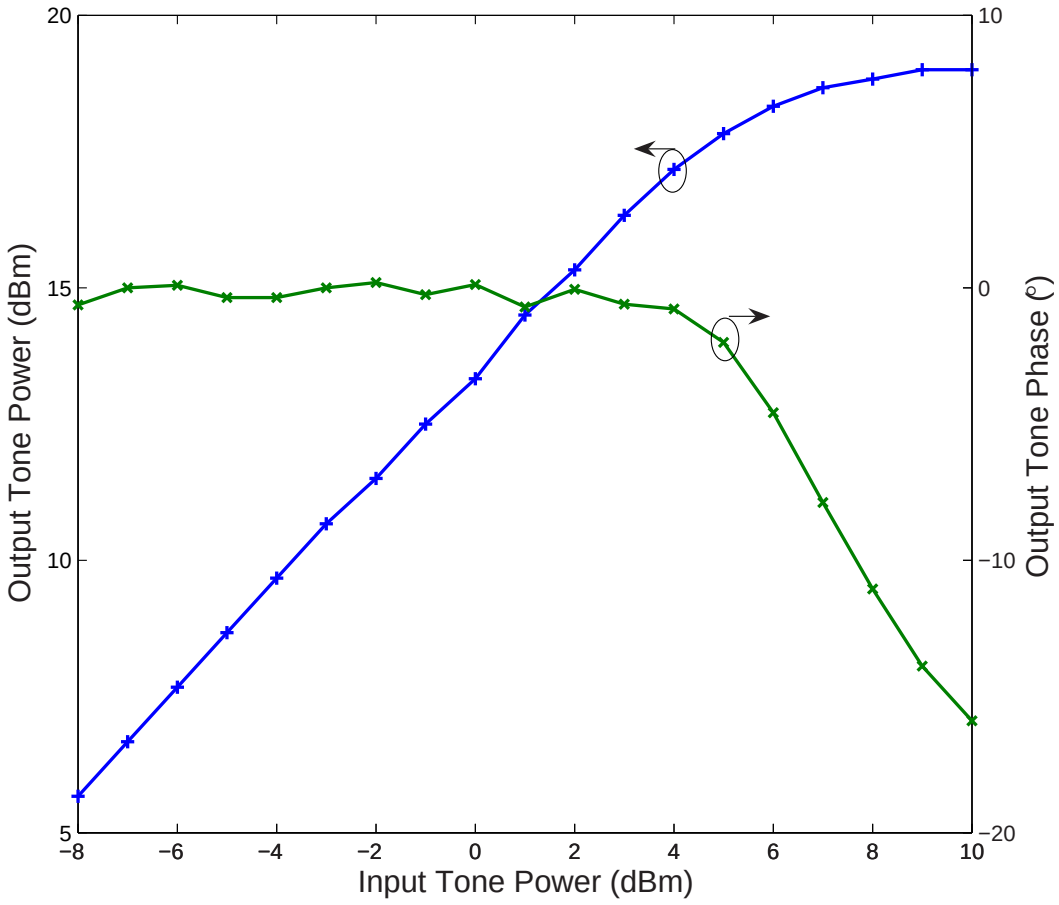


Figure 7.8: Measured AM-AM, AM-PM of the Hittite Darlington amplifier.

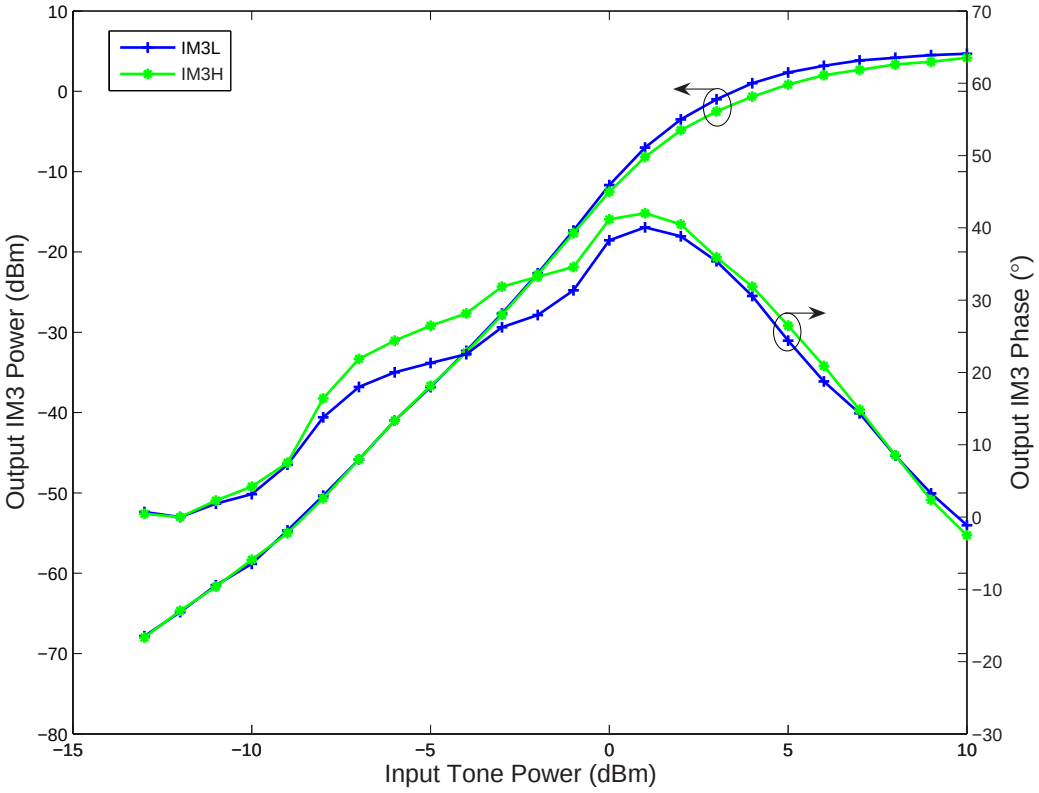


Figure 7.9: Measured magnitude and phase of the IM3 products for the Hittite amplifier.

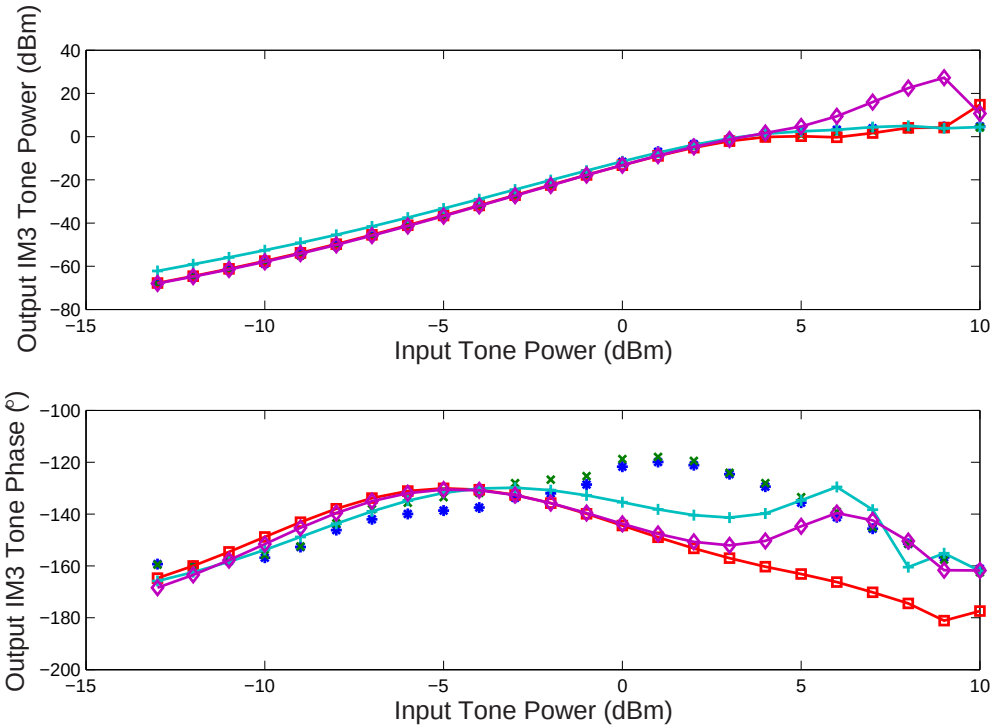


Figure 7.10: Measured and estimated magnitude and phase of Hittite amplifier IM3 products using various fitting techniques to extract the one-slice model, ($*$ - IM3L measured, $\times$ - IM3H measured, $\square$ - two-tone fit to a_3 and a_5, $+$ - nonlinear optimization fit to both data sets, $\diamond$ - GA fit to both data sets.)

Chapter 8

In-Circuit Filter Tuning Using a Nonlinear Distortion Response from a One-Port Measurement

8.1 Introduction

The cost of RF and mixed signal test in both time and money are an increasing component of the manufacturing cost of communications devices as discussed in Section 1.4. With increasing levels of integration and the corresponding increase in complexity this cost will also increase. Additional hurdles will arise due to higher levels of integration that will limit visibility into components within the design. In the digital world, the use of BIST and scan chains can alleviate this problem allowing functional testing of critical paths within a block of logic. The impact on system performance from this type of circuitry limits this solution for analog and microwave circuits. Utilizing the VSA-based VIMA instrument of Section 3.4.3, the metrology technique presented in this chapter sought to provide increased understanding of nonlinear operation as well as visibility into individual components within an integrated chain of components representing a simplified RF frontend.

8.2 Overview of Filter Tuning in RF Systems

Since all microwave communications products are bandpass systems, the architecture of the front-end circuitry will always contain at least one filter. These filters all require tuning during manufacturing in order to meet design specifications. Variations in manufacturing can be attributed to materials and packaging effects. Base station filters that use dielectric resonator cavity filters require significant tuning during manufacturing due to dielectric permittivity variation. The dielectrics used in the resonator elements have typical tolerances on the relative permittivity of ± 1 from batch to batch and ± 0.5 for the same batch. For $\epsilon_r = 29$, a variation of ± 1 in the permittivity translates to roughly ± 30 MHz of shift in the resonant frequency [123]. Since many modern communications standards have transmit and receive bandwidths similar in magnitude to this shift, it is clear that this variation has a tremendous impact that must be tuned out during final testing. In physically smaller filters the effects of packaging requires tuning during manufacturing. This is the case for passive structures implemented in a flip-chip form factor where the reactive loading effects of the package must be compensated by tuning.

The tuning of these filters takes different forms based on the filter implementation. Filters for base station applications are physically large and not produced in massive volumes so these tend to be tuned manually by a trained technician. Mass-produced filters that do not contain human-accessible tuning elements can be tuned using laser trimming processes that are common within the electronics industry [132]. Manual techniques require highly-skilled technologists and a tuning time from 15 minutes to 1 hour per filter [123]. The existing automated tuning techniques are based primarily on the 1950's era Dishal technique that requires shorting individual resonators and tuning each section of the filter in sequence [133]. This technique is not practical for embedded filters as it becomes difficult to short individual resonators in an integrated filter due to limited access to the filter elements.

The field of automated filter tuning techniques is outside the scope of this work, but this topic is growing in importance as wireless devices become more prevalent and it is necessary to build large volumes of highly selective filters. These methods use numerical optimization techniques (fuzzy logic or adaptive network models) to determine the elements that must be tuned [120] and [121]. For these techniques, two-port S-parameter measurements drive the tuning algorithms along with comparison to an approximate network model of the filters under test. In particular, the use phase of the S_{21} was shown to increase the

convergence of the adaptation routines [121]. Achieving equivalent two-port data from one-port measurements for an integrated filter to drive filter tuning algorithms such as these was the goal of the research presented here.

8.3 Overview of Filter Tuning Using an Intermodulation Distortion Response

The ability to extract two-port S-parameter data from an integrated filter using only single-port measurements has the ability to shorten tuning and therefore test time of future wireless communications products. The only existing tuning option for an integrated filter would be through S_{11} techniques, however these suffer from an inability to capture important transmission characteristics such as transmission zeros and transmission phase. The location of the transmission zero in the frequency domain is of critical importance to filter designers as an indicator of the tuning direction and stopband performance of the filter that cannot be determined from S_{11} measurements. A representative two-port response where the transmission zero occurs above the passband is shown in Fig. 8.1. In-circuit tuning of integrated components would also allow deviation from the typical constraint of matching all components to a 50 Ω impedance. Tuning to achieve optimal matching at more convenient impedances could potentially save considerable time and design effort and improve performance.

The measurement technique shown here utilizes the ability of the VSA-based VIMA instrument described in Section 3.4.3 to determine both the magnitude and phase of intermodulation products. For this application, the VIMA measures the reflected response of a filter loaded with a representative nonlinear device to simulate an integrated RF front-end.

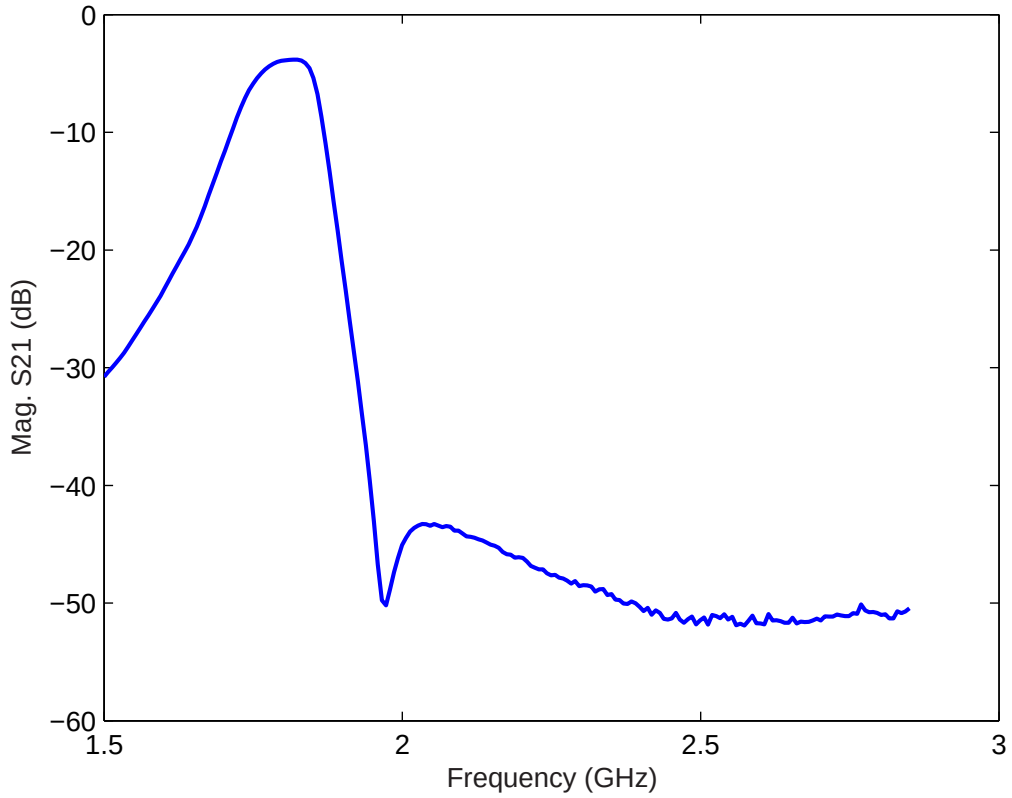


Figure 8.1: Typical two-port frequency response of a bandpass filter with a high-side transmission zero.

8.4 Filter Transmission Extraction via Intermodulation Distortion

The phenomenological study of the wireless characterization technique and the VIMA development forms the basis for extracting two-port transmission parameters from a one port measurement of reflected nonlinear distortion. As shown in Chapter 5 a nonlinearly loaded filter will transform the nonlinear reflection in a straightforward manner that allows extraction of the linear response of the filter. Only the nonlinear response of a stimulated circuit of this type can convey the transmission response of the filter since the reflected fundamental in a VNA-type measurement will contain both the filter and the linear reflection of the load response which cannot be separated. Extraction of the two-port

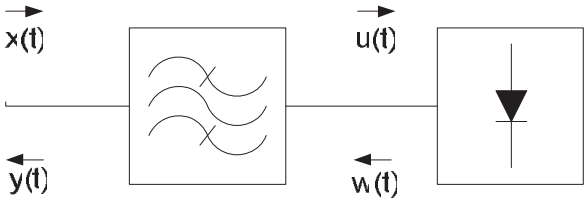


Figure 8.2: Model of nonlinearly loaded filter.

data begins by assuming the circuit under test can be approximated by the model in Fig. 8.2. The theoretical basis for the extraction follows that of the first slice of the multi-slice model with a truncated expression for the nonlinear load. The nonlinear representation can be truncated since the linear nature of the filter allows the extraction measurements at any convenient small-signal level. Ideally, considering only a third-order nonlinearity, the response to the circuit under two-tone stimulus can be described by the following signals throughout the chain:

$$x(t) = A(\cos(\omega_1 t + \phi_1) + \cos(\omega_2 t + \phi)) \ , \tag{8.1}$$

in phasor form around a carrier midway between the stimulus tones this becomes

$$X(\omega) = A \left(e^{j(\omega_\Delta t + \phi_1)} + e^{j(-\omega_\Delta t + \phi_2)} \right) \ , \tag{8.2}$$

where ω_Δ is the frequency separation between the carrier and the desired stimulus frequencies. This frequency separation is considered small such that phase and magnitude response of the filter are the same at the stimulus and intermodulation frequencies. Passing through the filter in the forward direction this becomes

$$W(\omega) = A \mid H(\omega) \mid \left(e^{j(\omega_\Delta t + \phi_1 + \phi_H)} + e^{j(-\omega_\Delta t + \phi_2 + \phi_H)} \right) \ , \tag{8.3}$$

where we are considering that the stimulus frequencies that pass through here are only the positive ones, thus the phase shift of the filter $H(\omega)$ is the same sign for both. Now after undergoing the third-order nonlinearity with coefficient a_3 representing the reflected nonlinear coefficient, the IM3 response is given as,

$$U(\omega) = \frac{3}{4} \, a_3 \, A^3 \mid H(\omega) \mid^3 \left(e^{j(-2\omega_\Delta t + 2\phi_2 + 2\phi_H - \omega_\Delta t - \phi_1 - \phi_H)} \right) \ . \tag{8.4}$$

Here the frequency of the measured IM product, IM3L, is $\omega_{\text{IM3L}} = 2\omega_2 - \omega_1$. The coefficient a_3 is the third-order coefficient, but its actual value is unimportant except for making absolute magnitude measurements to extract the loss of the filter. Combining terms in (8.4) yields,

$$U(\omega) = \frac{3}{4}\, a_3\, A^3 \mid H(\omega) \mid^3 \left(e^{j(-3\omega_\Delta t + 2\phi_2 - \phi_1 + \phi_H)}\right) . \tag{8.5}$$

Now the phasor form of the IM3L reflected response is

$$Y(\omega) = \frac{3}{4}\, a_3\, A^3 \mid H(\omega) \mid^4 \left(e^{j(-3\omega_\Delta t + 2\phi_2 - \phi_1 + 2\phi_H)}\right) , \tag{8.6}$$

and in the time domain

$$y_{\text{IM3L}}(t) = \frac{3}{4}\, a_3\, A^3 \mid H(\omega) \mid^4 \cos((\omega_c - 3\omega_\Delta) + 2\phi_2 - \phi_1 + 2\phi_H) . \tag{8.7}$$

The arrangement of the measurement apparatus changes slightly from that shown in Section 3.4.3, here the reflection measurement is arranged as in Fig. 8.3. The measurement performed by the VIMA provides the magnitude of the IM3L response, $\mid Y(\omega) \mid$, and the phase shift required to provide cancellation with Source 3 from an initial phase of $2\phi_2 - \phi_1 + \pi$. This shift in phase required to achieve maximum cancellation is taken as the phase shift of the interaction between the linear network and the nonlinear load. The output of Source 3 can then be written as

$$y_{S3}(t) = B\cos((\omega_c - 3\omega_\Delta) + 2\phi_2 - \phi_1 + \phi_{\text{shift}}) . \tag{8.8}$$

In general terms the reflected DUT response can be written as

$$y_{\text{IM3L}}(t) = \frac{3}{4}\, a_3\, A^3 \mid H(\omega) \mid^m \cos((\omega_c - 3\omega_\Delta) + 2\phi_2 - \phi_1 + p\, \phi_H) , \tag{8.9}$$

where m is the exponent of the filter magnitude response and p is the multiple of the phase response. Ideally the following would be true, $B = \frac{3}{4}\, a_3\, A^3 \mid H(\omega) \mid^m$ and $\phi_{\text{shift}} = p\, \phi_H + \pi$ with $m = 4$ and $p = 2$. In practice, the agreement between the predicted extraction coefficients m and p depends on the nonlinear region over which the load is stimulated during the measurements. It should be noted though, that deviations in the extraction coefficients from the theoretical values do not shift the center frequency of the extracted response, therefore the coefficients can be adjusted to extract the true response with no loss in information about the filter response. Application of this measurement and the extraction technique for a microwave coupled-line filter is presented in the next section.

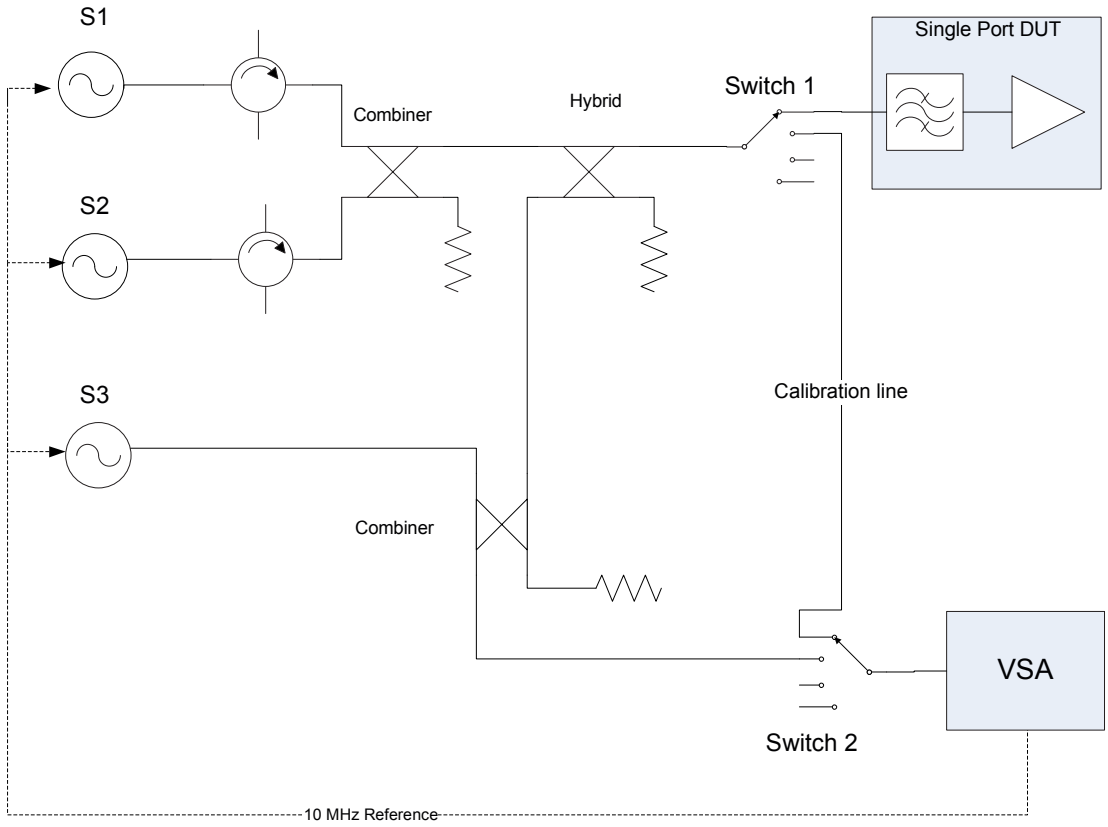


Figure 8.3: VSA-based VIMA reflection measurement setup.

8.5 Filter Tuning Example

Intermodulation measurements on a diode-loaded coupled-line microstrip filter using the VSA-based VIMA was performed to demonstrate the ability to tune a filter with extracted (S_{21}) parameters. A schematic of the circuit tested is shown in Fig. 8.4. The designed filter was constructed with variable capacitor-loaded resonators similar to microstrip filters in [124, 125]. The capacitors allowed the center frequency of the filter to be shifted between 1.7 – 2.0 GHz with an approximate fractional bandwidth of 10%. Two different cases were measured, that for a de-tuned filter and that for a properly tuned filter. The results of the tuning were verified by measuring the two-port transmission parameters with a VNA after extraction from the IMD measurements. The extraction equations and post-processing required to remove the phase shift introduced by the interconnect in Fig. 8.3 is

shown in Appendix E.

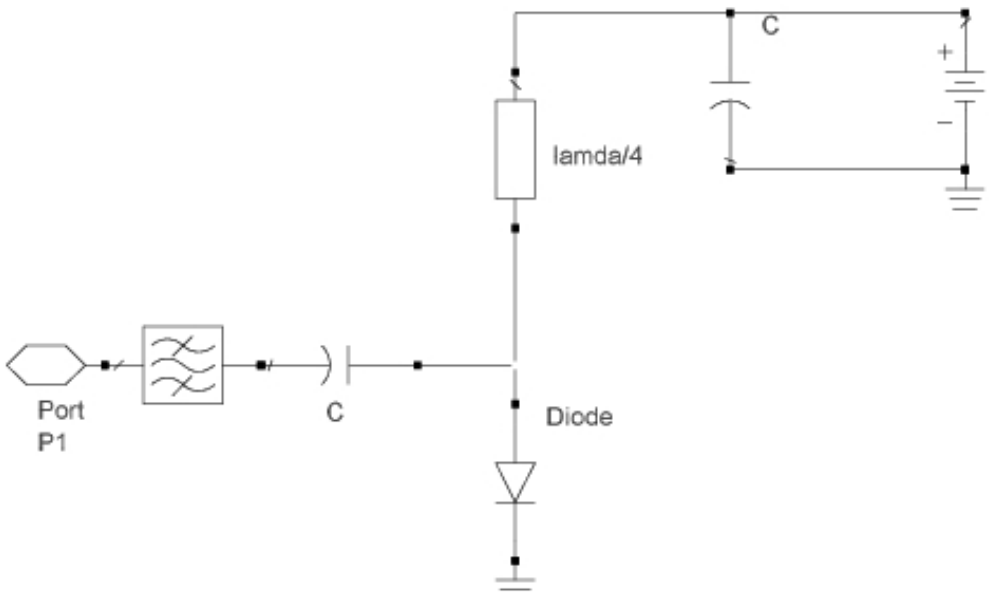


Figure 8.4: Schematic of nonlinearly loaded bandpass filter.

The de-tuned filter response is shown in Figs. 8.5 and 8.6 and the properly tuned response in Figs. 8.7 and 8.8 for the magnitude and phase respectively. The extracted magnitude and phase responses have both been normalized to match the desired result. This only introduces a vertical shift in the extracted values and thus does not affect the shape of the de-embedded transmission response. The ability of the IMD measurement to extract the magnitude and phase response is clearly shown for these two cases demonstrating that the

technique can provide the transmission response useful for tuning embedded filters. For the de-tuned case, the extracted magnitude showed errors between the measured S_{21} response of less than 2 dB, with phase errors of less than 20°. The extracted response for the properly tuned case exhibited the same magnitude discrepancies with a maximum phase error of 20°. Limitations of the measurement apparatus and uncompensated calibration errors are the main contributors of these errors. As shown in Figs. 8.5 and 8.7 the de-embedded response underestimated the measured transmission response over the frequency range from 1.7–1.8 GHz. This likely resulted from a standing wave due to impedance mismatches within the interconnect of the test setup that reduced the measured IMD magnitude. The phase errors for these measurements arose from the automation software which had a limit on the tone cancellation levels necessary for convergence on an estimate of the IMD phase. The limit used sought to increase the measurement speed by decreasing the phase accuracy. The raw measurements were also near the noise floor of the VSA receiver, thus insufficient SNR existed to achieve a lower phase uncertainty, which caused the large variation in extracted phase.

The magnitude response of the de-tuned setting also showed deviations between 1.85 and 1.9 GHz. This was related to a limitation of the dynamic range of the VSA receiver. In this frequency range, the S_{11} response of the de-tuned filter was very reflective resulting in large stimulus signals at the VSA input that reduced the ability to resolve the small nonlinear reflection generated by the filter. Correlation between the magnitude error and the measured S_{11} response is shown in Fig. 8.9, where significant amplitude errors occur when the return loss is less than 3 dB.

The extraction coefficients used to de-embed the filter response for the two cases were $m = 3$ and $p = 2.8$ and $m = 2.7$ and $p = 2.4$ for the de-tuned and tuned cases respectively. The phase coefficient p is relatively insensitive to deviations in the nonlinear response from the assumed third-order behavior since all odd order contributors to the IM3 product will produce the same relative phase for the shift due to the filter ($2\phi_H$) assuming no contribution to the IM3 phase by the nonlinear load itself. The magnitude coefficient m did show considerable deviation from the predicted $m = 4$ value for both tuning scenarios. This was due to the divergence from a third-order response of the nonlinearity at the input power levels used for the stimulus. A SPICE simulation of the diode IM3 response versus input power level shows that in the range of the stimulus used, (nominal power at the filter input of -15 dBm), the slope of the IM3 power output varies from less than 1:1 to 2.5:1, Fig.

8.10. As stated, this does not actually affect the center frequency of the extracted frequency response for the filter so the actual coefficient is optimized for the best fit to the desired response. The deviation of the magnitude and phase coefficients from the theoretical values may yield information about the nonlinear load itself. In an actual circuit implementation, this information may provide data on the bias of the terminating device, such as the input transistor(s) of an LNA for example.

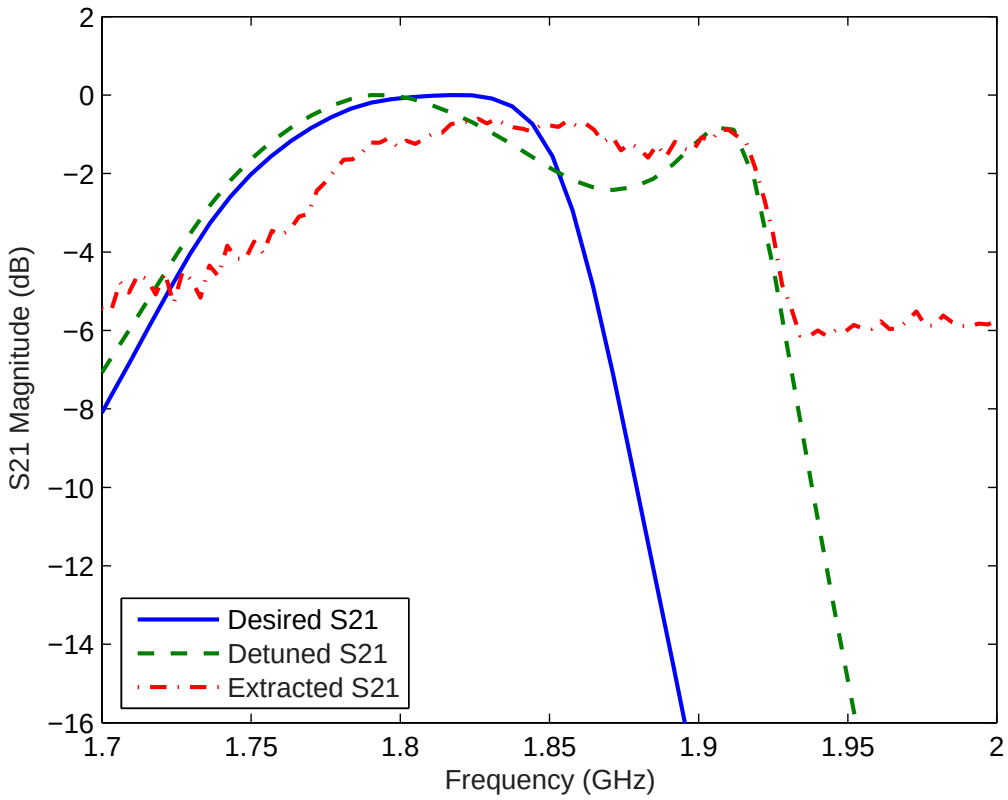


Figure 8.5: Desired and extracted S_{21} magnitude response for de-tuned filter.

8.6 Conclusion

The use of the VSA-based VIMA instrument and a suitable extraction technique was shown to accurately provide two-port transmission parameters for an embedded filter

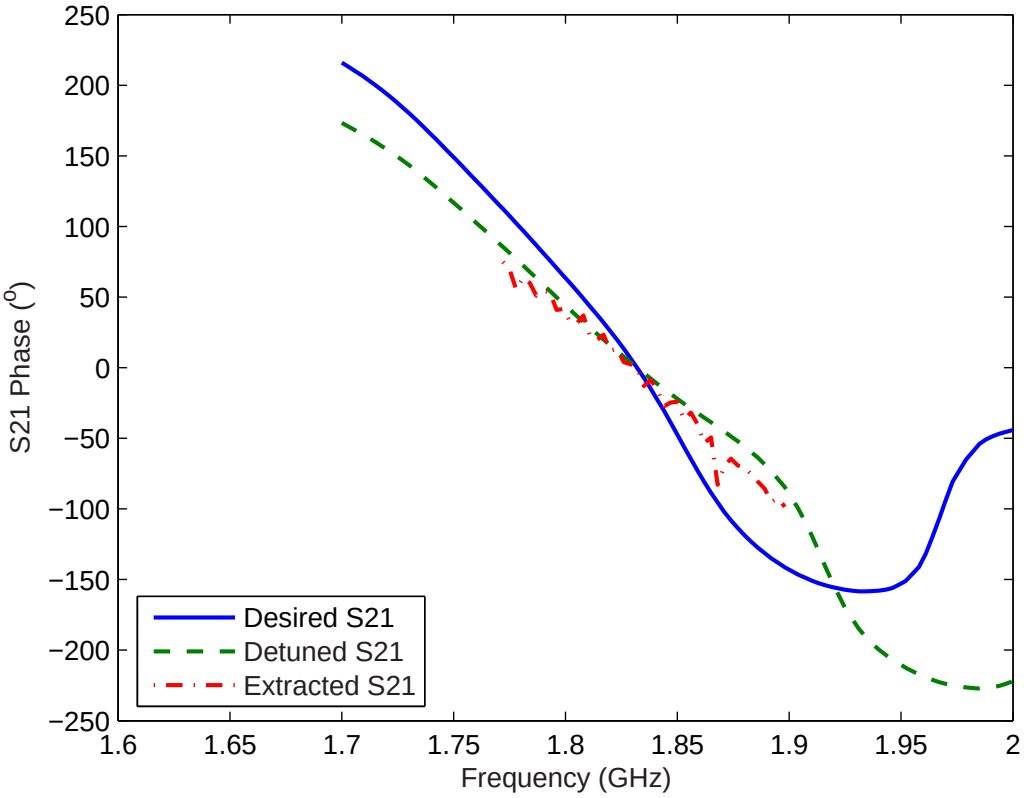


Figure 8.6: Desired and extracted S_{21} phase response for de-tuned filter.

from one-port measurements. The automated system establishes a baseline of performance that can be improved to increase the accuracy of the filter transmission response. Leveraging the high-dynamic range aspect of the system in Section 3.4.2 and possibly increasing this using the feed-forward cancellation system from Section 3.3 would allow extraction of the magnitude response into the stopband of the filter and potentially locate any transmission zeros. The successful implementation of this process advances current metrology techniques and provides information and capabilities not previously possible.

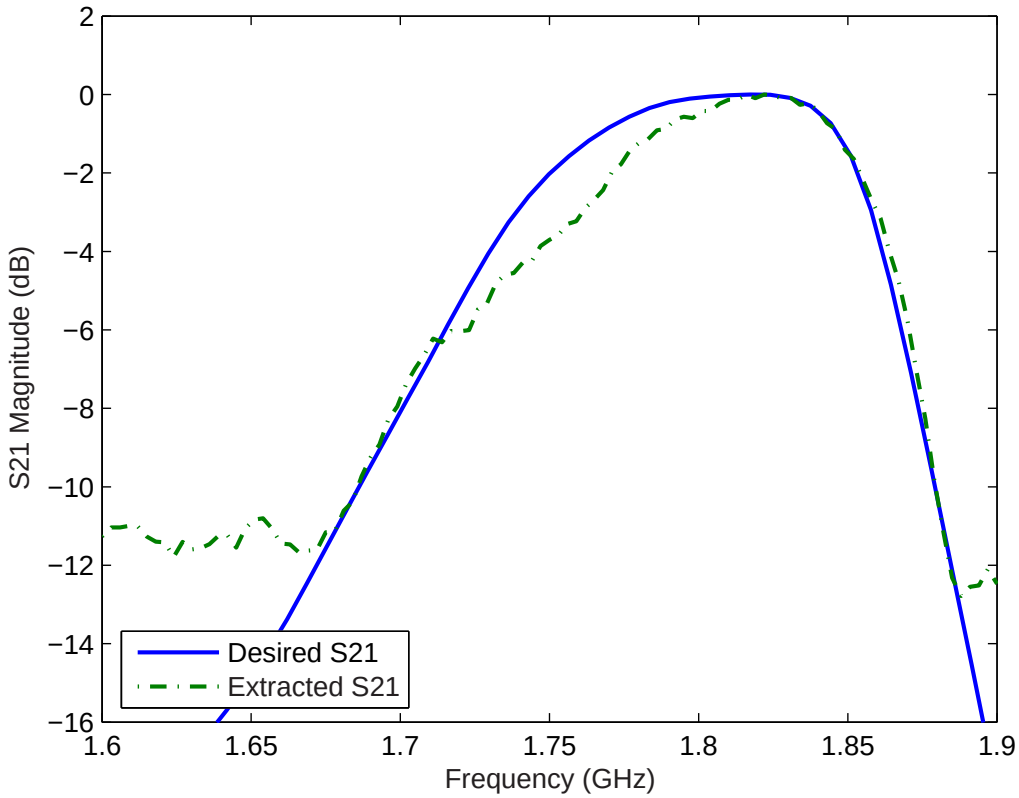


Figure 8.7: Desired and extracted S_{21} magnitude response for properly tuned filter.

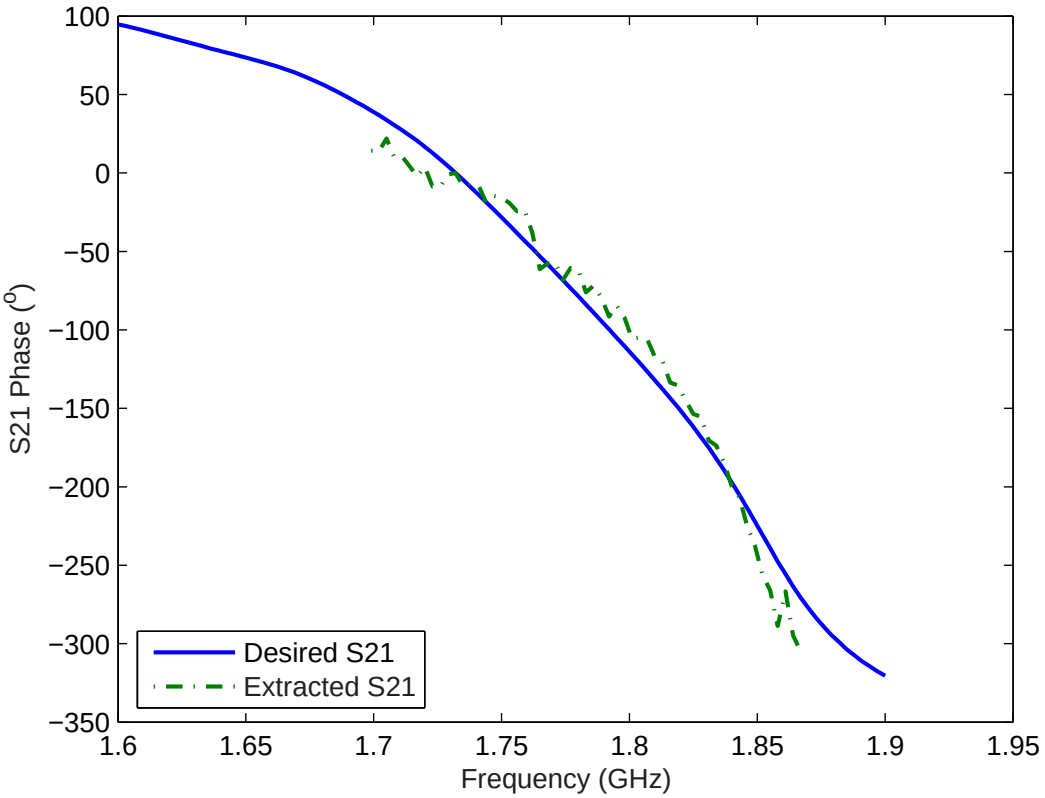


Figure 8.8: Desired and extracted S_{21} phase response for properly tuned filter.

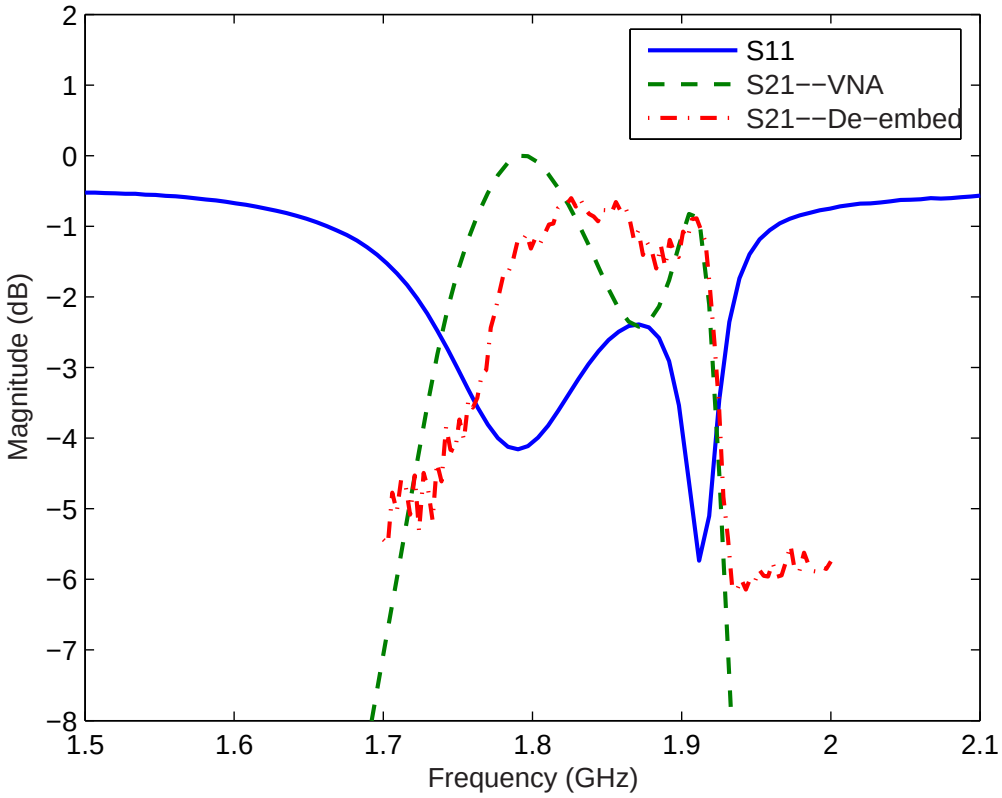


Figure 8.9: Comparison of extracted de-tuned filter S_{21} magnitude errors and the measured S_{11} response of the filter.

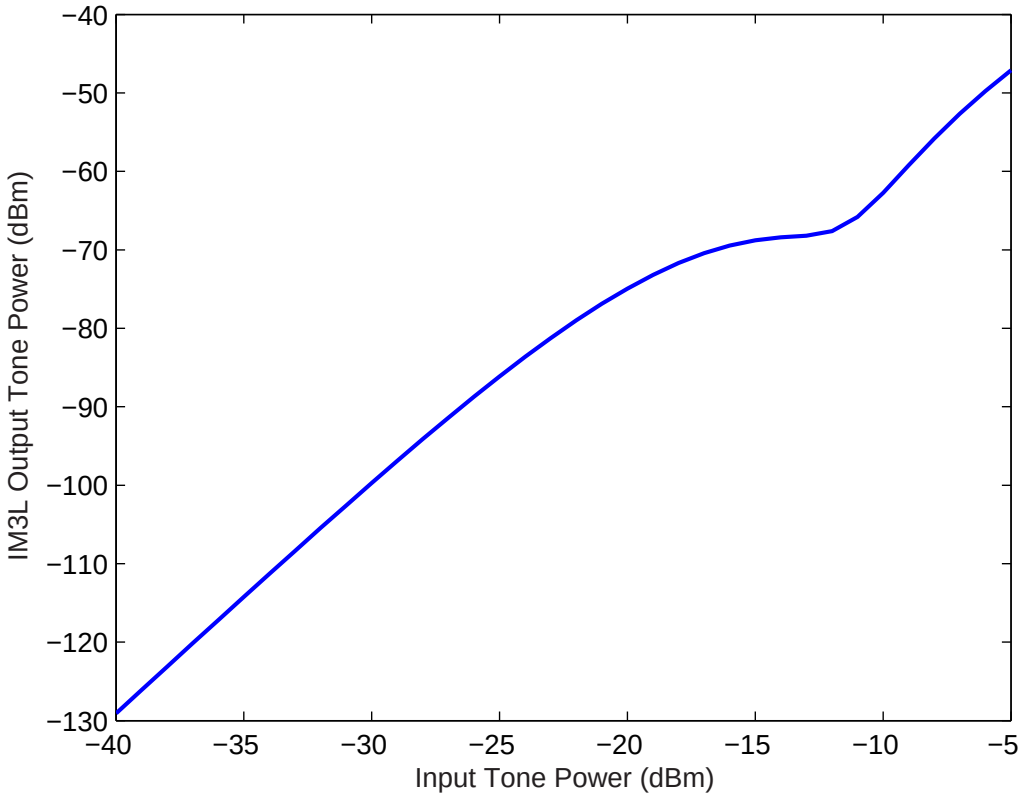


Figure 8.10: Simulated IM3L power output of the diode used as the nonlinear load for the filter under test.

Chapter 9

Conclusions and Future Work

9.1 Summary of Research and Original Contributions

The results of the research presented here represents significant advances in nonlinear metrology, modeling, and understanding of nonlinear processes in RF and microwave circuits. None of these areas stands alone since each impacts the design and manufacture of wireless products. A review of the original contributions and general knowledge gained are summarized here.

The novel VIMA systems in Sections 3.4.2 and 3.4.3 take advantage of a unique synthesizer architecture to eliminate the reference nonlinear device used in many of the existing intermodulation phase measurement systems. In addition, utilization of high-dynamic range receivers enables more accurate phase measurement and the ability to measure the nonlinear response of weak nonlinearities not possible in other systems. The dynamic range of these systems could benefit from the feed-forward cancellation system further enhancing the dynamic range performance. The VIMA of Section 3.4.2, which measures relative IM phase versus input power, showed for the first time the impact of baseband upconversion effects as a function of input power for nonlinear amplifiers. The VSA-based VIMA system in 3.4.3 can measure IM phase as a function of both stimulus power and frequency providing a valuable tool for nonlinear device and system characterization.

The multi-slice model in Chapter 4 was shown both in theory and experimentally

to allow the extraction of multiple parallel nonlinear processes separately. For the first time, a model of the even-order baseband contribution to IM generation as a function of stimulus level was derived from measurements. The modeling architecture was demonstrated to be effective in capturing both macro-level and micro-level memory effects for multiple amplifiers, which reinforces the applicability of this model construction and parameter extraction process. Incorporation of IM phase information from the VIMA systems advanced the state of behavioral modeling by reproducing IM generation that accounted for both amplitude and phase asymmetries. The model was able to capture the phase reference of individual contributors to the IM response and track their shift as a function of the stimulus level. Since the model was implemented with general components representing filters, mixers, and equation-based nonlinearities, it can be used in any type of simulator technology including harmonic balance, transient, or complex envelope.

The development of the remote characterization and nonlinear reflection parameter extraction metrology techniques provides valuable tools for the future of manufacturing test. The ability to determine transmission characteristics and tune embedded linear components (Chapter 8) in an integrated or assembled RF front-end could be leveraged in a manufacturing environment to reduce the number of tests and verify functionality without exercising full system-level performance. The remote characterization procedure in Chapter 5 also demonstrated the capability to determine the power conditions of active components in an RF front-end. This suggests that information about the bias levels in an integrated assembly could be determined yielding information that would otherwise be unobtainable due to the inaccessibility of internal components.

The large focus of this research on metrology techniques for nonlinear measurements established a large volume of practical considerations for performing good measurements on nonlinear systems. Realizing that nonlinear behavior can arise almost anywhere including the passive components, signal sources, and receiver of the measurement apparatus, one must guard against inadvertently establishing the conditions for generating nonlinear interactions that will corrupt the measurement. In this context, receiver dynamic range is the most important specification for nonlinear measurements as this component performs the actual measurements. In modern equipment, this parameter is limited to ~ 80 dB. This is the most critical component for systems designed for wired measurements of weak nonlinearities (as in PIM) or in remote characterization with weak responses from the DUT. For discrete tone systems, the dynamic range can be extended using feed-forward

cancellation in a 1:1 ratio of an increase in dynamic range for an increase in cancellation. The two-tone cancellation system designed here (Section 3.3) could therefore supply 20–30 dB of dynamic range improvement and potentially as much as 60 dB with improvements to system components.

9.2 Future Research

The results presented here has established a foundation for the continuation of research in nonlinear characterization and behavioral modeling using the developed metrology and measurement techniques. One of the areas that could be investigated is to determine the ability of the multi-slice model extracted from discrete tone stimulus to estimate nonlinear distortion for digitally modulated signals. Establishment of this would prove invaluable in a manufacturing environment by reducing the cost of test by using lower cost discrete tone signal sources and spectrum analyzer equipment instead of expensive vector modulation and demodulation equipment to measure digital modulation performance directly.

Another area for continued development would be the integration of the multi-tone cancellation system with either of the VIMA systems. This would require better implementation of the components within the cancellation system in terms of frequency and power range as well as improvements in the cancellation algorithm to increase the achievable cancellation. This would allow phase and magnitude characterization of weaker nonlinearities and potentially create a broadband instrument for in-band PIM measurements that are not currently possible.

Although the multi-slice model demonstrated a good ability to estimate IM generation for multiple amplifiers, the difficulties in handling the response of the Darlington amplifier showed the need to incorporate slices to handle feedback effects. For completeness, the model should also be extended to explain the observed phase shift in the amplifier response in the small-signal region.

Appendix A

Intermodulation Phase Measurement System Calibration

The RF signal generators used in the intermodulation phase measurement system shown in Fig. 3.9 had non-ideal operation with respect to phase injection as a function of output power. The following section outlines the corrections for their operation, applied to the measured data. The sources used in this measurement system injected phase shifts by two different mechanisms, at fixed attenuator switch points and a linear phase shift as a function of output power. The attenuator switch points only affects Source 3 in this setup since the two stimulus generators employ the attenuator lock function over their power output range used during data collection. Therefore the attenuator switch point phase shift does not propagate through the DUT. The attenuator switch point phase is simply an additive phase term that is subtracted from the measured phase indicated by Source 3 using calibration data taken on a network analyzer for the magnitude and direction of the phase shift at the switch points. The linear phase shift with output power affects all three Sources. Again for Source 3, this phase shift is additive and can be subtracted, for the stimulus Sources 1 and 2, the effect on the phase of the intermodulation products of interest is also additive as proven below.

A.1 Effect of Source Phase Shift on Measurements

The phase of the source output, ignoring the attenuator steps, can be expressed as

$$\phi_i(P) = m_s(\omega) \cdot P_{dB} + \phi_c \ , \tag{A.1}$$

where m_s is the linear phase shift with power and a weak dependence on output frequency ω and ϕ_c is a constant phase associated with the initial phase of the source when it locks to the desired frequency. In general, this constant phase can be taken as being zero. For the measurement system presented, the phase at some small-signal reference power is set to zero such that $\phi_i(P_{ref}) = 0$. Now the expression for the output of the sources becomes

$$x_i(t) = A_i \cos(\omega_i t + \phi_i(P)) \ , \tag{A.2}$$

Now putting the source output into complex exponential form

$$x_i(t) = \frac{1}{2} \sum_{\substack{q=-1 \\ q\neq 0}}^{1} A_q(P) e^{j(\omega_q t + \phi_q(P))} \ , \tag{A.3}$$

with $\omega_{-q} = -\omega_q$, $\phi_q(P) = m_s(\omega) \cdot P_{dB} + \phi_c$, and $\phi_{-q} = -\phi_q$.

With the source output in the form of (A.3), we can use the standard analysis of a nonlinearity represented as a power series with complex coefficients to determine the effect of the source phase shift on the DUT output. For an arbitrary number of stimulus tones Q, the DUT output for a particular polynomial order is given as

$$y_n(t) = a_n \left[\frac{1}{2} \sum_{\substack{q=-Q \\ q\neq 0}}^{Q} A_q(P) e^{j(\Phi_q)} \right]^n \ , \tag{A.4}$$

where $\Phi_q = \omega_q t + m_s(\omega_q) \cdot P_{dB} + \phi_c$. Now the response for the n-th order is

$$y_n(t) = \frac{a_n}{2^n} \sum_{q_1=-Q}^{Q} \cdots \sum_{q_n=-Q}^{Q} A_{q_1} \cdots A_{q_n} e^{j(\Phi_{q_1}(P) + \ldots + \Phi_{q_n}(P))} \ , \tag{A.5}$$

with the total response given by

$$y(t) = \sum_{n=0}^{N} y_n(t) \ . \tag{A.6}$$

Since the phase that is measured is a relative phase to that at a particular power, the expression for Φ_q becomes,

$$\Phi_q = \omega_q t + m_s(\omega_q) \cdot \Delta P_{dB} \tag{A.7}$$

Now expanding the first terms of (A.6), taking into account the effect of the complex polynomial coefficient we have

$$\begin{aligned} y(t) = & \\ & \frac{\mid a_3 \mid}{2^3} \sum_{q_1=-Q}^{Q} \sum_{q_2=-Q}^{Q} \sum_{q_3=-Q}^{Q} A_{q_1} A_{q_2} A_{q_3} e^{(\Phi_{q_1}+\Phi_{q_2}+\Phi_{q_3} \pm \phi_{a_3})} \\ + & \frac{\mid a_5 \mid}{2^5} \sum_{q_1=-Q}^{Q} \cdots \sum_{q_5=-Q}^{Q} A_{q_1} \cdots A_{q_5} e^{(\Phi_{q_1}+\cdots+\Phi_{q_5} \pm \phi_{a_5})} \\ + & \cdots , \end{aligned} \tag{A.8}$$

where $\pm\phi_n$ signifies that the phase of the complex coefficient contributes a complex conjugate term to the phase of positive and negative frequencies. The terms that contribute to the intermodulation product at $2\omega_1 - \omega_2$ for a two-tone stimulus have a form such that the frequency component of $\Phi_{q_1} + \cdots + \Phi_{q_n} = 2\omega_1 - \omega_2 + \phi$, where ϕ is a phase term dependent on the term under consideration. For a third-order contributor, there is only one set of q_i values that will produce this frequency (although there are multiple permutations of this set within the sum in (A.8)), i.e. $q_1 = 1$, $q_2 = 1$, $q_3 = -2$. For this set of mixing products, the phase component is

$$\begin{aligned} \Phi_{q_1} + \Phi_{q_2} + \Phi_{q_3} &= \omega_1 t + m_s(\omega_1) \cdot \Delta P_{1,dB} \\ &\quad + \omega_1 t + m_s(\omega_1) \cdot \Delta P_{1,dB} \\ &\quad - \omega_2 t - m_s(\omega_2) \cdot \Delta P_{2,dB} \\ \\ \Phi_{q_1} + \Phi_{q_2} + \Phi_{q_3} &= (2\omega_1 - \omega_2)t \\ &\quad + \Delta P_{dB} \ (2m_s(\omega_1) - m_s(\omega_2)) \\ \\ \Phi_{q_1} + \Phi_{q_2} + \Phi_{q_3} &= (2\omega_1 - \omega_2)t + \Delta P_{dB} \cdot m_s(\omega_1) \ , \end{aligned} \tag{A.9}$$

with the last step allowed since $m_s(\omega_1) \cong m_s(\omega_2)$. Similarly for one of the fifth-order contributors, $q_1 = 1$, $q_2 = 1$, $q_3 = 2$, $q_4 = -2$, $q_5 = -2$, the phase is

$$\Phi_{q_1} + \cdots + \Phi_{q_5} = (2\omega_1 - \omega_2)t + \Delta P_{dB} \cdot m_s(\omega_1) \ . \tag{A.10}$$

This demonstrates that for all of the odd-ordered contributors, the effect of the linear phase shift with power is the same and adds a single constant term to each of the phase components. Expressing the individual odd-ordered contributors in phasor form one obtains

$$\begin{aligned} Y_{\mathrm{IM3L}}(2\omega_1-\omega_2) = & \frac{\mid a_3 \mid}{2^{3-1}} c_{3,\mathrm{IM3L}}\ A^3 e^{j(\Delta P_{dB}\cdot m_s(\omega_1)+\phi_{a_3})} \\ & + \frac{\mid a_5 \mid}{2^{5-1}} c_{5,\mathrm{IM3L}}\ A^5 e^{j(\Delta P_{dB}\cdot m_s(\omega_1)+\phi_{a_5})} \\ & + \dots\ , \end{aligned} \tag{A.11}$$

or

$$\begin{aligned} Y_{\mathrm{IM3L}}(2\omega_1-\omega_2) = & e^{j(\Delta P_{dB}\cdot m_s(\omega_1))}\ . \\ & \Big[\ \frac{\mid a_3 \mid}{2^{3-1}} c_{3,\mathrm{IM3L}}\ A^3 e^{j\phi_{a_3}} \\ & + \frac{\mid a_5 \mid}{2^{5-1}} c_{5,\mathrm{IM3L}}\ A^5 e^{j\phi_{a_5}} \\ & + \dots\ \Big]\ . \end{aligned} \tag{A.12}$$

where $c_{n,k} = \sum_{l=0}^{\frac{n-3}{2}} \binom{n}{\frac{n-3}{2}-l,\, l+1,\, \frac{n+1}{2}-l}$ which represents the summation of all of the terms for each order that contribute to the frequency term k, here $\omega_k = 2\omega_1 - \omega_2$ [7]. From (A.12) it is clear that the effect of the phase shift in the linear sources is to introduce an additive phase term to that generated within the DUT. Therefore, the total relative phase at the DUT output at a particular power and frequency is

$$\phi_y = \phi_s + \phi_{DUT}\ . \tag{A.13}$$

where $\phi_s = \Delta P_{dB,s} \cdot m_s(\omega_1)$. Similarly, the total phase at the output of the cancellation source is

$$\phi_y = \phi_{meas} + \phi_c\ . \tag{A.14}$$

where ϕ_{meas} is the phase added by the cancellation source to achieve cancellation and $\phi_c = \Delta P_{dB,c} \cdot m_s(\omega_1)$ is the linear phase inserted by the cancellation source. Note the power terms in (A.13) and (A.14) are the power differences calculated from the small-signal reference power for both the stimulus sources and the cancellation source. Since the cancellation tone must be 180° out of phase with the desired frequency component at the DUT output, and the 0° reference cancellation phase was set at the small-signal power level, the actual phase of the frequency component under investigation is

$$\phi_{DUT} = \phi_c + \phi_{meas} - \phi_s\ . \tag{A.15}$$

Appendix B

Algorithm Details for the Multi-Tone Cancellation System

The algorithm executed by the microcontroller in the multi-channel feed-forward cancellation system is a straightforward approach to minimization of the power output at the hybrid combiners that serve to cancel the stimulus tones prior to the receiver input (Fig. 3.7). The algorithm determines the appropriate amplitude and phase shifts necessary to provide maximum tone cancellation separately and in sequence with the amplitude match performed first. Feedback to the microcontroller is provided by power detectors at various points within the two cancellation branches, the voltage output of these detectors is a logarithmic response producing a linearly increasing output for a linear increase in detected power in dB. Once the amplitude of the cancellation signal and the desired interfering tone has been match, the algorithm executes a binary search to determine the phase shift required for optimum cancellation. This alleviates the need to measure the phase of either the cancellation tone or the interfering stimulus tone and the search procedure can converge very quickly on the appropriate phase (generally less than fifteen iterations).

The current version of the software executes the algorithm based on the assumption that the two tones to be cancelled are not applied simultaneously during the cancellation of the first tone. Once the first tone has been cancelled, the software assumes that the

second tone is applied and then the adjustment of the cancellation tone for this channel is determined as above. The software to implement this algorithm is included below and still requires significant interaction with the user such that the code must be paused on the microcontroller in order to execute the cancellation algorithm. The software is not dynamic in its current form and does not continuously execute the cancellation procedure, i.e. it is a static implementation. The cancellation procedure must be executed directly by the user to accomplish cancellation and this control utilizes the debug interface of the software development environment for initiation of the software execution. The order of operations for executing the algorithm to achieve two-channel cancellation is listed here:

1. Power on the first stimulus generator.
2. Bring up the CodeWarrior debug environment with the cancellation software.
3. Execute the cancellation software by setting the channel select variable to channel 0, then set the *start* flag
4. After the algorithm executes, power on the second stimulus generator.
5. Execute the cancellation software by setting the channel select variable to channel 1, then set the *start* flag

At the completion of these steps, the cancellation hardware will stay in a static state with the achieved cancellation.

B.1 Algorithm Equations

The equation that guides the amplitude matching portion of the algorithm is based on the rejection equation, expressed in terms of cancellation level with amplitude (*alpha*) and phase error ($\phi - \pi$) as

$$\Delta P = P_1 - P_{\text{sum,avg}} = -10\log(\alpha^2 + 2\alpha\cos(\phi) + 1)\ . \tag{B.1}$$

The algorithm matches the amplitude of the tone to cancel by adjusting just the I control voltage with the Q control set to the minimum value. The computations necessary for

determining the proper I value begins by expressing the linear voltage gain of the modulator as

$$I_{\text{code}} \cdot a + I_{\text{offset}} = G_v \ , \tag{B.2}$$

where I_{code} is the binary representation of the modulator DAC input, a is the gain slope versus DAC code and I_{offset} is the gain offset. This equation is based on the measured linear voltage output of the modulator as a function of the I and Q control inputs. Now the form of the ADC response used to measure the output voltage of the power detectors is

$$P_{\text{x,code}} = P_{\text{x,dBm}} \cdot p_{\text{codeslope}} + C_{\text{codeoffset}} \ , \tag{B.3}$$

where $P_{\text{x,code}}$ is the binary code of the ADC for the power detector input power in dBm of $P_{\text{x,dBm}}$. Again the power detectors and the ADC both have linear responses to their respective inputs that yields the fit in (B.3) with a slope of $p_{\text{codeslope}}$ and offset of $C_{\text{codeoffset}}$. Now, the magnitude matching function uses the difference in power detected at the combining hybrids to determine the adjustment necessary for the modulator of the channel being matched. The delta ADC code value is

$$\Delta P_{\text{code}} = (P_{rx,\text{dBm}} - P_{\text{QM,dBm}}) p_{\text{codeslope}} \ , \tag{B.4}$$

where $P_{rx,\text{dBm}}$ and $P_{\text{QM,dBm}}$ are the power of the received tone to cancel and the output of the quadrature modulator respectively. Here the assumption is that the power detector and ADC responses are equal for the two feedback channels which is true for measured data of the components used in the system implementation (within an error of less than 1 dB).

The amplitude matching algorithm uses the estimated difference in power between the two tones at the cancellation hybrid input and the known value of the modulator I control value to determine the adjustment necessary for power match. Now at the power match goal, the target output of the modulator is

$$P_{QM,\text{lin,target}} = G^2_{v,\text{target}} P_{in,\text{lin}} = P_{rx,\text{lin}} \ , \tag{B.5}$$

where the target modulator voltage gain is $G_{v,\text{target}}$, the modulator input power is $P_{in,\text{lin}}$, and the received power of the tone to be cancelled is $P_{rx,\text{lin}}$. All of the powers are expressed in linear terms. Now from (B.2) we have

$$I_{\text{target}} \cdot a + I_{\text{offset}} = G_{v,\text{target}} \ , \tag{B.6}$$

for the target I value for achieving the power match condition. Now the current power output of the modulator can be expressed as

$$\begin{aligned} P_{QM,\text{current,dBm}} &= 20\log G_{v,\text{current}} + P_{in,\text{dBm}} \\ P_{QM,\text{current,dBm}} &= 20\log\left(I_{\text{current}}\cdot a + I_{\text{offset}}\right) + P_{in,\text{dBm}}\ . \end{aligned} \tag{B.7}$$

From (B.4) the output of the power output of the modulator is

$$\begin{aligned} P_{QM,\text{current,dBm}} &= P_{rx,\text{dBm}} - \frac{\Delta P_{\text{code}}}{p_{\text{codeslope}}} \\ &= 10\log\frac{G_{v,\text{target}}P_{in,\text{lin}}}{1\text{ mW}} - \frac{\Delta P_{\text{code}}}{p_{\text{codeslope}}} \\ P_{QM,\text{current,dBm}} &= 20\log\left(I_{\text{target}}\cdot a + I_{\text{offset}}\right) + P_{in,\text{dBm}} - \frac{\Delta P_{\text{code}}}{p_{\text{codeslope}}}\ . \end{aligned} \tag{B.8}$$

where the goal of achieving power match ($P_{rx,\text{dBm}} = P_{QM,\text{target,dBm}}$) was used to express the current modulator output power in terms of the target I control value. Now equating (B.7) and (B.8) and solving for I_{target} yields

$$I_{\text{target}} = 10^{\left(\frac{\Delta P_{\text{code}}}{20p_{\text{codeslope}}}\right)}\cdot I_{\text{current}} + \frac{I_{\text{offset}}}{a}\left(10^{\left(\frac{\Delta P_{\text{code}}}{20p_{\text{codeslope}}}\right)} - 1\right)\ . \tag{B.9}$$

The computations required in evaluating (B.9) are quite difficult for a fixed-point microprocessor. In particular the computation of the exponent term $10^{()}$ and the division $\frac{I_{\text{offset}}}{a}$ would consume considerable execution time if computed with a typical software math library. In the implementation of the algorithm on the 56F8323 processor, the solution used to overcome the challenge of computing these values directly was to use a lookup table for the exponent term and since the division term is constant simply inserting that term as a constant. The 56F8323 has a single-cycle hardware multiplier capable of both integer and fractional multiplication so performing multiplication is straightforward and fast.

The data needed for computation of (B.9) consists of the delta ADC value and the current I DAC control value. The constants $p_{\text{codeslope}}$ and a are determined from characterization of the ADC's and modulators. The delta ADC value is used to index a table of the previously computed exponent values for a range of delta values between -1500 and 1500. The limits on this argument of the exponent term was determined by analysis of the effect on the estimated correction needed to achieve power match. Outside of these limits the correction to the I control value exceeded the range of possible DAC settings. The exponent term table does not contain values for every delta ADC value in the above

range, rather the values are in 20 delta ADC steps. The proper value from the table to use is determined by averaging the bounding upper and lower values from the table performed by a simple binary search lookup routine. The table of exponent term values covers both fractional and integer values so the multiplication routines (Section B.2.8) for evaluating (B.9) are aware of the type of data for the argument in consideration.

The process for updating the I DAC value for matching power in a channel consists of computing the I_{target} value with (B.9), finding the difference between the target and current I value, then applying a scaled version of the difference to the current I value. The scaling factor is less than unity such that the algorithm executes the I adjustment in an iterative fashion until a minimum threshold value for the ADC code delta is reached or until a maximum number of iterations has occurred.

Once the magnitude matching process has finished, the next step is to adjust the phase of the quadrature modulator output to attain maximum cancellation. In this case, the feedback for the phase shift process is the power detector response at the output of the cancellation hybrid. Since it is not possible to measure the phase difference between the tones directly with the chosen hardware and extracting the required phase shift from the power measurements prior and after the cancellation hybrid would require inverse trigonometric function evaluation, a simple binary search routine was implemented to determine the optimum phase shift.

The phase search routine begins by rotating the cancellation tone to the cardinal directions of the I/Q domain and determining the two directions that produce the smallest measured power at the cancellation hybrid output. Determination of these two directions (which are always adjacent) sets the quadrant to search for the optimal phase. This has the effect of reducing the search space by a factor of four and greatly speeding up the phase shift process. With the quadrant identified, the phase shift algorithm uses a binary search routine to choose the best phase shift to maximize cancellation (minimize hybrid output power). Phase shift of the cancellation tone is achieved by keeping the modulator output power constant by using the I_{target} value from the amplitude routine and multiplying this with the appropriate sine and cosine terms for the current angle in the search pattern to set the I and Q control values. The sine and cosine functions are included in the 56F8323 math libraries. The observed response of this magnitude and phase adjustment procedure has shown convergence in approximately ten iterations for each procedure, per channel.

B.1.1 Miscellaneous Implementation Details

The quadrature modulator components used exhibited some deviation from ideal response shown in (B.2). In particular the offset values between the I and Q values for the two modulators varied both within each modulator (I offset different than Q) and between the modulators. The values of the offsets were determined by experimental characterization of the modulators and compensation for the effects are shown in the software in Section B.2.10.

B.1.2 Matlab Code for Computation of the Exponent Term Table

The following software was used in Matlab to generate the table values for the exponent term of (B.9).

```
% Variables
pcode_slope = 60.924;   % Linear fit slope for ADC response
a = 0.021494;           % Linear fit slope for QM gain vs I/Q value
IQ_offset = -11.098;    % Linear fit intercept for QM gain vs I/Q value
IQ_offset_div = floor(IQ_offset/a);
Imax = 1023;            % Maximum IQ value
Imin = 511;

% Form the lookup table.  The table is actually two tables, one for
% the index, here the deltaPcode value and the other table for the
% output value of 10^() for the particular deltaPcode values.  The
% table is restricted to deltaPcode values of +/-1500.  This range
% was observed to give the best coverage of the values of 10^()
% required.  The interpolation routine of the table output requires
% choosing the nearest two points (above and below) and taking the
% average value of this.  This is a very crude approximation, but
% can easily be performed in the micro.  The resolution chosen for
% this table is every 20 deltaPcode values.

% The values for the table output will be stored here as full precision,
% but for the computations they will be translated to their fixed
% precision equivalents.

% The table values for 10^() < 2 will be stored as the actual fraction
% value, since the largest unsigned fraction in the micro is ~2.
% The values > 2 will be stored as integers, rounded to nearest.

% delta ADC table values
```

```
tableindex = -1500:20:1500;
% Exponent term table values
tableoutput = 10.^(tableindex/(20*pcode_slope));
integerindex = find(tableoutput >=2);   % TABLEBREAK value is 95-1
tableoutput([integerindex]) = round(tableoutput([integerindex]));

function prod = fracmult(integer, frac)
%
% This function emulates a multiplication of a fixed precision
% fraction by an integer.  The precisioin of the fraction is
% fixed at 16 bits.
%
% The frac argument has already been converted from fractional
% decimal representation to integer fixed precision.

% Since the fractional component has been shifted by 15 left,
% the reverse is done after the multiplication.  This is done
% to preserve the precision of the operation by using an
% expanded result 32-bits instead of 16 for the multiplication
% result.
prod = floor(integer*frac/2^15);
```

B.2 C Code for Algorithm Execution on a Freescale 56F8323 DSP

This section includes all of the C code for executing a dual channel cancellation function with the hardware specified in Section 3.3 and the Freescale 56F8323 DSP. The software that implements the algorithm detailed in the previous section was written in the Metrowerks CodeWarrior development environment. The unique functions developed for this application are included below. Unlisted functions that appear in this software are part of the Metrowerks library for the 56F8323 processor.

B.2.1 Main Function

```
/**
###################################################################
**     Filename  : two_channel_cancel_test.C **     Project   :
two_channel_cancel_test **     Processor : 56F8323 **     Version
```

```
: Driver 01.03 **     Compiler  : Metrowerks DSP C Compiler **
Date/Time : 11/13/2004, 3:59 PM **     Abstract  : **         Main
module. **         Here is to be placed user's code. **
Settings  : **     Contents  : **         No public methods ** **
(c) Copyright UNIS, spol. s r.o. 1997-2002 **     UNIS, spol. s
r.o. **     Jundrovska 33 **     624 00 Brno **     Czech Republic
**     http      : www.processorexpert.com **     mail      :
info@processorexpert.com **
*********************************************************************/
/* MODULE two_channel_cancel_test */

/* Including used modules for compilling procedure */ #include
"Cpu.h" #include "Events.h" #include "AD1.h" #include "CHSEL.h"
#include "PWM1.h" #include "TFR1.h" #include "MFR1.h" #include
"MEM1.h" /* Include shared modules, which are used for whole
project */ #include "PE_Types.h" #include "PE_Error.h" #include
"PE_Const.h" #include "IO_Map.h"

// Include define macros
#include "C:\Metrowerks\one_ch_phase_shift_test\Code\defines.h"

// Include function prototypes
#include "functions.h"


// Global variables, used to pass data between main code and ISR's.
volatile int update;              // Indicates if I channel has
been
                                  // updated.
volatile int iqupdate;            // Indicates status of I and Q
values volatile int CHANSEL_val_next;    // Indicates which TxDAC
is selected

int index; int data_state, iqnew;            // State variable for
which I/Q channel is
                                  // active.
unsigned int Icurrent0, Qcurrent0;// Holding variables for I/Q
values, ch. 0 unsigned int Icurrent1, Qcurrent1;// Holding
variables for I/Q values, ch. 1


void main(void) {
  // Declare variable
  // GP indices
```

```
unsigned int i,k, iprev;
// Pointers to I/Q values to write to DAC
unsigned int *Icurrentptr0, *Qcurrentptr0;
unsigned int *Icurrentptr1, *Qcurrentptr1;
unsigned int *Icurrentptr, *Qcurrentptr;
unsigned int Icurrent, Qcurrent;

// Argument for I/Q target calculation
int deltaADC;
// ADC value arrays
unsigned int *AD_values, AD_just[5];

// Quadrant search parameters
unsigned int Ppos[4], srchquad;
Frac16 upper_ang, lower_ang, phi_est;
long phi_sum;
unsigned int pupper, plower, pbisect;
int deltaP;

// GP variables
byte error;
int amplooplimit, temp, start, chsel;
unsigned int IQoffset;
int ADRX, ADQM, ADHYB;
int qoffset, izero, qzero;

// Holding registers for debug
int deltaADCmat[100], Icurrentmat[100];

// Pointer to update flags set in the PWM ISR
volatile int *updptr, *iqupdptr, *iqnewptr;

/*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! ***/
PE_low_level_init();
/*** End of Processor Expert internal initialization.                    ***/

/* Write your code here */

// Call PESL init routine for Port A
PESL_init();

// Initialize pointers
updptr = &update;
*updptr = FALSE;
```

```
  iqupdptr = &iqupdate;
  *iqupdptr = FALSE;
  iqnewptr = &iqnew;
  *iqnewptr = FALSE;
  Icurrentptr0 = &Icurrent0;
  Qcurrentptr0 = &Qcurrent0;
  Icurrentptr1 = &Icurrent1;
  Qcurrentptr1 = &Qcurrent1;

  // Initialize values for I/Q
  Icurrent0 = 550;
  Qcurrent0 = QZERO_0;

  Icurrent1 = 550;
  Qcurrent1 = QZERO_1;

  iqnew = TRUE;

  while(*iqupdptr == FALSE){
    ;
  }

  // Wait for QM output to settle
  for(k=0;k<2000;++k){
    wait1();
  }


  AD_values = AD_just;

  start = FALSE;
  chsel = CH0;

  // Setup large loop for running through two tone cancellation.
  // Cancellation will not be automatic for both tones, rather this code
  // will need breakpoints to allow the user to determine which tone to
  // cancel.  The loop will repeat forever with break points for pauses

/*************************************************************************/
// Start of outer loop

  for(;;){

    if(start == TRUE){
```

```
  // Setup the test for the selected channel

  // Channel 0 selected for cancellation
  if(chsel == CH0){
    ADRX = ADRXCPL;
    ADQM = ADCH0CPL;
    ADHYB = ADCH0RX;
    Icurrentptr = Icurrentptr0;
    Qcurrentptr = Qcurrentptr0;
    izero = IZERO_0;
    qzero = QZERO_0;
    qoffset = QOFFSET0;
  }
  // Channel 1 selected for cancellation
  else{
    ADRX = ADCH0RX;
    ADQM = ADCH1CPL;
    ADHYB = ADCH1RX;
    Icurrentptr = Icurrentptr1;
    Qcurrentptr = Qcurrentptr1;
    izero = IZERO_1;
    qzero = QZERO_1;
    qoffset = QOFFSET1;
  }

  // Intialize Amplitude matching loop parameters
  amplooplimit = FALSE;
  i = 0;
  deltaADC = DELTAADCTHR + 1;

/*************************************************************************/

  // Begin initial amplitude correction loop.  Iterate on amplitude
  // correction until error between ADC values is below the
  // threshold.
  while( (deltaADC > DELTAADCTHR) | (deltaADC < -DELTAADCTHR)){

    // Poll the update flag to determine when the I value has been
    // written to the DAC, (actually half clock cycle later than
    // write).
    if(*iqupdptr == TRUE){

      // Wait for QM output to settle
      for(k=0;k<800;++k){
```

```
        wait1();
      }

      // Measure ADC channels 0 & 1, PE setup for 1 measurement
      // before report
      error = AD1_Measure(TRUE);
      // Fetch the ADC values
      error = AD1_GetValue16(AD_values);
      // Right justify data to appropriate 12-bit value
      AD_just[ADRX] = AD_just[ADRX] >> 4;
      AD_just[ADQM] = AD_just[ADQM] >> 4;

      // Compute difference between RX and QM output
      // Prx - Pqm
      deltaADC = AD_just[ADRX] - AD_just[ADQM];

      // Compute updated values of I
      temp = IQUpdateCompute(deltaADC, Icurrentptr, Qcurrentptr, izero, qzero);
      // Reset update flag to indicate new I/Q values are ready
      *iqupdptr = FALSE;
      iqnew = TRUE;
      // Store history of setting values
      Icurrentmat[i] = *Icurrentptr;
      deltaADCmat[i] = deltaADC;

      ++i;
      // Test for failed loop conditions, threshold not met,
      // but iteration limit reached
      if(i > AMPLOOPLIMIT){
        amplooplimit = TRUE;
        break;
      }  // end if(i > AMPLOOPLIMIT)
    } // end if(update == TRUE)
  } // End while(deltaADC > DELTAADCTHR)

/*************************************************************************/

  // Determine quadrant to search for maximum cancellation phase
  // setting.  Here the power at the output of the hybrid combiner
  // will be measured at each of the four cardinal points in
  // I/Q space to determine the boundaries for the quadrant to
  // search.

  // Compute offset from zero I/Q output for DAC
```

```
      IQoffset = *Icurrentptr - izero;

      // Measure power for phi_QM = 0

      error = AD1_Measure(TRUE);
      // Fetch the ADC values
      error = AD1_GetValue16(AD_values);
      // Right justify data to appropriate 12-bit value
//      AD_just[ADRXCPL] = AD_just[ADRXCPL] >> 4;
//      AD_just[ADCH0CPL] = AD_just[ADCH0CPL] >> 4;
      Ppos[0] = AD_just[ADHYB] >> 4;

      // Setup I/Q for phi_QM = pi/2
      *Qcurrentptr = qzero + IQoffset + qoffset;
      *Icurrentptr = izero;
      *iqupdptr = FALSE;
      iqnew = TRUE;
      temp = 800;

      // Wait for I/Q to be written to DAC
      while( (*iqupdptr) != TRUE ){
      ;
      }

      // Wait for QM output to settle
      for(k=0;k<temp;++k){
        wait1();
      }

      // Measure power for phi_QM = pi/2
      error = AD1_Measure(TRUE);
      // Fetch the ADC values
      error = AD1_GetValue16(AD_values);
      // Right justify data to appropriate 12-bit value
//      AD_just[ADRXCPL] = AD_just[ADRXCPL] >> 4;
//      AD_just[ADCH0CPL] = AD_just[ADCH0CPL] >> 4;
      Ppos[1] = AD_just[ADHYB] >> 4;

      // Setup I/Q for phi_QM = pi
      *Qcurrentptr = qzero;
      *Icurrentptr = izero - IQoffset;
      *iqupdptr = FALSE;
      iqnew = TRUE;
```

```
      // Wait for I/Q to be written to DAC
      while( (*iqupdptr) != TRUE ){
      ;
      }

      // Wait for QM output to settle
      for(k=0;k<temp;++k){
        wait1();
      }

      // Measure power for phi_QM = pi
      error = AD1_Measure(TRUE);
      // Fetch the ADC values
      error = AD1_GetValue16(AD_values);
      // Right justify data to appropriate 12-bit value
//      AD_just[ADRXCPL] = AD_just[ADRXCPL] >> 4;
//      AD_just[ADCH0CPL] = AD_just[ADCH0CPL] >> 4;
      Ppos[2] = AD_just[ADHYB] >> 4;

      // Setup I/Q for phi_QM = -pi/2
      *Qcurrentptr = qzero - IQoffset - qoffset;
      *Icurrentptr = izero;
      *iqupdptr = FALSE;
      iqnew = TRUE;

      // Wait for I/Q to be written to DAC
      while( (*iqupdptr) != TRUE ){
      ;
      }

      // Wait for QM output to settle
      for(k=0;k<temp;++k){
        wait1();
      }

      // Measure power for phi_QM = -pi/2
      error = AD1_Measure(TRUE);
      // Fetch the ADC values
      error = AD1_GetValue16(AD_values);
      // Right justify data to appropriate 12-bit value
//      AD_just[ADRXCPL] = AD_just[ADRXCPL] >> 4;
//      AD_just[ADCH0CPL] = AD_just[ADCH0CPL] >> 4;
      Ppos[3] = AD_just[ADHYB] >> 4;
```

```
  // Determine the quadrant to search and set the upper/lower angle
  // boundaries.  Angles will be expressed as signed fractions
  // representing phi_QM / pi.  Also set the values of the hybrid
  // power at the upper and lower boundaries from previous measurements.
  srchquad = findquad(Ppos);

  switch(srchquad){
    case QUADZERO:
      upper_ang = PLUSPIOVER2;
      lower_ang = ZERO;
      pupper = Ppos[1];
      plower = Ppos[0];
      break;
    case QUADONE:
      upper_ang = PLUSPI;
      lower_ang = PLUSPIOVER2;
      pupper = Ppos[2];
      plower = Ppos[1];
      break;
    case QUADTWO:
      upper_ang = MINUSPIOVER2;
      lower_ang = MINUSPI;
      pupper = Ppos[3];
      plower = Ppos[2];
      break;
    case QUADTHREE:
      upper_ang = ZERO;
      lower_ang = MINUSPIOVER2;
      pupper = Ppos[0];
      plower = Ppos[3];
      break;
  }  // end switch(srchquad)

  // Compute first angle to search
  phi_sum = (long) upper_ang + (long) lower_ang;
  phi_est = phi_sum >> 1;


/*************************************************************************/

  // Start initial phase correction loop.  Iterate on phase of QM
  // output until the power at the hybrid combiner output is
  // minimized.  Minimized here will be set by comparing the
  // powers at the upper and lower angle boundaries of the search
```

```
        // space.
        deltaP = DELTAPTHR+1;
        i = 0;
        while(deltaP > DELTAPTHR){

          // Compute I&Q for phase setting which has not been measured, will
          // only be one point for each iteration
          computeIQphase(phi_est, IQoffset, Icurrentptr,
                         Qcurrentptr, izero, qzero, qoffset);

          // Set I&Q
          *iqupdptr = FALSE;
          iqnew = TRUE;
          while((*iqupdptr != TRUE)){
          ;
          }

          // Wait for output of QM to settle
          for(k=0;k<800;++k){
            wait1();
          }

          // Read the power at the hybrid output
          error = AD1_Measure(TRUE);
          // Fetch the ADC values
          error = AD1_GetValue16(AD_values);
          // Right justify data to appropriate 12-bit value
//          AD_just[ADRXCPL] = AD_just[ADRXCPL] >> 4;
//          AD_just[ADCH0CPL] = AD_just[ADCH0CPL] >> 4;
          pbisect = AD_just[ADHYB] >> 4;


          // May need amplitude correction routine, not sure.

          // Determine the upper/lower limits for the next bisection
          // step.  The decision is based on the power of the upper
          // and lower boundaries.  Here upper refers to the physically
          // larger number, in the negative quadrants, the sign is
          // respected so -1 < -0.5.  Doesn't really matter, only value
          // of power is important.
          if( pupper < plower){
            // True value lies closer to upper, set lower angle as the
            // bisect angle, adjust power value as well.
            lower_ang = phi_est;
```

```
          deltaP = plower - pupper;
          plower = pbisect;
        }
        else{
          // True value lies closer to lower boundary, set upper angle
          // as the bisect angle and adjust power values.
          upper_ang = phi_est;
          deltaP = pupper - plower;
          pupper = pbisect;
        }

        // Compute the next bisection phase, phi_QM/pi
        phi_sum = (long) upper_ang + (long) lower_ang;
        phi_est = phi_sum >> 1;

        ++i;
        // Test for failed loop conditions, threshold not met,
        // but iteration limit reached
        if(i > PHASELOOPLIMIT){
          amplooplimit = TRUE;
          break;
        }  // end if(i > AMPLOOPLIMIT)

      }  // End of phase shift for maximum cancellation loop
      start = FALSE;
    }  // End of START==TRUE loop
  }  // End of master infinite loop
}

/* END two_channel_cancel_test */ /* **
###################################################################
** **     This file was created by UNIS Processor Expert 03.32 for
**     the Motorola 56800 series of microcontrollers. ** **
###################################################################
*/
```

B.2.2 Macro Definitions

```
/* This file contains definitions for the project */

#define Imask 0x0400
#define Qmask 0x03FF
#define SIZEOFTABLE 512
#define INDEXTABLEMAX 1500
```

```
#define INDEXTABLEMIN -1500
#define TABLESIZE 151
#define TABLEBREAK 94
#define IQOFFSETDIV 511
#define IQMAX 950
#define IZERO_0 508
#define QZERO_0 513
#define QOFFSET0 15
#define IZERO_1 504
#define QZERO_1 510
#define QOFFSET1 -28
#define DELTAADCTHR  10
#define DELTAPTHR 10
#define AMPLOOPLIMIT 30
#define PHASELOOPLIMIT 30
#define QUADZERO 0
#define QUADONE 1
#define QUADTWO 2
#define QUADTHREE 3
#define ZERO 0
#define PLUSPIOVER2  16384
#define PLUSPI       32767
#define MINUSPIOVER2 -16384
#define MINUSPI      32768

#define ADRXCPL  0
#define ADCH0CPL 1
#define ADCH0RX  2
#define ADCH1CPL 3
#define ADCH1RX  4

#define I0_UPDATED 2
#define I1_UPDATED 3
#define Q0_UPDATED 4
#define Q1_UPDATED 5

#define CH0 0
#define CH1 1

#define LOW 0
#define HIGH 1
```

B.2.3 Functions Header File

```
/*
 * Prototype for project functions
 */

#include "PE_Types.h" #ifndef _FUNCTIONS_H_ #define _FUNCTIONS_H_
 /*
  * findindex()
  *
  * Search routine for finding appropriate value of
  * exponential function for deltaADC.
  */
 int findindex(int, unsigned int *);

 /*
  * fracmultu()
  *
  * Multiplication routine for mixed unsigned fractional
  * and integer values.
  *
  */
 unsigned int fracmultu(unsigned int, unsigned int);

 /*
  * fracmultu()
  *
  * Multiplication routine for an unsigned fractional
  * and signed integer values.
  *
  */
 int fracmults(int, unsigned int);

 /*
  * PESL_init(void)
  *
  * This routine sets up Port A to act as a single bus and
  * enables the PWM reload interrupt.
  *
  */
 void PESL_init(void);

 /*
  * IQUpdateCompute()
  *
  * This function computes the updated I/Q values for the amplitude
```

```
 * adjustment routine.
 *
 */
int IQUpdateCompute(int, unsigned int *, unsigned int *, int, int);


/*
 * wait1()
 *
 * This function puts a delay of XX us into the code execution.
 *
 */
void wait1(void);

/*
 * findquad()
 *
 * This function finds the quadrant to search for choosing the
 * appropriate phase of the cancellation tone.
 *
 */

unsigned int findquad(unsigned int *);

/*
 * computeIQphase()
 *
 * This function computes the new values for I and Q to rotate the
 * QM output the to the supplied phase
 *
 */

void computeIQphase(Frac16, unsigned int, unsigned int *, unsigned int *, int, int,
#endif
```

B.2.4 Power Difference Exponent Search

The lookup table for the exponent value from (B.9) does not include all possible values for the difference between the ADC codes of the signals to be combined for cancellation. The encoded exponent data is indexed with delta ADC code spacing of 20 for code differences between ± 1500. This function takes the argument of the measured delta ADC value and returns the upper and lower bounding index values used to determine the

exponent value for use in (B.9).

```
/*
 * int findindex(int deltaADC, * int indices)
 *
 * This function takes the deltaADC value from the measurement of power
 * at two points in the circuit and determines the indices of the two
 * entries that bound the deltaADC value.  These indices are used to
 * estimate the exponential function for determining power correction
 * for cancellation.
 *
 * The two indices of the bounding terms are returned in an array to
 * the calling routine.  If the deltaADC value exceeds the bounds of
 * the table, then the indices are set at either the maximum or
 * minimum table index.
 *
 * The search routine is a simple binary search.
 *
 */

 #include "defines.h"
 #include "functions.h"

 int findindex(int deltaADC, unsigned int * indices)
 {
   // Declare variables
   int increment;               // Increment value, step size in binary search
   unsigned int top_index;      // Placeholder for current table position
   extern int deltaPcodeindex[TABLESIZE];
   int temp;

   // Setup search routine
   increment = TABLESIZE >> 2;  // Set as quarter table size
   top_index = TABLESIZE >> 1;  // Initial position at table midpoint

   // Test if deltaADC is at the bounds of the table
   if(deltaADC >= INDEXTABLEMAX)
   {
     // Set indices at maximum of table
     *indices = TABLESIZE - 1;
     *(indices+1) = TABLESIZE - 1;
   }
   else if(deltaADC <= INDEXTABLEMIN)
   {
     // Set indices at minimum of table
```

```
  *indices = 0;
  *(indices+1) = 0;
}
// Else if this not the case, start search loop
else
{
  // Start loop, broken by location of deltaADC bounds
  while(1)
  {
    temp = deltaPcodeindex[25];
    // Test if two points are both greater than deltaADC, if true
    // then decrease the top_index pointer
    if( (deltaPcodeindex[top_index] > deltaADC) &
        (deltaPcodeindex[top_index - 1] > deltaADC) )
    {
      // Step down in the table
      top_index = top_index - increment;
      // Decrease the increment size unless at 1
      if(increment > 1)
        increment = increment >> 1;
    }
    // Test if two points are both less than deltaADC, if true
    // then increase the top_index pointer
    else if( (deltaPcodeindex[top_index] < deltaADC) &
             (deltaPcodeindex[top_index - 1] < deltaADC) )
    {
      // Step up in the table
      top_index = top_index + increment;
      // Decrease the increment size unless at 1
      if(increment > 1)
        increment = increment >> 1;
    }
    // If these conditions are not true, then the bounds on
    // deltaADC have been found, break the loop.
    else
    {
      // Set indices values
      *indices = top_index;
      *(indices+1) = top_index - 1;

      break;
    }
  }
}
```

```
  return 0;
}
```

B.2.5 Multiplication Routine for Mixed Unsigned Fractional and Integer Values

```
/*
 * int fracmult(int, int)
 *
 * This function multiples two 16-bit numbers and returns a 16-bit value.
 * The multiplicands consist of one int which is a fractional value and
 * the other is an integer.  The multiplication in this manner requires
 * that the products is right shifted by the fractional multiplier, in
 * this case 2^15, or right shifted by 15 bits.
 */

#include "functions.h"
#include "intrinsics_56800E.h"
#include "dspfunc.h"


unsigned int fracmultu(integer, fraction)
{
  // Declare variables
  unsigned long product;
  unsigned int prodshift;

  // Set SA bit in OMR and wait for it to occur
  turn_on_sat();
  // Use macro that executes a mixed integer and fraction multiplication
  // with both multipicands unsigned.  The result of the multiplication
  // of the two numbers in 16-bit format is a 32-bit fractional
  product = impyuu(integer, fraction);
  // Use non-saturating 32-bit shift, with negative shift amounts
  // indicating right shift to produce the 16-bit integer value, rounded
  // to -infinity
  prodshift = L_shlftNs (product, -15);
  //prodshift = product >> 15;

  return(prodshift);
}
```

B.2.6 Multiplication Routine for an Unsigned Fractional and Signed Integer Values

```
/*
 * int fracmult(int, int)
 *
 * This function multiples two 16-bit numbers and returns a 16-bit value.
 * The multiplicands consist of one int which is a fractional value and
 * the other is an integer.  The multiplication in this manner requires
 * that the products is right shifted by the fractional multiplier, in
 * this case 2^15, or right shifted by 15 bits.
 */

 #include "functions.h"
 #include "intrinsics_56800E.h"
 #include "dspfunc.h"


 int fracmults(int fraction, unsigned int integer)
 {
   // Declare variables
   long product;
   int prodshift;

   // Set SA bit in OMR and wait for it to occur
   turn_on_sat();
   // Use L_mult intrinsic function to generate a 32-bit result
   product = impysu(fraction, integer);
   // Use non-saturating 32-bit shift, with negative shift amounts
   // indicating right shift.
   prodshift = L_shlftNs (product, -15);
   //prodshift = product >> 15;

   return(prodshift);
 }
```

B.2.7 Peripheral Initialization Function

Most of the software necessary for peripheral initialization is automatically generated by the CodeWarrior IDE, this function handles some initialization tasks not possible with the automated tools.

```
/*
```

```
* This file contains the PESL intialization function calls to drive Port A
* as a single bus and enable the PWM interrupt
*/
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include "PESLlib.h"

#include "functions.h"

void PESL_init(void){

 // Currently, the clock signal for the DAC is supplied by a PWM bean with
 // a 50% duty cycle.  However, the PWM pin uses is GPIO_A0, and Port A is
 // the largest IO port on the micro which I wanted to use for the parallel
 // data bus of the DAC.  When the PWM bean is used, PE will not allow Port
 // A to be used as a GP bus.  So, as a work around, I used the PESL routines
 // to drive the remaining 11 bits of Port A as a single bus for the data
 // and SEL line of the DAC I/F.

 // Initialize Port A bits 11-1 as GPIO
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_1 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_2 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_3 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_4 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_5 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_6 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_7 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_8 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_9 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_10 );
 PESL( GPIOA, GPIO_SETAS_GPIO, BIT_11 );

 // Set Port A bits 11-1 as outputs
 PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_1 );
 PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_2 );
 PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_3 );
 PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_4 );
 PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_5 );
 PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_6 );
 PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_7 );
 PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_8 );
 PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_9 );
```

```
  PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_10 );
  PESL( GPIOA, GPIO_SETAS_OUTPUT, BIT_11 );

  // Set Port A bits 11-1 as push-pull
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_1 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_2 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_3 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_4 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_5 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_6 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_7 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_8 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_9 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_10 );
  PESL( GPIOA, GPIO_SETAS_PUSHPULL, BIT_11 );

  // Enable PWM interrupt on timer reload
  PESL( PWMA, PWM_RELOAD_INT, PWM_ENABLE );
}
```

B.2.8 Quadrature Magnitude Update Function

```
/*
 * This function executes the IQ target calculation algorithm and
 * returns the IQ target values.
 */

// Included files
#include "defines.h" #include "functions.h" #include "tables2.h"
#include "dspfunc.h"


int IQUpdateCompute(int deltaADC, unsigned int *Icurrentptr,
                    unsigned int *Qcurrentptr, int izero, int qzero){

  // Variables for I/Q target calculation
  extern unsigned int logtermtable[TABLESIZE];
  unsigned int * indices;
  unsigned int top_index, bot_index, index[2];
  unsigned int IQcurrent, IQtarget;
  int temp;
  int logterm;                          // Function table output
  int mu_num;                             // mu update constant numerator
  long result1;
```

```
long result2;
long result3;


// Initialize the algorithm constants
mu_num = 5;
indices = index;
IQcurrent = *Icurrentptr;


// Find the index of the logterm table from the current value of
// the deltaADC term
temp = findindex(deltaADC, indices);
top_index = index[0];
bot_index = index[1];


// First compute average of the function table output
// for the two indices.  If at the table break point
// between fractional and integer, use a separate
// average procedure
if(top_index == TABLEBREAK)
  logterm = 2;
else
{
  result1 = logtermtable[top_index];
  result2 = logtermtable[bot_index];
  result1 = result1 + result2;
  logterm = result1 >> 1;
}

// Now compute two terms of IQtarget equation. If
// top_index >= TABLEBREAK, use an integer multiplication
// relationship.  Otherwise a mixed integer fraction multiplication
// must occur.  Hopefully, the compiler handles this part ok so
// there is no need for assembly to use the 36-bit registers to hold
// results.
// result3 needed so that could avoid a signed fractional, unsigned
// integer multiplication.  Since IQOFFSETDIV is really negative
// in the real computation, and it is multiplied by -1 in
// IQoffsetdiv *(10^() - 1), an unsigned multiplicaton of IQoffsetdiv
// and logterm is taken, then subtracted from IQoffsetdiv*(-1).  Note
// the macro for IQOFFSETDIV is taken as positive and all effects of
// the negative sign are taken into account by the operations on
```

```
  // result2 and result3.

  if(top_index >= TABLEBREAK)
  {
    result1 = logterm*IQcurrent;
    result2 = logterm*IQOFFSETDIV;
    result3 = IQOFFSETDIV - result2;
  }
  else
  {
    result1 = fracmultu(IQcurrent, logterm);
    result2 = fracmultu(IQOFFSETDIV, logterm);
    result3 = IQOFFSETDIV - result2;
  }

  // IQtarget is the sum of result1 and result3, but will be a 16-bit
  // integer value
  IQtarget = result1 + result3;

  // The IQcurrent update routine multiplies the difference between
  // IQcurrent and IQtarget and adds that to IQcurrent
  temp = impysu((IQtarget - IQcurrent), mu_num) >> 4;
  IQcurrent = IQcurrent + temp;

  // Clip IQcurrent to minimum/maximum of range
  if(IQcurrent > IQMAX)
    IQcurrent = IQMAX;

  if(IQcurrent < izero)
    IQcurrent = izero;

  // Assign I/Q values to pointers
  *Icurrentptr = IQcurrent;
  *Qcurrentptr = qzero;

  return(0);
 }
```

B.2.9 Quadrant Search Function for Quadrature Phase Adjustment

```
/*
 * This function finds the minimum two power points in the four
 * cardinal directions power measurements supplied as parameters.
 * The function returns an integer representing the quadrant that
```

```
 * the desired cancellation phase lies in.
 *
 */

#include "functions.h"

unsigned int findquad(unsigned int Ppos[]){

  // Declare variables
  int temp, low_index, i;
  unsigned int quadsum[3];

  // Find lowest pair of points in the array.  First form sums
  // of quadrants then find lowest sum.  This is the quadrant
  // to search.
  for(i=0;i<4;++i){
    quadsum[i] = Ppos[i] + Ppos[(i+1)%4];
  } // End for(i=0;i<4;++i)

  // Find lowest sum
  low_index = 0;
  for(i=1;i<4;++i){
    if( quadsum[i] < quadsum[low_index])
      low_index = i;
  }  // End for(i=0;i<4;++i)

  return(low_index);
}  // End findquad()
```

B.2.10 Quadrature Phase Update Function

```
/*
 * This function computes the I and Q values to place the QM output
 * at the supplied angle phi.  The magnitude of the output is held
 * constant by using the current value of I and Q to ensure that
 * sqrt(I^2 + Q^2) is the same for the new values.  The IQoffset value
 * is the magnitude of the original amplitude vector, which will aid
 * in computing a constant magnitude value for the new I and Q.
 *
 */

 #include "functions.h"
 #include "defines.h"
 #include "TFR1.h"
```

```
#include "PE_Types.h"


void computeIQphase(Frac16 phi, unsigned int IQoffset,
                    unsigned int * Icurrentptr, unsigned int * Qcurrentptr,
                    int izero, int qzero, int qoffset){

// Declare variables
unsigned int Inext, Qnext;
int Ioffset, Qoffset;
// Fractional result of sin() cos() to compute change in I/Q
Frac16 Inorm, Qnorm;

// Compute I adjustment factor
Inorm = TFR1_tfr16CosPIx(phi);
// Compute Q adjustment factor
Qnorm = TFR1_tfr16SinPIx(phi);

// Perform multiplication with signed fractional and unsigned
// integer.
Ioffset = fracmults(Inorm, IQoffset);
Qoffset = fracmults(Qnorm, IQoffset);

// Add offsets to IQZERO to compute new values
*Icurrentptr = Ioffset + izero;
if(Qoffset < 0){
  // Handle channel 1 offset differently so that it does not result
  // in oscillations around the Q channel zero point
  if(qoffset < 0){
    if(qoffset < Qoffset){
      *Qcurrentptr = Qoffset + qzero;
    }  // End of channel 1 if for qoffset < Qoffset
    else{
      *Qcurrentptr = Qoffset + qzero - qoffset;
    }  // End of channel 1 else for qoffset < Qoffset
  }  // End of channel 1 for Qoffset < 0
  else{
    *Qcurrentptr = Qoffset + qzero - qoffset;
  }  // End of channel 0 else for Qoffset < 0

}  // End of if for Qoffset < 0
else{
  // Handle channel 1 offset differently so that it does not result
  // in oscillations around the Q channel zero point
```

```
   if(qoffset < 0){
     if(-qoffset > Qoffset){
       *Qcurrentptr = Qoffset + qzero;
     }  // End of channel 1 if for qoffset < Qoffset
     else{
       *Qcurrentptr = Qoffset + qzero + qoffset;
     }  // End of channel 1 else for qoffset < Qoffset
   }  // End of channel 1 for Qoffset < 0
   else{
     *Qcurrentptr = Qoffset + qzero + qoffset;
   }  // End of channel 0 else for Qoffset < 0
 }  // End of else for Qoffset < 0
}
```

B.2.11 Interrupt Service Routines

```
/**
######################################################################
**     Filename  : Events.C **     Project   :
two_channel_cancel_test **     Processor : 56F8323 **     Beantype
: Events **     Version   : Driver 01.00 **     Compiler  :
Metrowerks DSP C Compiler **     Date/Time : 11/13/2004, 3:59 PM
**     Abstract  : **         This is user's event module. **
Put your event handler code here. **     Settings  : **
Contents  : **         PWM1_OnEnd - void PWM1_OnEnd(void); ** **
(c) Copyright UNIS, spol. s r.o. 1997-2002 **     UNIS, spol. s
r.o. **     Jundrovska 33 **     624 00 Brno **     Czech Republic
**     http      : www.processorexpert.com **     mail      :
info@processorexpert.com **
######################################################################*/
/* MODULE Events */

/*Including used modules for compilling procedure*/ #include
"Cpu.h" #include "Events.h" #include "AD1.h" #include "CHSEL.h"
#include "PWM1.h" #include "TFR1.h" #include "MFR1.h" #include
"MEM1.h"

/*Include shared modules, which are used for whole project*/
#include "PE_Types.h" #include "PE_Error.h" #include "PE_Const.h"
#include "IO_Map.h" #include "PE_Timer.h"

#include "C:\Metrowerks\one_ch_phase_shift_test\Code\defines.h"
#include "functions.h"
```

```
/* **
============================================================================
**     Event       :  PWM1_OnEnd (module Events) ** **     From
bean   :  PWM1 [PWM] **     Description : **         This event is
called when the specified number of cycles **         has been
generated. (only when the bean is enabled - **         Enable and
the events are enabled - EnableEvent). **     Parameters  : None
**     Returns     : Nothing **
============================================================================
*/ #pragma interrupt called void PWM1_OnEnd(void) {
  extern int data_state, iqnew;
  extern int Icurrent0, Qcurrent0;
  extern int Icurrent1, Qcurrent1;
  extern int update;
  extern int CHANSEL_val_next;
  volatile extern int iqupdate;
  extern int index;

  int value;
  // This interrupt routine writes the values to Port A to control
  // the I and Q inputs of the TxDAC for each of the cancellation
  // channels.  The 10-bit data bus is multiplexed four ways.  The
  // first two are on each of the TxDACs themselves, controlled by
  // the SEL line (bit 11 of the I/F, controlled with Imask and Qmask).
  // The other two selections are provided by the clock routing using
  // the external demux.  The clock line for each TxDAC is held low
  // when it is inactive while the clock line is driven for the active
  // TxDAC.

  // The value written to Port A must be masked with the active channel
  // on the TxDAC (either I or Q), then left-shifted by 1 since bit 0
  // of Port A is used by the PWM module.  The PWM module supplies a 50%
  // duty cycle clock that is used to clock and write the data into the
  // TxDAC on chip registers.

  // The writing procedure consists of the following.  First the I values
  // of channels 0 and 1 are written, then the Q values of the two
  // channels.  This sequence repeats.

  // Note: When the Q value is written to the TxDAC, both I and Q are
  // transferred into their respective interpolator filters.  Thus only
  // after both I and Q are written will the TxDAC outputs follow the
  // written values.
```

```
// Write values to Port A, first I, then Q
// data_state == 0 => I
// data_state == 1 => Q

// Setup write to I channels
if(data_state == 0){
  // Channel 0 selected
  if(CHANSEL_val_next == CH0){
    value = (Icurrent0|Imask) << 1;
    PESL( GPIOA, GPIO_WRITE_DATA, ~value );
    data_state = 0;
    // Drive CHANSEL low
    CHSEL_PutVal(LOW);
    CHANSEL_val_next = CH1;
    // If new values have been computed in main routine,
    // clear the new value flag and set the update flag
    // to indicate I ch. 0 has been updated
    if(iqnew == TRUE){
      iqnew = FALSE;
      iqupdate = I0_UPDATED;
    }

    // If Q ch. 1 has been updated (would have become active
    // on the previous rising edge of clock), then all four
    // I/Q values have been written to the DAC internal
    // registers.
    if(iqupdate == Q1_UPDATED)
      iqupdate = TRUE;
  }
  // Channel 1 selected
  else {
    value = (Icurrent1|Imask) << 1;
    PESL( GPIOA, GPIO_WRITE_DATA,  ~value );
    // Next pass will write to Q values
    data_state = 1;
    // Drive CHANSEL high
    CHSEL_PutVal(HIGH);
    CHANSEL_val_next = CH0;
    // If I ch. 0 has been updated, then set flag to
    // indicate that I chan 1. has been updated.
    if(iqupdate == I0_UPDATED)
      iqupdate = I1_UPDATED;
  }
}
```

```
  // Setup write to Q channels
  else{
    // Channel 0 selected
    if(CHANSEL_val_next == CH0){
      value = (Qcurrent0&Qmask) << 1;
      PESL( GPIOA, GPIO_WRITE_DATA, ~value );
      data_state = 1;
      // Drive CHANSEL low
      CHSEL_PutVal(LOW);
      CHANSEL_val_next = CH1;

      // If I ch. 1 has been updated, then set flag to
      // indicate that Q chan. 0 has been updated
      if(iqupdate == I1_UPDATED)
        iqupdate = Q0_UPDATED;
    }

    // Channel 1 selected
    else {
      value = (Qcurrent1&Qmask) << 1;
      PESL( GPIOA, GPIO_WRITE_DATA, ~value );
      data_state = 0;
      // Drive CHANSEL high
      CHSEL_PutVal(HIGH);
      CHANSEL_val_next = CH0;

      // If Q ch. 0 has been updated, set flag to indicate that
      // Q ch. 1 has been updated
      if(iqupdate == Q0_UPDATED)
        iqupdate = Q1_UPDATED;
    }
  }
  // Clear the PWM reload flag, enables the PWM to generate
  // a new interrupt.
  PESL( PWMA, PWM_CLEAR_RELOAD_FLAG, NULL );
}


/* END Events */

/* **
###################################################################
** **     This file was created by UNIS Processor Expert 03.32 for
```

```
**     the Motorola 56800 series of microcontrollers. ** **
#########################################################################
*/
```

Appendix C

Matlab Code for Magnitude-Only Multi-Slice Model Extraction

This section contains the Matlab routines necessary for extracting the model parameters and generating the model output for the magnitude-only based multi-slice model presented in Chapter 6. Comments within this software details the extraction procedure and the input data requirements.

C.1 Main Script

```
% This script computes the model of the complex data taken for
% the RFMD 2486 LNA and compares this to the actual measured
% data.   The measurements consist of complex a1 and b2 from
% a single tone AM-AM, AM-PM test on the 8510, and magnitude
% response measurements for a two-tone test taken with a
% spectrum analyzer.
%
% A series of scripts and functions are called
% to extract the pre and post-filter responses and the
% polynomial coefficients for the memoryless nonlinearity.  Then
% the modelled responses at the fundamental and the IM3L are
% computed and compared to the measured data.  A modification to
```

```
% the IM3 model is applied to the IM3H and IM3L models to
% account for the IM3 asymmetry.  It is believed at this time that
% the asymmetry is a function of multiple second order nonlinearites
% in sequence, thus producing a baseband component that is
% translated (through an ideal multiplication with the stimulus)
% to the IM3 tones.  The correction is additive for the IM3L
% term and subtractive for the IM3H term which was deduced by
% looking at the measured data.  There is also support for this
% type of correction based on the sign of the frequency component
% at the two different IM3 tones (postive for IM3L, and negative
% for IM3H).  The correction is based on  half of the measured
% difference between the IM3 components and it is assumed that
% the first slice AM-AM model falls exactly half-way between the
% two measured tones.  This is somewhat supported by the data,
% so any discrepancies will be explained by the failure of the
% AM-AM model to follow this type of behavior.

% First perform the filter coefficient extraction.  This is done
% at the small-signal point, here taken as -15 dBm input power
% in both the single-tone and two-tone cases.  Extracted filter
% responses in dB will be Hsm15 and Ksm15.

firstpassextract2

% Now compute the polynomial coefficients from the AM-AM data.  To attempt
% a better fit of IM3 products, first estimate the third order
% coefficient, from the two-tone measurements then subtract its
% contribution from the output of the single tone data, then fit the
% remaining data with a nth-order polynomial.

% Compute the magnitude gain from the complex a1, b2 data at 2 GHz.  This
% frequency is chosen since the minimum pre-filter attenuation occurs here
% and thus compression effects are more evident resulting in a higher
% quality polynomial model.
Gab_mag = abs((b2_real_mat(1,:)+sqrt(-1)*b2_imag_mat(1,:))./ ...
             (a1_real_mat(1,:)+sqrt(-1)*a1_imag_mat(1,:)));

% Estimate third-order coefficient from -15 dBm data
im3l_rfmdg_m15_corr = im3lcorr(im3l_rfmdg_m15(:,2), f1_ref(:,2),
...
                               f2_ref(:,2), f1_ref_nom, f2_ref_nom);
im3l_rfmdg_m15_corr_sub = im3l_rfmdg_m15_corr(1:(fup-flo)/n:end);
atten = 20.5; a3m15h = thirdgainextract(im3l_rfmdg_m15_corr_sub,
-15, ...
```

```
                          -15, Hsm15, Ksm15, atten);

% The extracted third-order coefficient is a3h so need to add effect of
% filters to get coefficient for one tone data since filters are not
% extracted.  For one tone instantaneous model, a3 needs to be converted to
% b3.
a3m15 = a3m15h +3*Hsm15(1) + Ksm15(1); b3m15 = a3m15 +
20*log10(3/4);

n = 7;      % Order of polynomial to extract
[coefa coefb] = amampolyfit3(20*log10(abs(a1_cplx_mat(1,:))), ...
                             Gab_mag, b3m15, n);

% Extract memoryless nonlinearity polynomial coefficients
coefah = mlesscoefextract(coefa, Hsm15(1), Ksm15(1), n);
coefah(end-3) = coefah(end-3)*(1);

% Find the difference between the measured IM3L and IM3H at a single
% frequency and compute the change to the polynomial coefficients
% for the IM3H tone.

im3_meas_diff_mat = []; im3_meas_sum_mat = [];

for A1 = -15:1:-1,
    n = 50;
    powstr1 = num2str(abs(A1));
    % Sample corrected IM3L output from two-tone data set.  Data is
    % corrected with third-order assumption and reference data at -20 dBm
    % input power.  Not an ideal correction, but no real other assumption
    % at hand.
    eval(['im3l_rfmdg_m' powstr1 '_corr = im3lcorr(im3l_rfmdg_m' powstr1...
          '(:,2), f1_ref(:,2), f2_ref(:,2), f1_ref_nom, f2_ref_nom);']);
    eval(['im3l_rfmdg_m' powstr1 '_corr_sub = im3l_rfmdg_m' powstr1 ...
          '_corr(1:(fup-flo)/n:end);'])
    eval(['im3h_rfmdg_m' powstr1 '_corr = im3lcorr(im3h_rfmdg_m' powstr1...
          '(:,2), f1_ref(:,2), f2_ref(:,2), f1_ref_nom, f2_ref_nom);']);
    eval(['im3h_rfmdg_m' powstr1 '_corr_sub = im3h_rfmdg_m' powstr1 ...
          '_corr(1:(fup-flo)/n:end);'])

    eval(['im3lv = sqrt(10.^((im3l_rfmdg_m' powstr1 ...
          '_corr_sub + atten)/10)*.1);'])
    eval(['im3hv = sqrt(10.^((im3h_rfmdg_m' powstr1 ...
          '_corr_sub + atten)/10)*.1);'])
    im3_meas_diff_mat = [im3_meas_diff_mat im3lv - im3hv];
```

```
    im3_meas_sum_mat = [im3_meas_sum_mat im3lv + im3hv];
end

findex = 25;        % Frequency index for extraction of even order fit
                    % with modifications to measured IM3 difference
p15 = -15:1:-1;     % Must use low resolution power since this scale is
                    % that which the IM3 components were measured
[coefah2 coefbeta] = im3difffit(p15, im3_meas_diff_mat(findex,:),
...
                                Hsm15(findex), Ksm15(findex));
coefah2(end-2) = coefah2(end-2);

% Compute the response of the three-box model with the filter responses
% Hs, Ks, and memoryless nonlinearity polynomial coefficients coefah

delta = 0.05;
n = 50;         % Number of frequency points in one-tone data
atten = 20.5;
f1_ref_nom = -20;   % Nominal power of f1 reference data
f2_ref_nom = -20;   % Nominal power of f2 reference data

% Create matrices for holding series of data sets
im3lmeas_sub = []; im3hmeas_sub = []; im3lfit_mat = [];
im3hfit_mat = []; f1fit_mat = []; f1fit2_mat = [];

im3l_modl_diff_mat = []; im3h_modl_diff_mat = []; im3lv_mat = [];
im3hv_mat = []; im3mlv_mat = []; im3mhv_mat = [];


for A1 = -15:1:-1,
    powstr1 = num2str(abs(A1));
%     Hs1 = zeros(size(Hsm15));
%     Ks1 = zeros(size(Ksm15));
    eval(['[f1fitm' powstr1 ', f1fit2m' powstr1 ', im3lfitm' powstr1 ...
          ', im3hfitm' powstr1 ...
          '] = polymodel4(coefah, coefah2, Hsm15, Ksm15, -' powstr1 ...
          ', -' powstr1 ');'])
    eval(['im3lfit_mat = [im3lfit_mat im3lfitm' powstr1 '];'])
    eval(['im3hfit_mat = [im3hfit_mat im3hfitm' powstr1 '];'])
    eval(['f1fit_mat = [f1fit_mat f1fitm' powstr1 '];'])
    eval(['f1fit2_mat = [f1fit2_mat f1fit2m' powstr1 '];'])

    % Now correct b2 data from single-tone with a1 data
    [val I] = find( (p401 >= (A1 - delta)) & (p401 <= (A1 + delta)));
```

```
    eval(['b2_m' powstr1 ...
          '_corr = f1corr(20*log10(abs(b2_cplx_mat(:,I(1)))),' ...
          '20*log10(abs(a1_cplx_mat(:,I(1)))), A1);'])

    % Sample corrected fundamental output from two-tone data set.
    % Reference data from correction is only available at -20 dBm.  Thus
    % use f1_ref_nom in correction function.
    eval(['f1_rfmdg_m' powstr1 '_corr = f1corr(f1_rfmdg_m' powstr1 ...
          '(:,2), f1_ref(:,2), f1_ref_nom);'])
    eval(['f1_rfmdg_m' powstr1 '_corr_sub = f1_rfmdg_m' powstr1 ...
          '_corr(1:(fup-flo)/n:end);'])
    % eval(['f1_rfmdg_m' powstr1 '_corr_sub = f1_rfmdg_m' powstr1
    % '(1:(fup-flo)/n:end,2);'])


    % Compute difference between measured upper and lower and model with
    % upper and lower IM3
    eval(['im3lmeas_sub = [im3lmeas_sub im3l_rfmdg_m' powstr1 ...
          '_corr_sub + atten];'])
    eval(['im3hmeas_sub = [im3hmeas_sub im3h_rfmdg_m' powstr1 ...
          '_corr_sub + atten];'])
    eval(['im3lv = sqrt(10.^((im3l_rfmdg_m' powstr1 ...
          '_corr_sub + atten)/10)*.1);'])
    eval(['im3hv = sqrt(10.^((im3h_rfmdg_m' powstr1 ...
          '_corr_sub + atten)/10)*.1);'])
%     im3_meas_diff_mat = [im3_meas_diff_mat im3lv - im3hv];
%     im3_meas_sum_mat = [im3_meas_sum_mat im3lv + im3hv];
    eval(['im3mlv = sqrt(10.^(im3lfitm' powstr1 '/10)*.1);'])
    eval(['im3mhv = sqrt(10.^(im3hfitm' powstr1 '/10)*.1);'])
    im3l_modl_diff_mat = [im3l_modl_diff_mat im3lv - im3mv];
    im3h_modl_diff_mat = [im3h_modl_diff_mat im3hv - im3mv];
    im3lv_mat = [im3lv_mat im3lv];
    im3hv_mat = [im3hv_mat im3hv];
    im3mlv_mat = [im3mlv_mat im3mlv];
    im3mhv_mat = [im3mhv_mat im3mhv];

    % Plot the fundamental model response versus the measured data,
    % one-tone
%     figure
%     eval(['plot(f50, [b2_m' powstr1 '_corr(:,1), f1fitm' powstr1 '])'])
%     tstr1 = 'Fundamental response for single tone test at -';
%     tstr2 = ' dBm';
%     ftitle([tstr1 powstr1 tstr2])
%     legend('Measured', 'Model')
```

```
%     xlabel('Frequency (MHz)')
%     ylabel('Tone Magnitude (dBm)')

    % Plot the fundamental model response versus the measured data,
    % two-tone
%     figure
%     eval(['plot(f50, [f1_rfmdg_m' powstr1 '_corr_sub + atten, f1fit2m'...
%     powstr1 '])'])
%     tstr3 = 'Fundamental response for two-tone test at -';
%     ftitle([tstr3 powstr1 tstr2])
%     legend('Measured', 'Model')
%     xlabel('Frequency (MHz)')
%     ylabel('Tone Magnitude (dBm)')

    % Plot the IM3L/IM3H model response versus the measured data, two-tone
%     figure
%     eval(['plot(f50, [im3l_rfmdg_m' powstr1 ...
%           '_corr_sub + atten, im3h_rfmdg_m' powstr1 ...
%           '_corr_sub + atten, im3lfitm' powstr1 ...
%           ', im3hfitm' powstr1 '])'])
%     tstr4 = 'IM3 response for two-tone test at -';
%     ftitle([tstr4 powstr1 tstr2])
%     legend('Measured IM3L', 'Measured IM3H', 'IM3L Model', 'IM3H Model')
%     xlabel('Frequency (MHz)')
%     ylabel('Tone Magnitude (dBm)')

end

% Plot im3l model response Pin vs Pout @ single frequency
% f_ind = 5;
% figure
% p15 = -15:1:-1;
% plot(p15, im3lfit_mat(f_ind,:))
% ftitle('IM3 Model Response at 3 GHz')
% xlabel('P_{in} (dBm)')
% ylabel('P_{out} (dBm)')

% Plot im3 model difference from measured im3l
% figure
% p15 = -15:1:-1;
% v15 = sqrt(10.^(p15/10)*.1);
% vim3l = im3l_modl_diff_mat(f_ind,:);
% vim3h = im3h_modl_diff_mat(f_ind,:);
% plot(v15, [vim3l;vim3h])
```

```
% xlabel('V_{in} (V)')
% ylabel('V_{MEASout} - V_{MODLout} (V)')
% ftitle('Voltage difference between measured IM3 and model IM3 at 3 GHz')
% legend('V_{IM3L}','V_{IM3H}',3)

% Plot im3 model difference from measured im3h
% figure
% p15 = -15:1:-1;
% v15 = sqrt(10.^(p15/10)*.1);
% vim3l = im3h_modl_diff_mat(25,:);
% plot(v15, vim3l)
% xlabel('V_{in} (V)')
% ylabel('V_{MEASout} - V_{MODLout} (V)')
% ftitle('Voltage difference between measured IM3H and model IM3 at 3 GHz')

% Plot im3 model difference from average measured im3
% figure
% p15 = -15:1:-1;
% v15 = sqrt(10.^(p15/10)*.1);
% vim3l=sqrt(10.^(im3lfit_mat(f_ind,:)/10)*.1)-im3_meas_sum_mat(f_ind,:)/2;
% plot(v15, vim3l)
% xlabel('V_{in} (V)')
% ylabel('V_{MODLout} - V_{AVGMEASout} (V)')
% ftitle(['Voltage difference between model IM3 and average measured' ...
%         'IM3 at 3 GHz'])

findex = 25; figure plot(v15, im3lv_mat(findex,:), v15,
im3hv_mat(findex,:), v15, ...
     im3mlv_mat(findex,:),'-+', v15, im3mhv_mat(findex,:),'-+')

figure plot(p15, im3lmeas_sub(findex,:),'-+', p15,
im3hmeas_sub(findex,:), ...
     '-x',p15, im3lfit_mat(findex,:), p15, im3hfit_mat(findex,:),'m-.',...
     'LineWidth',1.5 )
```

C.2 Correction of Measured IM3 Magnitude for Source Flatness

```
function corr = im3lcorr(im3l_meas, f1_ref, f2_ref, f1_ref_nom,
f2_ref_nom)
%
```

```
% This function returns the corrected IM3 power measurment based on the
% measured IM3 component and the correction factor calculated from the
% reference fundamental data.  The correction factor is taken to be a
% multiplicative percantage factor in the linear domain such that the
% logarithmic domain factor is additive.  This is likely a good estimate of
% the accuracy of the power sources since they are probably specified as
% a percentage of the true setting and not an absolute deviation for a
% given power.
%
% corr = im3lcorr(im3l_meas, f1_ref, f2_ref, f1_ref_nom, f2_ref_nom)
%
% im3l_meas - measured frequency domain of IM3 component at a single input
% power f1_ref - measured fundamental tone f1 at -20 dBm with no DUT
% attached
% f1_ref_nom - power setting of RF source for f1_ref data
% f2_ref - measured fundamental tone f2 at -20 dBm with no DUT attached
% f2_ref_nom - power setting of RF source for f2_ref data
% corr - power corrected IM3L data, compensated for source deviation from
% flatness, third order estimate only

% Calculate correction factors
f1_corr_fact = f1_ref_nom - f1_ref; f2_corr_fact = f2_ref_nom -
f2_ref;

% Apply to measured IM3
corr = im3l_meas + 2*f1_corr_fact + f2_corr_fact;
```

C.3 Extraction Script for Linear Networks of First Slice

```
% This script uses the small-signal extraction functions to
% obtain the three-box model parameters from complex and
% magnitude measurements from single-tone and two-tone tests.

% Since the single tone and two-tone data are of different lengths
% in frequency, need to take sub-sets of data to equalize.

n = 50;          % Number of frequency points in one-tone data
A1_1 = -15;      % Nominal power level of input tone for one tone data
A1_2 = -15;      % Nominal power level of f1 input tone for two tone data
A2_2 = -15;      % Nominal power level of f2 input tone for two tone data
f1_ref_nom = -20;   % Nominal power of f1 reference data
f2_ref_nom = -20;   % Nominal power of f2 reference data
atten = 0;     % Attenuation in measurement system of f1 one tone data
```

```
delta = 0.05;
fup = 4000;     % Upper frequency in MHz
flo = 2000;     % Lower frequency in MHz

% First correct the IM3L data for power flatness
powstr1 = num2str(abs(A1_1)); powstr2 = num2str(abs(A1_2));
eval(['im3l_rfmdg_m' powstr2 '_corr = im3lcorr(im3l_rfmdg_m'
powstr2 ...
      '(:,2), f1_ref(:,2), f2_ref(:,2), f1_ref_nom, f2_ref_nom);']);

% Smooth IM3 data
% eval(['im3l_smooth = ravgfilt(transpose(im3l_rfmdg_m' powstr2 ...
%       '_corr), 1);'])
% im3l_smooth = im3l_smooth';


% Now correct b2 data from single-tone with a1 data
[val I] = find( (p401 >= (A1_1 - delta)) & (p401 <= (A1_1 +
delta))); eval(['b2_m' powstr1 '_corr =
f1corr(20*log10(abs(b2_cplx_mat(:,I(1))))'...
      ', 20*log10(abs(a1_cplx_mat(:,I(1)))), A1_1);'])

% Take samples of IM3L data
eval(['im3l_rfmdg_m' powstr2 '_corr_sub = im3l_rfmdg_m' powstr2
...
      '_corr(1:(fup-flo)/n:end);'])
% eval(['im3l_rfmdg_m' powstr2 ...
%       '_corr_sub = im3l_smooth(1:(fup-flo)/n:end);'])


% Extract H(s)
eval(['f1_onetone = b2_m' powstr1 '_corr;']) [Hs, Hsx2] =
eval(['Hsextract(im3l_rfmdg_m' powstr2 ...
                  '_corr_sub, f1_onetone);']);
eval(['Hsm' powstr1 '= Hs;']); eval(['Hsx2m' powstr1 '= Hsx2;']);

% Extract K(s)
eval(['Ks = Ksextract(Hs, f1_onetone);']); eval(['Ksm' powstr1
'=Ks;']);

% Plot extracted filter responses
f50 = flo:(fup-flo)/n:fup-(fup-flo)/n; figure subplot(2,1,1)
plot(f50, Hs) ftitle('Pre/Post-filter Frequency Response')
xlabel('Frequency (MHz)') ylabel('Normalized Magnitude (dB)')
```

```
subplot(2,1,2) plot(f50, Ks) xlabel('Frequency (MHz)')
ylabel('Normalized Magnitude (dB)')
```

C.4 Pre-Linear Network Extraction Function

```
function [Hs, Hsx2] = Hsextract(im3l_corr, f1_corr)
%
% This function uses the small-signal extraction procedure to obtain
% the frequency response of the front end filter.  The response is
% normalized to its highest measured response and assumed to have
% no loss at that point.  The main assumption here is that the
% frequencies in the two-tone response and the one-tone response
% are close enough in frequency that they can be taken as equal.
%
% [Hs Hsx2] = Hsextract(im3l_corr, f1_corr)
%
% Hs - extracted magnitude of frequency response of pre-filter (dB)
% Hsx2 - extracted square magnitude of frequency response (dB)
% im3l_corr - flatness corrected IM3L measurement
% f1_corr - flatness corrected f1 measurement
%

% The linear equation used in the extraction is
%
% Vim3/Vf1 ~ (a3*V1^2*V2*|H(s)|^3*|K(s)|)/(a1*V1*|H(s)|*|K(s)|
%
% where the LHS ratio is a ratio of output average power (assume for
% now that the spectrum analyzer and VNA measures rms power, not peak power
% thus the voltage from v^2/R will be the rms voltage.
%
% The measured power data will need to be convered to rms voltage prior
% to taking the ratio.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Voltage-based formulation

% Convert output powers to peak voltage, system impedance is 50 Ohms
R = 50;

Pf1_lin = 10.^(f1_corr/10)*1e-3; Pim3l_lin =
```

```
10.^(im3l_corr/10)*1e-3;

Vf1_lin = sqrt(Pf1_lin * R * 2); Vim3l_lin = sqrt(Pim3l_lin * R *
2);

% Compute unormalized frequency response
Hs2_lin = Vim3l_lin./Vf1_lin;

% Normalize to the maximum value
% Hs_lin = sqrt(Hs2_lin);
Hs_lin = Hs2_lin.^(1/(3.0-1)); Hs_lin_norm = Hs_lin/max(Hs_lin);
% Hs_lin_norm = Hs_lin/Hs_lin(1);

% Return dB value of voltage transfer function
Hs = 20*log10(Hs_lin_norm); Hsx2 = Hs*2;
```

C.5 Post-Linear Network Extraction Function

```
function Ks = Ksextract(Hs, f1_corr)
%
% This function extracts the post filter of the three-box
% model according to the small-signal routine.  The computed
% pre-filter response is used under the assumption that the
% frequencies of K(s) and H(s) are the same for each data set.
% Here this is an exact equality if the H(s) is taken as the
% frequency of the fundamental f1.
%
% Ks - extracted magnitude of frequency response of post-filter
% Hs - extracted magnitude of frequency response of pre-filter
% f1_corr - flatness corrected measured f1 response from one-tone data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Voltage-based formulation

% Convert output powers to peak voltage, system impedance is 50 Ohms
R = 50;

Pf1_lin = 10.^(f1_corr/10)*1e-3; Hs_lin = 10.^(Hs/20);

Vf1_lin = sqrt(Pf1_lin * R * 2);

Ks_lin = Vf1_lin./Hs_lin;
```

```
% Normalize Ks for lossless at maximum
Ks_lin_norm = Ks_lin/max(Ks_lin);
 %Ks_lin_norm = Ks_lin/Ks_lin(1);

% Return dB value of voltage transfer function
Ks = 20*log10(Ks_lin_norm);
```

C.6 Third-Order Coefficient Extraction from Measured Two-Tone Response

```
function a3 = thirdgainextract(im3l_corr, A1, A2, Hs, Ks, atten)
%
% This function uses the small-signal extraction procedure results
% to compute the third order gain coefficient of the amplifier under
% test.
%
% a3 = thirdgainextract(im3l_corr, A1, A2, Hs, Ks, atten)
%
% a3 - computed third order gain coefficient
% im3l_corr - two tone flatness corrected IM3L
% A1 - power level of input tone (nominal, single value)
% A2 - power level of input tone (nominal, single value)
% Hs - extracted pre-filter normalized frequency response
% Ks - extracted post-filter normalize frequency response
% atten - attenuation of IM3L prior to measurement in system

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Voltage-based formulation

% Convert output powers to peak voltage, system impedance is 50 Ohms
R = 50;

Pim3l_lin = 10.^((im3l_corr+atten)/10)*1e-3; A1_lin =
10^(A1/10)*1e-3; A2_lin = 10^(A2/10)*1e-3;

Vim3l_lin = sqrt(Pim3l_lin * R * 2); Vin1_lin = sqrt(A1_lin * R *
2); Vin2_lin = sqrt(A2_lin * R * 2);

Hs_lin = 10.^(Hs/20); Ks_lin = 10.^(Ks/20);
```

```
a3_lin = Vim3l_lin./(.75 * Vin1_lin^2 * Vin2_lin * (Hs_lin).^3 .*
Ks_lin);

% Return dB form of voltage gain
a3 = 20*log10(a3_lin);
```

C.7 Modified Polynomial Fit Function

```
function [coefa, coefb] = amampolyfit3(pin, G, b3, n)
%
% This function fits a polynomial with real coefficients
% to the real gain G.   Before the fit, the precomputed
% response of the third-order component is removed from the
% output and the polynomial is fit to the result.  The input power
% and polynomial order is specified by the user.  Both the raw polynomial
% coefficients, coefb, and the coefficients for the instantaneous
% nonlinearity model are output.
%
% [coefa, coefb] = amampolyfit2(pin, G,a3, n)
%
% pin - input power, (dBm)
% G - complex gain, b2/a1
% b3 - previously computed third-order coefficient
% n - order of polynomial to fit to gain
% coefa - instantaneous model coefficients
% coefb - raw polynomial fit coefficients
%

% Convert input power to peak voltage, system impedance is 50 Ohms
R = 50;

pin_lin = 10.^((pin)/10)*1e-3; vin_lin = sqrt(pin_lin*R*2);

% Assume third-order is compressive
b3_lin = -10.^(b3/20);

% Compute output voltage with complex gain
vout_lin = vin_lin.*G;

% Remove third-order component
vout_lin_m = vout_lin - b3_lin(1)*vin_lin.^3;

% Generate bipolar voltage vectors
```

```
vin_lin_b = [fliplr(-vin_lin) 0 vin_lin]; vout_lin_b =
[fliplr(-vout_lin) 0 vout_lin]; vout_lin_bm = [fliplr(-vout_lin_m)
0 vout_lin_m];

% Generate raw gain polynomial fit
coefb = polyfit(vin_lin_b, vout_lin_bm, n); vout_lin_mod =
polyval(coefb, vin_lin_b); coeftest = coefb;
coeftest(end - 3) = b3_lin(1);% + coeftest(end-3);
vout_lin_test = polyval(coeftest, vin_lin_b);

coefb = coeftest;

figure plot(vin_lin_b, [vout_lin_bm; vout_lin_b; vout_lin_mod;
vout_lin_test]) legend('w/o 3^{rd} order', 'w 3^{rd} order', ...
       'Model w/o 3^{rd}','Model Test',2)

% Generate instantaneous model polynomial coefficients
coefa = zeros(size(coefb)); coefa(n) = coefb(n); for j = 1:2:n
    alpha = (n+1)/2 - (j+1)/2;
    alpha2 = alpha*2;
    coefa(j) = coefb(j)*2^alpha2*factorial(alpha)*factorial(1+alpha)/ ...
               factorial(1+alpha2);
end

% polycoefplot(coefb, vin_lin);
% polycoefplot(coeftest, vin_lin);
```

C.8 Memoryless Polynomial Coefficient Extraction Function

```
function coefah = mlesscoefextract(coefa, Hs, Ks, n)
%
% This function takes the polynomial coefficients found from
% a fit to single tone AM-AM data and de-embeds the coefficients
% for a true memoryless coefficient by using the filter
% responses found in the previous extraction effort.
%
% coefah = mlesscoefextract(coefa, Hs, Ks)
%
% coefa - polynomial coefficients in voltage form
% Hs - pre-filter magnitude response at a single frequency in dB form
% Ks - post-filter magnitude response at a single frequency in dB form
% coefah - memoryless coefficients
%
```

```
% Convert filter responses to linear voltage form

Hs_lin = 10^(Hs/20); Ks_lin = 10^(Ks/20);

% Compute coefah
Hs_lin_rep = Hs_lin*ones(1,n+1); Hs_lin_v = vander(Hs_lin_rep);
coefah = coefa./(Hs_lin_v(end,:)*Ks_lin);
```

C.9 Baseband Contribution Extraction Function

```
function [coefah2, coefbeta] = im3difffit(pin, im3_diff, Hs, Ks)
%
% This function computes the modified memoryless coefficients for
% the even-ordered polynomial that models the IM3 asymmetry.  It
% is believed that the difference in the two IM3 components is a
% result of differences in the translation of the baseband component
% to the IM3 tones.
%
% [coefahp, coefbeta] = im3difffit(pin, im3_diff, Hs, Ks)
%
% pin - input power, real (dBm)
% im3_diff - difference in IM3 components as a function of voltage
%            at a single frequency, vIM3L - vIM3H (V)
% Hs - pre-filter response at a specific frequency (in dB form)
% Ks - post-filter response at a specific frequency (in dB form)
% coefah2 - memory-less coefficients of even-order second-slice
% nonlinearity
% coefbeta - raw even-order coefficients of polynomial fit
%

% Convert filter responses to linear form

Hs_lin = 10^(Hs/20); Ks_lin = 10^(Ks/20);

% Convert input power to peak voltage, system impedance is 50 Ohms
R = 50;

pin_lin = 10.^(pin/10)*1e-3; vin_lin = sqrt(pin_lin*R*2);

% Only half of the difference will be applied to the AM-AM model
% to predict each tone magnitude
```

```
im3diff_half = im3_diff/2;

% Back out the effect of the post-filter response
im3diff_z = im3diff_half/Ks_lin;

% Back out the effect of the multiplication by the input stimulus
% both tones are equal power so the effect is the same for both
% tones
im3diff_u = im3diff_z./vin_lin;

% Now fit this term with an even order-polynomial, choose
% fourth order
% Find coefficients of polynomial to fit voltage difference, second
% order and higher only, no DC or linear components
n = 4; coefbeta = polyfitmod([-fliplr(vin_lin) 0 vin_lin], ...
           [fliplr(im3diff_u) 0 im3diff_u],n);
coefbeta = [coefbeta 0 0];

% Now use the polynomial fit and the theoretical baseband response to
% compute the instantaneous even-order coefficients.  Backing out
% the pre-filter and the coefficient for the number of frequency
% combinations that form the baseband component.
coefah2 = zeros(size(coefbeta)); coefah2(end - 2) = coefbeta(end -
2) / Hs_lin^2; coefah2(end - 4) = coefbeta(end - 4) /(2 *
Hs_lin^4);

figure plot(vin_lin, im3_diff) ftitle('IM3 voltage difference')

vim3difffit = polyval(coefbeta, vin_lin); figure plot(vin_lin,
[im3diff_u; vim3difffit]) xlabel('V_{in} (V)') ylabel('V_{IM3Lout}
- V_{IM3Hout} (V)') ftitle('Voltage difference between measured
IM3L, IM3H at 3 GHz') figure plot([-fliplr(vin_lin) 0 vin_lin],
...
     polyval(coefbeta, [-fliplr(vin_lin) 0 vin_lin]))
ftitle('Fit')
```

C.10 Model Output Generation Function

```
function [f1fit, f1fit2, im3lfit, im3hfit] =
                                polymodel2(coefah, coefah2, Hs, Ks, P1, P2)
%
% This function takes the extracted memoryless polynomical
```

```
% coefficients, filter responses, and the input powers of
% the stimulus and prepares a single tone fundamental response,
% two-tone fundamental response, and a two-tone IM3L/IM3H response.
%
% [f1fit, f1fit2, im3lfit, im3hfit] =
%                                 polymodel(coefah, coefahp, Hs, Ks, P1, P2)
%
% coefah - memoryless polynomial coefficients for first slice, AM-AM fit
% coefah2 - memoryless polynomial coefficients for second slice,
%           baseband component fit
% Hs - pre-filter amplitude response, dB form
% Ks - post-fitler amplitude response, dB form
% P1 - input power of first tone, dBm
% P2 - input power of second tone, dBm
% f1fit - model response for single tone input at fundamental, dBm
% f1fit2 - model response for two-tone input at fundamental, dBm
% im3lfit - model response for two-tone input at 2f1-f2, dBm
% im3hfit - model response for two-tone input at 2f2-f1, dBm
%

% Convert input powers to peak voltage, R = 50 Ohms
R = 50;

P1_lin = 10.^((P1)/10)*1e-3; P2_lin = 10.^((P2)/10)*1e-3;
P1IM3_lin = 10.^((P1-1)/10)*1e-3;

v1_lin = sqrt(P1_lin*R*2); v2_lin = sqrt(P2_lin*R*2); v1IM3_lin =
sqrt(P1IM3_lin*R*2);

% Convert filter responses to voltage form
Hs_lin = 10.^(Hs/20); Ks_lin = 10.^(Ks/20);

% Compute single tone fundamental response
n = size(coefah, 2) - 1; f1fit = 0; for j = 1:2:n,
    alpha = (n+1)/2 - (j+1)/2;
    alpha2 = alpha*2;
    coef = coefah(j)*factorial(1+alpha2)/(2^alpha2*factorial(alpha) ...
           *factorial(1+alpha));
    temp = coef*v1_lin^(n+1-j)*Hs_lin.^(n+1-j).*Ks_lin;
    f1fit = f1fit + temp;
end

f1fit = 10*log10(f1fit.^2/(2*R*1e-3));
```

```
% Compute two-tone fundamental response
% Khaled's formulation of the constants for each order in the double
% summation that takes each contribution of each frquency mix
% in each order of nonlinearity to IM3 and then sums all of the
% contributions of each order

% Since the input powers of the tones were equal, only a single input
% level will be used in the computation.
f1fit2 = 0;

for k = 1:2:n,
    for l = 0:(k-1)/2,
        m = [l ((k-1)/2-l) ((k+1)/2 - l) l];
        mco = multinomial(m, k);
        temp = coefah(n-k+1)*mco*(v1_lin^k)*(Hs_lin.^k).*Ks_lin;
        f1fit2 = f1fit2 + temp;
    end
end f1fit2 = 10*log10(f1fit2.^2/(2*R*1e-3));

% Compute two-tone IM3L and IM3H response
% Khaled's formulation for the first slice AM-AM model is used.  The
% baseband component that fits the half of the difference between the
% measured IM3 tones is added the IM3L component and subtracted from the
% IM3H component.

im3lfit1 = 0; im3hfit1 = 0;
% Slice one
for k = 3:2:n,
    for l = 0:(k-3)/2,
        m = [((k-3)/2-l) (l+1) l ((k+1)/2 - l)];
        mco = multinomial(m, k);
        temp = coefah(n-k+1)*mco*v1IM3_lin^k*Hs_lin.^(k).*Ks_lin;
        im3lfit1 = im3lfit1 + temp;
        temp = coefah(n-k+1)*mco*v1IM3_lin^k*Hs_lin.^(k).*Ks_lin;
        im3hfit1 = im3hfit1 + temp;
    end
end
% Slice 2, second order
slice2_2 = coefah2(end - 2)*v1IM3_lin^3*Hs_lin.^2.*Ks_lin;
% Slice 2, fourth order
slice2_4 = 2*coefah2(end - 4)*v1IM3_lin^5*Hs_lin.^4.*Ks_lin;
% Slice 2, total output
slice2 = slice2_2 + slice2_4;
%slice2 = 0;
```

```
% Add slice 2 to slice 1 for IM3L
im3lfit_total = abs(im3lfit1) + slice2;
% Subtract slice 2 from slice 1 for IM3H
im3hfit_total = abs(im3hfit1) - slice2;

im3lfit = 10*log10(im3lfit_total.^2/(2*R*1e-3)); im3hfit =
10*log10(im3hfit_total.^2/(2*R*1e-3));
```

Appendix D

Matlab Code for Complex Multi-Slice Model Extraction

This section contains the Matlab routines necessary for extracting the model parameters and generating the model output for the vector measurement based multi-slice model presented in Chapter 7. The routines used for each of the amplifiers presented in Chapter 7 are included as specific processing functions for the measured data required individual attention. Comments within this software details the extraction procedure and the input data requirements.

D.1 Scripts for GaAs MESFET Amplifier

D.1.1 RF Source Correction Script

```
% This script calls functions to process measured DUT data
% for single tone AM-AM, AM-PM and two-tone power sweep
% experiments.  The test setup in this case is the cancellation
% bridge where three sources, which share a 10 MHz reference
% frequency, for supplying the stimulus and the cancellation
% tone.  The cancellation tone source provides for the measurement
% of cancelled tone phase by allowing a carrier frequency shift
```

```
% in 0.09 degree increments.

% Data taken for this setup is not compensated for power loss
% in the apparatus network either at the input or output.  These
% losses were characterized and the losses will be compensated
% before further processing.

% Load single tone data, data is indicated power on source
% fpath = input('Enter file name for AM-AM stimulus power file:','s');
p_ref1_ind_amam = (-20:0)'; fpath = 'mc_amam_pout_cncl2.txt';
% Power measured after cancellation
p_cncl_meas_amam = load(fpath); fpath = 'mc_amam_pout2.txt';
% Power level to be cancelled, i.e. at DUT output
p_f1_meas_amam = load(fpath); fpath = 'mc_amam_pcncl2.txt';
% Power level indicated by cancellation source
p_cncl_src_amam = load(fpath);
%input('Enter file name for AM-PM phase measurement file:','s');
fpath = 'mc_amam_phase2.txt'; phase_meas_amam = load(fpath);

% Load two tone data
p_ref1_ind_mt = (-20:0)'; p_ref2_ind_mt = (-20:0)'; fpath =
'mc_pim3l_cncl_10k.txt';
% Power measured after cancellation
p_cncl_meas_mtL = load(fpath); fpath = 'mc_pim3h_cncl_10k.txt';
% Power measured after cancellation
p_cncl_meas_mtH = load(fpath); fpath = 'mc_pim3l_10k.txt';
% Power measured before cancellation
p_im3l_meas_mt = load(fpath); fpath = 'mc_pim3h_10k.txt';
% Power measured before cancellation
p_im3h_meas_mt = load(fpath); fpath = 'mc_im3l_pcncl_10k.txt';
% Power level indicated by cancellation source
p_cncl_src_mtL = load(fpath); fpath = 'mc_im3h_pcncl_10k.txt';
% Power level indicated by cancellation source
p_cncl_src_mtH = load(fpath);

fpath = 'mc_phase_im3l_10k.txt';
% Measured IM3L phase
phase_meas_mtL = load(fpath); fpath = 'mc_phase_im3h_10k.txt';
% Measured IM3H phase
phase_meas_mtH = load(fpath);


% Correct for loss in apparatus network for single tone data
  % 3.5 dB loss between source and DUT input (measured at several powers)
```

```
  p_refin_loss = 3.5;
  % 43.5 dB loss between DUT output and SA (40 dB pad + combiner)
  p_dutout_loss = 43.5;
  % 3.5 dB loss between cancellation source and combiner input
  p_cncl_loss = 3.5;

  % Correct power at DUT input
  p_ref1_in_amam = p_ref1_ind_amam - p_refin_loss;
  %save p_amam_in.txt p_ref1_in_amam -ASCII

  % Correct power at DUT output
  p_f1_dutout_amam  = p_f1_meas_amam + p_dutout_loss;
  % Save corrected AM-AM output power
  %save p_amam_out.txt p_f1_dutout_amam -ASCII

  % Indicated power of cancellation source for AM-AM measurment
  p_cncl_comb_amam = p_cncl_src_amam;

% Correct for loss in apparatus network for two-tone data
  % Correct power for tone 1 at DUT input
  p_ref1_in_mt = p_ref1_ind_mt - p_refin_loss;
  %save p_ref1_in_mt.txt p_ref1_in_mt -ASCII

  % Correct power for tone 2 at DUT input
  p_ref2_in_mt = p_ref2_ind_mt - p_refin_loss;
  %save p_ref2_in_mt.txt p_ref2_in_mt -ASCII

  % Correct power of IM3 products at DUT output
  p_im3l_dutout_mt = p_im3l_meas_mt + p_dutout_loss;
  p_im3h_dutout_mt = p_im3h_meas_mt + p_dutout_loss;
  %save p_im3l_out_10k.txt p_im3l_dutout_mt -ASCII
  %save p_im3h_out_10k.txt p_im3h_dutout_mt -ASCII

  % Indicated power of cancellation source for IM3L
  p_cncl_comb_mtL = p_cncl_src_mtL;
  % Indicated power of cancellation source for IM3H
  p_cncl_comb_mtH = p_cncl_src_mtH;

  % Compute cancellation levels for IM3 and AM-AM measurements
  p_im3l_cancellation = p_cncl_meas_mtL - p_im3l_meas_mt;
  p_im3h_cancellation = p_cncl_meas_mtH - p_im3h_meas_mt;
  p_amam_cancellation = p_cncl_meas_amam - p_f1_meas_amam;

% Correct for phase shifts introduced by attenuator steps.  This is a
```

```
% simple subtraction of the affect of the attenuators on the output phase
% from the signal generators.  It is only necessary to correct for the
% phase jumps produced by the cancellation generator for the two-tone tests
% since the stimulus generators are always used in attenuator lock mode and
% do not have the attenuator step affects.  In the single tone
% measurements, both the stimulus generator and the cancellation generator
% use attenuator lock and therefore do not need to be corrected.  The range
% of output power for which attenuator lock can be used is ~28 dB.

  % Phase correction will be performed in the time basis, i.e. normalized
  % phase shift with frequency, phi/omega.  In order to correct for the
  % attenuator shift, need to do the following:
  %     - determine the attenuator band that the reference power was
  %     measured in
  %     - segment the phase data by power to match the attenuator bands
  %     - look up the attenuator phase shifts in the calibration data
  %     - adjust entire power segments by the appropriate shift for
  %     cumulative effect of attenuator shifts

  % Load calibration data for size of attenuator steps at each of six
  % points in power sweep from -70 dBm to +13 dBm.  Data is in frequency
  % (row) vs attenuator step (col).  There are six steps at different power
  % outputs. format with the following indices.
  %     - f = 400e6:(2400e6-400e6)/200:2400e6;
  %     - pstep = [-58 -47 -36 -25 -14 -3];
  %     ***Note these steps are the powers
  %     where they occur when using the
  %     knob to control power, not arrows
  %     the difference is 1 dB in the switch
  %     points.
  atten_tstep = load('C:\Experiments\jan_05\tues_25\atten_phase_step.txt');

  % Cal data indices
  f = 400e6:(2400e6-400e6)/200:2400e6;
  % !!!! This was changed since the MC data fell right at the switch points
  pstep = [-59 -48 -37 -26 -15 -4];

  % Generate interpolated vector of step sizes for frequency of operation
  f_op_IM3L = 450e6 - 10e3;
  f_op_IM3H = 450e6 + 2*10e3;
  atten_tstep_mtL = interp1(f,atten_tstep,f_op_IM3L);
  atten_tstep_mtH = interp1(f,atten_tstep,f_op_IM3H);

  % Determine attenuator band in which the reference power was taken.  All
```

```
% attenuator effects are based on this power representing zero phase for
% the cancellation source.  The reference power for Mini-circuits
% amplifier is -10 dBm indicated.  The index of this point will be found
% and used to find the corresponding IM3 cancellation power index.
p_ind_ref = -10;
ref_index = find(p_ref1_ind_mt == -10);
p_cncl_amam_ref = p_cncl_comb_amam(ref_index);
p_im3l_ref = p_cncl_comb_mtL(ref_index);
p_im3h_ref = p_cncl_comb_mtH(ref_index);

% Assume reference is not in band 0, this step places the reference
% position in one of the attenuator bands 1-6.
atten_bnd_im3l = floor(interp1(pstep,1:6,p_im3l_ref));
atten_bnd_im3h = floor(interp1(pstep,1:6,p_im3h_ref));

% Compute the cancellation power outputs that fall within each attenuator
% band for each of the IM3 products
I_atten0_mtL = find((p_cncl_comb_mtL < pstep(1)));
I_atten1_mtL = find((p_cncl_comb_mtL >=  pstep(1)) & ...
                    (p_cncl_comb_mtL <pstep(2)));
I_atten2_mtL = find((p_cncl_comb_mtL >=  pstep(2)) & ...
                    (p_cncl_comb_mtL <pstep(3)));
I_atten3_mtL = find((p_cncl_comb_mtL >=  pstep(3)) & ...
                    (p_cncl_comb_mtL <pstep(4)));
I_atten4_mtL = find((p_cncl_comb_mtL >=  pstep(4)) & ...
                    (p_cncl_comb_mtL <pstep(5)));
I_atten5_mtL = find((p_cncl_comb_mtL >=  pstep(5)) & ...
                    (p_cncl_comb_mtL <pstep(6)));
I_atten6_mtL = find((p_cncl_comb_mtL >=  pstep(6)));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I_atten0_mtH = find((p_cncl_comb_mtH < pstep(1)));
I_atten1_mtH = find((p_cncl_comb_mtH >=  pstep(1)) & ...
                    (p_cncl_comb_mtH <pstep(2)));
I_atten2_mtH = find((p_cncl_comb_mtH >=  pstep(2)) & ...
                    (p_cncl_comb_mtH <pstep(3)));
I_atten3_mtH = find((p_cncl_comb_mtH >=  pstep(3)) & ...
                    (p_cncl_comb_mtH <pstep(4)));
I_atten4_mtH = find((p_cncl_comb_mtH >=  pstep(4)) & ...
                    (p_cncl_comb_mtH <pstep(5)));
I_atten5_mtH = find((p_cncl_comb_mtH >=  pstep(5)) & ...
                    (p_cncl_comb_mtH <pstep(6)));
I_atten6_mtH = find((p_cncl_comb_mtH >=  pstep(6)));
```

```
% Correction
I_atten3_mtL = [10 11 12 13 14 15];
I_atten4_mtL = [16 17 18 19 20 12];
I_atten2_mtH = [6 7 8 9];
I_atten3_mtH = [10 11 12 13 14 15];

% Now correct for effects of attenuator switch points away from the
% reference cancellation power, for the IM3 phase only
  % Normalize the measured phase shift with frequency of operation,
  % operate on radian measurement
  phase_meas_norm_mtL = pi/180*phase_meas_mtL / (2*pi*f_op_IM3L);
  phase_meas_norm_mtH = pi/180*phase_meas_mtH / (2*pi*f_op_IM3H);

  % Since the reference power is at some intermediate attenuator band,
  % adjust the entire measured phase by the full attenuator band
  % cumulative sum and then use the sum up to the band where the
  % reference power point lies and back out this amount as the correction
  % for the zero degree reference point.

  step_correct_im3l = zeros(size(phase_meas_norm_mtL));
  step_correct_im3h = zeros(size(phase_meas_norm_mtH));
  step_correct_im3l([I_atten1_mtL(1), I_atten2_mtL(1), ...
                     I_atten3_mtL(1), I_atten4_mtL(1)]) = ...
                     atten_tstep_mtL(1:4);
  step_correct_im3h([I_atten1_mtH(1), I_atten2_mtH(1), ...
                     I_atten3_mtH(1), I_atten4_mtH(1)]) = ...
                     atten_tstep_mtH(1:4);

  phase_meas_norm_flat_mtL = phase_meas_norm_mtL + ...
                              cumsum(step_correct_im3l,1);
  phase_meas_norm_flat_mtH = phase_meas_norm_mtH + ...
                              cumsum(step_correct_im3h,1);

  % Now back out the cumulative sum of the reference power point
  phase_meas_norm_adj_mtL = phase_meas_norm_flat_mtL - ...
                            sum(atten_tstep_mtL(1:atten_bnd_im3l));
  phase_meas_norm_adj_mtH = phase_meas_norm_flat_mtH - ...
                            sum(atten_tstep_mtH(1:atten_bnd_im3h));

%     figure
%     plot(p_ref1_ind_mt, phase_meas_norm_mtL, '-*', p_ref1_ind_mt, ...
%          cumsum(step_correct_im3l,1), p_ref1_ind_mt, ...
%          phase_meas_norm_adj_mtL,'-x')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% At this point, the data should be completely corrected for the effects
% of the phase insertion caused by the attenuator switch points.  Now the
% effect of the linear phase shift with power of the sources needs to be
% addressed.

% Correction of phase shift with change in source power follows the
% theoretical development
%
% -- load calibration data for slope of phase shift vs frequency
% -- interpolate table for operating frequency, both stimulus source and
% cancellation source
% -- use equations for corrections developed in theoretical work
% -- plot raw data versus attenuator and generator phase shift data

  % Load calibration data for slope of phase shift vs frequency.  Data is
  % in vector of slope vs frequency for 201 points, linearly spaced from
  % 400 - 2400 MHz.
  m_phi = load('C:\Experiments\jan_05\tues_25\phase_slope.txt');

  % Interpolate for the frequencies used in the sources for the single
  % tone data
  f_op_amam = 450e6;
  m_phi_ref1_amam = interp1(f, m_phi, f_op_amam);
  m_phi_cncl_amam = m_phi_ref1_amam;

  % Interpolate for the frequencies used in the two tone data
  f_op_ref1_mt = 450e6;
  f_op_ref2_mt = 450e6 + 10e3;

  m_phi_ref1_mt = interp1(f, m_phi, f_op_ref1_mt);
  m_phi_ref2_mt = interp1(f, m_phi, f_op_ref2_mt);
  m_phi_cncl_mtL = interp1(f, m_phi, f_op_IM3L);
  m_phi_cncl_mtH = interp1(f, m_phi, f_op_IM3H);

  % Correct measured phase for AM-AM AM-PM data
  % Power difference for sources is that of indicated power from the
  % reference power since
  % indicated power was that chosen as the reference.
  phase_corr_amam = phase_meas_amam*pi/180/(2*pi*f_op_amam) + ...
       m_phi_cncl_amam*(p_cncl_comb_amam - p_cncl_amam_ref) - ...
       m_phi_ref1_amam*(p_ref1_ind_amam - p_ind_ref);
  %save phase_amam.txt phase_corr_amam -ASCII
```

```
%     figure
%     corram1 = m_phi_cncl_amam*(p_cncl_comb_amam - p_cncl_amam_ref);
%     corram2 = m_phi_ref1_amam*(p_ref1_ind_amam - p_ind_ref);
%     data = phase_meas_amam*pi/180/(2*pi*f_op_amam);
%     plot(p_ref1_ind_amam, corram1, p_ref1_ind_amam, corram2, ...
%          p_ref1_ind_amam, data, p_ref1_ind_amam, data+corram1-corram2)

    % Correct measured phase for two-tone data
    phase_corr_mtL = phase_meas_norm_adj_mtL + ...
                     m_phi_cncl_mtL*(p_cncl_comb_mtL - p_im3l_ref) - ...
                     m_phi_ref1_mt*(p_ref1_ind_mt - p_ind_ref);
    phase_corr_mtH = phase_meas_norm_adj_mtH + ...
                     m_phi_cncl_mtH*(p_cncl_comb_mtH - p_im3h_ref) - ...
                     m_phi_ref2_mt*(p_ref1_ind_mt - p_ind_ref);
    %save phase_mtL_10k.txt phase_corr_mtL -ASCII
    %save phase_mtH_10k.txt phase_corr_mtH -ASCII

%     figure
%     corr1 = m_phi_cncl_mtL*(p_cncl_comb_mtL - p_im3l_ref);
%     corr2 = m_phi_ref1_mt*(p_ref1_ind_mt - p_ind_ref);
%     corr3 = cumsum(step_correct_im3l,1);
%     plot(p_ref1_ind_mt, corr1, p_ref1_ind_mt, corr2, ...
%          p_ref1_ind_mt, phase_meas_norm_adj_mtL, ...
%          p_ref1_ind_mt, corr1-corr2+corr3+phase_meas_norm_mtL)


    % Now plot raw measured data and corrected data
    figure
    [AX, H1, H2] = plotyy(p_ref1_in_amam, ...
         [phase_meas_amam phase_corr_amam*2*pi*f_op_amam*180/pi], ...
         p_ref1_in_amam, p_f1_dutout_amam);
    ftitle('Mini-Circuits Raw vs Calibrated phase for AM-AM')
    axes(AX(1))
    axis([-24 -2 -10 5])
    xlabel('Input power (dBm)')
    ylabel('Phase (degrees)')
    axes(AX(2))
    axis([-24 -2 20 45])
    ylabel('Output power (dBm)')
    legend([H1;H2], 'Uncorrected Phase', 'Corrected Phase', 'Mag. f_1',2)
    set(H1(1),'LineWidth', 1.5, 'LineStyle', '-')
    set(H1(2),'LineWidth', 1.5, 'LineStyle', '--')
    set(H2(1),'LineWidth', 1.5, 'LineStyle', '-.')
```

```
figure
plot(p_ref1_in_mt, phase_meas_mtL, '+', p_ref1_in_mt, ...
                phase_corr_mtL*2*pi*f_op_IM3L*180/pi, 'LineWidth', 1.5)
ftitle('Mini-Circuits Raw vs Calibrated phase for Two-tone IM3L')
xlabel('Input power (dBm)')
ylabel('Phase (degrees)')
legend('Measured', 'Corrected')

figure
plot(p_ref1_in_mt, phase_meas_mtH, '+', p_ref1_in_mt, ...
                phase_corr_mtH*2*pi*f_op_IM3H*180/pi, 'LineWidth', 1.5)
ftitle('Mini-Circuits Raw vs Calibrated phase for Two-tone IM3H')
xlabel('Input power (dBm)')
ylabel('Phase (degrees)')
legend('Measured', 'Corrected')

figure
[AX, H1, H2] = plotyy(p_ref1_in_mt, ...
                     [phase_corr_mtL*2*pi*f_op_IM3L*180/pi, ...
                      phase_corr_mtH*2*pi*f_op_IM3H*180/pi],...
                      p_ref1_in_mt, ...
                     [p_im3l_dutout_mt, p_im3h_dutout_mt]);
ftitle('Mini-Circuits Calibrated phase for IM3L and IM3H')
axes(AX(1))
axis([-21 1 -30 10])
ylabel('Phase (degrees)')
axes(AX(2))
axis([-21 1 -25 25])
ylabel('IM3 power (dBm)')
xlabel('Input power (dBm)')
legend([H1 H2], 'Phase IM3L', 'Phase IM3H', 'Mag. IM3L', 'Mag. IM3H',4)
set(H1(1),'LineWidth', 1.5, 'LineStyle', '-')
set(H1(2),'LineWidth', 1.5, 'LineStyle', '--')
set(H2(1),'LineWidth', 1.5, 'LineStyle', '-.')
set(H2(2),'LineWidth', 1.5, 'LineStyle', ':')
```

D.1.2 Model Parameter Extraction and Output Generation Script

```
% This script processes the 10 kHz separation two-tone data taken at
% 450 MHz on the Mini-Circuits PA.  The script plots the measured IM3
% magnitude and phase as polar plots with comparison to the
% expected output based on single tone measurements.
```

```
% In this version, the extrapolation of the single tone complex gain is
% used to provide a better fit to the IM3 data based on the single tone
% data.  This is necessary since I did not sweep the single-tone and the
% two-tone data over the same average power level.


% Read in the corrected measured magnitude and phase data.  Magnitude
% data is that corrected for path loss from DUT to spectrum analyzer
% and corrected phase data takes into account the attenuator and
% phase insertion with power of the IFR sources.
clear all, close all f_op_amam = 450e6; deltaf = 10e3; f_op_IM3L =
450e6 - deltaf; f_op_IM3H = 450e6 + 2*deltaf; read_path =
'C:\Experiments\mar_05\wed_23\mc\'; im3l_mag_p =
load([read_path,'p_im3l_out_10k.txt']); im3h_mag_p =
load([read_path,'p_im3h_out_10k.txt']); im3l_phase =
load([read_path,'phase_mtL_10k.txt']); im3h_phase =
load([read_path,'phase_mtH_10k.txt']); ref1_mag_p =
load([read_path,'p_ref1_in_mt.txt']); ref2_mag_p =
load([read_path,'p_ref2_in_mt.txt']); amam_mag_p =
load([read_path,'p_amam_out.txt']); amam_phase =
load([read_path,'phase_amam.txt']); amam_magin_p =
load([read_path,'p_amam_in.txt']);

amam_cncl_p = load([read_path,'amam_cncl_dB.txt']); im3l_cncl_p =
load([read_path,'im3l_cncl_dB_10k.txt']); im3h_cncl_p =
load([read_path,'im3h_cncl_dB_10k.txt']);

% Convert power in dBm to voltage magnitude, assume 50 Ohm
% impedance
im3l_mag_v = sqrt(10.^(im3l_mag_p/10)/10); im3h_mag_v =
sqrt(10.^(im3h_mag_p/10)/10); f1_mag_v =
sqrt(10.^(ref1_mag_p/10)/10); f2_mag_v =
sqrt(10.^(ref2_mag_p/10)/10); amam_mag_v =
sqrt(10.^(amam_mag_p/10)/10); amam_magin_v =
sqrt(10.^(amam_magin_p/10)/10);

% Convert phase data to radians, saved as frequency normalized
phase_amam_rad = amam_phase * 2*pi*f_op_amam;
phase_amamin_rad = zeros(size(phase_amam_rad));  % Assume input phase is 0
im3l_phase_rad = im3l_phase * 2*pi*f_op_IM3L; im3h_phase_rad =
im3h_phase * 2*pi*f_op_IM3H;

% Plot magnitude and phase of IM3 products on polar plot
```

```
% figure
% polar(im3l_phase_rad, im3l_mag_v)
% hold
% polar(im3h_phase_rad, im3h_mag_v, 'g')

% Fit complex polynomial to AM-AM, AM-PM data in the form of a complex
% gain.

    % Convert polar form of data to complex form
    [x_amam, y_amam] = pol2cart(phase_amam_rad, amam_mag_v);
    vout_cplx_amam = complex(x_amam, y_amam);
    [x_amamin, y_amamin] = pol2cart(phase_amamin_rad, amam_magin_v);
    vin_cplx_amam = complex(x_amamin, y_amamin);

    % Compute complex voltage gain
    G_cplx = vout_cplx_amam./vin_cplx_amam;

    % Plot complex gain for both voltage and power
%     figure
%     plot(abs(vin_cplx_amam), abs(G_cplx))
    figure
    subplot(2,1,1)
    plot(amam_magin_p, amam_mag_p - amam_magin_p,'-^');
    xlabel('Input Tone Power (dBm)')
    ylabel('Power Gain (dB)')
    subplot(2,1,2)
    plot(amam_magin_p, phase_amam_rad*180/pi,'-^');
    xlabel('Input Tone Power (dBm)')
    ylabel('Output Phase ( \circ)')
    ftitle('Measured Magnitude of Complex Gain')

    % Extrapolate the complex gain, magnitude and phase separately
    % Currently these are fit with linear polynomials over their entire
    % range in the voltage domain.
        % Perform linear fit
        lin_fit_mag = polyfit(abs(vin_cplx_amam), abs(G_cplx),1);
        lin_fit_ang = polyfit(abs(vin_cplx_amam), angle(G_cplx),1);
        % Generate extended voltage input
        p_ext = -2.5:5;
        v_ext = sqrt(10.^(p_ext/10)/10);
        % Generate gain magnitude extension
        G_mag_ext = polyval(lin_fit_mag, v_ext);
        % Generate gain phase extension
        G_ang_ext = polyval(lin_fit_ang, v_ext);
```

```
    % Form the extended voltage gain expression
    G_cplx_ext = [real(vout_cplx_amam./vin_cplx_amam); ...
                  (G_mag_ext.*cos(G_ang_ext))'] + ...
                  sqrt(-1)*[imag(vout_cplx_amam./vin_cplx_amam);
                  (G_mag_ext.*sin(G_ang_ext))'];
    % Form the extended input voltage
    vin_cplx_amam_ext = [vin_cplx_amam; v_ext'];
    % Form the extended output voltage
    vout_cplx_amam_ext = vin_cplx_amam_ext.*G_cplx_ext;


% Fit a complex polynomial to the voltage gain.
%
% First form the negative voltage data by negating the
% complex input and output voltages, forcing the output voltage
% to be zero at zero input.  By negating the positive only
% input/output voltage, the transfer function will automatically
% be odd.
%
% Next use the polyfit function to determine the coefficients.
%
% Finally, use polyval and plot the results to visually determine
% if a polynomial of significant order was used.

N = 15;      % Order of polynomial
v_in_amam_lr = [fliplr(-vin_cplx_amam_ext); 0; vin_cplx_amam_ext];
v_out_amam_lr = [fliplr(-vout_cplx_amam_ext); 0; vout_cplx_amam_ext];
G_coefb = polyfit(v_in_amam_lr, v_out_amam_lr, N);

% Plot the vector form of the odd-order polynomial coefficients
figure
ftitle('Vector plot of odd-order complex coefficients')
h = compass(G_coefb(1:2:end));
color = ['m','c','r','g','b','k','m','c','r','g'];
K = size(color,2);
for k = 1:(N+1)/2
    set(h(k),'Color', color(k))
end
legend('9','7','5','3','1',0)

% Evaluate the polynomial fit for the positive voltage data
% and compare to the measured data
v_fit_amam = polyval(G_coefb, vin_cplx_amam);
figure
```

```
    ftitle('Single tone fit')
    plot(abs(vin_cplx_amam), abs(vout_cplx_amam), '-*', ...
         abs(vin_cplx_amam), abs(v_fit_amam))
    legend('Meas.','Est.',0)

    v_fit_amam_ext = polyval(G_coefb, vin_cplx_amam_ext);
    figure
    subplot(2,1,1)
    ext = size(vin_cplx_amam,1);
    ftitle('Single tone fit extrapolated')
    plot(abs(vin_cplx_amam), abs(vout_cplx_amam),'d', ...
         abs(vin_cplx_amam_ext(ext+1:end)), ...
         abs(vout_cplx_amam_ext(ext+1:end)), '^', ...
         abs(vin_cplx_amam_ext), abs(v_fit_amam_ext),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Peak Voltage (V)')
    legend('Meas.','Ext.','Est.',0)
%     title('Extrapolated Single-tone Fit')
    subplot(2,1,2)
    plot(abs(vin_cplx_amam), angle(vout_cplx_amam),'d', ...
         abs(vin_cplx_amam_ext(ext+1:end)), ...
         angle(vout_cplx_amam_ext(ext+1:end)), '^', ...
         abs(vin_cplx_amam_ext), angle(v_fit_amam_ext),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Peak Voltage (V)')
    legend('Meas.','Ext.','Est.',0)


%
%     figure
%     polar(phase_amam_rad, amam_mag_v)
%     hold
%     polar(angle(v_fit_amam), abs(v_fit_amam),'g')

% Now use the complex coefficients to estimate the IM3 product magnitude
% and phase.
%
% First need to take into account that the coefficients that were fit, were
% the average coefficients and so need to be backed out to the
% instantaneous coefficients. Following work done in RAWCON 2004 paper and
% that of Gard:Gutierrez:Steer:1999 and Gharaibeh:Steer:2005

    % Compute instantaneous coefficients
    % Only the odd coefficients will be retained since the even terms
```

```
    % are approximately zero from fit.
    G_coefa = zeros(size(G_coefb));
    G_coefa(N) = G_coefb(N);          % DC components are equal
    for j = 1:2:N
        alpha = (N+1)/2 - (j+1)/2;
        alpha2 = alpha*2;
        G_coefa(j) = G_coefb(j)*2^alpha2*factorial(alpha)*...
                     factorial(1+alpha)/factorial(1+alpha2);
    end

    % Repeat of computation of single tone fit using development in
    % Khaled's paper wiht instantaneous coefficients
    v_fit_amam2 = 0;
    v_fit_amam_comp = zeros(size(vin_cplx_amam,1),size(1:2:N,2));
    for n = 1:2:N,
        m = [ 0 0 (n-1)/2 (n+1)/2];
        mco = multinomial(m,n);
        temp = G_coefa(N-n+1)*mco*(vin_cplx_amam).^n;
        v_fit_amam2 = v_fit_amam2 + temp;
        v_fit_amam_comp(:,(n+1)/2) = temp;
    end

    % Plots comparing two sets of coefficients fit, expected identical, and
    % the phase of the single tone measured and expected data.
%     figure
%     plot(abs(vin_cplx_amam), abs(v_fit_amam),'+-',...
%          abs(vin_cplx_amam), abs(v_fit_amam2))
%     figure
%     plot(abs(vin_cplx_amam), phase(v_fit_amam2), ...
%          abs(vin_cplx_amam), phase_amam_rad)
%     figure
%     plot(amam_magin_p, 10*log10(abs(v_fit_amam_comp).^2*10))
%     legend('1','3','5','7','9',0)

    % Now compute complex IM3 from coefficients and formulation based on
    % Gharaibeh:Steer:2005
    im3lfit1 = 0;
    im3hfit1 = 0;
    im3l_order_comp = zeros(size(f1_mag_v,1),size(3:2:N,2));
%     G_coefa([1,3]) = 0;
    for n = 3:2:N,
        %im3lfit1 =0;
        for l = 0:(n-3)/2,
            % Note, the - terms are L's not one's
```

```
        m = [((n-3)/2-l) (l+1) l ((n+1)/2 - l)];
        mco = multinomial(m, n);
        temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
        im3lfit1 = im3lfit1 + temp;
        temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
        im3hfit1 = im3hfit1 + temp;
        im3l_order_comp(:,(n-1)/2) = im3l_order_comp(:,(n-1)/2) + temp;
    end
end

figure
ftitle('Individual odd-order contributions to IM3 product')
plot(ref1_mag_p, 10*log10(abs(im3l_order_comp).^2*10))
legend('3','5','7','9',0)

figure
ftitle('Individual odd-order contributions to IM3 product, polar')
polar(angle(im3l_order_comp), abs(im3l_order_comp))
legend('3','5','7','9',0)

% Compute phase difference between measured IM3 phase (at 0 degree
% reference power) and the estimated phase from the fit
im3l_phase_fit = unwrap(angle(im3lfit1));
diff = im3l_phase_fit(11)*180/pi - im3l_phase_rad(11)*180/pi;


% Compute error bars for measured IM3 phase data, assume equal power
% between tone and cancellation tone
alpha = 1.0;
im3l_phase_error = 180 - 180/pi*acos((10.^(im3l_cncl_p./10) - ...
                   alpha.^2 - 1)./(2.*alpha));
im3h_phase_error = 180 - 180/pi*acos((10.^(im3h_cncl_p./10) - ...
                   alpha.^2 - 1)./(2.*alpha));
amam_phase_error = 180 - 180/pi*acos((10.^(amam_cncl_p./10) - ...
                   alpha.^2 - 1)./(2.*alpha));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generate expressions for fitting the parameters of the second slice
% to account for the IM3 phase asymmetry.
v_im3l_cplx = im3l_mag_v.*cos(im3l_phase_rad+diff*pi/180) + ...
              sqrt(-1)*im3l_mag_v.*sin(im3l_phase_rad+diff*pi/180);
v_im3h_cplx = im3h_mag_v.*cos(im3h_phase_rad+diff*pi/180) + ...
              sqrt(-1)*im3h_mag_v.*sin(im3h_phase_rad+diff*pi/180);
% Compute rotated IM3 fit, takes into account different phase reference
```

```
    % between measured IM3 and the estimated IM3 from the extrapolated
    % single tone fit
    v_im3fit1 = abs(im3lfit1).*cos(angle(im3lfit1)) + ...
                sqrt(-1)*abs(im3lfit1).*sin(angle(im3lfit1));

%   Since the phase reference of the second-slice component is that of the
%   output of the first-slice, (exhibited by measurements, unproven in
%   theory), the difference vector needs to be centered at 0 degrees, then
%   added to the phase of the single-tone fit.  So rotate the two complex
%   difference vectors by the difference between the phase reference
%   points.
    v_im3l_diff = v_im3l_cplx - v_im3fit1;
    v_im3h_diff = v_im3h_cplx - v_im3fit1;
    v_im3l_diff_rot = abs(v_im3l_diff).*...
                      cos(angle(v_im3l_diff)-diff*pi/180) + ...
                      sqrt(-1)*abs(v_im3l_diff).*...
                      sin(angle(v_im3l_diff)-diff*pi/180);
    v_im3h_diff_rot = abs(v_im3h_diff).*...
                      cos(unwrap(angle(v_im3h_diff))-diff*pi/180) + ...
                      sqrt(-1)*abs(v_im3h_diff).*...
                      sin(unwrap(angle(v_im3h_diff))-diff*pi/180);

    % Plot the difference in phase and magnitude between the estimated and
    % the measured data
    figure
    plotyy(f1_mag_v, [(angle(v_im3l_diff))*180/pi ...
                       unwrap(angle(v_im3h_diff))*180/pi], ...
           f1_mag_v, [abs(v_im3l_diff) abs(v_im3h_diff)])
    figure
    plotyy(f1_mag_v, [(angle(v_im3l_diff_rot))*180/pi ...
                     -unwrap(angle(v_im3h_diff_rot))*180/pi], ...
           f1_mag_v, [abs(v_im3l_diff) abs(v_im3h_diff)])
   ftitle('Difference between measured IM3L & IM3H and single slice model')

    % Compute average difference, use conjugate of IM3H.
    v_im3_diff_avg = 1/2*(v_im3l_diff_rot + conj(v_im3h_diff_rot));
    % Divide the difference by the amplitude of the input to account for
    % the ideal multiplier in the second slice
    temp = v_im3_diff_avg./abs(f1_mag_v);
    v_im3_diff_avg = temp;
%     figure
%     plotyy(f1_mag_v, angle(v_im3_diff_avg)*180/pi, ...
%     f1_mag_v, abs(v_im3_diff_avg))
```

```
    % Now fit even order polynomial to data difference data to get second
    % slice model
    v_in_two_tone_lr = [-fliplr(vin_cplx_amam); 0; vin_cplx_amam];
    vdiff_lr = [fliplr(v_im3_diff_avg); 0; v_im3_diff_avg];
    G2_coefb = polyfit(v_in_two_tone_lr, vdiff_lr, 8);

    figure
    subplot(2,1,1)
    plot(f1_mag_v, abs(v_im3_diff_avg), ...
         f1_mag_v, abs(polyval(G2_coefb, f1_mag_v)))
    subplot(2,1,2)
    plot(f1_mag_v, angle(v_im3_diff_avg)*180/pi, ...
         f1_mag_v, angle(polyval(G2_coefb, f1_mag_v))*180/pi)

    % Add difference to IM3L, add conjugate to IM3H, remember to account
    % for ideal multiplier effect
    temp = polyval(G2_coefb, f1_mag_v).*f1_mag_v;
    v_im3l_slice2 = abs(temp).*cos(angle(temp)+diff*pi/180) + ...
                    sqrt(-1)*abs(temp).*sin(angle(temp)+diff*pi/180);
    v_im3h_slice2 = abs(temp).*cos(angle(conj(temp))+diff*pi/180) + ...
                    sqrt(-1)*abs(temp).*sin(angle(conj(temp))+diff*pi/180);
    v_im3l_est = v_im3fit1 + v_im3l_slice2;
    v_im3h_est = v_im3fit1 + v_im3h_slice2;
    % Plot result
%     figure
%     plotyy(f1_mag_v, [angle(v_im3l_cplx)*180/pi ...
%                       angle(v_im3h_cplx)*180/pi ...
%                       angle(v_im3l_est)*180/pi ...
%                       angle(v_im3h_est)*180/pi],
%            f1_mag_v, [abs(v_im3l_cplx) abs(v_im3h_cplx) ...
%                       abs(v_im3l_est) abs(v_im3h_est)])

    % Plot the result of the two slice fit for the IM3L and IM3H products
    % separating phase and amplitude
    figure
    ftitle('Multi-slice Fit to Mini-circuits IM3 Data')
    subplot(2,1,1)
    plot(lin2logmag(f1_mag_v), lin2logmag(v_im3l_cplx),'x', ...
         lin2logmag(f1_mag_v), lin2logmag(v_im3l_est),'-^', ...
         lin2logmag(f1_mag_v), lin2logmag(v_im3h_cplx),'+', ...
         lin2logmag(f1_mag_v), lin2logmag(v_im3h_est),'-d')
    xlabel('Input Tone Power (dBm)')
    ylabel('Output IM3 Tone Power (dBm)')
    legend('IM3L Mag. Meas.', 'IM3L Mag. Est.','IM3H Mag. Meas.',...
```

```
        'IM3H Mag. Est.',0)
    subplot(2,1,2)
%     figure
%     ftitle('Multi-slice Fit to Mini-circuits IM3 Phase Data')
    plot(lin2logmag(f1_mag_v), angle(v_im3l_cplx)*180/pi,'x', ...
         lin2logmag(f1_mag_v), angle(v_im3l_est)*180/pi,'-^', ...
         lin2logmag(f1_mag_v), angle(v_im3h_cplx)*180/pi,'+', ...
         lin2logmag(f1_mag_v), unwrap(angle(v_im3h_est))*180/pi,'-d')
    xlabel('Input Tone Power (dBm)')
    ylabel('Relative Phase of IM3 Products ( \circ)')
    legend('IM3L Phase Meas.', 'IM3L Phase Est.','IM3H Phase Meas.', ...
        'IM3H Phase Est.',0)


    figure
    ftitle('IM3L measured and estimated')
    plot(10*log10(abs(vin_cplx_amam).^2*10), 10*log10(im3l_mag_v.^2*10),...
         10*log10(abs(vin_cplx_amam).^2*10), ...
         10*log10(abs(im3lfit1).^2*10),'+')
    legend('Meas.', 'Est.',0)

    figure
    subplot(2,1,1)
    ftitle('IM3 phase measured and errorbars')
%     plot(ref1_mag_p, im3l_phase_rad*180/pi, ...
%          ref1_mag_p, unwrap(angle(im3lfit1))*180/pi)
    errorbar(ref1_mag_p, im3l_phase_rad*180/pi, im3l_phase_error);
    xlabel('Input Tone Power (dBm)')
    ylabel('Measured IM3L Phase ( \circ)')
%     hold
%     plot(ref1_mag_p, unwrap(angle(im3lfit1))*180/pi - diff,'r')
%     legend('Meas.', 'Est.',0)
    title('Measured Relative IM3 Phase and Error')
    subplot(2,1,2)
    errorbar(ref1_mag_p, im3h_phase_rad*180/pi, im3h_phase_error);
    xlabel('Input Tone Power (dBm)')
    ylabel('Measured IM3L Phase ( \circ)')
%     hold
%     plot(ref1_mag_p, unwrap(angle(im3lfit1))*180/pi - diff,'r')
%     legend('Meas.', 'Est.',0)

    figure
    ftitle('IM3H measured and estimated')
    plot(10*log10(abs(vin_cplx_amam).^2*10), 10*log10(im3h_mag_v.^2*10),...
```

```
     10*log10(abs(vin_cplx_amam).^2*10), 10*log10(abs(im3hfit1).^2*10))
legend('Meas.', 'Est.',0)

figure
ftitle('Single tone measured and estimated')
plot(amam_magin_p, amam_mag_p, ...
     amam_magin_p, 10*log10(abs(v_fit_amam).^2*10))
legend('Meas.','Est.',0)

% Now compare estimated first slice IM3 with measured IM3 products in
% complex form on a polar plot
figure
ftitle('IM3 measured compared to IM3 fit from single-tone data')
polar(im3l_phase_rad+diff*pi/180, im3l_mag_v)
hold
polar(im3h_phase_rad+diff*pi/180, im3h_mag_v, 'g')
polar(angle(im3lfit1), abs(im3lfit1),'r')

% Plot fit of single slice model to IM3 data
figure
subplot(2,1,1)
plot(ref1_mag_p, im3l_mag_p,'x-', ref1_mag_p, im3h_mag_p,'+-', ...
     ref1_mag_p, 10*log10(abs(im3lfit1).^2*10), 'o-')
xlabel('Input Tone Power (dBm)')
ylabel('Measured IM3 Mag. (dBm)')
legend('IM3L', 'IM3H', 'IM3fit',0)
subplot(2,1,2)
plot(ref1_mag_p, im3l_phase_rad*180/pi + diff,'x-', ...
     ref1_mag_p, im3h_phase_rad*180/pi + diff,'+-', ...
     ref1_mag_p, angle(im3lfit1)*180/pi, 'o-')
xlabel('Input Tone Power (dBm)')
ylabel('Measured IM3 Phase ( \circ)')
legend('IM3L', 'IM3H', 'IM3fit',0)
ftitle('Single slice model of two-tone data')

% Plot the measured and estimated phase of the IM3 products using the
% fit from the single tone data
figure
ftitle('Measured and modeled phase of IM3 products')
%   plot(ref1_mag_p, im3l_phase_rad*180/pi,
%        ref1_mag_p, im3h_phase_rad*180/pi, ...
%        ref1_mag_p, unwrap(angle(im3lfit1))*180/pi-diff)
plot(ref1_mag_p, im3l_phase_rad*180/pi,'-x', ...
     ref1_mag_p, im3h_phase_rad*180/pi,'-+', ...
```

```
        ref1_mag_p, angle(v_im3l_est)*180/pi, '-^', ...
        ref1_mag_p, angle(v_im3h_est)*180/pi,'-d')
    axis([ref1_mag_p(1) ref1_mag_p(end) -40 20])
%     legend('\phi_{IM3L,meas}','\phi_{IM3H,meas}','\phi_{IM3,est}',0)
    xlabel('Input Tone Power (dBm)')
    ylabel('Relative Phase of IM3 Products ( \circ)')
    legend('\phi_{IM3L,meas}','\phi_{IM3H,meas}', ...
           '\phi_{IM3L,est}','\phi_{IM3H,est}',0)

    % Now compare estimated IM3 with measured IM3 products in complex form
    % on a polar plot with adjustment for difference
%     figure
%    ftitle('IM3 measured compared to IM3 fit from single-tone data, diff')
%     polar(im3l_phase_rad, im3l_mag_v)
%     hold
%     polar(im3h_phase_rad, im3h_mag_v,'g')
%     polar(unwrap(angle(im3lfit1))-diff*pi/180, abs(im3lfit1),'r')

    figure
    ftitle('IM3 measured compared to IM3 fit from multi-slice model, diff')
    polar(im3l_phase_rad-diff*pi/180, im3l_mag_v,'-+')
    hold
    polar(im3h_phase_rad-diff*pi/180, im3h_mag_v,'-gx')
    polar(angle(v_im3l_est), abs(v_im3l_est),'-r^')
    polar(angle(v_im3h_est), abs(v_im3h_est),'-cd')


    % Now compare estimated IM3 with measured IM3 products in complex form
    % on a polar plot with adjustment for difference and offset so the
    % magnitude can be plotted in dB
%     figure
%     ftitle('IM3 measured compared to IM3 fit from single-tone data, dB')
%     polar(im3l_phase_rad, 10*log10(im3l_mag_v.^2*10)+25)
%     hold
%     polar(im3h_phase_rad, 10*log10(im3h_mag_v.^2*10)+25,'g')
%     polar(unwrap(angle(im3lfit1))-diff*pi/180,
%           10*log10(abs(im3lfit1).^2*10)+25,'r')

    figure
    ftitle('IM3 measured compared to IM3 fit from multislice model, dB')
    polar(im3l_phase_rad+diff*pi/180, 10*log10(im3l_mag_v.^2*10)+25,'-+')
    hold
    polar(im3h_phase_rad+diff*pi/180, 10*log10(im3h_mag_v.^2*10)+25,'-gx')
    polar(angle(v_im3l_est), 10*log10(abs(v_im3l_est).^2*10)+25,'-r^')
```

```
polar(angle(v_im3h_est), 10*log10(abs(v_im3h_est).^2*10)+25,'-cd')
legend('IM3L  Meas.', 'IM3H Meas.','IM3L Est.', 'IM3H Est.')


figure
plot(ref1_mag_p, im3l_mag_p,'x-', ref1_mag_p, im3h_mag_p,'+-', ...
     ref1_mag_p, 10*log10(abs(im3l_mag_v).^2*10), 'o-', ...
     ref1_mag_p, 10*log10(abs(v_im3h_est).^2*10), '*-')
xlabel('Input Tone Power (dBm)')
ylabel('IM3 Mag. (dBm)')
legend('IM3L  Meas.', 'IM3H Meas.','IM3L Est.', 'IM3H Est.')

% Plot figures for inclusion in papers
  % Plot single-tone measured data
  figure
  [AX H1 H2] = plotyy(amam_magin_p, amam_mag_p, ...
                      amam_magin_p, phase_amam_rad*180/pi);
  set(H1, 'Marker', '+')
  set(H2, 'Marker', 'x')
  xlabel('Input Tone Power (dBm)')
  % Set left Y axis label
  set(get(AX(1),'Ylabel'), 'String', 'Output Tone Power (dBm)')
  set(AX(1),'Ycolor', 'k')
  % Set right Y axis label
  set(get(AX(2),'Ylabel'), 'String', 'Output Tone Phase (\circ)')
  set(AX(2),'Ycolor', 'k')
  ftitle('Measured AM-AM, AM-PM Data')

  % Plot two-tone measured data, both upper and lower IM3
  figure
  [AX H1 H2] = plotyy(ref1_mag_p, [im3l_mag_p im3h_mag_p], ...
               ref1_mag_p, [im3l_phase_rad im3h_phase_rad]*180/pi);
  set(H1(1), 'Marker', '+'), set(H1(1), 'Color', 'b')
  set(H1(2), 'Marker', '*'), set(H1(2), 'Color', 'g')
  set(H2(1), 'Marker', '+'), set(H2(1), 'Color', 'b')
  set(H2(2), 'Marker', '*'), set(H2(2), 'Color', 'g')
  xlabel('Input Tone Power (dBm)')
  % Set left Y axis label
  set(get(AX(1), 'Ylabel'), 'String', 'Output IM3 Power (dBm)')
  set(AX(1),'Ycolor', 'k')
  legend('IM3L', 'IM3H')
  % Set right Y axis label
  set(get(AX(2),'Ylabel'), 'String', 'Output IM3 Phase (\circ)')
  set(AX(2),'Ycolor', 'k')
```

```
ftitle('Measured IM3 Two-tone Data')
```

D.2 Scripts for MOSFET Amplifier

D.2.1 RF Source Correction Script

```
% This script calls functions to process measured DUT data
% for single tone AM-AM, AM-PM and two-tone power sweep
% experiments.  The test setup in this case is the cancellation
% bridge where three sources, which share a 10 MHz reference
% frequency, for supplying the stimulus and the cancellation
% tone.  The cancellation tone source provides for the measurement
% of cancelled tone phase by allowing a carrier frequency shift
% in 0.09 degree increments.

% Data taken for this setup is not compensated for power loss
% in the apparatus network either at the input or output.  These
% losses were characterized and the losses will be compensated
% before further processing.

% Load single tone data, data is indicated power on source
p_ref1_ind_amam = (-20:0)'; fpath = 'ar_amam_pout_cncl.txt';
% Power measured after cancellation
p_cncl_meas_amam = load(fpath);
% Power level to be cancelled, i.e. at DUT output
fpath = 'ar_amam_pout.txt'; p_f1_meas_amam = load(fpath); fpath =
'ar_amam_pcncl.txt';
% Power level indicated by cancellation source
p_cncl_src_amam = load(fpath);
% Meausred phase for AM-PM
fpath = 'ar_amam_phase.txt'; phase_meas_amam = load(fpath);

% Load two tone data
p_ref1_ind_mt = (-14:0)'; p_ref2_ind_mt = (-14:0)'; fpath =
'ar_pim3l_cncl_10k.txt';
% Power measured after cancellation
p_cncl_meas_mtL = load(fpath); fpath = 'ar_pim3h_cncl_10k.txt';
% Power measured after cancellation
p_cncl_meas_mtH = load(fpath); fpath = 'ar_pim3l_10k.txt';
% Power measured before cancellation
p_im3l_meas_mt = load(fpath); fpath = 'ar_pim3h_10k.txt';
% Power measured before cancellation
```

```
p_im3h_meas_mt = load(fpath); fpath = 'ar_im3l_pcncl_10k.txt';
% Power level indicated by cancellation source
p_cncl_src_mtL = load(fpath); fpath = 'ar_im3h_pcncl_10k.txt';
% Power level indicated by cancellation source
p_cncl_src_mtH = load(fpath);

fpath = 'ar_phase_im3l_10k.txt';
% Measured IM3L phase
phase_meas_mtL = load(fpath); fpath = 'ar_phase_im3h_10k.txt';
% Measured IM3H phase
phase_meas_mtH = load(fpath);


% Correct for loss in apparatus network for single tone data
  % 3 dB loss between source and DUT input (measured at several powers)
  p_refin_loss = 3;
  % 43 dB loss between DUT output and SA (40 dB pad + combiner)
  p_dutout_loss = 43;
  % 3.33 dB loss between cancellation source and combiner input
  p_cncl_loss = 3.33;

  % Correct power at DUT input
  p_ref1_in_amam = p_ref1_ind_amam - p_refin_loss;
  %save p_amam_in.txt p_ref1_in_amam -ASCII

  % Correct power at DUT output
  p_f1_dutout_amam  = p_f1_meas_amam + p_dutout_loss;
  %save p_amam_out.txt p_f1_dutout_amam -ASCII

  % Indicated power of cancellation source for AM-AM measurement
  p_cncl_comb_amam = p_cncl_src_amam;

% Correct for loss in apparatus network for two-tone data
  % Correct power for tone 1 at DUT input
  p_ref1_in_mt = p_ref1_ind_mt - p_refin_loss;
  %save p_ref1_in_mt.txt p_ref1_in_mt -ASCII

  % Correct power for tone 2 at DUT input
  p_ref2_in_mt = p_ref2_ind_mt - p_refin_loss;
  %save p_ref2_in_mt.txt p_ref2_in_mt -ASCII

  % Correct power of IM3 products at DUT output
  p_im3l_dutout_mt = p_im3l_meas_mt + p_dutout_loss;
  p_im3h_dutout_mt = p_im3h_meas_mt + p_dutout_loss;
```

```
  %save p_im3l_out_10k.txt p_im3l_dutout_mt -ASCII
  %save p_im3h_out_10k.txt p_im3h_dutout_mt -ASCII

  % Indicated power of cancellation source for two-tone measurement
  p_cncl_comb_mtL = p_cncl_src_mtL;
  p_cncl_comb_mtH = p_cncl_src_mtH;

  % Compute cancellation levels for IM3 and AM-AM measurements
  p_im3l_cancellation = p_cncl_meas_mtL - p_im3l_meas_mt;
  p_im3h_cancellation = p_cncl_meas_mtH - p_im3h_meas_mt;
  p_amam_cancellation = p_cncl_meas_amam - p_f1_meas_amam;

% Correct for phase shifts introduced by attenuator steps.  This is a
% simple subtraction of the affect of the attenuators on the output phase
% from the signal generators.  It is only necessary to correct for the
% phase jumps produced by the cancellation generator for the two-tone tests
% since the stimulus generators are always used in attenuator lock mode and
% do not have the attenuator step affects.  In the single tone
% measurements, both the stimulus generator and the cancellation generator
% use attenuator lock and therefore do not need to be corrected.  The range
% of output power for which attenuator lock can be used is ~28 dB.

  % Phase correction will be performed in the time basis, i.e. normalized
  % phase shift with frequency, phi/omega.  In order to correct for the
  % attenuator shift, need to do the following:
  %     - determine the attenuator band that the reference power was
  %     measured in
  %     - segment the phase data by power to match the attenuator bands
  %     - look up the attenuator phase shifts in the calibration data
  %     - adjust entire power segments by the appropriate shift for
  %     cumulative effect of attenuator shifts

  % Load calibration data for size of attenuator steps at each of six
  % points in power sweep from -70 dBm to +13 dBm.  Data is in frequency
  % (row) vs attenuator step (col).  There are six steps at different power
  % outputs. format with the following indices.
  %     - f = 400e6:(2400e6-400e6)/200:2400e6;
  %     - pstep = [-58 -47 -36 -25 -14 -3];
  %     ***Note these steps are the powers
  %     where they occur when using the
  %     knob to control power, not arrows
  %     the difference is 1 dB in the switch points.
  %
  atten_tstep = load('C:\Experiments\jan_05\tues_25\atten_phase_step.txt');
```

```
% Cal data indices
f = 400e6:(2400e6-400e6)/200:2400e6;
% !!!! This was changed since the AR data fell below the calibrated range
% of data, i.e. cancellation source power below -70 dBm.
pstep = [-70 -58 -47 -36 -25 -14];

% Generate interpolated vector of step sizes for frequency of operation
f_op_IM3L = 450e6 - 10e3;
f_op_IM3H = 450e6 + 2*10e3;
atten_tstep_mtL = interp1(f,atten_tstep,f_op_IM3L);
atten_tstep_mtH = interp1(f,atten_tstep,f_op_IM3H);
%!!!!!!!!!!!!!!!!!!!!!!!!
% Compensate for the attenuator phase step for data below -70 dBm
% indicated power on the cancellation source.  Since the attenuator step
% phase insertion appears cyclic, will use the next step size (average of
% two that occur in the calibration set taken, steps 3 and 6)
atten_step_correct_mtL = (atten_tstep_mtL(3) + atten_tstep_mtL(6))/2;
atten_step_correct_mtH = (atten_tstep_mtH(3) + atten_tstep_mtH(6))/2;

atten_tstep_mtL = [atten_step_correct_mtL atten_tstep_mtL];
atten_tstep_mtH = [atten_step_correct_mtH atten_tstep_mtH];

% Determine attenuator band in which the reference power was taken.  All
% attenuator effects are based on this power representing zero phase for
% the cancellation source.  The reference power for th AR amplifier is
% -10 dBm indicated.  The index of this point will be found and used to
% find the corresponding IM3 cancellation power index.
p_ind_ref = -10;
ref_index = find(p_ref1_ind_mt == -10);
p_cncl_amam_ref = p_cncl_comb_amam(ref_index);
p_im3l_ref = p_cncl_comb_mtL(ref_index);
p_im3h_ref = p_cncl_comb_mtH(ref_index);

% Assume reference is not in band 0, this step places the reference
% position in one of the attenuator bands 1-6.
atten_bnd_im3l = floor(interp1(pstep,1:6,p_im3l_ref));
atten_bnd_im3h = floor(interp1(pstep,1:6,p_im3h_ref));

% Compute the cancellation power outputs that fall within each attenuator
% band for each of the IM3 products
I_atten0_mtL = find((p_cncl_comb_mtL < pstep(1)));
I_atten1_mtL = find((p_cncl_comb_mtL >=  pstep(1)) & ...
                    (p_cncl_comb_mtL <pstep(2)));
```

```
I_atten2_mtL = find((p_cncl_comb_mtL >=  pstep(2)) & ...
                    (p_cncl_comb_mtL <pstep(3)));
I_atten3_mtL = find((p_cncl_comb_mtL >=  pstep(3)) & ...
                    (p_cncl_comb_mtL <pstep(4)));
I_atten4_mtL = find((p_cncl_comb_mtL >=  pstep(4)) & ...
                    (p_cncl_comb_mtL <pstep(5)));
I_atten5_mtL = find((p_cncl_comb_mtL >=  pstep(5)) & ...
                    (p_cncl_comb_mtL <pstep(6)));
I_atten6_mtL = find((p_cncl_comb_mtL >=  pstep(6)));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I_atten0_mtH = find((p_cncl_comb_mtH < pstep(1)));
I_atten1_mtH = find((p_cncl_comb_mtH >=  pstep(1)) & ...
                    (p_cncl_comb_mtH <pstep(2)));
I_atten2_mtH = find((p_cncl_comb_mtH >=  pstep(2)) & ...
                    (p_cncl_comb_mtH <pstep(3)));
I_atten3_mtH = find((p_cncl_comb_mtH >=  pstep(3)) & ...
                    (p_cncl_comb_mtH <pstep(4)));
I_atten4_mtH = find((p_cncl_comb_mtH >=  pstep(4)) & ...
                    (p_cncl_comb_mtH <pstep(5)));
I_atten5_mtH = find((p_cncl_comb_mtH >=  pstep(5)) & ...
                    (p_cncl_comb_mtH <pstep(6)));
I_atten6_mtH = find((p_cncl_comb_mtH >=  pstep(6)));

% Correction for points that fell close to the switch points
I_atten2_mtL = [8 9 10];
I_atten3_mtL = [11 12 13 14];

% Now correct for effects of attenuator switch points away from the
% reference cancellation power, for the IM3 phase only
  % Normalize the measured phase shift with frequency of operation,
  % operate on radian measurement
  phase_meas_norm_mtL = pi/180*phase_meas_mtL / (2*pi*f_op_IM3L);
  phase_meas_norm_mtH = pi/180*phase_meas_mtH / (2*pi*f_op_IM3H);

  % Since the reference power is at some intermediate attenuator band,
  % adjust the entire measured phase by the full attenuator band
  % cumulative sum and then use the sum up to the band where the
  % reference power point lies and back out this amount as the correction
  % for the zero degree reference point.

  step_correct_im3l = zeros(size(phase_meas_norm_mtL));
  step_correct_im3h = zeros(size(phase_meas_norm_mtH));
```

```
    step_correct_im3l([I_atten1_mtL(1), I_atten2_mtL(1), ...
                       I_atten3_mtL(1),I_atten4_mtL(1)]) = ...
                       atten_tstep_mtL(1:4);
    step_correct_im3h([I_atten1_mtH(1), I_atten2_mtH(1), ...
                       I_atten3_mtH(1),I_atten4_mtL(1)]) = ...
                       atten_tstep_mtH(1:4);

    phase_meas_norm_flat_mtL = phase_meas_norm_mtL + ...
                               cumsum(step_correct_im3l,1);
    phase_meas_norm_flat_mtH = phase_meas_norm_mtH + ...
                               cumsum(step_correct_im3h,1);

    % Now back out the cumulative sum of the reference power point
    phase_meas_norm_adj_mtL = phase_meas_norm_flat_mtL - ...
                              sum(atten_tstep_mtL(1:atten_bnd_im3l));
    phase_meas_norm_adj_mtH = phase_meas_norm_flat_mtH - ...
                              sum(atten_tstep_mtH(1:atten_bnd_im3h));

%     figure
%     plot(p_ref1_ind_mt, phase_meas_norm_mtL, '-*', ...
%          p_ref1_ind_mt, cumsum(step_correct_im3l,1), ...
%          p_ref1_ind_mt, phase_meas_norm_adj_mtL,'-x')

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % At this point, the data should be completely corrected for the effects
  % of the phase insertion caused by the attenuator switch points.  Now the
  % effect of the linear phase shift with power of the sources needs to be
  % addressed.

  % Correction of phase shift with change in source power follows the
  % theoretical development
  %
  % -- load calibration data for slope of phase shift vs frequency
  % -- interpolate table for operating frequency, both stimulus source and
  % cancellation source
  % -- use equations for corrections developed in theoretical work
  % -- plot raw data versus attenuator and generator phase shift data

    % Load calibration data for slope of phase shift vs frequency.  Data is
    % in vector of slope vs frequency for 201 points, linearly spaced from
    % 400 - 2400 MHz.
    m_phi = load('C:\Experiments\jan_05\tues_25\phase_slope.txt');

    % Interpolate for the frequencies used in the sources for the single
```

```
    % tone data
    f_op_amam = 450e6;
    m_phi_ref1_amam = interp1(f, m_phi, f_op_amam);
    m_phi_cncl_amam = m_phi_ref1_amam;

    % Interpolate for the frequencies used in the two tone data
    f_op_ref1_mt = 450e6;
    f_op_ref2_mt = 450e6 + 10e3;

    m_phi_ref1_mt = interp1(f, m_phi, f_op_ref1_mt);
    m_phi_ref2_mt = interp1(f, m_phi, f_op_ref2_mt);
    m_phi_cncl_mtL = interp1(f, m_phi, f_op_IM3L);
    m_phi_cncl_mtH = interp1(f, m_phi, f_op_IM3H);

    % Correct measured phase for AM-AM AM-PM data Power difference for
    % sources is that of indicated power from the reference power since
    % indicated power was that chosen as the reference.
    phase_corr_amam = phase_meas_amam*pi/180/(2*pi*f_op_amam) + ...
                   m_phi_cncl_amam*(p_cncl_comb_amam - p_cncl_amam_ref)...
                   - m_phi_ref1_amam*(p_ref1_ind_amam - p_ind_ref);
    %save phase_amam.txt phase_corr_amam -ASCII

%     figure
%     corram1 = m_phi_cncl_amam*(p_cncl_comb_amam - p_cncl_amam_ref);
%     corram2 = m_phi_ref1_amam*(p_ref1_ind_amam - p_ind_ref);
%     data = phase_meas_amam*pi/180/(2*pi*f_op_amam);
%     plot(p_ref1_ind_amam, corram1, p_ref1_ind_amam, corram2, ...
%          p_ref1_ind_amam, data, p_ref1_ind_amam, data+corram1-corram2)

    % Correct measured phase for two-tone data
    phase_corr_mtL = phase_meas_norm_adj_mtL + m_phi_cncl_mtL*...
                     (p_cncl_comb_mtL - p_im3l_ref) - ...
                     m_phi_ref1_mt*(p_ref1_ind_mt - p_ind_ref);
    phase_corr_mtH = phase_meas_norm_adj_mtH + m_phi_cncl_mtH*...
                     (p_cncl_comb_mtH - p_im3h_ref) - ...
                     m_phi_ref2_mt*(p_ref1_ind_mt - p_ind_ref);
    %save phase_mtL_10k.txt phase_corr_mtL -ASCII
    %save phase_mtH_10k.txt phase_corr_mtH -ASCII

%     figure
%     corr1 = m_phi_cncl_mtL*(p_cncl_comb_mtL - p_im3l_ref);
%     corr2 = m_phi_ref1_mt*(p_ref1_ind_mt - p_ind_ref);
%     corr3 = cumsum(step_correct_im3l,1);
%     plot(p_ref1_ind_mt, corr1, p_ref1_ind_mt, corr2, ...
```

```
%          p_ref1_ind_mt, phase_meas_norm_adj_mtL, ...
%          p_ref1_ind_mt, corr1-corr2+corr3+phase_meas_norm_mtL)


    % Now plot raw measured data and corrected data
    figure
    [AX, H1, H2] = plotyy(p_ref1_ind_amam, ...
              [phase_meas_amam phase_corr_amam*2*pi*f_op_amam*180/pi], ...
              p_ref1_ind_amam, p_f1_dutout_amam);
    ftitle('AR Raw vs Calibrated phase for AM-AM')
    axes(AX(1))
    xlabel('Input power (dBm)')
    ylabel('Phase (degrees)')
    axes(AX(2))
    ylabel('Output power (dBm)')
    legend([H1;H2], 'Uncorrected Phase', 'Corrected Phase', 'Mag. f_1',2)
    set(H1(1),'LineWidth', 1.5, 'LineStyle', '-')
    set(H1(2),'LineWidth', 1.5, 'LineStyle', '--')
    set(H2(1),'LineWidth', 1.5, 'LineStyle', '-.')


    figure
    plot(p_ref1_ind_mt, phase_meas_mtL, '+', ...
         p_ref1_ind_mt, phase_corr_mtL*2*pi*f_op_IM3L*180/pi, ...
         'LineWidth', 1.5)
    ftitle('AR Raw vs Calibrated phase for Two-tone IM3L')
    xlabel('Input power (dBm)')
    ylabel('Phase (degrees)')
    legend('Measured', 'Corrected')

    figure
    plot(p_ref1_ind_mt, phase_meas_mtH, '+', ...
         p_ref1_ind_mt, phase_corr_mtH*2*pi*f_op_IM3H*180/pi, ...
         'LineWidth', 1.5)
    ftitle('AR Raw vs Calibrated phase for Two-tone IM3H')
    xlabel('Input power (dBm)')
    ylabel('Phase (degrees)')
    legend('Measured', 'Corrected')

    figure
    [AX, H1, H2] = plotyy(p_ref1_ind_mt, ...
                          [phase_corr_mtL*2*pi*f_op_IM3L*180/pi, ...
                           phase_corr_mtH*2*pi*f_op_IM3H*180/pi], ...
                           p_ref1_ind_mt, ...
```

```
                        [p_im3l_dutout_mt, p_im3h_dutout_mt]);
    ftitle('AR Calibrated phase for IM3L and IM3H')
    axes(AX(1))
    axis([-15 1 -10 10])
    ylabel('Phase (degrees)')
    axes(AX(2))
    axis([-15 1 -60 10])
    ylabel('IM3 power (dBm)')
    xlabel('Input power (dBm)')
    legend([H1 H2], 'Phase IM3L', 'Phase IM3H', 'Mag. IM3L', 'Mag. IM3H',2)
    set(H1(1),'LineWidth', 1.5, 'LineStyle', '-')
    set(H1(2),'LineWidth', 1.5, 'LineStyle', '--')
    set(H2(1),'LineWidth', 1.5, 'LineStyle', '-.')
    set(H2(2),'LineWidth', 1.5, 'LineStyle', ':')
```

D.2.2 Model Parameter Extraction and Output Generation Script

```
% This script processes the 10 kHz separation two-tone data taken at
% 450 MHz on the AR Amplifier.  The script plots the measured IM3
% magnitude and phase as polar plots with comparison to the
% expected output based on single tone measurements.


% Read in the corrected measured magnitude and phase data.  Magnitude
% data is that corrected for path loss from DUT to spectrum analyzer
% and corrected phase data takes into account the attenuator and
% phase insertion with power of the IFR sources.
clear all, close all, cd C:\Experiments\mar_05\mon_7 f_op_amam =
450e6; deltaf = 10e3; f_op_IM3L = 450e6 - deltaf; f_op_IM3H =
450e6 + 2*deltaf; read_path = 'C:\Experiments\mar_05\wed_23\ar\';
im3l_mag_p = load([read_path,'p_im3l_out_10k.txt']); im3h_mag_p =
load([read_path,'p_im3h_out_10k.txt']); im3l_phase =
load([read_path,'phase_mtL_10k.txt']); im3h_phase =
load([read_path,'phase_mtH_10k.txt']); ref1_mag_p =
load([read_path,'p_ref1_in_mt.txt']); ref2_mag_p =
load([read_path,'p_ref2_in_mt.txt']); amam_mag_p =
load([read_path,'p_amam_out.txt']); amam_phase =
load([read_path,'phase_amam.txt']); amam_magin_p =
load([read_path,'p_amam_in.txt']);

amam_cncl_p = load([read_path,'amam_cncl_dB.txt']); im3l_cncl_p =
load([read_path,'im3l_cncl_dB_10k.txt']); im3h_cncl_p =
load([read_path,'im3h_cncl_dB_10k.txt']);
```

```
% Convert power in dBm to voltage magnitude, assume 50 Ohm
% impedance
im3l_mag_v = sqrt(10.^(im3l_mag_p/10)/10); im3h_mag_v =
sqrt(10.^(im3h_mag_p/10)/10); f1_mag_v =
sqrt(10.^(ref1_mag_p/10)/10); f2_mag_v =
sqrt(10.^(ref2_mag_p/10)/10); amam_mag_v =
sqrt(10.^(amam_mag_p/10)/10); amam_magin_v =
sqrt(10.^(amam_magin_p/10)/10);

% Convert phase data to radians, saved as frequency normalized
phase_amam_rad = amam_phase * 2*pi*f_op_amam;
phase_amamin_rad = zeros(size(phase_amam_rad));  % Assume input phase is 0
im3l_phase_rad = im3l_phase * 2*pi*f_op_IM3L; im3h_phase_rad =
im3h_phase * 2*pi*f_op_IM3H;

% Plot magnitude and phase of IM3 products on polar plot
% figure
% polar(im3l_phase_rad, im3l_mag_v)
% hold
% polar(im3h_phase_rad, im3h_mag_v, 'g')

% Fit complex polynomial to AM-AM, AM-PM data in the form of a complex
% gain.

    % Convert polar form of data to complex form
    [x_amam, y_amam] = pol2cart(phase_amam_rad, amam_mag_v);
    vout_cplx_amam = complex(x_amam, y_amam);
    [x_amamin, y_amamin] = pol2cart(phase_amamin_rad, amam_magin_v);
    vin_cplx_amam = complex(x_amamin, y_amamin);

    f1_mag_vcplx = complex(f1_mag_v, zeros(size(f1_mag_v)));

    % Compute complex voltage gain
    G_cplx = vout_cplx_amam./vin_cplx_amam;

    % Plot complex gain for both voltage and power
%     figure
%     plot(abs(vin_cplx_amam), abs(G_cplx))
    figure
    subplot(2,1,1)
    plot(amam_magin_p, amam_mag_p - amam_magin_p,'-^');
    xlabel('Input Tone Power (dBm)')
    ylabel('Power Gain (dB)')
```

```
subplot(2,1,2)
plot(amam_magin_p, phase_amam_rad*180/pi,'-^');
xlabel('Input Tone Power (dBm)')
ylabel('Output Phase ( \circ)')
ftitle('Measured Magnitude of Complex Gain')

% Extrapolate the complex gain, magnitude and phase separately
% Currently these are fit with linear polynomials over their entire
% range in the voltage domain.
    % Perform linear fit
    lin_fit_mag = polyfit(abs(vin_cplx_amam), abs(G_cplx),1);
    lin_fit_ang = polyfit(abs(vin_cplx_amam), angle(G_cplx),1);
    % Generate extended voltage input
    p_ext = -3:5;
    v_ext = sqrt(10.^(p_ext/10)/10);
    % Generate gain magnitude extension
    G_mag_ext = polyval(lin_fit_mag, v_ext);
    % Generate gain phase extension
    G_ang_ext = polyval(lin_fit_ang, v_ext);
    % Form the extended voltage gain expression
    G_cplx_ext = [real(vout_cplx_amam./vin_cplx_amam); ...
                  (G_mag_ext.*cos(G_ang_ext))'] + ...
                  sqrt(-1)*[imag(vout_cplx_amam./vin_cplx_amam); ...
                  (G_mag_ext.*sin(G_ang_ext))'];
    % Form the extended input voltage
    vin_cplx_amam_ext = [vin_cplx_amam; v_ext'];
    % Form the extended output voltage
    vout_cplx_amam_ext = vin_cplx_amam_ext.*G_cplx_ext;


% Fit a complex polynomial to the voltage gain.
%
% First form the negative voltage data by negating the
% complex input and output voltages, forcing the output voltage
% to be zero at zero input.  By negating the positive only
% input/output voltage, the transfer function will automatically
% be odd.
%
% Next use the polyfit function to determine the coefficients.
%
% Finally, use polyval and plot the results to visually determine
% if a polynomial of significant order was used.

N = 5;      % Order of polynomial
```

```
    v_in_amam_lr = [fliplr(-vin_cplx_amam_ext); 0; vin_cplx_amam_ext];
    v_out_amam_lr = [fliplr(-vout_cplx_amam_ext); 0; vout_cplx_amam_ext];
    G_coefb = polyfit(v_in_amam_lr, v_out_amam_lr, N);

    % Plot the vector form of the odd-order polynomial coefficients
    figure
    ftitle('Vector plot of odd-order complex coefficients')
    h = compass(G_coefb(1:2:end));
    color = ['m','c','r','g','b','k','m','c','r','g'];
    K = size(color,2);
    for k = 1:(N+1)/2
        set(h(k),'Color', color(k))
    end
    legend('9','7','5','3','1',0)

    % Evaluate the polynomial fit for the positive voltage data
    % and compare to the measured data
    v_fit_amam = polyval(G_coefb, vin_cplx_amam);
    figure
    ftitle('Single tone fit')
    plot(abs(vin_cplx_amam), abs(vout_cplx_amam), '-*', ...
         abs(vin_cplx_amam), abs(v_fit_amam))
    legend('Meas.','Est.',0)

    v_fit_amam_ext = polyval(G_coefb, vin_cplx_amam_ext);
    figure
    subplot(2,1,1)
    ext = size(vin_cplx_amam,1);
    ftitle('Single tone fit extrapolated')
%     plot(abs(vin_cplx_amam), abs(vout_cplx_amam),'d', ...
%          abs(vin_cplx_amam_ext(ext+1:end)), ...
%          abs(vout_cplx_amam_ext(ext+1:end)), '^', ...
%          abs(vin_cplx_amam_ext), abs(v_fit_amam_ext),'--')
    plot(10*log10(abs(vin_cplx_amam).^2*10), ...
         10*log10(abs(vout_cplx_amam).^2*10),'d', ...
         10*log10(abs(vin_cplx_amam_ext(ext+1:end)).^2*10), ...
         10*log10(abs(vout_cplx_amam_ext(ext+1:end)).^2*10), '^', ...
         10*log10(abs(vin_cplx_amam_ext).^2*10), ...
         10*log10(abs(v_fit_amam_ext).^2*10),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Peak Voltage (V)')
    legend('Meas.','Ext.','Est.',0)
%     title('Extrapolated Single-tone Fit')
    subplot(2,1,2)
```

```
%     plot(abs(vin_cplx_amam), angle(vout_cplx_amam),'d', ...
%          abs(vin_cplx_amam_ext(ext+1:end)), ...
%          angle(vout_cplx_amam_ext(ext+1:end)), '^', ...
%          abs(vin_cplx_amam_ext), angle(v_fit_amam_ext),'--')
    plot(10*log10(abs(vin_cplx_amam).^2*10), angle(vout_cplx_amam),'d', ...
         10*log10(abs(vin_cplx_amam_ext(ext+1:end)).^2*10), ...
         angle(vout_cplx_amam_ext(ext+1:end)), '^', ...
         10*log10(abs(vin_cplx_amam_ext).^2*10),angle(v_fit_amam_ext),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Phase ( \circ)')
    legend('Meas.','Ext.','Est.',0)


% Now use the complex coefficients to estimate the IM3 product magnitude
% and phase.
%
% First need to take into account that the coefficients that were fit, were
% the average coefficients and so need to be backed out to the
% instantaneous coefficients. Following work done in RAWCON 2004 paper and
% that of Gard:Gutierrez:Steer:1999 and Gharaibeh:Steer:2005

    % Compute instantaneous coefficients
    % Only the odd coefficients will be retained since the even terms
    % are approximately zero from fit.
    G_coefa = zeros(size(G_coefb));
    G_coefa(N) = G_coefb(N);          % DC components are equal
    for j = 1:2:N
        alpha = (N+1)/2 - (j+1)/2;
        alpha2 = alpha*2;
        G_coefa(j) = G_coefb(j)*2^alpha2*factorial(alpha)*...
                     factorial(1+alpha)/factorial(1+alpha2);
    end

    % Repeat of computation of single tone fit using development in
    % Khaled's paper wiht instantaneous coefficients
    v_fit_amam2 = 0;
    v_fit_amam_comp = zeros(size(vin_cplx_amam,1),size(1:2:N,2));
    for n = 1:2:N,
        m = [ 0 0 (n-1)/2 (n+1)/2];
        mco = multinomial(m,n);
        temp = G_coefa(N-n+1)*mco*(vin_cplx_amam).^n;
        v_fit_amam2 = v_fit_amam2 + temp;
        v_fit_amam_comp(:,(n+1)/2) = temp;
    end
```

```
    % Plots comparing two sets of coefficients fit, expected identical, and
    % the phase of the single tone measured and expected data.
    figure
    ftitle(['Comparison of fit between non-extrapolated single tone' ...
            'using polyval and that using the envelope formulation'])
    plot(abs(vin_cplx_amam), abs(v_fit_amam),'+-',...
         abs(vin_cplx_amam), abs(v_fit_amam2))
    figure
    ftitle(['Comparison of phase fit between envelope formulation and' ...
            'measured phase of AM-AM data'])
    plot(abs(vin_cplx_amam), angle(v_fit_amam2)*180/pi, ...
         abs(vin_cplx_amam), phase_amam_rad*180/pi)
    figure
    ftitle(['Magnitude of individual odd-order contributors from ' ...
            'envelope formulation of AM-AM data'])
    plot(amam_magin_p, 10*log10(abs(v_fit_amam_comp).^2*10))
    legend('1','3','5','7','9',0)

    % Now compute complex IM3 from coefficients and formulation based on
    % Gharaibeh:Steer:2005
    im3lfit1 = 0;
    im3hfit1 = 0;
    im3l_order_comp = zeros(size(f1_mag_v,1),size(3:2:N,2));
%     G_coefa([1,3]) = 0;
    for n = 3:2:N,
        %im3lfit1 =0;
        for l = 0:(n-3)/2,
            % Note, the - terms are L's not one's
            m = [((n-3)/2-l) (l+1) l ((n+1)/2 - l)];
            mco = multinomial(m, n);
            temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
            im3lfit1 = im3lfit1 + temp;
            temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
            im3hfit1 = im3hfit1 + temp;
            im3l_order_comp(:,(n-1)/2) = im3l_order_comp(:,(n-1)/2) + temp;
        end
    end

    figure
    ftitle('Individual odd-order contributions to IM3 product')
    plot(ref1_mag_p, 10*log10(abs(im3l_order_comp).^2*10))
    legend('3','5','7','9',0)
```

```
figure
ftitle('Individual odd-order contributions to IM3 product, polar')
polar(angle(im3l_order_comp), abs(im3l_order_comp))
legend('3','5','7','9',0)

% Compute phase difference between measured IM3 phase (at 0 degree
% reference power) and the estimated phase from the fit
im3l_phase_fit = unwrap(angle(im3lfit1));
diff = im3l_phase_fit(5)*180/pi - im3l_phase_rad(5)*180/pi;

figure
ftitle('Comparison of IM3 magnitude measured and single-slice estimate')
plot(ref1_mag_p, 10*log10(abs(im3lfit1).^2*10), ...
     ref1_mag_p, im3l_mag_p, ref1_mag_p, im3h_mag_p)

figure
ftitle('Comparison of IM3 phase measured and single-slice estimate')
plot(ref1_mag_p, angle(im3lfit1)*180/pi, ...
     ref1_mag_p, (im3l_phase_rad+diff*pi/180)*180/pi, ...
     ref1_mag_p, (im3h_phase_rad+diff*pi/180)*180/pi)

% Compute error bars for measured IM3 phase data, assume equal power
% between tone and cancellation tone
alpha = 1.0;
im3l_phase_error = 180 - 180/pi*acos((10.^(im3l_cncl_p./10) - ...
                   alpha.^2 - 1)./(2.*alpha));
im3h_phase_error = 180 - 180/pi*acos((10.^(im3h_cncl_p./10) - ...
                   alpha.^2 - 1)./(2.*alpha));
amam_phase_error = 180 - 180/pi*acos((10.^(amam_cncl_p./10) - ...
                   alpha.^2 - 1)./(2.*alpha));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generate expressions for fitting the parameters of the second slice
% to account for the IM3 phase asymmetry.
v_im3l_cplx = im3l_mag_v.*cos(im3l_phase_rad+diff*pi/180)  + ...
              sqrt(-1)*im3l_mag_v.*sin(im3l_phase_rad+diff*pi/180);
v_im3h_cplx = im3h_mag_v.*cos(im3h_phase_rad+diff*pi/180) + ...
              sqrt(-1)*im3h_mag_v.*sin(im3h_phase_rad+diff*pi/180);
% Compute rotated IM3 fit, takes into account different phase reference
% between measured IM3 and the estimated IM3 from the extrapolated
% single tone fit
v_im3fit1 = abs(im3lfit1).*cos(angle(im3lfit1)) + ...
            sqrt(-1)*abs(im3lfit1).*sin(angle(im3lfit1));
```

```
%   Since the phase reference of the second-slice component is that of the
%   output of the first-slice, (exhibited by measurements, unproven in
%   theory), the difference vector needs to be centered at 0 degrees, then
%   added to the phase of the single-tone fit.  So rotate the two complex
%   difference vectors by the difference between the phase reference
%   points.
    v_im3l_diff = v_im3l_cplx - v_im3fit1;
    v_im3h_diff = v_im3h_cplx - v_im3fit1;
    v_im3l_diff_rot = abs(v_im3l_diff).*...
                      cos(angle(v_im3l_diff)-diff*pi/180) + ...
                      sqrt(-1)*abs(v_im3l_diff).*...
                      sin(angle(v_im3l_diff)-diff*pi/180);
    v_im3h_diff_rot = abs(v_im3h_diff).*...
                      cos(unwrap(angle(v_im3h_diff))-diff*pi/180) + ...
                      sqrt(-1)*abs(v_im3h_diff).*...
                      sin(unwrap(angle(v_im3h_diff))-diff*pi/180);

    % Plot the difference in phase and magnitude between the estimated and
    % the measured data
    figure
    plotyy(f1_mag_v, [(unwrap(angle(v_im3l_diff)))*180/pi ...
                       unwrap(angle(v_im3h_diff))*180/pi], ...
           f1_mag_v, [abs(v_im3l_diff) abs(v_im3h_diff)])
    ftitle(['Comparison of difference between IM3L & IM3H and ' ...
            'single slice model'])
    figure
    plotyy(f1_mag_v, ...
           [(unwrap(angle(v_im3l_diff_rot)))*180/pi - ...
             unwrap(angle(v_im3h_diff_rot))*180/pi], ...
           f1_mag_v, [abs(v_im3l_diff) abs(v_im3h_diff)])
   ftitle('Difference between measured IM3L & IM3H and single slice model')

    % Compute average difference, use conjugate of IM3H.
    v_im3_diff_avg = 1/2*(v_im3l_diff_rot + conj(v_im3h_diff_rot));
    % Divide the difference by the amplitude of the input to account for
    % the ideal multiplier in the second slice
    temp = v_im3_diff_avg./abs(f1_mag_v);
    v_im3_diff_avg = temp;
%     figure
%     plotyy(f1_mag_v, angle(v_im3_diff_avg)*180/pi, ...
%            f1_mag_v, abs(v_im3_diff_avg))

    % Now fit even order polynomial to data difference data to get second
    % slice model
```

```
    v_in_two_tone_lr = [-fliplr(f1_mag_vcplx); 0; f1_mag_vcplx];
    vdiff_lr = [fliplr(v_im3_diff_avg); 0; v_im3_diff_avg];
    G2_coefb = polyfit(v_in_two_tone_lr, vdiff_lr, 8);

    figure
    subplot(2,1,1)
    plot(f1_mag_v, abs(v_im3_diff_avg), ...
         f1_mag_v, abs(polyval(G2_coefb, f1_mag_v)))
    subplot(2,1,2)
    plot(f1_mag_v, unwrap(angle(v_im3_diff_avg))*180/pi, ...
         f1_mag_v, unwrap(angle(polyval(G2_coefb, f1_mag_v)))*180/pi)
    ftitle('Fit of average difference between IM3L/H and measured data')

    % Add difference to IM3L, add conjugate to IM3H, remember to account
    % for ideal multiplier effect
    temp = polyval(G2_coefb, f1_mag_v).*f1_mag_v;
    v_im3l_slice2 = abs(temp).*cos(angle(temp)+diff*pi/180) + ...
                    sqrt(-1)*abs(temp).*sin(angle(temp)+diff*pi/180);
    v_im3h_slice2 = abs(temp).*cos(angle(conj(temp))+diff*pi/180) + ...
                    sqrt(-1)*abs(temp).*sin(angle(conj(temp))+diff*pi/180);
    v_im3l_est = v_im3fit1 + v_im3l_slice2;
    v_im3h_est = v_im3fit1 + v_im3h_slice2;

    % Compute the root square error of the single slice and two slice fit
    fit_error_oneslice = sum(abs(v_im3fit1 - v_im3l_cplx).^2./f1_mag_v)+...
                         sum(abs(v_im3fit1 - v_im3h_cplx).^2./f1_mag_v)
    fit_error_twoslice =sum(abs(v_im3l_est - v_im3l_cplx).^2./f1_mag_v)+...
                        sum(abs(v_im3h_est - v_im3h_cplx).^2./f1_mag_v)

    % Plot the result of the two slice fit for the IM3L and IM3H products
    % separating phase and amplitude
    figure
    ftitle('Multi-slice Fit to AR IM3 Data')
    subplot(2,1,1)
    plot(lin2logmag(f1_mag_v),lin2logmag(v_im3l_cplx),'x', ...
         lin2logmag(f1_mag_v),lin2logmag(v_im3l_est),'-^', ...
         lin2logmag(f1_mag_v),lin2logmag(v_im3h_cplx),'+', ...
         lin2logmag(f1_mag_v),lin2logmag(v_im3h_est),'-d')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Peak Voltage (V)')
    legend('IM3L Mag. Meas.', 'IM3L Mag. Est.','IM3H Mag. Meas.', ...
           'IM3H Mag. Est.',0)
    subplot(2,1,2)
%     figure
```

```
%     ftitle('Multi-slice Fit to AR IM3 Phase Data')
    plot(lin2logmag(f1_mag_v), angle(v_im3l_cplx)*180/pi,'x', ...
         lin2logmag(f1_mag_v), angle(v_im3l_est)*180/pi,'-^', ...
         lin2logmag(f1_mag_v), angle(v_im3h_cplx)*180/pi, '+', ...
         lin2logmag(f1_mag_v), unwrap(angle(v_im3h_est))*180/pi,'-d')
    xlabel('Input Peak Voltage (V)')
    ylabel('Relative Phase of IM3 Products ( \circ)')
    legend('IM3L Phase Meas.', 'IM3L Phase Est.','IM3H Phase Meas.', ...
           'IM3H Phase Est.',0)

    figure
    ftitle('IM3L measured and estimated')
    plot(10*log10(abs(f1_mag_vcplx).^2*10), 10*log10(im3l_mag_v.^2*10),...
         10*log10(abs(f1_mag_vcplx).^2*10), ...
         10*log10(abs(im3lfit1).^2*10),'+')
    legend('Meas.', 'Est.',0)

    figure
    subplot(2,1,1)
    ftitle('IM3 phase measured and errorbars')
%     plot(ref1_mag_p, im3l_phase_rad*180/pi, ...
%          ref1_mag_p, unwrap(angle(im3lfit1))*180/pi)
    errorbar(ref1_mag_p, im3l_phase_rad*180/pi, im3l_phase_error);
    xlabel('Input Tone Power (dBm)')
    ylabel('Measured IM3L Phase ( \circ)')
%     hold
%     plot(ref1_mag_p, unwrap(angle(im3lfit1))*180/pi - diff,'r')
%     legend('Meas.', 'Est.',0)
    title('Measured Relative IM3 Phase and Error')
    subplot(2,1,2)
    errorbar(ref1_mag_p, im3h_phase_rad*180/pi, im3h_phase_error);
    xlabel('Input Tone Power (dBm)')
    ylabel('Measured IM3L Phase ( \circ)')
%     hold
%     plot(ref1_mag_p, unwrap(angle(im3lfit1))*180/pi - diff,'r')
%     legend('Meas.', 'Est.',0)

    figure
    ftitle('IM3H measured and estimated')
    plot(10*log10(abs(f1_mag_vcplx).^2*10), 10*log10(im3h_mag_v.^2*10),...
         10*log10(abs(f1_mag_vcplx).^2*10), 10*log10(abs(im3hfit1).^2*10))
    legend('Meas.', 'Est.',0)

    figure
```

```
    ftitle('Single tone measured and estimated')
    plot(amam_magin_p, amam_mag_p, ...
         amam_magin_p, 10*log10(abs(v_fit_amam).^2*10))
    legend('Meas.','Est.',0)

    % Now compare estimated first slice IM3 with measured IM3 products in
    % complex form on a polar plot
    figure
    ftitle('IM3 measured compared to IM3 fit from single-tone data')
    polar(im3l_phase_rad+diff*pi/180, im3l_mag_v)
    hold
    polar(im3h_phase_rad+diff*pi/180, im3h_mag_v, 'g')
    polar(angle(im3lfit1), abs(im3lfit1),'r')
    figure
    subplot(2,1,1)
    plot(ref1_mag_p, im3l_mag_p,'x-', ref1_mag_p, im3h_mag_p,'+-', ...
         ref1_mag_p, 10*log10(abs(im3lfit1).^2*10), 'o-')
    xlabel('Input Tone Power (dBm)')
    ylabel('Measured IM3 Mag. (dBm)')
    legend('IM3L', 'IM3H', 'IM3fit',0)
    subplot(2,1,2)
    plot(ref1_mag_p, im3l_phase_rad*180/pi + diff,'x-', ...
         ref1_mag_p, im3h_phase_rad*180/pi + diff,'+-', ...
         ref1_mag_p, angle(im3lfit1)*180/pi, 'o-')
    xlabel('Input Tone Power (dBm)')
    ylabel('Measured IM3 Phase ( \circ)')
    legend('IM3L', 'IM3H', 'IM3fit',0)

    % Plot the measured and estimated phase of the IM3 products using the
    % fit from the single tone data
    figure
    ftitle('Measured and modeled phase of IM3 products')
%     plot(ref1_mag_p, im3l_phase_rad*180/pi,...
%          ref1_mag_p, im3h_phase_rad*180/pi, ...
%          ref1_mag_p, unwrap(angle(im3lfit1))*180/pi-diff)
    plot(ref1_mag_p, im3l_phase_rad*180/pi,'-x', ...
         ref1_mag_p, im3h_phase_rad*180/pi,'-+', ...
         ref1_mag_p, angle(v_im3l_est)*180/pi, '-^', ...
         ref1_mag_p, angle(v_im3h_est)*180/pi,'-d')
    axis([ref1_mag_p(1) ref1_mag_p(end) -40 20])
%     legend('\phi_{IM3L,meas}','\phi_{IM3H,meas}','\phi_{IM3,est}',0)
    xlabel('Input Tone Power (dBm)')
    ylabel('Relative Phase of IM3 Products ( \circ)')
    legend('\phi_{IM3L,meas}','\phi_{IM3H,meas}','\phi_{IM3L,est}', ...
```

```
        '\phi_{IM3H,est}',0)

    % Now compare estimated IM3 with measured IM3 products in complex form
    % on a polar plot with adjustment for difference
%     figure
%    ftitle('IM3 measured compared to IM3 fit from single-tone data, diff')
%     polar(im3l_phase_rad, im3l_mag_v)
%     hold
%     polar(im3h_phase_rad, im3h_mag_v,'g')
%     polar(unwrap(angle(im3lfit1))-diff*pi/180, abs(im3lfit1),'r')

    figure
    ftitle('IM3 measured compared to IM3 fit from multi-slice model, diff')
    polar(im3l_phase_rad-diff*pi/180, im3l_mag_v,'-+')
    hold
    polar(im3h_phase_rad-diff*pi/180, im3h_mag_v,'-gx')
    polar(angle(v_im3l_est), abs(v_im3l_est),'-r^')
    polar(angle(v_im3h_est), abs(v_im3h_est),'-cd')


    % Now compare estimated IM3 with measured IM3 products in complex form
    % on a polar plot with adjustment for difference and offset so the
    % magnitude can be plotted in dB
%     figure
%     ftitle('IM3 measured compared to IM3 fit from single-tone data, dB')
%     polar(im3l_phase_rad, 10*log10(im3l_mag_v.^2*10)+25)
%     hold
%     polar(im3h_phase_rad, 10*log10(im3h_mag_v.^2*10)+25,'g')
%     polar(unwrap(angle(im3lfit1))-diff*pi/180, ...
%           10*log10(abs(im3lfit1).^2*10)+25,'r')

    figure
    ftitle('IM3 measured compared to IM3 fit from multislice model, dB')
    polar(im3l_phase_rad+diff*pi/180, 10*log10(im3l_mag_v.^2*10)+45,'-+')
    hold
    polar(im3h_phase_rad+diff*pi/180, 10*log10(im3h_mag_v.^2*10)+45,'-gx')
    polar(angle(v_im3l_est), 10*log10(abs(v_im3l_est).^2*10)+45,'-r^')
    polar(angle(v_im3h_est), 10*log10(abs(v_im3h_est).^2*10)+45,'-cd')
    legend('IM3L  Meas.', 'IM3H Meas.','IM3L Est.', 'IM3H Est.')


    figure
    subplot(2,1,1)
%     plot(ref1_mag_p, im3l_mag_p,'x-', ref1_mag_p, im3h_mag_p,'+-', ...
```

```
%         ref1_mag_p, 10*log10(abs(im3l_mag_v).^2*10), 'o-', ...
%         ref1_mag_p, 10*log10(abs(v_im3h_est).^2*10), '*-')
    plot(ref1_mag_p, im3l_mag_p,'x-', ...
         ref1_mag_p, 10*log10(abs(v_im3l_est).^2*10), 'o-')
    xlabel('Input Tone Power (dBm)')
    ylabel('IM3 Mag. (dBm)')
    legend('IM3L  Meas.', 'IM3L Est.')
    subplot(2,1,2)
    plot(ref1_mag_p, im3h_mag_p,'+-', ...
         ref1_mag_p, 10*log10(abs(v_im3h_est).^2*10), '*-')
    xlabel('Input Tone Power (dBm)')
    ylabel('IM3 Mag. (dBm)')
    legend('IM3H Meas.',  'IM3H Est.')

    % Plot figures for inclusion in papers
      % Plot single-tone measured data
      figure
      [AX H1 H2] = plotyy(amam_magin_p, amam_mag_p, ...
                          amam_magin_p, phase_amam_rad*180/pi);
      set(H1, 'Marker', '+')
      set(H2, 'Marker', 'x')
      xlabel('Input Tone Power (dBm)')
      % Set left Y axis label
      set(get(AX(1),'Ylabel'), 'String', 'Output Tone Power (dBm)')
      set(AX(1),'Ycolor', 'k')
      % Set right Y axis label
      set(get(AX(2),'Ylabel'), 'String', 'Output Tone Phase (\circ)')
      set(AX(2),'Ycolor', 'k')
      ftitle('Measured AM-AM, AM-PM Data')

      % Plot two-tone measured data, both upper and lower IM3
      figure
      [AX H1 H2] = plotyy(ref1_mag_p, [im3l_mag_p im3h_mag_p], ...
                          ref1_mag_p, [im3l_phase_rad im3h_phase_rad]*180/pi);
      set(H1(1), 'Marker', '+'), set(H1(1), 'Color', 'b')
      set(H1(2), 'Marker', '*'), set(H1(2), 'Color', 'g')
      set(H2(1), 'Marker', '+'), set(H2(1), 'Color', 'b')
      set(H2(2), 'Marker', '*'), set(H2(2), 'Color', 'g')
      xlabel('Input Tone Power (dBm)')
      % Set left Y axis range
      axis(AX(1), [-18 -2 -50 10])
      set(AX(1), 'YTickMode', 'auto')
      set(AX(2), 'YTickMode', 'auto')
      % Set left Y axis label
```

```
set(get(AX(1), 'Ylabel'), 'String', 'Output IM3 Power (dBm)')
set(AX(1),'Ycolor', 'k')
legend('IM3L', 'IM3H',2)
% Set right Y axis label
set(get(AX(2),'Ylabel'), 'String', 'Output IM3 Phase (\circ)')
set(AX(2),'Ycolor', 'k')
ftitle('Measured IM3 Two-tone Data')
```

D.3 Scripts for Darlington HBT Amplifier

D.3.1 RF Source Correction Script

```
% This script calls functions to process measured DUT data
% for single tone AM-AM, AM-PM and two-tone power sweep
% experiments.  The test setup in this case is the cancellation
% bridge where three sources, which share a 10 MHz reference
% frequency, for supplying the stimulus and the cancellation
% tone.  The cancellation tone source provides for the measurement
% of cancelled tone phase by allowing a carrier frequency shift
% in 0.09 degree increments.

% Data taken for this setup is not compensated for power loss
% in the apparatus network either at the input or output.  These
% losses were characterized and the losses will be compensated
% before further processing.  The actual indicated power of the
% cancellation source was not recorded for the Hittite data
% set so the correction is fixed based on measured loss in the
% interconnect.

clear all, close all
% Load single tone data, data is indicated power on source
pre_fname = ''; pos_fname = '10k'; deltaf = 10e3; amampath =
'C:\Experiments\jan_05\thur_06\';
% twotonepath = 'C:\Experiments\dec_04\thur_16\';
twotonepath = 'C:\Experiments\may_05\tues_31\'; savepath =
'C:\Experiments\may_05\tues_31\'; p_ref1_ind_amam = (-5:13)';
fpath = strcat(amampath,pre_fname,'p_cncl_meas_amam.txt');
p_cncl_meas_amam = load(fpath);     % Power measured after cancellation
fpath = strcat(amampath,pre_fname,'p_f1_meas_amam.txt');
p_f1_meas_amam = load(fpath);       % Power level to be cancelled, i.e. at DUT output
%fpath = 'hittite_amam_pcncl.txt';
```

```
%p_cncl_src_amam = load(fpath);      % Power level indicated by cancellation source
fpath = strcat(amampath,pre_fname,'phase_meas_amam.txt');
phase_meas_amam = load(fpath);

% Load two tone data
p_ref1_ind_mt = (-10:13)'; p_ref2_ind_mt = (-10:13)'; fpath =
strcat(twotonepath,pre_fname,'p_cncl_im3l_',pos_fname,'.txt');
% Power measured after cancellation
p_cncl_meas_mtL = load(fpath); fpath =
strcat(twotonepath,pre_fname,'p_cncl_im3h_',pos_fname,'.txt');
% Power measured after cancellation
p_cncl_meas_mtH = load(fpath); fpath =
strcat(twotonepath,pre_fname,'p_im3l_meas_',pos_fname,'.txt');
% Power measured before cancellation
p_im3l_meas_mt = load(fpath); fpath =
strcat(twotonepath,pre_fname,'p_im3h_meas_',pos_fname,'.txt');
% Power measured before cancellation
p_im3h_meas_mt = load(fpath); fpath =
strcat(twotonepath,pre_fname,'im3l_pcncl_',pos_fname,'.txt');
% Power level indicated by cancellation source
p_cncl_src_mtL = load(fpath); fpath =
strcat(twotonepath,pre_fname,'im3h_pcncl_',pos_fname,'.txt');
% Power level indicated by cancellation source
p_cncl_src_mtH = load(fpath);

% Measured IM3L phase
fpath =
strcat(twotonepath,pre_fname,'phase_im3l_',pos_fname,'.txt');
phase_meas_mtL = load(fpath);
% Measured IM3H phase
fpath =
strcat(twotonepath,pre_fname,'phase_im3h_',pos_fname,'.txt');
phase_meas_mtH = load(fpath);


% Correct for loss in apparatus network for single tone data
  % 3 dB loss between source and DUT input (measured at several powers)
  p_refin_loss = 3;
  % 19 dB loss between DUT output and SA (16 dB pad + combiner)
  p_dutout_loss = 19;
  % 3.33 dB loss between cancellation source and combiner input
  p_cncl_loss = 3.33;

  % Correct power at DUT input
```

```
  p_ref1_in_amam = p_ref1_ind_amam - p_refin_loss;
  fpath = strcat(savepath,'p_amam_in.txt');
  %save(fpath,'p_ref1_in_amam','-ASCII')

  % Correct power at DUT output
  p_f1_dutout_amam  = p_f1_meas_amam + p_dutout_loss;
  fpath = strcat(savepath,'p_amam_out.txt');
  %save(fpath,'p_f1_dutout_amam','-ASCII')

  % Correct power for that produced by the cancellation source.  Power
  % output by this source equals that for the fundamental measured at the
  % SA for the DUT response plus the loss in the combiner.
  p_cncl_comb_amam = p_f1_meas_amam + p_cncl_loss;

% Correct for loss in apparatus network for two-tone data
  % Correct power for tone 1 at DUT input
  p_ref1_in_mt = p_ref1_ind_mt - p_refin_loss;
  fpath = strcat(savepath,'p_ref1_in_mt.txt');
  save(fpath,'p_ref1_in_mt','-ASCII')

  % Correct power for tone 2 at DUT input
  p_ref2_in_mt = p_ref2_ind_mt - p_refin_loss;
  fpath = strcat(savepath,'p_ref2_in_mt.txt');
  save(fpath,'p_ref2_in_mt','-ASCII')

  % Correct power of IM3 products at DUT output
  p_im3l_dutout_mt = p_im3l_meas_mt + p_dutout_loss;
  p_im3h_dutout_mt = p_im3h_meas_mt + p_dutout_loss;
  fpath = strcat(savepath,'p_im3l_out_',pos_fname,'.txt');
  save(fpath,'p_im3l_dutout_mt', '-ASCII')
  fpath = strcat(savepath,'p_im3h_out_',pos_fname,'.txt');
  save(fpath,'p_im3h_dutout_mt', '-ASCII')

  % Indicated power of cancellation source for two-tone measurements
  p_cncl_comb_mtL = p_cncl_src_mtL;
  p_cncl_comb_mtH = p_cncl_src_mtH;

  % Compute cancellation levels for IM3 and AM-AM measurements
  p_im3l_cancellation = p_cncl_meas_mtL - p_im3l_meas_mt;
  p_im3h_cancellation = p_cncl_meas_mtH - p_im3h_meas_mt;
  p_amam_cancellation = p_cncl_meas_amam - p_f1_meas_amam;
  save([savepath,'im3l_cncl_dB_',pos_fname,'.txt'],...
        'p_im3l_cancellation', '-ASCII')
  save([savepath,'im3h_cncl_dB_',pos_fname,'.txt'],...
```

```
      'p_im3h_cancellation', '-ASCII')
  %save([savepath,'amam_cncl_dB','.txt'],'p_amam_cancellation', '-ASCII')

% Correct for phase shifts introduced by attenuator steps.  This is a
% simple subtraction of the affect of the attenuators on the output phase
% from the signal generators.  It is only necessary to correct for the
% phase jumps produced by the cancellation generator for the two-tone tests
% since the stimulus generators are always used in attenuator lock mode and
% do not have the attenuator step affects.  In the single tone
% measurements, both the stimulus generator and the cancellation generator
% use attenuator lock and therefore do not need to be corrected.  The range
% of output power for which attenuator lock can be used is ~28 dB.

  % Phase correction will be performed in the time basis, i.e. normalized
  % phase shift with frequency, phi/omega.  In order to correct for the
  % attenuator shift, need to do the following:
  %     - determine the attenuator band that the reference power was
  %     measured in
  %     - segment the phase data by power to match the attenuator bands
  %     - look up the attenuator phase shifts in the calibration data
  %     - adjust entire power segments by the appropriate shift for
  %     cumulative effect of attenuator shifts

  % Load calibration data for size of attenuator steps at each of six
  % points in power sweep from -70 dBm to +13 dBm.  Data is in frequency
  % (row) vs attenuator step (col).  There are six steps at different power
  % outputs. format with the following indices.
  %     - f = 400e6:(2400e6-400e6)/200:2400e6;
  %     - pstep = [-58 -47 -36 -25 -14 -3];
  %     ***Note these steps are the powers where they occur when using the
  %     knob to control power, not arrows the difference is 1 dB in the
  %     switch points.
  atten_tstep = load('C:\Experiments\jan_05\tues_25\atten_phase_step.txt');

  % Cal data indices
  f = 400e6:(2400e6-400e6)/200:2400e6;
  pstep = [-80 -69 -58 -47 -36 -25 -14 -3];

  % Generate interpolated vector of step sizes for frequency of operation
  f_op_IM3L = 450e6 - deltaf;
  f_op_IM3H = 450e6 + 2*deltaf;
  atten_tstep_mtL = interp1(f,atten_tstep,f_op_IM3L);
  atten_tstep_mtH = interp1(f,atten_tstep,f_op_IM3H);
  %!!!!!!!!!!!!!!!!!!!!!!!!
```

```
% Compensate for the attenuator phase step for data below -70 dBm
% indicated power on the cancellation source.  Since the attenuator step
% phase insertion appears cyclic, will use the next step size (average of
% two that occur in the calibration set taken).  For this data, the
% cancellation power source hits the attenuator switch at both -70 and
% -81 dBm corresponding to steps 3 and 6, and 2 and 5, respectively in
% the measured calibration data.
atten_step_correct_mtL=(atten_tstep_mtL([2,3])+atten_tstep_mtL([5,6]))/2;
atten_step_correct_mtH=(atten_tstep_mtH([2,3])+atten_tstep_mtH([5,6]))/2;

atten_tstep_mtL = [atten_step_correct_mtL atten_tstep_mtL];
atten_tstep_mtH = [atten_step_correct_mtH atten_tstep_mtH];


% Determine attenuator band in which the reference power was taken.  All
% attenuator effects are based on this power representing zero phase for
% the cancellation source.  The reference power for Hittite amplifier is
% 0 dBm indicated.  The index of this point will be found and used to
% find the corresponding IM3 cancellation power index.
p_ind_ref = 0;
ref_index = find(p_ref1_ind_amam == 0);
mt_ref_index = find(p_ref1_ind_mt == 0);
p_cncl_amam_ref = p_cncl_comb_amam(ref_index);
p_im3l_ref = p_cncl_comb_mtL(mt_ref_index);
p_im3h_ref = p_cncl_comb_mtH(mt_ref_index);

% Assume reference is not in band 0, this step places the reference
% position in one of the attenuator bands 1-6.
atten_bnd_im3l = floor(interp1(pstep,1:8,p_im3l_ref));
atten_bnd_im3h = floor(interp1(pstep,1:8,p_im3h_ref));

% Compute the cancellation power outputs that fall within each attenuator
% band for each of the IM3 products
I_atten0_mtL = find((p_cncl_comb_mtL < pstep(1)));
I_atten1_mtL = find((p_cncl_comb_mtL >=  pstep(1)) & ...
                    (p_cncl_comb_mtL <pstep(2)));
I_atten2_mtL = find((p_cncl_comb_mtL >=  pstep(2)) & ...
                    (p_cncl_comb_mtL <pstep(3)));
I_atten3_mtL = find((p_cncl_comb_mtL >=  pstep(3)) & ...
                    (p_cncl_comb_mtL <pstep(4)));
I_atten4_mtL = find((p_cncl_comb_mtL >=  pstep(4)) & ...
                    (p_cncl_comb_mtL <pstep(5)));
I_atten5_mtL = find((p_cncl_comb_mtL >=  pstep(5)) & ...
                    (p_cncl_comb_mtL <pstep(6)));
```

```
  I_atten6_mtL = find((p_cncl_comb_mtL >=  pstep(6)) & ...
                      (p_cncl_comb_mtL <pstep(7)));
  I_atten7_mtL = find((p_cncl_comb_mtL >=  pstep(7)) & ...
                      (p_cncl_comb_mtL <pstep(8)));
  I_atten8_mtL = find((p_cncl_comb_mtL >=  pstep(8)));


  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % For the Hittite 10 kHz separation data, there is a problem with the fifth data
  % point such that it should fall within the next higher attenuator band and thus
  % require a shift in attenuator phase at the fifth point.

  I_atten0_mtH = find((p_cncl_comb_mtH < pstep(1)));
  I_atten1_mtH = find((p_cncl_comb_mtH >=  pstep(1)) & ...
                      (p_cncl_comb_mtH <pstep(2)));
  I_atten2_mtH = find((p_cncl_comb_mtH >=  pstep(2)) & ...
                      (p_cncl_comb_mtH <pstep(3)));
  I_atten3_mtH = find((p_cncl_comb_mtH >=  pstep(3)) & ...
                      (p_cncl_comb_mtH <pstep(4)));
  I_atten4_mtH = find((p_cncl_comb_mtH >=  pstep(4)) & ...
                      (p_cncl_comb_mtH <pstep(5)));
  I_atten5_mtH = find((p_cncl_comb_mtH >=  pstep(5)) & ...
                      (p_cncl_comb_mtH <pstep(6)));
  I_atten6_mtH = find((p_cncl_comb_mtH >=  pstep(6)) & ...
                      (p_cncl_comb_mtH <pstep(7)));
  I_atten7_mtH = find((p_cncl_comb_mtH >=  pstep(7)) & ...
                      (p_cncl_comb_mtH <pstep(8)));
  I_atten8_mtH = find((p_cncl_comb_mtH >=  pstep(8)));
%   I_atten6_mtH = find((p_cncl_comb_mtH >=  pstep(6)));

  % Correction for 100k data
%   I_atten0_mtL = [1];
%   I_atten1_mtL = [2 3 4];
%   I_atten2_mtL = [5 6 7];
%   I_atten4_mtL = [10 11 12 13];
%   I_atten5_mtL = [14 15 16 17 18 19];
%   I_atten1_mtH = [3 4];
%   I_atten2_mtH = [5 6 7];
%   I_atten4_mtH = [10 11 12 13];
%   I_atten5_mtH = [14 15 16 17 18 19];

  % Correction for 50k data
%   I_atten4_mtL = [10 11 12 13];
%   I_atten5_mtL = [14 15 16 17 18 19];
```

```
%   I_atten1_mtH = [3 4];
%   I_atten2_mtH = [5 6 7];
%   I_atten4_mtH = [10 11 12 13];
%   I_atten5_mtH = [14 15 16 17 18 19];

  % Correction for extended 10k data
  I_atten1_mtL = [2 3 4];
  I_atten2_mtL = [5 6 7];
  I_atten3_mtL = [8 9];
  I_atten4_mtL = [10 11 12];
  I_atten1_mtH = [2 3 4];
  I_atten2_mtH = [5 6 7];
  I_atten3_mtH = [8 9];
  I_atten4_mtH = [10 11 12];

  % Correction for 10k data
%    I_atten1_mtH = [3 4];
%    I_atten2_mtH = [5 6 7];
  % Correction for 5k data
%     I_atten1_mtL = [3 4];
%     I_atten2_mtL = [5 6 7];
%     I_atten1_mtH = [3 4];
%     I_atten2_mtH = [5 6 7];
  % Correction for 1k data, none

  % Now correct for effects of attenuator switch points away from the
  % reference cancellation power, for the IM3 phase only
    % Normalize the measured phase shift with frequency of operation,
    % operate on radian measurement
    phase_meas_norm_mtL = pi/180*phase_meas_mtL / (2*pi*f_op_IM3L);
    phase_meas_norm_mtH = pi/180*phase_meas_mtH / (2*pi*f_op_IM3H);

    % Since the reference power is at some intermediate attenuator band,
    % adjust the entire measured phase by the full attenuator band
    % cumulative sum and then use the sum up to the band where the
    % reference power point lies and back out this amount as the correction
    % for the zero degree reference point.

    step_correct_im3l = zeros(size(phase_meas_norm_mtL));
    step_correct_im3h = zeros(size(phase_meas_norm_mtH));
    step_correct_im3l([I_atten1_mtL(1), I_atten2_mtL(1), ...
                       I_atten3_mtL(1), I_atten4_mtL(1), ...
                       I_atten5_mtL(1), I_atten6_mtL(1), ...
                       I_atten7_mtL(1)]) = atten_tstep_mtL(1:7);
```

```
    step_correct_im3h([I_atten1_mtH(1), I_atten2_mtH(1), ...
                       I_atten3_mtH(1), I_atten4_mtH(1), ...
                       I_atten5_mtH(1), I_atten6_mtH(1), ...
                       I_atten7_mtH(1)]) = atten_tstep_mtH(1:7);

    phase_meas_norm_flat_mtL = phase_meas_norm_mtL + ...
                               cumsum(step_correct_im3l,1);
    phase_meas_norm_flat_mtH = phase_meas_norm_mtH + ...
                               cumsum(step_correct_im3h,1);

    % Now back out the cumulative sum of the reference power point
    phase_meas_norm_adj_mtL = phase_meas_norm_flat_mtL - ...
                              sum(atten_tstep_mtL(1:atten_bnd_im3l));
    phase_meas_norm_adj_mtH = phase_meas_norm_flat_mtH - ...
                              sum(atten_tstep_mtH(1:atten_bnd_im3h));

%     figure
%     plot(p_ref1_ind_mt, phase_meas_norm_mtL, '-*', ...
%     p_ref1_ind_mt, cumsum(step_correct_im3l,1), ...
%     p_ref1_ind_mt, phase_meas_norm_adj_mtL,'-x')

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % At this point, the data should be completely corrected for the effects
  % of the phase insertion caused by the attenuator switch points.  Now the
  % effect of the linear phase shift with power of the sources needs to be
  % addressed.

  % Correction of phase shift with change in source power follows the
  % theoretical development
  %
  % -- load calibration data for slope of phase shift vs frequency
  % -- interpolate table for operating frequency, both stimulus source and
  % cancellation source
  % -- use equations for corrections developed in theoretical work
  % -- plot raw data versus attenuator and generator phase shift data

    % Load calibration data for slope of phase shift vs frequency.  Data is
    % in vector of slope vs frequency for 201 points, linearly spaced from
    % 400 - 2400 MHz.
    m_phi = load('C:\Experiments\jan_05\tues_25\phase_slope.txt');

    % Interpolate for the frequencies used in the sources for the single
    % tone data
    f_op_amam = 450e6;
```

```
    m_phi_ref1_amam = interp1(f, m_phi, f_op_amam);
    m_phi_cncl_amam = m_phi_ref1_amam;

    % Interpolate for the frequencies used in the two tone data
    f_op_ref1_mt = 450e6;
    f_op_ref2_mt = 450e6 + deltaf;

    m_phi_ref1_mt = interp1(f, m_phi, f_op_ref1_mt);
    m_phi_ref2_mt = interp1(f, m_phi, f_op_ref2_mt);
    m_phi_cncl_mtL = interp1(f, m_phi, f_op_IM3L);
    m_phi_cncl_mtH = interp1(f, m_phi, f_op_IM3H);

    % Correct measured phase for AM-AM AM-PM data
    % Power difference for sources is that of indicated power from the
    % reference power since indicated power was that chosen as the
    % reference.
    phase_corr_amam = phase_meas_amam*pi/180/(2*pi*f_op_amam) + ...
              m_phi_cncl_amam*(p_cncl_comb_amam - p_cncl_amam_ref) - ...
              m_phi_ref1_amam*(p_ref1_ind_amam - p_ind_ref);
    fpath = strcat(savepath,'phase_amam.txt');
    %save(fpath,'phase_corr_amam','-ASCII');

%     figure
%     corram1 = m_phi_cncl_amam*(p_cncl_comb_amam - p_cncl_amam_ref);
%     corram2 = m_phi_ref1_amam*(p_ref1_ind_amam - p_ind_ref);
%     data = phase_meas_amam*pi/180/(2*pi*f_op_amam);
%     plot(p_ref1_ind_amam, corram1, p_ref1_ind_amam, corram2, ...
%          p_ref1_ind_amam, data, p_ref1_ind_amam, data+corram1-corram2)

    % Correct measured phase for two-tone data
    phase_corr_mtL = phase_meas_norm_adj_mtL + ...
                     m_phi_cncl_mtL*(p_cncl_comb_mtL - p_im3l_ref) - ...
                     m_phi_ref1_mt*(p_ref1_ind_mt - p_ind_ref);
    phase_corr_mtH = phase_meas_norm_adj_mtH + ...
                     m_phi_cncl_mtH*(p_cncl_comb_mtH - p_im3h_ref) - ...
                     m_phi_ref2_mt*(p_ref1_ind_mt - p_ind_ref);

    % !!! Change reference phase point to lower input power
    phase_corr_mtL = phase_corr_mtL - phase_corr_mtL(2);
    phase_corr_mtH = phase_corr_mtH - phase_corr_mtH(2);

    fpath = strcat(savepath,'phase_mtL_',pos_fname,'.txt');
    save(fpath,'phase_corr_mtL','-ASCII')
    fpath = strcat(savepath,'phase_mtH_',pos_fname,'.txt');
```

```
    save (fpath,'phase_corr_mtH', '-ASCII')


%     figure
%     corr1 = m_phi_cncl_mtL*(p_cncl_comb_mtL - p_im3l_ref);
%     corr2 = m_phi_ref1_mt*(p_ref1_ind_mt - p_ind_ref);
%     corr3 = cumsum(step_correct_im3l,1);
%     plot(p_ref1_ind_mt, corr1, p_ref1_ind_mt, corr2, ...
%          p_ref1_ind_mt, phase_meas_norm_adj_mtL, ...
%          p_ref1_ind_mt, corr1-corr2+corr3+phase_meas_norm_mtL)


    % Now plot raw measured data and corrected data
    figure
    [AX, H1, H2] = plotyy(p_ref1_ind_amam, ...
         [phase_meas_amam phase_corr_amam*2*pi*f_op_amam*180/pi], ...
                          p_ref1_ind_amam, p_f1_dutout_amam);
    ftitle('Hittite Raw vs Calibrated phase for AM-AM')
    axes(AX(1))
    xlabel('Input power (dBm)')
    ylabel('Phase (degrees)')
    axes(AX(2))
    ylabel('Output power (dBm)')
    legend([H1;H2], 'Uncorrected Phase', 'Corrected Phase', 'Mag. f_1',2)
    set(H1(1),'LineWidth', 1.5, 'LineStyle', '-')
    set(H1(2),'LineWidth', 1.5, 'LineStyle', '--')
    set(H2(1),'LineWidth', 1.5, 'LineStyle', '-.')


    figure
    plot(p_ref1_ind_mt, phase_meas_mtL, '+', ...
         p_ref1_ind_mt, phase_corr_mtL*2*pi*f_op_IM3L*180/pi, ...
         'LineWidth', 1.5)
    ftitle('Hittite Raw vs Calibrated phase for Two-tone IM3L')
    xlabel('Input power (dBm)')
    ylabel('Phase (degrees)')
    legend('Measured', 'Corrected')

    figure
    plot(p_ref1_ind_mt, phase_meas_mtH, '+', ...
         p_ref1_ind_mt, phase_corr_mtH*2*pi*f_op_IM3H*180/pi, ...
         'LineWidth', 1.5)
    ftitle('Hittite Raw vs Calibrated phase for Two-tone IM3H')
    xlabel('Input power (dBm)')
```

```
ylabel('Phase (degrees)')
legend('Measured', 'Corrected')

figure
[AX, H1, H2] = plotyy(p_ref1_ind_mt,
                      [phase_corr_mtL*2*pi*f_op_IM3L*180/pi, ...
                       phase_corr_mtH*2*pi*f_op_IM3H*180/pi], ...
                      p_ref1_ind_mt, ...
                      [p_im3l_dutout_mt, p_im3h_dutout_mt]);
ftitle('Hittite Calibrated phase for IM3L and IM3H')
axes(AX(1))
axis([-11 14 -10 50])
ylabel('Phase (degrees)')
axes(AX(2))
axis([-11 14 -90 10])
ylabel('IM3 power (dBm)')
xlabel('Input power (dBm)')
legend([H1 H2], 'Phase IM3L', 'Phase IM3H', 'Mag. IM3L', 'Mag. IM3H',0)
set(H1(1),'LineWidth', 1.5, 'LineStyle', '-')
set(H1(2),'LineWidth', 1.5, 'LineStyle', '--')
set(H2(1),'LineWidth', 1.5, 'LineStyle', '-.')
set(H2(2),'LineWidth', 1.5, 'LineStyle', ':')
```

D.3.2 Model Parameter Extraction and Output Generation Script for Manual Selection of Third and Fifth-Order Coefficients

```
% This script processes the 10 kHz separation two-tone data taken at
% 450 MHz on the Hittite PA.  This is the second set of data taken
% on this amplifier with the hybrid combiner used at the
% output of the DUT.  The script plots the measured IM3
% magnitude and phase as polar plots with comparison to the
% expected output based on single tone measurements.

% In this version, the a3 and a5 coefficients of the single-tone polynomial
% are determined by using the two tone measured data for the IM3 products
% at small input power where the third and fifth order terms dominate.  The
% contributions of these terms to the fundamental are removed prior to
% fitting the a polynomial to the single tone data.  This should improve
% the estimation of the IM3 data by the single-tone fit.

% Read in the corrected measured magnitude and phase data.  Magnitude
% data is that corrected for path loss from DUT to spectrum analyzer
% and corrected phase data takes into account the attenuator and
```

```
% phase insertion with power of the IFR sources.
clear all, close all, cd C:\Experiments\mar_05\mon_7 f_op_amam =
450e6; deltaf = 10e3; refindex = 2; f_op_IM3L = 450e6 - deltaf;
f_op_IM3H = 450e6 + 2*deltaf;
% read_path = 'C:\Experiments\mar_05\wed_23\hittite\';
amam_read_path = 'C:\Experiments\mar_05\wed_23\hittite\';
read_path = 'C:\Experiments\may_05\tues_31\'; im3l_mag_p =
load([read_path,'p_im3l_out_10k.txt']); im3h_mag_p =
load([read_path,'p_im3h_out_10k.txt']); im3l_phase =
load([read_path,'phase_mtL_10k.txt']); im3h_phase =
load([read_path,'phase_mtH_10k.txt']); ref1_mag_p =
load([read_path,'p_ref1_in_mt.txt']); ref2_mag_p =
load([read_path,'p_ref2_in_mt.txt']); amam_mag_p =
load([amam_read_path,'p_amam_out.txt']); amam_phase =
load([amam_read_path,'phase_amam.txt']); amam_magin_p =
load([amam_read_path,'p_amam_in.txt']);

amam_cncl_p = load([amam_read_path,'amam_cncl_dB.txt']);
im3l_cncl_p = load([read_path,'im3l_cncl_dB_10k.txt']);
im3h_cncl_p = load([read_path,'im3h_cncl_dB_10k.txt']);

% Convert power in dBm to voltage magnitude, assume 50 Ohm
% impedance
im3l_mag_v = sqrt(10.^(im3l_mag_p/10)/10); im3h_mag_v =
sqrt(10.^(im3h_mag_p/10)/10); f1_mag_v =
sqrt(10.^(ref1_mag_p/10)/10); f2_mag_v =
sqrt(10.^(ref2_mag_p/10)/10); amam_mag_v =
sqrt(10.^(amam_mag_p/10)/10); amam_magin_v =
sqrt(10.^(amam_magin_p/10)/10);

% Convert phase data to radians, saved as frequency normalized
phase_amam_rad = amam_phase * 2*pi*f_op_amam;
phase_amamin_rad = zeros(size(phase_amam_rad));  % Assume input phase is 0
im3l_phase_rad = im3l_phase * 2*pi*f_op_IM3L; im3h_phase_rad =
im3h_phase * 2*pi*f_op_IM3H;

% Plot magnitude and phase of IM3 products on polar plot
% figure
% polar(im3l_phase_rad, im3l_mag_v)
% hold
% polar(im3h_phase_rad, im3h_mag_v, 'g')

% Fit complex polynomial to AM-AM, AM-PM data in the form of a complex
% gain.
```

```
% Convert polar form of data to complex form
[x_amam, y_amam] = pol2cart(phase_amam_rad, amam_mag_v);
vout_cplx_amam = complex(x_amam, y_amam);
[x_amamin, y_amamin] = pol2cart(phase_amamin_rad, amam_magin_v);
vin_cplx_amam = complex(x_amamin, y_amamin);

f1_mag_vcplx = complex(f1_mag_v, zeros(size(f1_mag_v)));

% Compute complex voltage gain
G_cplx = vout_cplx_amam./vin_cplx_amam;

% Plot complex gain for both voltage and power
figure
subplot(2,1,1)
plot(amam_magin_p, amam_mag_p - amam_magin_p,'-^');
xlabel('Input Tone Power (dBm)')
ylabel('Power Gain (dB)')
subplot(2,1,2)
plot(amam_magin_p, phase_amam_rad*180/pi,'-^');
xlabel('Input Tone Power (dBm)')
ylabel('Output Phase ( \circ)')
ftitle('Measured Magnitude of Complex Gain')

% Extrapolate the complex gain, magnitude and phase separately
% Currently these are fit with linear polynomials over their entire
% range in the voltage domain.
    % Perform linear fit
    lin_fit_mag = polyfit(abs(vin_cplx_amam), abs(G_cplx),1);
    lin_fit_ang = polyfit(abs(vin_cplx_amam), angle(G_cplx),1);
    % Generate extended voltage input
    p_ext = 11:16;
    v_ext = sqrt(10.^(p_ext/10)/10);
    % Generate gain magnitude extension
    G_mag_ext = polyval(lin_fit_mag, v_ext);
    % Generate gain phase extension
    G_ang_ext = polyval(lin_fit_ang, v_ext);
    % Form the extended voltage gain expression
    G_cplx_ext = [real(vout_cplx_amam./vin_cplx_amam); ...
                  (G_mag_ext.*cos(G_ang_ext))'] + ...
                  sqrt(-1)*[imag(vout_cplx_amam./vin_cplx_amam); ...
                  (G_mag_ext.*sin(G_ang_ext))'];
    % Form the extended input voltage
    vin_cplx_amam_ext = [vin_cplx_amam; v_ext'];
```

```
        % Form the extended output voltage
        vout_cplx_amam_ext = vin_cplx_amam_ext.*G_cplx_ext;



    % Fit a complex polynomial to the voltage gain.
    %
    % First form the negative voltage data by negating the
    % complex input and output voltages, forcing the output voltage
    % to be zero at zero input.  By negating the positive only
    % input/output voltage, the transfer function will automatically
    % be odd.
    %
    % Next use the polyfit function to determine the coefficients.
    %
    % Finally, use polyval and plot the results to visually determine
    % if a polynomial of significant order was used.

    % !!! Test to determine if backing out third and fifth order
    % coefficients from measured IM3 data can improve the IM3 fit with a
    % single slice.  Coefficients extracted over a range of IM3 data where
    % the contributions of the particular order dominate.

    % Compute G_coefa third order term
    c3 = multinomial([0 1 2 0 ], 3);
%     [X, Y] = pol2cart((im3l_phase_rad)/1, (im3l_mag_v)/1);
%     im3_cplx_v = complex(X,Y);
%     a3_3 = (im3_cplx_v)./(c3*f1_mag_v.^3);
    a3_3 = abs(im3l_mag_v)./(c3*f1_mag_v.^3);
    a3_3_avg = sum(a3_3(1:4))/size(a3_3(1:4),1);

    % Compute G_coefa fifth order term
    c5 = multinomial([1 1 3 0], 5) + multinomial([2 0 2 1], 5);
    a3_5 = abs(im3l_mag_v)./(abs(f1_mag_v).^5 * c5);
    a3_5_avg = sum(a3_5(6:11))/size(a3_5(6:11),1);

    % Determine angles for the a3 and a5 coefficients.  The goal here
    % was to use the measured IM3 phase data to get an estimate of what
    % the average phase difference between the fifth order and third
    % order terms should be over a range of input powers.  A grid of
    % a3 and a5 angles was used to plot the difference between the
    % estimated phase in the region dominated by the third and fifth
    % to see the minimum error point and thus determine the phase
    % difference.  Since the two components can only contribute a constant
```

```
% phase, it was hoped that this exercise would reveal the actual
% phase difference needed between the two components.  However, for
% the input power range from Pin = -13 to -8 dBm, the delta phase
% for the a3 and a5 was not constant or even close to constant.
% Therefore this technique did not readily appear to yield the
% information I sought.

C = tan(im3l_phase_rad);
phi = -pi:pi/40:pi;
m = 6;
[phi_a3 phi_a5] = meshgrid(phi, phi);
R = abs(a3_3_avg)*c3*cos(phi_a3)*abs(f1_mag_v(m)).^3 + ...
    abs(a3_5_avg)*c5*cos(phi_a5)*abs(f1_mag_v(m)).^5;
I = abs(a3_3_avg)*c3*sin(phi_a3)*abs(f1_mag_v(m)).^3 + ...
    abs(a3_5_avg)*c5*sin(phi_a5)*abs(f1_mag_v(m)).^5;

error = C(m)*R - I;

% Plot the mesh grid error function
figure
mesh(phi_a3, phi_a5, abs(error))
xlabel('\phi_{a_3}')
ylabel('\phi_{a_5}')

% Find the angles of a3 and a5 that resulted in the minimum error.
[Y J] = min(abs(error),[],1);
[X K] = min(abs(error),[],2);
[mina3 a3ind] = min(Y);
[mina5 a5ind] = min(X);
phi_a3(a5ind,a3ind)*180/pi, phi_a5(a5ind,a3ind)*180/pi,

%     C(m)*R(a5ind,a3ind) - I(a5ind,a3ind), error(a5ind, a3ind)
% Here set the angles of the coefficients manually.  No alternate
% method seemed available to setting the appropriate values.
a3_ang = 175*pi/180;
[X,Y] = pol2cart(a3_ang,abs(a3_3_avg));
a3 = complex(X,Y);
a5_ang_adj = (280)*pi/180;
[X,Y] = pol2cart(a5_ang_adj,abs(a3_5_avg));
a5 = complex(X,Y);

% Test to determine what the order of a polynomial to fit the
% IM3 data only is.  The result was a 41 order fit where the
% majority of the orders was needed to fit the phase of the IM3
```

```
    % products.
%     figure
%     im3l_cplx = complex(im3l_mag_v.*cos(im3l_phase_rad), ...
%                         im3l_mag_v.*sin(im3l_phase_rad));
%     q = 24;
%     im3coef = polyfit([-fliplr(abs(f1_mag_v(1:q))); 0 ;...
%                        abs(f1_mag_v(1:q))], ...
%                       [-fliplr(im3l_cplx(1:q)); 0 ;im3l_cplx(1:q)], 41);
%     im3fit = polyval(im3coef, abs(f1_mag_v(1:q)));
%     subplot(2,1,1)
%     plot(10*log10(abs(f1_mag_v(1:q)).^2*10), ...
%          10*log10(im3l_mag_v(1:q).^2*10),'-+', ...
%          10*log10(abs(f1_mag_v(1:q)).^2*10), 10*log10(abs(im3fit).^2*10))
%     subplot(2,1,2)
%     plot(10*log10(abs(f1_mag_v(1:q)).^2*10), ...
%          im3l_phase_rad(1:q)*180/pi, '-+', ...
%          10*log10(abs(f1_mag_v(1:q)).^2*10), angle(im3fit)*180/pi)
%     angle(im3coef)*180/pi;

    % Compute G_coefb from G_coefa terms to allow backing out the
    % single tone effects of these terms.
    ax = [a3 a5];
    bx = zeros(size(ax));
    for n = 1:2
        alpha = n;
        alpha2 = 2*alpha;
        bx(n) = ax(n) * factorial(1+alpha2)/(2^alpha2*factorial(alpha)*...
                factorial(1+alpha));
    end

    % Subtract b3*vin.^3 + b5*vin.^5 from the single tone data
    vout_cplx_amam_ext_m3m5 = vout_cplx_amam_ext - ...
                              bx(1)*vin_cplx_amam_ext.^3 - ...
                              bx(2)*vin_cplx_amam_ext.^5;

    N = 17;      % Order of polynomial
    v_in_amam_lr = [fliplr(-vin_cplx_amam_ext); 0; vin_cplx_amam_ext];
    v_out_amam_lr = [fliplr(-vout_cplx_amam_ext_m3m5); 0; ...
                     vout_cplx_amam_ext_m3m5];
%     G_coefb = polyfit(v_in_amam_lr, v_out_amam_lr, N);
%     G_coefb = polyfit_no_third_aw(v_in_amam_lr, v_out_amam_lr, N);
    G_coefb = polyfit_no_third_fifth_aw(v_in_amam_lr, v_out_amam_lr, N);
    G_coefb(N-2) = bx(1);   % Third order term
    G_coefb(N-4) = bx(2);   % Fifth order term
```

```
    % Evaluate the polynomial fit for the positive voltage data
    % and compare to the measured data
    v_fit_amam = polyval(G_coefb, vin_cplx_amam);
    figure
    ftitle('Single tone fit')
    plot(10*log10(abs(vin_cplx_amam).^2*10), ...
         10*log10(abs(vout_cplx_amam).^2*10), '-*', ...
         10*log10(abs(vin_cplx_amam).^2*10), ...
         10*log10(abs(v_fit_amam).^2*10))
    legend('Meas.','Est.',0)

    v_fit_amam_ext = polyval(G_coefb, vin_cplx_amam_ext);
    figure
    subplot(2,1,1)
    ext = size(vin_cplx_amam,1);
    ftitle('Single tone fit extrapolated')
%     plot(abs(vin_cplx_amam), abs(vout_cplx_amam),'d', ...
%          abs(vin_cplx_amam_ext(ext+1:end)), ...
%          abs(vout_cplx_amam_ext(ext+1:end)), '^', ...
%          abs(vin_cplx_amam_ext), abs(v_fit_amam_ext),'--')
    plot(10*log10(abs(vin_cplx_amam).^2*10), ...
         10*log10(abs(vout_cplx_amam).^2*10),'d', ...
         10*log10(abs(vin_cplx_amam_ext(ext+1:end)).^2*10), ...
         10*log10(abs(vout_cplx_amam_ext(ext+1:end)).^2*10), '^', ...
         10*log10(abs(vin_cplx_amam_ext).^2*10), ...
         10*log10(abs(v_fit_amam_ext).^2*10),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Peak Voltage (V)')
    legend('Meas.','Ext.','Est.',0)
%     title('Extrapolated Single-tone Fit')
    subplot(2,1,2)
%     plot(abs(vin_cplx_amam), angle(vout_cplx_amam),'d', ...
%          abs(vin_cplx_amam_ext(ext+1:end)), ...
%          angle(vout_cplx_amam_ext(ext+1:end)), '^', ...
%          abs(vin_cplx_amam_ext), angle(v_fit_amam_ext),'--')
    plot(10*log10(abs(vin_cplx_amam).^2*10), angle(vout_cplx_amam),'d', ...
         10*log10(abs(vin_cplx_amam_ext(ext+1:end)).^2*10), ...
         angle(vout_cplx_amam_ext(ext+1:end)), '^', ...
         10*log10(abs(vin_cplx_amam_ext).^2*10),angle(v_fit_amam_ext),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Phase ( \circ)')
    legend('Meas.','Ext.','Est.',0)
```

```
% Now use the complex coefficients to estimate the IM3 product magnitude
% and phase.
%
% First need to take into account that the coefficients that were fit, were
% the average coefficients and so need to be backed out to the
% instantaneous coefficients. Following work done in RAWCON 2004 paper and
% that of Gard:Gutierrez:Steer:1999 and Gharaibeh:Steer:2005

    % Compute instantaneous coefficients
    % Only the odd coefficients will be retained since the even terms are
    % approximately zero from fit.
    G_coefa = zeros(size(G_coefb));

    G_coefa(N+1) = G_coefb(N+1);          % DC components are equal
    for j = 1:2:N
        alpha = (N+1)/2 - (j+1)/2;
        alpha2 = alpha*2;
        G_coefa(j) = G_coefb(j)*2^alpha2*factorial(alpha)*...
                     factorial(1+alpha)/factorial(1+alpha2);
    end

    % Repeat of computation of single tone fit using development in
    % Khaled's paper wiht instantaneous coefficients
    v_fit_amam2 = 0;
    v_fit_amam_comp = zeros(size(vin_cplx_amam,1),size(1:2:N,2));
    for n = 1:2:N,
        m = [ 0 0 (n-1)/2 (n+1)/2];
        mco = multinomial(m,n);
        temp = G_coefa(N-n+1)*mco*(vin_cplx_amam).^n;
        v_fit_amam2 = v_fit_amam2 + temp;
        v_fit_amam_comp(:,(n+1)/2) = temp;
    end

    % Plots comparing two sets of coefficients fit, expected identical, and
    % the phase
    % of the single tone measured and expected data.
    figure
    ftitle(['Comparison of fit between non-extrapolated single tone '...
            'using polyval and that using the envelope formulation'])
    plot(abs(vin_cplx_amam), abs(v_fit_amam),'+-',...
         abs(vin_cplx_amam), abs(v_fit_amam2))
    figure
    ftitle(['Comparison of phase fit between envelope formulation and ' ...
```

```
       'measured phase of AM-AM data'])
plot(abs(vin_cplx_amam), angle(v_fit_amam2)*180/pi, ...
     abs(vin_cplx_amam), phase_amam_rad*180/pi)
figure
ftitle(['Magnitude of individual odd-order contributors from '...
       'envelope formulation of AM-AM data'])
plot(amam_magin_p, 10*log10(abs(v_fit_amam_comp).^2*10))
legend('1','3','5','7','9',0)

% Now compute complex IM3 from coefficients and formulation based on
% Gharaibeh:Steer:2005
im3lfit1 = 0;
im3hfit1 = 0;
im3l_order_comp = zeros(size(f1_mag_v,1),size(3:2:N,2));
for n = 3:2:N,
    %im3lfit1 =0;
    for l = 0:(n-3)/2,
        % Note, the - terms are L's not one's
        m = [((n-3)/2-l) (l+1) l ((n+1)/2 - l)];
        mco = multinomial(m, n);
        temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
        im3lfit1 = im3lfit1 + temp;
        temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
        im3hfit1 = im3hfit1 + temp;
        im3l_order_comp(:,(n-1)/2) = im3l_order_comp(:,(n-1)/2) + temp;
    end
end

figure
ftitle('Individual odd-order contributions to IM3 product')
plot(ref1_mag_p, 10*log10(abs(im3l_order_comp).^2*10))
legend('3','5','7','9',0)

figure
ftitle('Individual odd-order contributions to IM3 product, polar')
polar(angle(im3l_order_comp), abs(im3l_order_comp))
legend('3','5','7','9',0)

% Compute phase difference between measured IM3 phase (at 0 degree
% reference power) and the estimated phase from the fit
im3l_phase_fit = unwrap(angle(im3lfit1));
diff = im3l_phase_fit(refindex)*180/pi - ...
       im3l_phase_rad(refindex)*180/pi;
```

```
 figure
ftitle('Comparison of IM3 magnitude measured and single-slice estimate')
 plot(ref1_mag_p, 10*log10(abs(im3lfit1).^2*10), ...
      ref1_mag_p, im3l_mag_p, ref1_mag_p, im3h_mag_p)

 figure
 ftitle('Comparison of IM3 phase measured and single-slice estimate')
 plot(ref1_mag_p, unwrap(angle(im3lfit1))*180/pi, ...
      ref1_mag_p, unwrap(im3l_phase_rad+diff*pi/180)*180/pi, ...
      ref1_mag_p, unwrap(im3h_phase_rad+diff*pi/180)*180/pi)

 figure
 plot(10*log10(abs(vin_cplx_amam_ext).^2*10), ...
      angle(vout_cplx_amam_ext)*180/pi, ...
      10*log10(abs(vin_cplx_amam_ext).^2*10), ...
      angle(vout_cplx_amam_ext_m3m5)*180/pi)
```

D.3.3 Model Parameter Extraction and Output Generation Script for Genetic Algorithm Fitting Procedure

```
% This script processes the 10 kHz separation two-tone data taken at
% 450 MHz on the Hittite PA.  This is the second set of data taken
% on this amplifier with the hybrid combiner used at the
% output of the DUT.  The script plots the measured IM3
% magnitude and phase as polar plots with comparison to the
% expected output based on single tone measurements.

% In this version, I will employ the GA toolbox to determine
% a better fit to the two-tone phase data using both the
% single tone and two tone data sets.


% Read in the corrected measured magnitude and phase data.  Magnitude
% data is that corrected for path loss from DUT to spectrum analyzer
% and corrected phase data takes into account the attenuator and
% phase insertion with power of the IFR sources.
clear all, close all, cd C:\Experiments\mar_05\mon_7

% Declare globals
global vout_cplx_amam; global vin_cplx_amam; global f1_mag_v;
global im3_meas_avg_cplx; global nvars; global popsize;
```

```
f_op_amam = 450e6; deltaf = 10e3; refindex = 2; f_op_IM3L = 450e6
- deltaf; f_op_IM3H = 450e6 + 2*deltaf;
% read_path = 'C:\Experiments\mar_05\wed_23\hittite\';
amam_read_path = 'C:\Experiments\mar_05\wed_23\hittite\';
read_path = 'C:\Experiments\may_05\tues_31\'; im3l_mag_p =
load([read_path,'p_im3l_out_10k.txt']); im3h_mag_p =
load([read_path,'p_im3h_out_10k.txt']); im3l_phase =
load([read_path,'phase_mtL_10k.txt']); im3h_phase =
load([read_path,'phase_mtH_10k.txt']); ref1_mag_p =
load([read_path,'p_ref1_in_mt.txt']); ref2_mag_p =
load([read_path,'p_ref2_in_mt.txt']); amam_mag_p =
load([amam_read_path,'p_amam_out.txt']); amam_phase =
load([amam_read_path,'phase_amam.txt']); amam_magin_p =
load([amam_read_path,'p_amam_in.txt']);

amam_cncl_p = load([amam_read_path,'amam_cncl_dB.txt']);
im3l_cncl_p = load([read_path,'im3l_cncl_dB_10k.txt']);
im3h_cncl_p = load([read_path,'im3h_cncl_dB_10k.txt']);

% Convert power in dBm to voltage magnitude, assume 50 Ohm
% impedance
im3l_mag_v = sqrt(10.^(im3l_mag_p/10)/10); im3h_mag_v =
sqrt(10.^(im3h_mag_p/10)/10); f1_mag_v =
sqrt(10.^(ref1_mag_p/10)/10); f2_mag_v =
sqrt(10.^(ref2_mag_p/10)/10); amam_mag_v =
sqrt(10.^(amam_mag_p/10)/10); amam_magin_v =
sqrt(10.^(amam_magin_p/10)/10);

% Convert phase data to radians, saved as frequency normalized
phase_amam_rad = amam_phase * 2*pi*f_op_amam;
phase_amamin_rad = zeros(size(phase_amam_rad));   % Assume input phase is 0
im3l_phase_rad = im3l_phase * 2*pi*f_op_IM3L; im3h_phase_rad =
im3h_phase * 2*pi*f_op_IM3H;

% Plot magnitude and phase of IM3 products on polar plot
% figure
% polar(im3l_phase_rad, im3l_mag_v)
% hold
% polar(im3h_phase_rad, im3h_mag_v, 'g')

% Fit complex polynomial to AM-AM, AM-PM data in the form of a complex
% gain.
```

```
% Convert polar form of data to complex form
[x_amam, y_amam] = pol2cart(phase_amam_rad, amam_mag_v);
vout_cplx_amam = complex(x_amam, y_amam);
[x_amamin, y_amamin] = pol2cart(phase_amamin_rad, amam_magin_v);
vin_cplx_amam = complex(x_amamin, y_amamin);

f1_mag_vcplx = complex(f1_mag_v, zeros(size(f1_mag_v)));

% Compute complex voltage gain
G_cplx = vout_cplx_amam./vin_cplx_amam;

% Plot complex gain for both voltage and power
figure
subplot(2,1,1)
plot(amam_magin_p, amam_mag_p - amam_magin_p,'-^');
xlabel('Input Tone Power (dBm)')
ylabel('Power Gain (dB)')
subplot(2,1,2)
plot(amam_magin_p, phase_amam_rad*180/pi,'-^');
xlabel('Input Tone Power (dBm)')
ylabel('Output Phase ( \circ)')
ftitle('Measured Magnitude of Complex Gain')

% Extrapolate the complex gain, magnitude and phase separately
% Currently these are fit with linear polynomials over their entire
% range in the voltage domain.
    % Perform linear fit
    lin_fit_mag = polyfit(abs(vin_cplx_amam), abs(G_cplx),1);
    lin_fit_ang = polyfit(abs(vin_cplx_amam), angle(G_cplx),1);
    % Generate extended voltage input
    p_ext = 11:16;
    v_ext = sqrt(10.^(p_ext/10)/10);
    % Generate gain magnitude extension
    G_mag_ext = polyval(lin_fit_mag, v_ext);
    % Generate gain phase extension
    G_ang_ext = polyval(lin_fit_ang, v_ext);
    % Form the extended voltage gain expression
    G_cplx_ext = [real(vout_cplx_amam./vin_cplx_amam); ...
                  (G_mag_ext.*cos(G_ang_ext))'] + ...
                  sqrt(-1)*[imag(vout_cplx_amam./vin_cplx_amam); ...
                  (G_mag_ext.*sin(G_ang_ext))'];
    % Form the extended input voltage
    vin_cplx_amam_ext = [vin_cplx_amam; v_ext'];
    % Form the extended output voltage
```

```
    vout_cplx_amam_ext = vin_cplx_amam_ext.*G_cplx_ext;


% Fit a complex polynomial to the voltage gain.
%
% First form the negative voltage data by negating the
% complex input and output voltages, forcing the output voltage
% to be zero at zero input.  By negating the positive only
% input/output voltage, the transfer function will automatically
% be odd.
%
% Next use the polyfit function to determine the coefficients.
%
% Finally, use polyval and plot the results to visually determine
% if a polynomial of significant order was used.

% !!! Test to determine if backing out third and fifth order
% coefficients from measured IM3 data can improve the IM3 fit with a
% single slice.  Coefficients extracted over a range of IM3 data where
% the contributions of the particular order dominate.

% Compute G_coefa third order term
c3 = multinomial([0 1 2 0 ], 3);
%   [X, Y] = pol2cart((im3l_phase_rad)/1, (im3l_mag_v)/1);
%   im3_cplx_v = complex(X,Y);
%   a3_3 = (im3_cplx_v)./(c3*f1_mag_v.^3);
a3_3 = abs(im3l_mag_v)./(c3*f1_mag_v.^3);
a3_3_avg = sum(a3_3(1:4))/size(a3_3(1:4),1);

% Compute G_coefa fifth order term
c5 = multinomial([1 1 3 0], 5) + multinomial([2 0 2 1], 5);
a3_5 = abs(im3l_mag_v)./(abs(f1_mag_v).^5 * c5);
a3_5_avg = sum(a3_5(6:11))/size(a3_5(6:11),1);

% Determine angles for the a3 and a5 coefficients.  The goal here
% was to use the measured IM3 phase data to get an estimate of what
% the average phase difference between the fifth order and third
% order terms should be over a range of input powers.  A grid of
% a3 and a5 angles was used to plot the difference between the
% estimated phase in the region dominated by the third and fifth
% to see the minimum error point and thus determine the phase
% difference.  Since the two components can only contribute a constant
% phase, it was hoped that this exercise would reveal the actual
```

```
% phase difference needed between the two components.  However, for
% the input power range from Pin = -13 to -8 dBm, the delta phase
% for the a3 and a5 was not constant or even close to constant.
% Therefore this technique did not readily appear to yield the
% information I sought.

C = tan(im3l_phase_rad);
phi = -pi:pi/40:pi;
m = 6;
[phi_a3 phi_a5] = meshgrid(phi, phi);
R = abs(a3_3_avg)*c3*cos(phi_a3)*abs(f1_mag_v(m)).^3 + ...
    abs(a3_5_avg)*c5*cos(phi_a5)*abs(f1_mag_v(m)).^5;
I = abs(a3_3_avg)*c3*sin(phi_a3)*abs(f1_mag_v(m)).^3 + ...
    abs(a3_5_avg)*c5*sin(phi_a5)*abs(f1_mag_v(m)).^5;

error = C(m)*R - I;

% Plot the mesh grid error function
figure
mesh(phi_a3, phi_a5, abs(error))
xlabel('\phi_{a_3}')
ylabel('\phi_{a_5}')

% Find the angles of a3 and a5 that resulted in the minimum error.
[Y J] = min(abs(error),[],1);
[X K] = min(abs(error),[],2);
[mina3 a3ind] = min(Y);
[mina5 a5ind] = min(X);
phi_a3(a5ind,a3ind)*180/pi, phi_a5(a5ind,a3ind)*180/pi,

%     C(m)*R(a5ind,a3ind) - I(a5ind,a3ind), error(a5ind, a3ind)
% Here set the angles of the coefficients manually.  No alternate
% method seemed available to setting the appropriate values.
a3_ang = 175*pi/180;
[X,Y] = pol2cart(a3_ang,abs(a3_3_avg));
a3 = complex(X,Y);
a5_ang_adj = (280)*pi/180;
[X,Y] = pol2cart(a5_ang_adj,abs(a3_5_avg));
a5 = complex(X,Y);

% Compute G_coefb from G_coefa terms to allow backing out the
% single tone effects of these terms.
ax = [a3 a5];
bx = zeros(size(ax));
```

```
    for n = 1:2
        alpha = n;
        alpha2 = 2*alpha;
        bx(n) = ax(n) * factorial(1+alpha2)/...
                (2^alpha2*factorial(alpha)*factorial(1+alpha));
    end

    % Subtract b3*vin.^3 + b5*vin.^5 from the single tone data
    vout_cplx_amam_ext_m3m5 = vout_cplx_amam_ext - ...
                              bx(1)*vin_cplx_amam_ext.^3 - ...
                              bx(2)*vin_cplx_amam_ext.^5;

    N = 17;      % Order of polynomial
    v_in_amam_lr = [fliplr(-vin_cplx_amam_ext); 0; vin_cplx_amam_ext];

    % Settings for using extrapolated single tone data for fit
%     v_out_amam_lr = [fliplr(-vout_cplx_amam_ext); 0; vout_cplx_amam_ext];
%     G_coefb = polyfit(v_in_amam_lr, v_out_amam_lr, N);
%     G_coefb = polyfit_no_third_aw(v_in_amam_lr, v_out_amam_lr, N);

    % Settings for using estimated a3 and a5 coefficients from two tone
    % data
    v_out_amam_lr = [fliplr(-vout_cplx_amam_ext_m3m5); 0; ...
                     vout_cplx_amam_ext_m3m5];
    G_coefb = polyfit_no_third_fifth_aw(v_in_amam_lr, v_out_amam_lr, N);
    G_coefb(N-2) = bx(1);   % Third order term
    G_coefb(N-4) = bx(2);   % Fifth order term

    % Evaluate the polynomial fit for the positive voltage data
    % and compare to the measured data
    v_fit_amam = polyval(G_coefb, vin_cplx_amam);
%     figure
%     ftitle('Single tone fit')
%     plot(10*log10(abs(vin_cplx_amam).^2*10), ...
%          10*log10(abs(vout_cplx_amam).^2*10), '-*', ...
%          10*log10(abs(vin_cplx_amam).^2*10), ...
%          10*log10(abs(v_fit_amam).^2*10))
%     legend('Meas.','Est.',0)

    v_fit_amam_ext = polyval(G_coefb, vin_cplx_amam_ext);
    figure
    subplot(2,1,1)
    ext = size(vin_cplx_amam,1);
    ftitle('Single tone fit extrapolated')
```

```
%     plot(abs(vin_cplx_amam), abs(vout_cplx_amam),'d', ...
%          abs(vin_cplx_amam_ext(ext+1:end)), ...
%          abs(vout_cplx_amam_ext(ext+1:end)), '^', ...
%          abs(vin_cplx_amam_ext), abs(v_fit_amam_ext),'--')
    plot(10*log10(abs(vin_cplx_amam).^2*10), ...
         10*log10(abs(vout_cplx_amam).^2*10),'d', ...
         10*log10(abs(vin_cplx_amam_ext(ext+1:end)).^2*10), ...
         10*log10(abs(vout_cplx_amam_ext(ext+1:end)).^2*10), '^', ...
         10*log10(abs(vin_cplx_amam_ext).^2*10), ...
         10*log10(abs(v_fit_amam_ext).^2*10),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Peak Voltage (V)')
    legend('Meas.','Ext.','Est.',0)
%     title('Extrapolated Single-tone Fit')
    subplot(2,1,2)
%     plot(abs(vin_cplx_amam), angle(vout_cplx_amam),'d', ...
%          abs(vin_cplx_amam_ext(ext+1:end)), ...
%          angle(vout_cplx_amam_ext(ext+1:end)), '^', ...
%          abs(vin_cplx_amam_ext), angle(v_fit_amam_ext),'--')
    plot(10*log10(abs(vin_cplx_amam).^2*10), angle(vout_cplx_amam),'d',...
         10*log10(abs(vin_cplx_amam_ext(ext+1:end)).^2*10), ...
         angle(vout_cplx_amam_ext(ext+1:end)), '^', ...
         10*log10(abs(vin_cplx_amam_ext).^2*10),angle(v_fit_amam_ext),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Phase ( \circ)')
    legend('Meas.','Ext.','Est.',0)


% Now use the complex coefficients to estimate the IM3 product magnitude
% and phase.
%
% First need to take into account that the coefficients that were fit, were
% the average coefficients and so need to be backed out to the
% instantaneous coefficients. Following work done in RAWCON 2004 paper and
% that of Gard:Gutierrez:Steer:1999 and Gharaibeh:Steer:2005

    % Compute instantaneous coefficients
    % Only the odd coefficients will be retained since the even terms are
    % approximately zero from fit.
    G_coefa = zeros(size(G_coefb));

    G_coefa(N+1) = G_coefb(N+1);          % DC components are equal
    for j = 1:2:N
        alpha = (N+1)/2 - (j+1)/2;
```

```
    alpha2 = alpha*2;
    G_coefa(j) = G_coefb(j)*2^alpha2*factorial(alpha)*...
                 factorial(1+alpha)/factorial(1+alpha2);
end

% Repeat of computation of single tone fit using development in
% Khaled's paper wiht instantaneous coefficients
v_fit_amam2 = 0;
v_fit_amam_comp = zeros(size(vin_cplx_amam,1),size(1:2:N,2));
for n = 1:2:N,
    m = [ 0 0 (n-1)/2 (n+1)/2];
    mco = multinomial(m,n);
    temp = G_coefa(N-n+1)*mco*(vin_cplx_amam).^n;
    v_fit_amam2 = v_fit_amam2 + temp;
    v_fit_amam_comp(:,(n+1)/2) = temp;
end

% Plots comparing two sets of coefficients fit, expected identical, and
% the phase of the single tone measured and expected data.
figure
ftitle('Single tone fit')
subplot(2,1,1)
plot(10*log10(abs(vin_cplx_amam).^2*10), ...
     10*log10(abs(vout_cplx_amam).^2*10), '-*', ...
     10*log10(abs(vin_cplx_amam).^2*10), ...
     10*log10(abs(v_fit_amam2).^2*10))
subplot(2,1,2)
plot(10*log10(abs(vin_cplx_amam).^2*10), phase_amam_rad*180/pi, ...
     10*log10(abs(vin_cplx_amam).^2*10), angle(v_fit_amam2)*180/pi)
figure
ftitle(['Magnitude of individual odd-order contributors from '...
       'envelope formulation of AM-AM data'])
plot(amam_magin_p, 10*log10(abs(v_fit_amam_comp).^2*10))
legend('1','3','5','7','9',0)

% Now compute complex IM3 from coefficients and formulation based on
% Gharaibeh:Steer:2005
im3lfit1 = 0;
im3hfit1 = 0;
im3l_order_comp = zeros(size(f1_mag_v,1),size(3:2:N,2));
for n = 3:2:N,
    %im3lfit1 =0;
    for l = 0:(n-3)/2,
        % Note, the - terms are L's not one's
```

```
            m = [((n-3)/2-l) (l+1) l ((n+1)/2 - l)];
            mco = multinomial(m, n);
            temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
            im3lfit1 = im3lfit1 + temp;
            temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
            im3hfit1 = im3hfit1 + temp;
            im3l_order_comp(:,(n-1)/2) = im3l_order_comp(:,(n-1)/2) + temp;
        end
    end

    figure
    ftitle('Individual odd-order contributions to IM3 product')
    plot(ref1_mag_p, 10*log10(abs(im3l_order_comp).^2*10))
    legend('3','5','7','9',0)

    figure
    ftitle('Individual odd-order contributions to IM3 product, polar')
    polar(angle(im3l_order_comp), abs(im3l_order_comp))
    legend('3','5','7','9',0)

    % Compute phase difference between measured IM3 phase (at 0 degree
    % reference power) and the estimated phase from the fit
    im3l_phase_fit = unwrap(angle(im3lfit1));
    diff = im3l_phase_fit(refindex)*180/pi-im3l_phase_rad(refindex)*180/pi;

    figure
   ftitle('Comparison of IM3 magnitude measured and single-slice estimate')
    plot(ref1_mag_p, 10*log10(abs(im3lfit1).^2*10), ...
         ref1_mag_p, im3l_mag_p, ref1_mag_p, im3h_mag_p)

    figure
    ftitle('Comparison of IM3 phase measured and single-slice estimate')
    plot(ref1_mag_p, unwrap(angle(im3lfit1))*180/pi, ...
         ref1_mag_p, unwrap(im3l_phase_rad+diff*pi/180)*180/pi, ...
         ref1_mag_p, unwrap(im3h_phase_rad+diff*pi/180)*180/pi)

    figure
    plot(10*log10(abs(vin_cplx_amam_ext).^2*10), ...
         angle(vout_cplx_amam_ext)*180/pi, ...
         10*log10(abs(vin_cplx_amam_ext).^2*10), ...
         angle(vout_cplx_amam_ext_m3m5)*180/pi)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% !! Now use the results from the single tone polynomial fit and the
```

```
% correction for a3 and a5 from the IM3 data to set the initial conditions
% for the GA optimization of the odd order coefficients polynomial
% coefficients.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Form complex version of measure IM3L and IM3H data
  [X Y] = pol2cart(unwrap(im3l_phase_rad+diff*pi/180), im3l_mag_v);
  im3l_meas_cplx = complex(X, Y);
  [X Y] = pol2cart(unwrap(im3h_phase_rad+diff*pi/180), im3h_mag_v);
  im3h_meas_cplx = complex(X, Y);

  im3_meas_avg_cplx = (im3l_meas_cplx + im3h_meas_cplx)/2;

  % Execute the Genetic Algorithm (non-vectorized) based on user settings

  % First set the options for the GA function
  % Start with default options
  options = gaoptimset;   % start with the default option settings

  % number of iterations for each section 1, 2, 3 & 4 terms

  iter  = 300;

  options = gaoptimset(options,'EliteCount', 10);
  % stop if error function reaches 0
  options = gaoptimset(options,'FitnessLimit' ,0);
  options = gaoptimset(options,'StallGenLimit' ,Inf);
  % never stall
  options = gaoptimset(options,'StallTimeLimit' ,Inf);
  % update plot every 10 iterations
  options = gaoptimset(options,'PlotInterval' ,10);
  popsize = 100;
  % Used for separate magnitude and phase ind. variables
  options = gaoptimset(options,'PopulationSize' ,popsize);
  options = gaoptimset(options,'SelectionFcn' , @selectionstochunif);
  options = gaoptimset(options,'MutationFcn' ,{ @mutationuniform 0.3 });
  options =gaoptimset(options,'CrossoverFcn',{@crossoverintermediate,0.5});
  options = gaoptimset(options,'Display' ,'off');
  options = gaoptimset(options,'PlotFcns' ,{ @gaplotbestf });

  % Fitness function, i.e. error function
  fitnessFunction = @one_tone_two_tone_fit_cplx;

  % Number of Variables, assumes N odd
```

```
% Used for separate magnitude and phase independent variables for
% polynomial coefficients
nvars = (N+1);
% Form vector of odd order coefficients
G_coefa_odd = G_coefa(1:2:N);
% Independent magnitude and phase variables
a_init = [abs(G_coefa_odd), angle(G_coefa_odd)];
% Determine the initial population limits based on the fit to the
% polynomial
G_coefa_odd_mag_limit = 0.01*ones(size(G_coefa_odd));
G_coefa_odd_mag_limit([1:nvars/2 - 6]) = 0.01;
% Upper and lower bounds on initial estimates of initial population
G_coefa_odd_lb = [abs(G_coefa_odd) - ...
                  G_coefa_odd_mag_limit.*abs(G_coefa_odd), ...
                  angle(G_coefa_odd) - pi/40];
G_coefa_odd_ub = [abs(G_coefa_odd) + ...
                  G_coefa_odd_mag_limit.*abs(G_coefa_odd), ...
                  angle(G_coefa_odd) + pi/40];
%Modify initial population values
options = gaoptimset(options,'Generations' ,iter);
options = gaoptimset(options,'PopInitRange', ...
                     [G_coefa_odd_lb; G_coefa_odd_ub]);

% Run GA
[A,FVAL,REASON,OUTPUT,POPULATION,SCORES] = ...
                                        ga(fitnessFunction,nvars,options);

% Form the coefficient vector for GA fit
G_coefa_ga = zeros(size(G_coefa));

% Independent magnitude and angle components
[X Y] = pol2cart(A(nvars/2+1:end), A(1:nvars/2));    % pol2cart(Theta, R)
G_coefa_ga([1:2:N]) = complex(X, Y);

% Compute the fundamental response
v_fit_amam_ga = 0;
for n = 1:2:N,
    m = [ 0 0 (n-1)/2 (n+1)/2];
    mco = multinomial(m,n);
    temp = G_coefa_ga(N-n+1)*mco*(vin_cplx_amam).^n;
    v_fit_amam_ga = v_fit_amam_ga + temp;
end

% Compute the IM3 response
```

```
im3lfit_ga = 0;
im3hfit_ga = 0;
for n = 3:2:N,
    for l = 0:(n-3)/2,
        % Note, the - terms are L's not one's
        m = [((n-3)/2-l) (l+1) l ((n+1)/2 - l)];
        mco = multinomial(m, n);
        temp = G_coefa_ga(N-n+1)*mco*(f1_mag_v).^n;
        im3lfit_ga = im3lfit_ga + temp;
        im3hfit_ga = im3lfit_ga;
    end
end

% Compute the error function values for the results
error_poly = one_tone_two_tone_fit_cplx(a_init)
error_ga = one_tone_two_tone_fit_cplx(A)

% Plot results of GA fit for both single tone and
figure
subplot(2,1,1)
ftitle('Single tone GA fit')
plot(10*log10(abs(vin_cplx_amam).^2*10), ...
     10*log10(abs(vout_cplx_amam).^2*10), '-*', ...
     10*log10(abs(vin_cplx_amam).^2*10), ...
     10*log10(abs(v_fit_amam_ga).^2*10))
legend('Meas.','Est.',0)
subplot(2,1,2)
plot(10*log10(abs(vin_cplx_amam).^2*10), phase_amam_rad*180/pi, ...
     10*log10(abs(vin_cplx_amam).^2*10), angle(v_fit_amam_ga)*180/pi)
legend('Meas.','Est.',0)

figure
subplot(2,1,1)
ftitle('IM3 Mag. and Phase GA fit')
plot(ref1_mag_p, im3l_mag_p, ref1_mag_p, im3h_mag_p, ...
    ref1_mag_p, 10*log10(abs(im3lfit_ga).^2*10))
legend('IM3L Mag. Meas.', 'IM3H Mag. Meas.', 'IM3 Mag. Est.',0)
subplot(2,1,2)
plot(ref1_mag_p, unwrap(im3l_phase_rad+diff*pi/180)*180/pi, ...
     ref1_mag_p, unwrap(im3h_phase_rad+diff*pi/180)*180/pi, ...
     ref1_mag_p, unwrap(angle(im3lfit_ga))*180/pi, ...
     ref1_mag_p, unwrap(angle(im3lfit1))*180/pi);
legend('\phi_{IM3L} Meas.', '\phi_{IM3H} Meas.', '\phi_{IM3} Est.',0)
```

Genetic Algorithm Error Function

```
function error_func = one_tone_two_tone_fit_cplx(a);
%
% Error function for GA optimization.  This error function
% uses both the single tone and the two tone data sets.  In
% this version, the coefficients are separated into magnitude and phase
% components so that they can vary over a more complete range.
%
% Note: this function is called from a script so the
% data sets are in that workspace and are treated as
% globals for the purposes of this function.
%

global vout_cplx_amam; global vin_cplx_amam; global f1_mag_v_ext;
global im3_meas_avg_cplx; global nvars; global popsize;

% Compute the estimate of the IM3 products and fundamental using
% the coefficient vector a.

% Independent real and imaginary components
% G_coefa_ga = complex(a(1:nvars/2), a(nvars/2+1:end));

% Independent magnitude and angle components
[X Y] = pol2cart(a(nvars/2+1:end), a(1:nvars/2)); G_coefa_ga =
complex(X, Y);

    % Compute fundamental
    v_fit_amam_ga = 0;
    for n = 1:2:(nvars-1),
        m = [ 0 0 (n-1)/2 (n+1)/2];
        mco = multinomial(m,n);
        temp = G_coefa_ga(nvars/2-(n-1)/2)*mco*(vin_cplx_amam).^n;
        v_fit_amam_ga = v_fit_amam_ga + temp;
    end

    % Compute IM3 product, symmetric single slice version Now compute
    % complex IM3 from coefficients and formulation based on
    % Gharaibeh:Steer:2005
    im3lfit_ga = 0;
    for n = 3:2:(nvars-1),
        for l = 0:(n-3)/2,
            % Note, the - terms are L's not one's
            m = [((n-3)/2-l) (l+1) l ((n+1)/2 - l)];
            mco = multinomial(m, n);
```

```
            temp = G_coefa_ga(nvars/2-(n-1)/2)*mco*(f1_mag_v_ext).^n;
            im3lfit_ga = im3lfit_ga + temp;
        end
    end

% Compute the errors for the fundamental and IM3 products.  The error
% functions are weighted to give the phase data more importance

error_one_tone_abs = abs(abs(vout_cplx_amam) - ...
                     abs(v_fit_amam_ga))./abs(vout_cplx_amam);
error_one_tone_ang = abs(angle(vout_cplx_amam) -
angle(v_fit_amam_ga)); error_two_tone_abs =
abs(abs(im3_meas_avg_cplx) - ...
                     abs(im3lfit_ga))./abs(im3_meas_avg_cplx);
error_two_tone_ang = abs(angle(im3_meas_avg_cplx) -
angle(im3lfit_ga)); error_weight_one_mag =
ones(size(vout_cplx_amam)); error_weight_one_mag([end-5:end]) = 1;
error_weight_two_mag = ones(size(im3_meas_avg_cplx));
error_weight_two_mag([end-5:end]) = 1; error_weight_one =
100*ones(size(vout_cplx_amam)); error_weight_one([end-5:end]) =
100; error_weight_two = 100*ones(size(im3_meas_avg_cplx));
error_weight_two([end-5:end]) = 100;
% Since these error values are vectors based on the sweep on the input
% power and the GA technique requires a scalar, the errors need to
% be summed.
error_func = sum(error_weight_one_mag.*error_one_tone_abs) + ...
             sum(error_weight_one.*error_one_tone_ang) + ...
             sum(error_weight_two_mag.*error_two_tone_abs) + ...
             sum(error_weight_two.*error_two_tone_ang);
```

D.3.4 Model Parameter Extraction and Output Generation Script for Levenberg-Marquadt Fitting Procedure

```
% This script processes the 10 kHz separation two-tone data taken at
% 450 MHz on the Hittite PA.  This is the second set of data taken
% on this amplifier with the hybrid combiner used at the
% output of the DUT.  The script plots the measured IM3
% magnitude and phase as polar plots with comparison to the
% expected output based on single tone measurements.

% In this version, I will employ the Optimization toolbox to determine
% a better fit to the two-tone phase data using both the
```

```
% single tone and two tone data sets.


% Read in the corrected measured magnitude and phase data.  Magnitude
% data is that corrected for path loss from DUT to spectrum analyzer
% and corrected phase data takes into account the attenuator and
% phase insertion with power of the IFR sources.
clear all, close all, cd C:\Experiments\mar_05\mon_7

% Declare globals
global vout_cplx_amam; global vin_cplx_amam; global f1_mag_v;
global im3_meas_avg_cplx; global nvars; global popsize;


f_op_amam = 450e6; deltaf = 10e3; refindex = 2; f_op_IM3L = 450e6
- deltaf; f_op_IM3H = 450e6 + 2*deltaf;
% read_path = 'C:\Experiments\mar_05\wed_23\hittite\';
amam_read_path = 'C:\Experiments\mar_05\wed_23\hittite\';
read_path = 'C:\Experiments\may_05\tues_31\'; im3l_mag_p =
load([read_path,'p_im3l_out_10k.txt']); im3h_mag_p =
load([read_path,'p_im3h_out_10k.txt']); im3l_phase =
load([read_path,'phase_mtL_10k.txt']); im3h_phase =
load([read_path,'phase_mtH_10k.txt']); ref1_mag_p =
load([read_path,'p_ref1_in_mt.txt']); ref2_mag_p =
load([read_path,'p_ref2_in_mt.txt']); amam_mag_p =
load([amam_read_path,'p_amam_out.txt']); amam_phase =
load([amam_read_path,'phase_amam.txt']); amam_magin_p =
load([amam_read_path,'p_amam_in.txt']);

amam_cncl_p = load([amam_read_path,'amam_cncl_dB.txt']);
im3l_cncl_p = load([read_path,'im3l_cncl_dB_10k.txt']);
im3h_cncl_p = load([read_path,'im3h_cncl_dB_10k.txt']);

% Convert power in dBm to voltage magnitude, assume 50 Ohm
% impedance
im3l_mag_v = sqrt(10.^(im3l_mag_p/10)/10); im3h_mag_v =
sqrt(10.^(im3h_mag_p/10)/10); f1_mag_v =
sqrt(10.^(ref1_mag_p/10)/10); f2_mag_v =
sqrt(10.^(ref2_mag_p/10)/10); amam_mag_v =
sqrt(10.^(amam_mag_p/10)/10); amam_magin_v =
sqrt(10.^(amam_magin_p/10)/10);

% Convert phase data to radians, saved as frequency normalized
phase_amam_rad = amam_phase * 2*pi*f_op_amam;
```

```
phase_amamin_rad = zeros(size(phase_amam_rad));  % Assume input phase is 0
im3l_phase_rad = im3l_phase * 2*pi*f_op_IM3L; im3h_phase_rad =
im3h_phase * 2*pi*f_op_IM3H;

% Plot magnitude and phase of IM3 products on polar plot
% figure
% polar(im3l_phase_rad, im3l_mag_v)
% hold
% polar(im3h_phase_rad, im3h_mag_v, 'g')

% Fit complex polynomial to AM-AM, AM-PM data in the form of a complex
% gain.

    % Convert polar form of data to complex form
    [x_amam, y_amam] = pol2cart(phase_amam_rad, amam_mag_v);
    vout_cplx_amam = complex(x_amam, y_amam);
    [x_amamin, y_amamin] = pol2cart(phase_amamin_rad, amam_magin_v);
    vin_cplx_amam = complex(x_amamin, y_amamin);

    f1_mag_vcplx = complex(f1_mag_v, zeros(size(f1_mag_v)));

    % Compute complex voltage gain
    G_cplx = vout_cplx_amam./vin_cplx_amam;

    % Plot complex gain for both voltage and power
    figure
    subplot(2,1,1)
    plot(amam_magin_p, amam_mag_p - amam_magin_p,'-^');
    xlabel('Input Tone Power (dBm)')
    ylabel('Power Gain (dB)')
    subplot(2,1,2)
    plot(amam_magin_p, phase_amam_rad*180/pi,'-^');
    xlabel('Input Tone Power (dBm)')
    ylabel('Output Phase ( \circ)')
    ftitle('Measured Magnitude of Complex Gain')

    % Extrapolate the complex gain, magnitude and phase separately
    % Currently these are fit with linear polynomials over their entire
    % range in the voltage domain.
        % Perform linear fit
        lin_fit_mag = polyfit(abs(vin_cplx_amam), abs(G_cplx),1);
        lin_fit_ang = polyfit(abs(vin_cplx_amam), angle(G_cplx),1);
        % Generate extended voltage input
        p_ext = 11:16;
```

```
    v_ext = sqrt(10.^(p_ext/10)/10);
    % Generate gain magnitude extension
    G_mag_ext = polyval(lin_fit_mag, v_ext);
    % Generate gain phase extension
    G_ang_ext = polyval(lin_fit_ang, v_ext);
    % Form the extended voltage gain expression
    G_cplx_ext = [real(vout_cplx_amam./vin_cplx_amam); ...
                  (G_mag_ext.*cos(G_ang_ext))'] + ...
                  sqrt(-1)*[imag(vout_cplx_amam./vin_cplx_amam); ...
                  (G_mag_ext.*sin(G_ang_ext))'];
    % Form the extended input voltage
    vin_cplx_amam_ext = [vin_cplx_amam; v_ext'];
    % Form the extended output voltage
    vout_cplx_amam_ext = vin_cplx_amam_ext.*G_cplx_ext;



% Fit a complex polynomial to the voltage gain.
%
% First form the negative voltage data by negating the
% complex input and output voltages, forcing the output voltage
% to be zero at zero input.  By negating the positive only
% input/output voltage, the transfer function will automatically
% be odd.
%
% Next use the polyfit function to determine the coefficients.
%
% Finally, use polyval and plot the results to visually determine
% if a polynomial of significant order was used.

% !!! Test to determine if backing out third and fifth order coefficients from
% measured IM3 data can improve the IM3 fit with a single slice.  Coefficients
% extracted over a range of IM3 data where the contributions of the particular
% order dominate.

% Compute G_coefa third order term
c3 = multinomial([0 1 2 0 ], 3);
%     [X, Y] = pol2cart((im3l_phase_rad)/1, (im3l_mag_v)/1);
%     im3_cplx_v = complex(X,Y);
%     a3_3 = (im3_cplx_v)./(c3*f1_mag_v.^3);
a3_3 = abs(im3l_mag_v)./(c3*f1_mag_v.^3);
a3_3_avg = sum(a3_3(1:4))/size(a3_3(1:4),1);

% Compute G_coefa fifth order term
```

```
c5 = multinomial([1 1 3 0], 5) + multinomial([2 0 2 1], 5);
a3_5 = abs(im3l_mag_v)./(abs(f1_mag_v).^5 * c5);
a3_5_avg = sum(a3_5(6:11))/size(a3_5(6:11),1);

% Determine angles for the a3 and a5 coefficients.  The goal here
% was to use the measured IM3 phase data to get an estimate of what
% the average phase difference between the fifth order and third
% order terms should be over a range of input powers.  A grid of
% a3 and a5 angles was used to plot the difference between the
% estimated phase in the region dominated by the third and fifth
% to see the minimum error point and thus determine the phase
% difference.  Since the two components can only contribute a constant
% phase, it was hoped that this exercise would reveal the actual
% phase difference needed between the two components.  However, for
% the input power range from Pin = -13 to -8 dBm, the delta phase
% for the a3 and a5 was not constant or even close to constant.
% Therefore this technique did not readily appear to yield the
% information I sought.

C = tan(im3l_phase_rad);
phi = -pi:pi/40:pi;
m = 6;
[phi_a3 phi_a5] = meshgrid(phi, phi);
R = abs(a3_3_avg)*c3*cos(phi_a3)*abs(f1_mag_v(m)).^3 + abs(a3_5_avg)*c5*cos(phi_a
I = abs(a3_3_avg)*c3*sin(phi_a3)*abs(f1_mag_v(m)).^3 + abs(a3_5_avg)*c5*sin(phi_a

error = C(m)*R - I;

% Plot the mesh grid error function
figure
mesh(phi_a3, phi_a5, abs(error))
xlabel('\phi_{a_3}')
ylabel('\phi_{a_5}')

% Find the angles of a3 and a5 that resulted in the minimum error.
[Y J] = min(abs(error),[],1);
[X K] = min(abs(error),[],2);
[mina3 a3ind] = min(Y);
[mina5 a5ind] = min(X);
phi_a3(a5ind,a3ind)*180/pi, phi_a5(a5ind,a3ind)*180/pi,

%     C(m)*R(a5ind,a3ind) - I(a5ind,a3ind), error(a5ind, a3ind)
% Here set the angles of the coefficients manually.  No alternate
% method seemed available to setting the appropriate values.
```

```
    a3_ang = 175*pi/180;
    [X,Y] = pol2cart(a3_ang,abs(a3_3_avg));
    a3 = complex(X,Y);
    a5_ang_adj = (280)*pi/180;
    [X,Y] = pol2cart(a5_ang_adj,abs(a3_5_avg));
    a5 = complex(X,Y);

    % Compute G_coefb from G_coefa terms to allow backing out the
    % single tone effects of these terms.
    ax = [a3 a5];
    bx = zeros(size(ax));
    for n = 1:2
        alpha = n;
        alpha2 = 2*alpha;
        bx(n) = ax(n) * factorial(1+alpha2)/...
               (2^alpha2*factorial(alpha)*factorial(1+alpha));
    end

    % Subtract b3*vin.^3 + b5*vin.^5 from the single tone data
    vout_cplx_amam_ext_m3m5 = vout_cplx_amam_ext - ...
                              bx(1)*vin_cplx_amam_ext.^3 - ...
                              bx(2)*vin_cplx_amam_ext.^5;

    N = 17;      % Order of polynomial
    v_in_amam_lr = [fliplr(-vin_cplx_amam_ext); 0; vin_cplx_amam_ext];

    % Settings for using extrapolated single tone data for fit
%     v_out_amam_lr = [fliplr(-vout_cplx_amam_ext); 0; vout_cplx_amam_ext];
%     G_coefb = polyfit(v_in_amam_lr, v_out_amam_lr, N);
%     G_coefb = polyfit_no_third_aw(v_in_amam_lr, v_out_amam_lr, N);

    % Settings for using estimated a3 and a5 coefficients from two tone
    % data
    v_out_amam_lr = [fliplr(-vout_cplx_amam_ext_m3m5); 0; ...
                     vout_cplx_amam_ext_m3m5];
    G_coefb = polyfit_no_third_fifth_aw(v_in_amam_lr, v_out_amam_lr, N);
    G_coefb(N-2) = bx(1);   % Third order term
    G_coefb(N-4) = bx(2);   % Fifth order term

    % Evaluate the polynomial fit for the positive voltage data
    % and compare to the measured data
    v_fit_amam = polyval(G_coefb, vin_cplx_amam);
%     figure
%     ftitle('Single tone fit')
```

```
%     plot(10*log10(abs(vin_cplx_amam).^2*10), ...
%          10*log10(abs(vout_cplx_amam).^2*10), '-*', ...
%          10*log10(abs(vin_cplx_amam).^2*10), ...
%          10*log10(abs(v_fit_amam).^2*10))
%     legend('Meas.','Est.',0)

    v_fit_amam_ext = polyval(G_coefb, vin_cplx_amam_ext);
    figure
    subplot(2,1,1)
    ext = size(vin_cplx_amam,1);
    ftitle('Single tone fit extrapolated')
%     plot(abs(vin_cplx_amam), abs(vout_cplx_amam),'d', ...
%          abs(vin_cplx_amam_ext(ext+1:end)), ...
%          abs(vout_cplx_amam_ext(ext+1:end)), '^', ...
%          abs(vin_cplx_amam_ext), abs(v_fit_amam_ext),'--')
    plot(10*log10(abs(vin_cplx_amam).^2*10), ...
         10*log10(abs(vout_cplx_amam).^2*10),'d', ...
         10*log10(abs(vin_cplx_amam_ext(ext+1:end)).^2*10), ...
         10*log10(abs(vout_cplx_amam_ext(ext+1:end)).^2*10), '^', ...
         10*log10(abs(vin_cplx_amam_ext).^2*10), ...
         10*log10(abs(v_fit_amam_ext).^2*10),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Peak Voltage (V)')
    legend('Meas.','Ext.','Est.',0)
%     title('Extrapolated Single-tone Fit')
    subplot(2,1,2)
%     plot(abs(vin_cplx_amam), angle(vout_cplx_amam),'d', ...
%          abs(vin_cplx_amam_ext(ext+1:end)), ...
%          angle(vout_cplx_amam_ext(ext+1:end)), '^', ...
%          abs(vin_cplx_amam_ext), angle(v_fit_amam_ext),'--')
    plot(10*log10(abs(vin_cplx_amam).^2*10), angle(vout_cplx_amam),'d',...
         10*log10(abs(vin_cplx_amam_ext(ext+1:end)).^2*10), ...
         angle(vout_cplx_amam_ext(ext+1:end)), '^', ...
         10*log10(abs(vin_cplx_amam_ext).^2*10),angle(v_fit_amam_ext),'--')
    xlabel('Input Peak Voltage (V)')
    ylabel('Output Phase ( \circ)')
    legend('Meas.','Ext.','Est.',0)


% Now use the complex coefficients to estimate the IM3 product magnitude
% and phase.
%
% First need to take into account that the coefficients that were fit, were
% the average coefficients and so need to be backed out to the
```

```
% instantaneous coefficients. Following work done in RAWCON 2004 paper and
% that of Gard:Gutierrez:Steer:1999 and Gharaibeh:Steer:2005

    % Compute instantaneous coefficients
    % Only the odd coefficients will be retained since the even terms are
    % approximately zero from fit.
    G_coefa = zeros(size(G_coefb));

    G_coefa(N+1) = G_coefb(N+1);          % DC components are equal
    for j = 1:2:N
        alpha = (N+1)/2 - (j+1)/2;
        alpha2 = alpha*2;
        G_coefa(j) = G_coefb(j)*2^alpha2*factorial(alpha)*...
                     factorial(1+alpha)/factorial(1+alpha2);
    end

    % Repeat of computation of single tone fit using development in
    % Khaled's paper wiht instantaneous coefficients
    v_fit_amam2 = 0;
    v_fit_amam_comp = zeros(size(vin_cplx_amam,1),size(1:2:N,2));
    for n = 1:2:N,
        m = [ 0 0 (n-1)/2 (n+1)/2];
        mco = multinomial(m,n);
        temp = G_coefa(N-n+1)*mco*(vin_cplx_amam).^n;
        v_fit_amam2 = v_fit_amam2 + temp;
        v_fit_amam_comp(:,(n+1)/2) = temp;
    end

    % Plots comparing two sets of coefficients fit, expected identical, and the phase
    % of the single tone measured and expected data.
    figure
    ftitle('Single tone fit')
    subplot(2,1,1)
    plot(10*log10(abs(vin_cplx_amam).^2*10), ...
         10*log10(abs(vout_cplx_amam).^2*10), '-*', ...
         10*log10(abs(vin_cplx_amam).^2*10), ...
         10*log10(abs(v_fit_amam2).^2*10))
    subplot(2,1,2)
    plot(10*log10(abs(vin_cplx_amam).^2*10), phase_amam_rad*180/pi, ...
         10*log10(abs(vin_cplx_amam).^2*10), angle(v_fit_amam2)*180/pi)
    figure
    ftitle(['Magnitude of individual odd-order contributors from ' ...
           'envelope formulation of AM-AM data'])
    plot(amam_magin_p, 10*log10(abs(v_fit_amam_comp).^2*10))
```

```
legend('1','3','5','7','9',0)

% Now compute complex IM3 from coefficients and formulation based on
% Gharaibeh:Steer:2005
im3lfit1 = 0;
im3hfit1 = 0;
im3l_order_comp = zeros(size(f1_mag_v,1),size(3:2:N,2));
for n = 3:2:N,
    %im3lfit1 =0;
    for l = 0:(n-3)/2,
        % Note, the - terms are L's not one's
        m = [((n-3)/2-l) (l+1) l ((n+1)/2 - l)];
        mco = multinomial(m, n);
        temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
        im3lfit1 = im3lfit1 + temp;
        temp = G_coefa(N-n+1)*mco*(f1_mag_v).^n;
        im3hfit1 = im3hfit1 + temp;
        im3l_order_comp(:,(n-1)/2) = im3l_order_comp(:,(n-1)/2) + temp;
    end
end

figure
ftitle('Individual odd-order contributions to IM3 product')
plot(ref1_mag_p, 10*log10(abs(im3l_order_comp).^2*10))
legend('3','5','7','9',0)

figure
ftitle('Individual odd-order contributions to IM3 product, polar')
polar(angle(im3l_order_comp), abs(im3l_order_comp))
legend('3','5','7','9',0)

% Compute phase difference between measured IM3 phase (at 0 degree
% reference power) and the estimated phase from the fit
im3l_phase_fit = unwrap(angle(im3lfit1));
diff = im3l_phase_fit(refindex)*180/pi-im3l_phase_rad(refindex)*180/pi;

figure
ftitle('Comparison of IM3 magnitude measured and single-slice estimate')
plot(ref1_mag_p, 10*log10(abs(im3lfit1).^2*10), ...
     ref1_mag_p, im3l_mag_p, ref1_mag_p, im3h_mag_p)

figure
ftitle('Comparison of IM3 phase measured and single-slice estimate')
plot(ref1_mag_p, unwrap(angle(im3lfit1))*180/pi, ...
```

```
          ref1_mag_p, unwrap(im3l_phase_rad+diff*pi/180)*180/pi, ...
          ref1_mag_p, unwrap(im3h_phase_rad+diff*pi/180)*180/pi)

    figure
    plot(10*log10(abs(vin_cplx_amam_ext).^2*10), ...
          angle(vout_cplx_amam_ext)*180/pi, ...
          10*log10(abs(vin_cplx_amam_ext).^2*10), ...
          angle(vout_cplx_amam_ext_m3m5)*180/pi)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% !! Now use the results from the single tone polynomial fit and the
% correction for a3 and a5 from the IM3 data to set the initial conditions
% for various nonlinear optimization functions within Matlab
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Form complex version of measure IM3L and IM3H data
  [X Y] = pol2cart(unwrap(im3l_phase_rad+diff*pi/180), im3l_mag_v);
  im3l_meas_cplx = complex(X, Y);
  [X Y] = pol2cart(unwrap(im3h_phase_rad+diff*pi/180), im3h_mag_v);
  im3h_meas_cplx = complex(X, Y);

  im3_meas_avg_cplx = (im3l_meas_cplx + im3h_meas_cplx)/2;

  % Form vector of odd order coefficients, this is initial value of
  % parameters to solve for
  G_coefa_odd = G_coefa(1:2:N);
  % Independent magnitude and phase variables
  a_orig = [abs(G_coefa_odd), angle(G_coefa_odd)];
  a_init = [abs(G_coefa_odd) - 0.005*abs(G_coefa_odd), ...
            angle(G_coefa_odd)-rand(size(G_coefa_odd))*pi/50];

  % Number of Variables, assumes N odd
  % Used for separate magnitude and phase independent variables for
  % polynomial coefficients
  nvars = (N+1);

  % Set nonlinear optimization options
  options = optimset('MaxFunEvals', 30000);
  options = optimset('LargeScale', 'off');

  % Execute nonlinear optimization of function to minimize
   % Unconstrained nonlinear optimization
%  [A, fval, exitflag, output]=fminunc(@nl_multitone_fit, a_init, options);
```

```
  % Form upper and lower bound of independent variables
  lb = [zeros(size(abs(G_coefa_odd))), -pi*ones(size(abs(G_coefa_odd)))];
  ub = [100*ones(size(abs(G_coefa_odd))), pi*ones(size(abs(G_coefa_odd)))];
  % Constrained nonlinear optimization
%   [A, fval, exitflag, output] = fmincon(@nl_multitone_fit, ...
%                           a_init,[],[],[],[],lb,ub, @confun_nl, options);

  % Form input data sequence for Levenberg-Marquadt optimization
  vin_data = [abs(vin_cplx_amam); angle(vin_cplx_amam); ...
              abs(f1_mag_v); zeros(size(f1_mag_v))];
  % Form output data sequence for Levenberg-Marquadt optimization
  vout_data = [abs(vout_cplx_amam); angle(vout_cplx_amam); ...
               abs(im3_meas_avg_cplx); angle(im3_meas_avg_cplx)];

   % Nonlinear least squares curve fit (Levenberg-Marquadt)
   [A, resnorm] = lsqcurvefit(@nl_lsq_multitone_fit, a_init, ...
                              vin_data, vout_data, lb, ub,options);

  % Form the coefficient vector for GA fit
  G_coefa_nl = zeros(size(G_coefa));

  % Independent magnitude and angle components
  [X Y] = pol2cart(A(nvars/2+1:end), A(1:nvars/2));    % pol2cart(Theta, R)
  G_coefa_nl([1:2:N]) = complex(X, Y);

  % Compute the fundamental response
  v_fit_amam_nl = 0;
  for n = 1:2:N,
      m = [ 0 0 (n-1)/2 (n+1)/2];
      mco = multinomial(m,n);
      temp = G_coefa_nl(N-n+1)*mco*(vin_cplx_amam).^n;
      v_fit_amam_nl = v_fit_amam_nl + temp;
  end

  % Compute the IM3 response
  im3lfit_nl = 0;
  im3hfit_nl = 0;
  for n = 3:2:N,
      for l = 0:(n-3)/2,
          % Note, the - terms are L's not one's
          m = [((n-3)/2-l) (l+1) l ((n+1)/2 - l)];
          mco = multinomial(m, n);
          temp = G_coefa_nl(N-n+1)*mco*(f1_mag_v).^n;
          im3lfit_nl = im3lfit_nl + temp;
```

```
        im3hfit_nl = im3lfit_nl;
    end
end

% Compute the error function values for the results
error_poly = nl_multitone_fit(a_orig)
error_nl = nl_multitone_fit(A)

% Plot results of NL fit for both single tone and
figure
subplot(2,1,1)
ftitle('Single tone NL fit')
plot(10*log10(abs(vin_cplx_amam).^2*10), ...
     10*log10(abs(vout_cplx_amam).^2*10), '-*', ...
     10*log10(abs(vin_cplx_amam).^2*10), ...
     10*log10(abs(v_fit_amam_nl).^2*10))
legend('Meas.','Est.',0)
subplot(2,1,2)
plot(10*log10(abs(vin_cplx_amam).^2*10), phase_amam_rad*180/pi, ...
     10*log10(abs(vin_cplx_amam).^2*10), angle(v_fit_amam_nl)*180/pi)
legend('Meas.','Est.',0)

figure
subplot(2,1,1)
ftitle('IM3 Mag. and Phase NL fit')
plot(ref1_mag_p, im3l_mag_p, ref1_mag_p, im3h_mag_p, ...
     ref1_mag_p, 10*log10(abs(im3lfit_nl).^2*10))
legend('IM3L Mag. Meas.', 'IM3H Mag. Meas.', 'IM3 Mag. Est.',0)
subplot(2,1,2)
plot(ref1_mag_p, unwrap(im3l_phase_rad+diff*pi/180)*180/pi, ...
     ref1_mag_p, unwrap(im3h_phase_rad+diff*pi/180)*180/pi, ...
     ref1_mag_p, unwrap(angle(im3lfit_nl))*180/pi, ...
     ref1_mag_p, unwrap(angle(im3lfit1))*180/pi);
legend('\phi_{IM3L} Meas.', '\phi_{IM3H} Meas.', '\phi_{IM3} Est.',0)
```

Nonlinear Optimization Error Function

```
function f = nl_multitone_fit(a_init, vin_data);

% This function is used with the nonlinear least square curve fitting
% routine. The vin_data argument contains the concatenated data for the
% input voltage for both the single and two-tone data.  The output data is
% the concatenated output from each of the single tone and two-tone
% solutions.
```

```
global nvars;

% Compute the estimate of the IM3 products and fundamental using
% the coefficient vector a.

% Form the single and two-tone input voltages from vin_data
[X Y] = pol2cart(vin_data(20:20+19-1), vin_data(1:19));
vin_cplx_amam = complex(X, Y); [X Y] =
pol2cart(vin_data(20+19+24:end), vin_data(20+19:20+19+24-1));
f1_mag_v = complex(X, Y);

% Independent magnitude and angle components
[X Y] = pol2cart(a_init(nvars/2+1:end), a_init(1:nvars/2));
G_coefa_nl = complex(X, Y);

    % Compute fundamental
    v_fit_amam_nl = 0;
    for n = 1:2:(nvars-1),
        m = [ 0 0 (n-1)/2 (n+1)/2];
        mco = multinomial(m,n);
        temp = G_coefa_nl(nvars/2-(n-1)/2)*mco*(vin_cplx_amam).^n;
        v_fit_amam_nl = v_fit_amam_nl + temp;
    end

    % Compute IM3 product, symmetric single slice version Now compute
    % complex IM3 from coefficients and formulation based on
    % Gharaibeh:Steer:2005
    im3lfit_nl = 0;
    for n = 3:2:(nvars-1),
        for l = 0:(n-3)/2,
             % Note, the - terms are L's not one's
            m = [((n-3)/2-l) (l+1) l ((n+1)/2 - l)];
            mco = multinomial(m, n);
            temp = G_coefa_nl(nvars/2-(n-1)/2)*mco*(f1_mag_v).^n;
            im3lfit_nl = im3lfit_nl + temp;
        end
    end

% Form function output, the concatenated solutions
f = [abs(v_fit_amam_nl); angle(v_fit_amam_nl); ...
     abs(im3lfit_nl); angle(im3lfit_nl)];
%
```

Appendix E

Post-Processing Details for Raw Vector Measurements in the In-Circuit Filter Tuning Application

This section contains the Matlab code for extracting the linear response (both magnitude and phase) of an embedded filter using vector intermodulation distortion measurements. The processing accounts for the phase shift introduced by the Marconi sources as the cancellation source (Source 3) is swept in power to achieve cancellation, the phase shift effect introduced by the interconnect cables and switches with respect to the measured phase shift required for maximum cancellation, and the phase shift of the . The equation that corrects for the interconnect cable and switch effects is

$$\phi_{\text{shift}} = \phi_{\text{shift,S3,meas}} + (\phi_{\text{P1,P5}} - \phi_{\text{P1,P3}} + \phi_{\text{l}_2}) + \phi_{\text{l}_2} - \phi_{\text{P3,P6}} - \phi_{\text{P4,P5}} + \phi_{\text{P4,P6}} \; . \qquad \text{(E.1)}$$

The definitions of the variables in this equation are:

- ϕ_{shift} - true phase shift of DUT versus two-tone stimulus frequency,
- $\phi_{\text{shift,S3,meas}}$ - raw measured phase shift of DUT versus two-tone stimulus frequency,
- $\phi_{\text{P1,P5}}$ - interconnect phase shift from Source 1 output port to VSA input port,
- $\phi_{\text{P1,P3}}$ - interconnect phase shift from Source 1 output port to DUT input port,
- ϕ_{l_2} - interconnect phase shift of feed line from Switch 1 to DUT input port,
- $\phi_{\text{P3,P6}}$ - interconnect phase shift from DUT input port to cancellation combiner output port,
- $\phi_{\text{P4,P5}}$ - interconnect phase shift from Source 3 output to VSA input,
- $\phi_{\text{P4,P6}}$ - interconnect phase shift from Source 3 output to cancellation combiner output port.

A diagram of the measurement apparatus with these ports labelled is shown in Fig. E.1.

E.1 Matlab Code for Execution of Post-Processing

```
  % Script for displaying S parameters and IM3 response data for filter 4.
clear all

% Defines
f_im3l_delta = -3*10e3;

% For properly tuned Filter 4
% div = 2.7;
% plot_limit = 34:90;
% phidiv = 2.4;

% For mis-tuned Filter 4
div = 3; plot_limit = 25:101; phidiv = 2.8;


% Set paths
  % S-parameters for diode with zero ohm series resistor, Vbias = 280 mV
  S1_path = 'C:\Experiments\oct_05\fri_07\Sparam_filter4\DD_DIOD0.txt';
  % S-parameters for diode with 10 ohm series resistor, Vbias = 280 mV
  S2_path = 'C:\Experiments\oct_05\fri_07\Sparam_filter4\DD_DIOD10.txt';
```

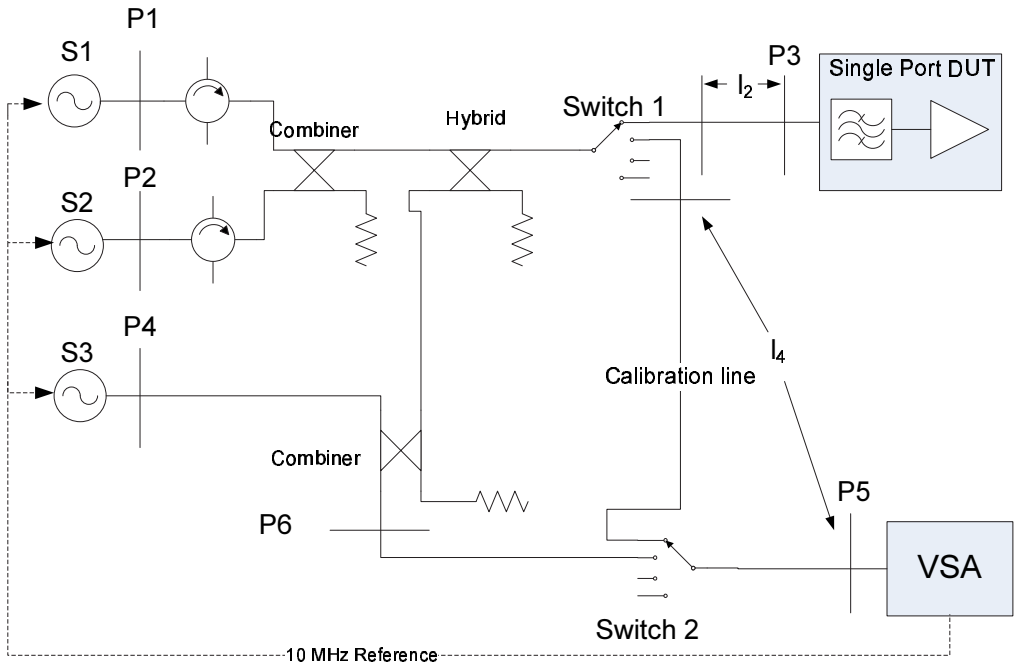


Figure E.1: Measurement system block diagram labelled with ports and interconnect designations for phase shift correction equation.

```
  % S-parameters for Filter 4, original tuned for fc=1.82 GHz, two port
  S3_path = 'C:\Experiments\oct_05\fri_07\Sparam_filter4\DD_FILT4.txt';
  % S-parameters for Filter 4, original tuning, with diode load, zero ohm
  % series resistor with Vbias = 280 mV
  S4_path = 'C:\Experiments\oct_05\fri_07\Sparam_filter4\DD_FILTDN.txt';
  % S-parameters for Filter 4, zero ohm diode load, filter tuned to
  % correct for change in S11 from original tuning after diode load added
  S5_path = 'C:\Experiments\oct_05\fri_07\Sparam_filter4\DD_FILTDT.txt';
  % S-parameters for Filter 4, after diode tuning, two-port
  S6_path = 'C:\Experiments\oct_05\fri_07\Sparam_filter4\DD_FILTT.txt';
  % S-parameters for Filter 4, after two-port mis-tuning
  S7_path = ...
      'C:\Experiments\oct_05\fri_14\filter4_Sparam_detune\DD_F4DETUN.txt';

% Read files and extract S parameters and frequency range
[S11_1, S21_1, S12_1, S22_1, freq] = vna8510_read(S1_path);
[S11_2, S21_2, S12_2, S22_2, freq] = vna8510_read(S2_path);
[S11_3, S21_3, S12_3, S22_3, freq] = vna8510_read(S3_path);
```

```
[S11_4, S21_4, S12_4, S22_4, freq] = vna8510_read(S4_path);
[S11_5, S21_5, S12_5, S22_5, freq] = vna8510_read(S5_path);
[S11_6, S21_6, S12_6, S22_6, freq] = vna8510_read(S6_path);
[S11_7, S21_7, S12_7, S22_7, freq] = vna8510_read(S7_path);

freqG = freq/1e9;

% Plot S21 for Filter 4 pre-diode and after diode tuning
figure
% subplot(2,1,1)
% plot(freqG, 20*log10(abs(S21_3)),'-', freqG, 20*log10(abs(S21_6)),'--');
plot(freqG, 20*log10(abs(S21_6))); xlabel('Frequency (GHz)')
ylabel('Mag. S21 (dB)') ftitle('S_{21} Magnitude for Pre-diode and
Post-diode Filter 4')
% legend('|S_{21}| Pre-diode', '|S_{21}| Post-diode')


% Plot S11 for Filter 4 pre-diode, with diode before tuning, with diode
% after tuning, and without diode after tuning
figure
% subplot(2,2,1)
plot(freqG, 20*log10(abs(S11_3)), ...
     freqG, 20*log10(abs(S11_4)), ...
     freqG, 20*log10(abs(S11_5)), ...
     freqG, 20*log10(abs(S11_7)))
legend('S11 initial w/o diode', ...
       'S11 initial w/ diode',  ...
       'S11 tuned w/ diode',    ...
       'S11 tuned w/o diode')
xlabel('Frequency (GHz)') ylabel('Mag. S11 (dB)')

% Read in IM3L magnitude and phase as well as the initial and final
% indicated power settings of the cancellation source
% im3l_mag_file = ...
%   load('C:\Experiments\oct_05\fri_07\filter4_im3l_data_2\im3l_mag.txt');
% im3l_phase_file = ...
%  load('C:\Experiments\oct_05\fri_07\filter4_im3l_data_2\im3l_phase.txt');
% s3_init_file = ...
%load('C:\Experiments\oct_05\fri_07\filter4_im3l_data_2\s3_init_power.txt');
% s3_final_file = ...
%load('C:\Experiments\oct_05\fri_07\filter4_im3l_data_2\s3_final_power.txt');

im3l_mag_file = ...
  load('C:\Experiments\oct_05\fri_14\filter4_im3l_data3\im3l_mag.txt');
```

```
im3l_phase_file = ...
  load('C:\Experiments\oct_05\fri_14\filter4_im3l_data3\im3l_phase.txt');
s3_init_file = ...
load('C:\Experiments\oct_05\fri_14\filter4_im3l_data3\s3_init_power.txt');
s3_final_file = ...
load('C:\Experiments\oct_05\fri_14\filter4_im3l_data3\s3_final_power.txt');

% Extract frequency vector for IM3L data
f_im3l = im3l_mag_file(1,:);
% Compute mean values
im3l_mag = im3l_mag_file(2:end,:); im3l_phase =
im3l_phase_file(2:end,:); im3l_mag_mean = mean(im3l_mag,1);
s3_init_power = mean(s3_init_file,1); s3_final_power =
mean(s3_final_file,1);

figure plotyy(f_im3l/1e9, im3l_phase, f_im3l/1e9, im3l_mag_mean)

% figure
% plot(f_im3l/1e9, im3l_phase_mean)
% ftitle('Measured IM3L Phase Response, uncorrected')
% xlabel('Frequency (GHz)')
% ylabel('Phase (^\circ)')

figure plot(f_im3l/1e9, im3l_mag_mean) ftitle('Measured IM3L
Magnitude Response, uncorrected') xlabel('Frequency (GHz)')
ylabel('Magnitude (dBm)')


% Correct the phase shifts in the measured IM3L data
  im3l_phase_mod1 = im3l_phase;

  figure
  plot(f_im3l/1e9, im3l_phase_mod1)
  ftitle('Raw phase data')

  % Unwrap data with conventional technique.  The phase unwrap does not
  % seem to work as expected so two unwrap steps need to occur, no change
  % observed when dimension parameter is changed between row-wise and
  % column-wise.
  im3l_phase_mod2 = 180/pi*unwrap(im3l_phase_mod1*pi/180,1);
  im3l_phase_mod3 = mean(im3l_phase_mod2,1);
  im3l_phase_mod4 = 180/pi*unwrap(im3l_phase_mod3*pi/180);

  figure
```

```
  plot(f_im3l/1e9, im3l_phase_mod4)
  ftitle('Measured IM3L Phase Response, unwrapped')
  xlabel('Frequency (GHz)')
  ylabel('Magnitude (dBm)')


  % Plot comparisons with measurements and theory
  s21_mag = 20*log10(abs(S21_6));
  s21_mag_norm = -max(s21_mag) + s21_mag;
    % Tuned for S11 response without diode
%     s21_mag_tuned = 20*log10(abs(S21_6));
    % Mis-tuned for effect
    s21_mag_tuned = 20*log10(abs(S21_7));

  s21_mag_tuned_norm = -max(s21_mag_tuned) + s21_mag_tuned;
  im3l_mag_norm = -max(im3l_mag_mean) + im3l_mag_mean;

  % Convert magnitude to linear
  s21_mag_norm_lin = sqrt(10.^(s21_mag_norm/10));
  im3l_mag_norm_lin = sqrt(10.^(im3l_mag_norm/10));

  figure
  plot(freqG, s21_mag_norm_lin, f_im3l/1e9, im3l_mag_norm_lin);

  figure
  subplot(2,1,1)
  plot(freqG, 180/pi*unwrap(angle(S21_3)), ...
       freqG, 180/pi*unwrap(angle(S21_6)), ...
       f_im3l/1e9, (im3l_phase_mod4-max(im3l_phase_mod4))/6.7+91.26)
  subplot(2,1,2)
  plot(freqG, s21_mag_norm, ...
       freqG, s21_mag_tuned_norm, ...
       f_im3l/1e9, im3l_mag_norm/2.56)
  ftitle('De-embedded S21 Response')

% De-embed the attenuator phase shift steps that are encountered from the
% power level at which the calibration phase is measured and the power
% level needed for cancellation.  This is only done for the cancellation
% source.  The attenuator steps occur at 11 dB intervals.  Currently only
% data from -70 - +13 dBm is available, so the steps will be extrapolated
% since the step size repeats with a modulus of three.

  % Read in attenuator step data for 10 MHz steps 400-2400 MHz, 1 dB
  % increments.  Data is in form of frequency(row) vs atten step(col) for
```

```
% six steps between -70 - +13 dBm (seven intervals).  The step data is
% in a normalized frequency form, i.e. the phase data is normalized by
% the frequency at which it was taken, thus it is in terms of a delay.
% The normalization is done in the radians form so the frequency term
% used for normalization needs to be in radians.
atten_step = load('C:\Experiments\jan_05\tues_25\atten_phase_step.txt');

% Generate frequency matrix for phase step data
f_step = [400e6:10e6:2400e6];
f_step_mat = repmat(f_step', 1, size(atten_step,2));

% Generate power vector for where the attenuator steps occur.  The data
% is truncated at the top end omitting the final step at +8 dBm.
pstep = [-92 -81 -70 -58 -47 -36 -25 -14 -3];

% Generate average value of step shifts over the three step sizes for
% data Pout < -70 dBm.  This avoids need to generate calibration data for
% these points.  Probably necessary anyways since the 8510 cannot measure
% very small signals.  The extension extends by a full cycle of the
% three phase step amounts.  The extension will be done by taking the
% average of the measured two cycles.
atten_step_low = (atten_step(:,1:3) + atten_step(:,4:6))/2;
% Extend attenuator phase shift step data
atten_step_ext = [atten_step_low, atten_step];

% Find the attenuator bands that are crossed when the cancellation source
% is adjusted from the calibration phase measurement step to the power
% level required for cancellation.  histc command uses bins defined by
% edges and not center values.
[n, bins_init] = histc(s3_init_power, pstep);
[n, bins_final] = histc(s3_final_power, pstep);

% Determine attenuator phase shift step adjustment for each point in the
% frequency sweep.
step_corr = zeros(size(im3l_phase_mod4));
for n=1:size(f_im3l,2)
    % Interpolate between frequency points in calibration data to
    % determine attenuator step phase shift as a function of IM3L
    % frequency.
    atten_step_fn = interp1(f_step, atten_step_ext, f_im3l(n));

    % Sum phase shifts over the attenuator bands crossed by the power
    % change from initial to final.  There is an adjustment to the bins
    % since the pstep vector defines the upper edge of the bin where the
```

```
    % phase shift occurs, i.e. for a 0 bin designation, the phase shift
    % in "bin" 1 needs to be applied.  For the upper limit at the initial
    % power, only the attenuator steps below this power need to be added.
    step_corr([n]) = sum(atten_step_ext(bins_final(n)+1:bins_init(n)));
end
% Apply attenuator step correction, result is normalized phase shift,
% i.e. delay.
im3l_phase_stepcorr = pi/180*im3l_phase_mod4./(2*pi*f_im3l) - step_corr;
% Remove effect of the linear phase shift with power decrease.

% Load the phase shift versus power change slope.  Calibration data
% stored as phase shift slope versus power vs frequency.
m_phi = load('C:\Experiments\jan_05\tues_25\phase_slope.txt');
% Interpolate the slope data for the proper frequency
m_phi_im3l = interp1(f_step, m_phi, f_im3l);

% Compute the indicated power shift of the cancellation source.
pdelta = s3_final_power - s3_init_power;

% Compute phase shift introduced from power shift
slope_corr = pdelta.*m_phi_im3l;
figure
plot(f_im3l,[pi/180*im3l_phase_mod4./(2*pi*f_im3l);step_corr;slope_corr])

% Apply the correction, slope is negative so add the amount which will
% subtract off the positive phase shift that occurred with decreasing the
% power.
im3l_phase_slopecorr = im3l_phase_stepcorr - slope_corr;
im3l_phase_corr = im3l_phase_slopecorr.*f_im3l*2*pi*180/pi;
figure
plot(f_im3l, im3l_phase_corr)
ftitle('IM3L Phase Response Corrected for Source Phase Shift w/Power')

% Now plot the de-embedded corrected phase versus the measured S21 phase,
% both normalized to the same point
s21_phase_untune = interp1(freqG*1e9,180/pi*unwrap(angle(S21_6)),f_im3l);
  % Filter tuned for two-port S11 response
%    s21_phase_tune=interp1(freqG*1e9,180/pi*unwrap(angle(S21_6)), f_im3l);
   % Filter mis-tuned for effect
 s21_phase_tune = interp1(freqG*1e9, 180/pi*unwrap(angle(S21_7)), f_im3l);

s21_phase_untune_norm = s21_phase_untune - s21_phase_untune(45);
s21_phase_tune_norm = s21_phase_tune - s21_phase_tune(45);
```

```
  im3l_phase_corr_norm = im3l_phase_corr - im3l_phase_corr(45);

%   figure
%   plot(f_im3l/1e9, s21_phase_untune_norm, ...
%        f_im3l/1e9, s21_phase_tune_norm, ...
%        f_im3l/1e9, im3l_phase_corr_norm/6)
%
%   axis([f_im3l(1)/1e9 f_im3l(end)/1e9 -400 100])
%   xlabel('Frequency (GHz)')
%   ylabel('S21 Phase (^o)')
%   ftitle('De-embedded Phase Response of Filter 4')
%   legend('Measured S21 w/o tuning', ...
%          'Measured S21 w/ tuning', ...
%          'De-embedded S21',0)
  figure
  % Plot the mis-tuned magnitude data
%   plot(freqG, s21_mag_norm, '-', ...
%        freqG, s21_mag_tuned_norm, '--', ...
%        f_im3l/1e9, im3l_mag_norm/div, '-.')
  % Plot the properly tuned magnitude data
  plot(freqG, s21_mag_norm, '-', ...
       f_im3l/1e9, im3l_mag_norm/div, '-.')
  axis([f_im3l(1)/1e9 f_im3l(end)/1e9 -16 2])
  xlabel('Frequency (GHz)')
  ylabel('S21 Magnitude (dB)')
  ftitle('De-embedded Magnitude Response of Filter 4')
%   legend('Desired S21', ...
%          'Detuned S21', ...
%          'Extracted S21',3)
  legend('Desired S21', ...
         'Extracted S21',3)

% Now need to remove the effect of the interconnect phase insertion on the
% measurements.  These calculations remove the effect of the difference in
% the phase planes used in the measurement system and those needed to
% extract the true phase shift of the DUT response.

% Load files
interconnect_phase_path =
'C:\Experiments\oct_05\fri_07\Sparam_filter4\';
% Phase shift from Port 1 to Port 5
phiP1P5 = load([interconnect_phase_path, 'phiP1P5.txt']);
% Phase shift from Port 1 to Port 3
phiP1P3 = load([interconnect_phase_path, 'phiP1P3.txt']);
```

```
% Phase shift on line 2, SW1 to DUT
phil2 = load([interconnect_phase_path, 'phil2.txt']);
% Phase shift from Port 3 to Port 6
phiP3P6 = load([interconnect_phase_path, 'phiP3P6.txt']);
% Phase shift from Port 4 to Port 5
phiP4P5 = load([interconnect_phase_path, 'phiP4P5.txt']);
% Phase shift from Port 4 to Port 6
phiP4P6 = load([interconnect_phase_path, 'phiP4P6.txt']);
% Phase shift from filter board output to diode board input, i.e. line 5 in
% for Filter 4.
phil5 =
load('C:\Experiments\oct_05\thur_20\S_param_l4_l5\phil5.txt');

% Interpolate for frequency of IM3L response, this likely needs to be
% updated for the actual frequency of each correction term.
phiP1P5_interp = interp1(freqG*1e9, phiP1P5, f_im3l);
phiP1P3_interp = interp1(freqG*1e9, phiP1P3, f_im3l); phil2_interp
= interp1(freqG*1e9, phil2, f_im3l); phiP3P6_interp =
interp1(freqG*1e9, phiP3P6, f_im3l); phiP4P5_interp =
interp1(freqG*1e9, phiP4P5, f_im3l); phiP4P6_interp =
interp1(freqG*1e9, phiP4P6, f_im3l); phil5_interp =
interp1(freqG*1e9, phil5, f_im3l);

phil4_interp = phiP1P5_interp - phiP1P3_interp + phil2_interp;

% Apply correction equation
im3l_phase_true = (pi/180*im3l_phase_corr) + phiP1P5_interp ...
                  - phiP1P3_interp + 2*phil2_interp - phiP3P6_interp ...
                  - phiP4P5_interp + phiP4P6_interp;
im3l_post_inter_correct = (pi/180*im3l_phase_corr) +
phiP1P5_interp ...
                  - phiP1P3_interp + 2*phil2_interp - phiP3P6_interp ...
                  - phiP4P5_interp + phiP4P6_interp;
% This code corrects for the phase shift introduced by the diode.  This was
% extracted from ADS simulation of the diode using the manufacturer's SPICE
% model.  In the simulation, the diode IM3 phase experienced a shift
% of 245 to -330 degrees over the swept frequency range of 1.6 - 1.9 GHz.
% The additional feed line used between the filter and the diode in
% the Filter 4 implementation is also removed (l5 - Line 5).
im3_diode_correct = interp1([1.6e9 1.9e9], [245 -330], f_im3l);
im3l_post_diode_correct = im3l_phase_true -
im3_diode_correct*pi/180; im3l_phase_true = im3l_phase_true -
im3_diode_correct*pi/180;
% Diode coax feed line (line 5) correction
```

```
im3_feed_correct = (phidiv)*phil5_interp; im3l_post_feed_correct =
im3l_phase_true - im3_feed_correct; im3l_phase_true =
im3l_phase_true - im3_feed_correct;

% figure
% plot(f_im3l, im3l_post_inter_correct*180/pi, ...
%      f_im3l, im3l_post_diode_correct*180/pi, ...
%      f_im3l, im3l_post_feed_correct*180/pi)

% Adjust phase shift
im3l_phase_true_norm = im3l_phase_true - im3l_phase_true(45);

figure
% Plot mis-tuned phase data
plot(f_im3l/1e9, s21_phase_untune_norm, '-', ...
    f_im3l/1e9, s21_phase_tune_norm, '--', ...
    f_im3l([plot_limit])/1e9,  ...
    im3l_phase_true_norm([plot_limit])*180/pi/(phidiv), '-.')
% Plot properly tuned phase data
% plot(f_im3l/1e9, s21_phase_untune_norm, '-', ...
%     f_im3l([plot_limit])/1e9,  ...
%     im3l_phase_true_norm([plot_limit])*180/pi/(phidiv), '-.')
xlabel('Frequency (GHz)') ylabel('S21 Phase (^o)')
ftitle('De-embedded Phase Response of Filter 4') legend('Desired
S21', ...
       'Detuned S21', ...
       'Extracted S21',3)
% legend('Desired S21', ...
%        'Extracted S21',3)


% Plot VNA S11, S21, de-embedded S21
figure plot(freqG, 20*log10(abs(S11_7)), '-', ...
       freqG, s21_mag_tuned_norm, '--', ...
       f_im3l/1e9, im3l_mag_norm/div -0.6, '-.')
xlabel('Frequency (GHz)') ylabel('Magnitude (dB)') legend('S11',
'S21--VNA', 'S21--De-embed') axis([1.5 2.1 -8 2])
```

Bibliography

[1] J. Ferrario, R. Wolf, and S. Moss, "Architecting millisecond test solutions for wireless phone RFIC's," *Proceedings International Test Conference*, vol. 1, pp. 1352–1332, Oct. 2003.

[2] A. Halder, S. Bhattacharya, and A. Chatterjee, "Automatic multitone alternate test generation for RF circuits using behavioral models," *Proc. Intl. Test Conf.*, pp. 665–673, 2003.

[3] S. S. Akbay, A Halder, A. Chatterjee, and D. Keezer, "Low-cost test of embedded RF/analog/mixed-signal circuits in SOPs," *IEEE Trans. Adv. Packag.*, vol. 27, pp. 352–363, May 2004.

[4] Y. Zorian, "Testing the monster chip," *IEEE Spectr.*, pp. 54–60, July 1999.

[5] A. Walker, M. Steer, K. Gard, and K. Gharaibeh, "Multi-slice behavioral model of RF systems and devices," *IEEE Radio and Wireless Conference (RAWCON)* 2004, pp. 71–74, Sept. 2004.

[6] N. B. de Carvalho, J. C. Pedro, "A comprehensive explanation of distortion sideband asymmetries," *IEEE Trans. on Microwave Theory and Techn.*, vol. 50, pp. 2090–2101, Sept. 2002.

[7] K. Gharaibeh, and M. Steer, "Modeling distortion in multi-channel communication systems," *IEEE Trans. on Microwave Theory and Techn.*, vol. 53, pp. 1682–1692, May 2005.

[8] J. H. K. Vuolevi, T. Rahkonen, J. P. A. Manninen, "Measurement technique for char-

acterizing memory effects in RF power amplifiers," *IEEE Trans. on Microwave Theory and Techn.*, vol. 49, pp. 1383–1389, Aug. 2001.

[9] H. Ku and J. S. Kenney, "Behavioral modeling of nonlinear RF power amplifiers considering memory effects," *IEEE Trans. on Microwave Theory and Techn.*, vol. 51, pp. 2495–2504, Dec. 2003.

[10] K.M. Gharaibeh, K.G. Gard and M.B. Steer, "Accurate estimation of digital communication system metrics - SNR, EVM and ρ in a nonlinear amplifier environment," 64^{th} *Automated RF Techniques Group Conf. Digest*, pp. 41–44, Dec. 2004.

[11] S. Boyd, Y. S. Tang, and L. O. Chua, "Measuring volterra kernels," *IEEE Trans. on Circuits and Systems*, vol. CAS-30, pp. 571–577, Aug. 1983.

[12] A. Zhu, W. Tianhai, and T. Brazil, "Narrowband and Volterra-based behavioral models of high frequency amplifiers," 58^{th} *Automated RF Techniques Group Conf. Digest*, Nov. 2001.

[13] J. F. Sevic, M. B. Steer and A. M. Pavio, "Nonlinear analysis methods for the simulation of digital wireless communication systems," *Intl. Journal of Microwave and Millimeter-Wave Computer-Aided Engineering*, vol. 6, pp. 197–216, May 1996.

[14] D. D. Weiner and J. F. Spina, *Sinusoidal Analysis and Modeling of Weakly Nonlinear Circuits*, New York: Van Nostrand Reinhold, 1980.

[15] S. Maas, *Nonlinear Microwave and RF Circuits*, Artech House Publishers, 2003.

[16] G. L. Heiter, "Characterization of nonlinearities in microwave devices and systems," *IEEE Trans. on Microwave Theory and Techn.*, vol. MTT-21, pp. 797–805, Dec. 1973.

[17] M. Steer, P. J. Khan and R. S. Tucker, "Relationship between volterra series and generalized power series," *Proc. of the IEEE*, vol. 71, pp. 1453–1454, Dec. 1983.

[18] J. F. Sevic, K. L. Burger and M. B. Steer, "A novel envelope-termination load-pull method for ACPR optimization of RF/microwave power amplifiers," 1998 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 2, pp. 723–726, June 1998.

[19] H. Ku, M. D. McKinley and J. S. Kenney, "Extraction of accurate behavioral models for power amplifiers with memory effects using two-tone measurements," 2002 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 1, pp. 139–142, June 2002.

[20] M. Maqusi, "Characterization of nonlinear distortion in HRC multiplexed cable television systems," *IEEE Trans. on Circuits and Systems*, vol. CAS-32, pp. 605–609, June 1985.

[21] S. A. Mass, "How to model intermodulation distortion," 1991 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 1, pp. 149–151, June 1991.

[22] S. A. Maas, "Third-order intermodulation distortion in cascaded stages," *IEEE Microwave and Guided Wave Lett.*, vol. 5, pp. 189–191, June 1995.

[23] S. A. Maas, B. L. Nelson, and D. L. Tait, "Intermodulation in heterojunction bipolar transistors," *IEEE Trans. on Microwave Theory and Techn.*, vol. 40, pp. 442–448, March 1992.

[24] C. Evans, D. Rees, L. Jones, and M. Weiss, "Periodic signals for measuring nonlinear volterra kernels," *IEEE Trans. Instrum. Meas*, vol 45, pp. 362–371, April 1996.

[25] T. Wang and T. J. Brazil, "A Volterra mapping-based S-parameter behavioral model for nonlinear RF and microwave circuits and systems," 1999 *IEEE MTT-S Int. Microwave Symposium Digest*, pp. 783–786, June 1999.

[26] N. B. Carvalho and J. C. Pedro, "Two-tone IMD asymmetry in microwave power amplifiers," 2000 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 1, pp. 445–448, June 2000.

[27] F. Verbeyst and M.V. Bossche, "The Volterra input-output map of a high frequency amplifier as a practical alternative to load-pull measurements," *Conf. Proc. Instrumentation and Measurement Technology Conference*, vol. 1, pp. 283–286, May 1994.

[28] F. Verbeyst and M. Vanden Bossche, "VIOMAP, 16 QAM and spectral regrowth: enhanced prediction and predistortion based on two-tone black-box model extraction", 45^th *Automated RF Techniques Group Conf. Digest*, May 1995.

[29] B. Razavi, *RF Microelectronics*, Prentice Hall, 1997.

[30] , M. C. Jeruchim, P. Balaban, and K. Sam Shanmugan, *Simulation of Communication Systems Modeling, Methodology, and Techniques*, 2^{nd} Ed., Kluwer Academic/Plenum Publishers, New York, 2000.

[31] K. G. Gard, H. M. Gutierrez, M. B. Steer, "Characterization of spectral regrowth in microwave amplifiers based on the nonlinear transformation of a complex gaussian process," *IEEE Trans. on Microwave Theory and Techn.*, vol. 47, pp. 1059–1069, July 1999.

[32] H. Chen, "Modeling and identification of parallel nonlinear systems: structural classification and parameter estimation methods," *Proc. of IEEE*, vol. 83, pp. 39–66, Jan. 1995.

[33] M. Weiss, C. Evans, and D. Rees, "Identification of nonlinear cascade systems using paired multisine signals," *IEEE Instrumentation and Measurement Technology Conference*, vol. 1, pp. 765–770, May 1997.

[34] J. Tsimbinos and K. V. Lever, "Measurement, composition and inversion of orthogonal representations of memoryless nonlinearities," *IEEE Intl. Symp. on Circuits and Systems*, vol. 4, pp. 2494–2497, May 1993.

[35] T. P. Dobrowiecki and J. Schoukens, "Practical choices in the FRF measurement in presence of nonlinear distortions," *IEEE Trans. on Instrumentation and Measurement*, vol. 50, pp. 2–7, Feb. 2001.

[36] D. Nesic, "Observability for simple wiener and simple Wiener-Hammerstein systems," *Proc. of the American Control Conference*, vol. 4, pp. 2349–2353, June 1998.

[37] P. Crama and J. Schoukens, "Wiener-Hammerstein system estimator initialisation using a random multisine excitation," 58^{th} *Automated RF Techniques Group Conf. Digest*, Nov. 2001.

[38] K. M. Gharaibeh, K. Gard, H. Gutierrez, M. B. Steer, "The importance of nonlinear order in modeling intermodulation distortion and spectral regrowth," 2002 *Radio and Wireless Conference (RAWCON)*, pp. 161–164, Aug. 2002.

[39] H. Ku, M. D. McKinley, J. S. Kenney, "Quantifying memory effects in RF power amplifiers," *IEEE Trans. on Microwave Theory and Techn.*, vol. 50, pp. 2843–2849, Dec. 2002.

[40] K. M. Gharaibeh K. Gard, and M. B. Steer, "The impact on nonlinear amplification on the performance of CDMA systems, *2004 IEEE Radio and Wireless Conference (RAWCON)*, Sep. 2004.

[41] J. C. Pedro, N. B. Carvalho, "Nonlinear modeling of RF/microwave circuits for multi-tone signal analysis," 58^{th} *Automated RF Techniques Group Conf. Digest*, Nov. 2001.

[42] J. C. Pedro, N. B. Carvalho, P. M. Lavrador, "Modeling nonlinear behavior of band-pass memoryless and dynamic systems," 2003 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 3, pp. 2133–2136, June 2003.

[43] N. B. Carvalho, J. C. Pedro, "A comprehensive explanation of distortion sideband asymmetries," *IEEE Trans. on Microwave Theory and Techn.*, vol. 50, pp. 2090–2101, Sep. 2002.

[44] J. C. Pedro and S. A. Maas, "A comparative overview of microwave and wireless power-amplifier behavioral modeling approaches," *IEEE Trans. on Microwave Theory and Techn.*, vol. 53, pp. 1150–1163, June 1972.

[45] J. F. Sevic, and M. B. Steer, "Analysis of GaAs MESFET spectrum regeneration driven by a $\pi/4$-DQPSK modulated source," 1995 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 3, pp. 1375–1378, June 1995.

[46] K. Barkley, "Two-tone IMD measurement techniques," *RF Design*, June 2001.

PIM

[47] J. C. Booth, K. Leong, S. A. Schima, J. A. Jargon, D. C. DeGroot, and R. Schwall, "Phase-sensitive measurement of nonlinearity in high-temperature superconductor thin films," *IEEE Trans. on Appl. Superconduct.*, vol. 15, pp. 1000–1003, June 2005.

[48] A. K. Brown, "Passive intermodulation products in antennas - an overview," *IEE Colloquium on Passive Intermodulation Products in Antennas and Related Structures*, pp. 1/1–1/3, Jun. 1989.

[49] P. L. Lui and A. D. Rawlins, "The design and improvement of PIMP measurement facilities and the measurement of pimp in antenna structures," *IEE Colloquium on Passive Intermodulation Products in Antennas and Related Structures*, pp. 7/1–7/8, Jun. 1989.

[50] P. L. Lui and A. D. Rawlins, "The field measurement of passive intermodulation products," *Intl. Conf. on Mobile Radio and Personal Communications*, pp. 199–203, Dec. 1989.

[51] P. L. Lui and A. D. Rawlins, "Passive non-linearities in antenna systems," *IEE Colloquium on Passive Intermodulation Products in Antennas and Related Structures*, pp. 6/1–6/7, Jun. 1989.

[52] Y. Patenaude, J. Dallaire, F. Menard, S. Richard, "Antenna PIM measurements and associated test facilities," *Antennas and Propagation Society Intl. Symp.*, vol. 4, pp. 620–623, July 2001.

[53] W. B. Mieth, "A cost-effective solution to measurement of hull-generated intermodulation interference on U.S. Navy ships," *IEEE National Symp. on Electromagn. Compat.*, pp. 186–189, May 1989.

[54] J. Jargon, "Measurement comparison of a low-intermodulation termination for the U.S. wireless industry," NIST Tech. Note 1521, July 2001.

[55] J. A. Jargon, D. C. DeGroot, and K. L. Reed, "NIST passive intermodulation measurement comparison for wireless base station equipment," *Automated Rf Techniques Group Conf. Digest*, pp. 128–139, Dec. 1998.

[56] P. L. Liu, A. D. Rawlins, and D. W. Watts, "Measurement of intermodulation products generated by structural components," *Electronics Letters*, vol. 24, pp. 1005–1007, Aug. 1988.

[57] P. L. Liu, "Passive intermodulation interference in communication systems," *Electronics and Communication Engineering Journal*, vol. 2, pp. 109–118, June 1990.

[58] K. Chamberlin, "Quantitative analysis of intermodulation product interference," *IEEE Trans. Electromagn. Compat.*, vol. 31, pp. 316–319, Aug. 1989.

[59] V. Golikov, S. Hienonen, and P. Vainikainen, "Passive intermodulation distortion measurements in mobile communication antennas," *Vehicular Technology Conf.*, vol. 4, pp. 2623–2625, Oct. 2001.

[60] S. Hienonen and A. V. Raisanen, "Passive intermodulation near-field measurements on microstrip lines," 34^{th} *European Micro. Conf. Digest*, pp. 1041–1044, Oct. 2004.

[61] S. Hienonen, V. Golikov, P. Vainikainen, and A. V. Raisanen, "Near-field scanner fro the detection of passive intermodulationsources in base station antennas," *IEEE Trans. Electromagn. Compat*, vol. 46, pp. 661–667, Nov. 2004.

[62] B. G. M. Helme, "Passive intermodulation of ict components," *IEE Colloquium on Screening Effectiveness Measurements*, pp. 1/1–1/8, May 1998.

[63] D. Weinstein, "Passive intermodulation distortion in connectors, cable and cable assemblies," Amphenol application note.

[64] B. Rosenberger, "The measurement of intermodulation products on passive components and transmission lines," *IEEE Symposium on Technologies for Wireless Applications Digest*, pp. 57–62, Feb. 1999.

[65] J. Sanford, "Passive intermodulation in antenna design," *Antennas and Propagation Society International Symposium Digest*, vol. 3, pp. 1651–1654, June 1993.

[66] M. Vladimirescu, R. Kwiatkowski, and K. Engel, "Tunnel conduction consequences in high frequency microcontacts; passive intermodulation effect," 50^{th} *IEEE Holm Conference on Electrical Contacts*, pp. 152–159, 2004.

[67] Agilent, "Signal Studio for Enhanced Multitone Software A.01.00 through A.03.01 Release Descriptions," 2003.

[68] T. Van den Broeck and Jan Verspecht, "Calibrated vectorial nonlinear-network analyzers," 1994 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 2, pp. 1069–1072, May 1994.

[69] J. Verspecht, P. Debi, A. Barel, and L. Martens, "Accurate on wafer measurement of phase and amplitude of the spectral components ofincident and scattered voltage waves at the signal ports of a nonlinear microwave device," 1995 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 3, pp. 1029–1032, June 1995.

[70] J. Jargon, K.C. Gupta, D. Schreurs, K. Remley, and D. DeGroot, "A method of developing frequency-domain models for nonlinear circuits based on large-signal measurements," 58^{th} *Automated RF Techniques Group Conf. Digest*, Nov. 2001.

[71] C. van Niekerk, D. Schreurs, and P. Meyer, "Recent developments in non-linear device modeling techniques," *IEEE AFRICON 1999*,vol. 2, pp. 1105–1110, Sept. 1999.

[72] N. Tufillaro, D. Usikov, L. Barford, D. M. Walker, and D. Schreurs, "Measurement driven models of nonlinear electronic components," 55^{th} *Automated RF Techniques Group Conf. Digest*, pp. 1–8, June 2000.

[73] K. A. Remley, D. C. DeGroot, J. A. Jargon, and K.C. Gupta, "A Method to compare vector nonlinear network analyzers," 2001 *IEEE MTT-S Int. Microwave Symposium Digest*, 2001, vol. 3, pp. 1667–1670, June 2001.

[74] S. Vandenplas, F. Verbeyst, E. vandamme, and M. Vanden Bossche, "Calibration issues for the large signal network analyzer (LSNA)," 60^{th} *Automated RF Techniques Group Conf. Digest*, pp. 99–106, Dec. 2002.

[75] D. M. Schreurs, J. Verspecht, S. Vandenberghe, and E. Vandamme, "Straightforward and accurate nonlinear device model parameter-estimation method based on vectorial large-siganl meausrements," *IEEE Trans. on Microwave Theory and Techn.*, vol. 50, pp. 2315–2319, Oct. 2002.

[76] S. K. Myoung, X. Cui, D. Chaillot, P. Roblin, F. Berbeyst, M. Vanden Bossche, S. J. Doo, and W. Dai, "Large signal network analzyer with trigger for baseband & RF system characterization with application to K-modeling & output baseband modulation linearation," 64^{th} *Automated RF Techniques Group Conf. Digest*, pp. 189–195, Dec. 2004.

[77] K. A. Remley, D. F. Williams, D. M. Schreurs, G. Loglio, and A. Cidronali, "Phase detrending for measured multisine signals," 61^{st} *Automated RF Techniques Group Conf. Digest*, pp. 73–83, June 2003.

[78] W. Grabinski, E. Vandamme, D. Schreurs, H. Maeder, O. Pilloud, and C. C. McAndrew, "5.5 GHz LSNA MOSFET modeling for RF CMOS circuit design," 60^{th} *Automated RF Techniques Group Conf. Digest*, pp. 39–47, Dec. 2002.

[79] K. Remley, D. Williams, D. Schreurs, and J. Wood, "Simplifying and interpreting two-tone measurements," *IEEE Trans. on Microwave Theory and Techn.*, vol. 52, pp. 2576–2584, Nov. 2004.

[80] D. M. Pozar, *Microwave Engineering*, John Wiley and Sons, Inc., 1998.

[81] Microlab FXR website.

[82] P. D. L. Beasley, A. G. Stove, B. J. Reits, and B-O. As, "Solving the problems of a single antenna frequency modulated CW radar," *IEEE Intl. Radar Conference*, pp. 391–395, May 1990.

[83] J. K. Cavers, "Adaptation behavior of a feedforward amplifier linearizer," *IEEE Trans. on Vehicular Technology*, vol. 44, pp. 31–40, Feb. 1995.

[84] A. M. Smith and J. K. Cavers, "A wideband architecture for adaptive feedforward linearization," *IEEE Vehicular Conf. Digest*, vol. 3, pp. 2488–2492, May 1998.

[85] "250–1000 MHz Quadrature Modulator," Analog Devices datasheet.

[86] "50 Hz to 2.7 GHz 60 dB TruPwr Detector," Analog Devices datasheet.

[87] "Dual 10–Bit TxDAC+ with 2 x Interpolation Filters," Analog Devices datasheet.

[88] A. Walker, M. Steer, K. Gard, "Simple, broadband relative phase measurement of intermodulation products," 65^{th} *Automated RF Techniques Group Conf. Digest*, June 2005.

[89] U. Lott, "A method for measuring magnitude and phase of harmonics generated in nonlinear microwave two-ports," 1998 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 1, pp. 225–228, May 1998.

[90] P. Heymann, R. Doerner, and M. Rudolph, "Multiharmonic generators for relative phase calibration of nonlinear network analyzers," *IEEE Trans. on Instrumentation and Measurement*, vol. 50, pp. 129–134, Feb. 2001.

[91] N. Suematsu, T. Shigematsu, Y. Iyama, and O. Ishida, "Transfer characteristic of IM3 relative phase for a GaAs FET amplifier," 1997 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 2, pp. 901–904, June 1997.

[92] Y. Yang, J. Yi, J. Nam, B. Kim, and M. Park, "Measurement of two-tone transer characteristics of high-power amplifiers," *IEEE Trans. on Microwave Theory and Techn.*, vol. 49, pp. 568–571, Mar. 2001.

[93] D. J. Williams, J. Leckey, and P. J. Tasker, "A study of the effect of envelope impedance on intermodulation asymmetry using a two-tone time domain measurement system," 2002 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 3, pp. 1841–1844, June 2002.

[94] J. Dunsmore and D. Goldberg, "Novel two-tone intermodulation phase measurement for evaluating amplifier memory effects," 33^{rd} *European Microwave Conference*, vol. 1, pp. 235–238, Oct. 2003.

[95] T. R. Turlington, "Behavioral modeling of nonlinear RF and microwave devices," Artech House Publishers, 1999.

[96] N. Mizusawa and S. Kusunoki, "Third and fifth order base-band component injection for linearization of the power amplifier in a cellular phone," 2005 *IEEE MTT-S Int. Microwave Symposium Digest*, pp. 1565–1568, June 2005.

[97] L. Gustafsson, G. H. Hansson, and K. I. Lundstrom, "On the use of describing functions in the study of nonlinear active microwave circuits," *IEEE Trans. on Microwave Theory and Techn.*, vol. 20, pp. 402–409, June 1972.

[98] D. Haigh, D. Webster, R. Kaunisto, C. Nduijuba, A. Khanifar, M. Darvishzadeh, T. Parker, J. Scott, and I. Thayne, "Developments in RF circuit design," *IEE Colloquium on RF Design Scene*, pp. 4/1–4/10, Feb. 1996.

[99] C. Borchers, L. Hedrich and E Barke, "Equation-based behavioral model generation for nonlinear analog circuits," *Design Automation Conference Proc.*, pp. 236–239, June 1996.

[100] N. B. Tufillaro and D. M. Walker, "Behavioral models of microwave circuits with fading memory," 56^{th} *Automated RF Techniques Group Conf. Digest*, Dec. 2000.

[101] C. Cho, W. R. Eisenstadt, B. Stengel, and E. Ferrer, "IIP3 estimation from the gain compression curve," *IEEE Trans. on Microwave Theory and Techn.*, vol. 53, pp. 1197–1201, April 2005.

[102] R. S. Tucker, "Third-order intermodulation distortion and gain compression in GaAs FET's," *IEEE Trans. on Microwave Theory and Techn.*, vol. 27, pp. 400–408, May 1979.

[103] A. Saleh, " Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Trans. on Commun.*, vol. 29, pp. 1715–1720, Nov. 1981.

[104] H. Bosma, "A general model for junction circulators; choice of magnetization and bias field," IEEE Trans. on Magn., vol. MAG-4, pp. 587–596, Sept. 1968.

[105] Amplifier Research, "Operating and Service Manual (Model 10W1000C)," 1997.

[106] K. H. Sann, "The measurement of near-carrier noise in microwave amplifiers," *IEEE Trans. on Microwave Theory and Techn.*, vol. 9, pp. 761–766, Sep. 1968.

[107] F. L. Walls, "Suppressed carrier based PM and AM noise measurement techniques," *IEEE Intl. Freq. Cntrl. Symp.*, pp. 485–492, May 1997.

[108] E. Rubiola and V. Giordano, "Dual carrier suppression interferometer for measurement of phase noise," *Electronics Letters*, vol. 36, pp. 2073–2075, Sep. 2000.

[109] E. Rubiola and V. Giordano, "Advanced interferometric phase and amplitude noise measurements," *Review of Scientific Instruments*, vol. 73, pp. 2445–2457, June 2002.

[110] F. L. Walls, "PM and AM noise of combined signal sources," *IEEE Intl. Freq. Ctrl. Symp. Digest*, pp. 532–540, May, 2003.

[111] E. C. Ifeachor, B. W. Jervis, *Digital Signal Processing A Practical Approach,* Prentice Hall, 2^{nd} edition, 2001.

[112] Maury Microwave Corporation, "Large signal network analyzer technology preliminary product overview," May, 2003.

[113] ARFTG, "Minutes of the $4th$ ARFTG NVNA Users' Forum," June, 2004.

[114] Agilent Technologies, "Calibration Guide Agilent Technologies 8560 E-Series and EC-Series Spectrum Analyzers,", vol. I and II, February 2002.

[115] C. Crespo–Cadenas, J. Reina–Tosina, and M. J. Madero–Ayora, "Phase characterization of two–tone intermodulation distortion," 2005 *IEEE MTT-S Int. Microwave Symposium Digest*, June 2005.

[116] J. C. Pedro, J. P. Martins, and P. M. Cabral, "New method for phase characterization of nonlinear distortion products," 2005 *IEEE MTT-S Int. Microwave Symposium Digest*, June 2005.

[117] S. A. Maas and D. Tait, "Parameter-extraction method for heterojunction bipolar transistors," *IEEE Microwave and Guided Wave Lett.*, vol. 2, pp. 502–504, Dec. 1992.

[118] C. T. Armijo and R. G. Meyer, "A new wide-band Darlington amplifier," *J. Solid-State Circuits*, vol. 24, pp. 1105–1109, Aug. 1989.

[119] A. L. Berman and C. E. Mahle, "Nonlinear phase shift in traveling-wave tubes as applied to multiple access communications satellites," *Trans. on Commun. Technol.*, vol. 18, pp. 37–48, Feb. 1970.

[120] V. Miraftab and R. R. Mansour, "A robust fuzzy-logic technique for computer-aided diagnosis of microwave filters," 2005 *IEEE MTT-S Int. Microwave Symposium Digest*, June 2005.

[121] P. Harscher and R. Vahldieck, "Automated computer-controlled tuning of waveguide filters using adaptive network models," *IEEE Trans. on Microwave Theory and Techn.*, vol. 49, pp. 2125–2130, Nov. 2001.

[122] L. H. Chua and D Mirshekar-Syahkal, "Accurate and direct characterization of high-Q microwave resonators using one-port measurement," *IEEE Trans. on Microwave Theory and Techn.*, vol. 51, pp. 978–985, Mar. 2003.

[123] R. R. Mansour, "Filter technologies for wireless basestations," *IEEE Microwave*, pp. 68–74, Mar. 2004.

[124] G. Matthaei, E. M. T. Jones, L. Young, *Microwave Filters, Impedance-Matching Networks, and Coupling Structures,* Artech House Publishers, Feb. 1980.

[125] I. C. Hunter, *Theory and Design of Microwave Filters, Institution of Electrical Engineers*, London, UK, 2001.

[126] J. M. Webber, "RF test requirements for cellular formats: a comparitive analysis," *IEEE Systems Readiness Technology Conf. (AUTOTESTCON)*, pp. 630–635, Sept. 2003.

[127] J. Lukez, "Novel techniques for wideband RF test," *IEEE Electronics Manufacturing Technology Symposium*, pp. 423–425, July 2002.

[128] D. Brown, J. Ferrario, R. Wolf, J. Li, and J. Bhagat, "RF testing on a mixed signal tester," *Proc. Intl. Test Conf.*, pp. 793–800, 2004.

[129] J. Ferrario, R. Wolf, and H. Ding, "Moving from mixed signal to RF test hardware development," *Proc. Intl. Test Conf.*, pp. 948–956, 2001.

[130] R. Voorakaranam, S. Cherubal, and A. Chatterjee, "A signature test framework for rapid production testing of RF circuits," *Proc. Design, Automation and Test in Europe*, pp. 186–191, Mar. 2002.

[131] J. E. Reinhardt, A. V. Fogle, and d. E. Dunstone, "Automated process cuts filter tuning time from hours to minutes," *Microwaves and RF*, pp. 103–104, June 2001.

[132] "LASERtrim tuning application notes," Johanson Technology Datasheet LZT983, 1998.

[133] M. Dishal, "Alignment and adjustment of synchronously tuned multiple-resonant-circuit filters," *PVOC. IRE*, vol. 39, pp. 1448-1455, Nov. 1951.

[134] L. Accatino, G. Bertin, M. Mongiardo, and G. Resnati, "Dual-mode filters with grooved/splitted dielectric resonators for cellular-radio base stations," *IEEE Trans. on Microwave Theory and Techn.*, vol. 50, pp. 2882–2889, Dec. 2002.

[135] "Hints for making better spectrum analyzer measurements," Agilent application note, AN 1286-1.

[136] P. Stepanek, D. Kurtz, and J. Gorin, "Coherent addition of intermodulation distortion in spectrum analyzers," 2003 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 2, pp. 1415–1417, June 2003.

[137] A. Telegdy, "Extending the dynamic range of RF/microwave intermodulation measuremnts by multiple-carrier cancellation," *IEEE Intl. Persona, Indoor and Mobile Radio Communications*, vol. 1, pp. 251–255, Sep. 2000.

[138] S. N. Stitzer, H. Goldie, and P.R. Emtage, "Frequency selective high power YIG limiters," 1976 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 76, pp. 272–274, June 1976.

[139] S. N. Stitzer, P.S. Carter, and H. Goldie, "A high power X-band frequency selective passive YIG limter," 1977 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 77, pp. 528–531, June 1977.

[140] S. N. Stitzer and H. Goldie, "A multi-octave frequency selective limiter," 1983 *IEEE MTT-S Int. Microwave Symposium Digest*, vol. 83, pp. 326–328, May 1983.

[141] A. Nag and K. R. Rao, "Narrow band noise suppression scheme for improving signal to noise ratio," *Proceedings Intl. Conf. on VLSI Design*, pp. 25–29, Jan. 2003.

[142] J. D. Adam, L. E. Davis, G. F. Dionne, E. F. Schloemann, and S. N. Stitzer, "Ferrite Devices and Materials," *IEEE Trans. on Microwave Theory and Techn.*, vol. 50, pp. 721–737, Mar. 2002.

[143] M. Soma, W. Heaileselassie, and J. Sherrid, "Measurement of phase and frequency variations in radio-frequency signals," *IEEE VLSI Test Symposium*, pp. 203–208, May, 2003.

[144] Stephen Boyd and Leon O. Chua, "Uniqueness of a basic nonlinear structure," *IEEE Trans. Circuits Systems,* vol. cas-30, pp. 648–651, 1983.

[145] Stephen P. Boyd and Leon O. Chua, "Uniqueness of circuits and systems containing one nonlinearity," *IEEE Trans. Automatic Control,* vol. ac-30, pp. 674–681, 1985.

[146] H. Schuman, "Time-domain scattering from a nonlinearly loaded wire," *IEEE Trans. on Antennas and Propagation*, vol. 22, pp. 611–613, July 1974.

[147] T. K. Liu, F. M. Tesche, "Analysis of antennas and scatters with nonlinear loads," *IEEE Trans. on Antennas and Propagation*, vol. AP-24, pp. 131–139, Mar. 1976.

[148] T. K. Sarkar, "Scattering analysis of nonlinearly loaded antennas," *IEEE Trans. on Antennas and Propagation*, vol. AP-24, pp. 125–131, Mar. 1976.

[149] M. Kanda, "Analytical and numerical techniques for analyzing an electrically short dipole with a nonlinear load," *IEEE Trans. on Antennas and Propagation*, vol. AP-28, pp. 71–77, Jan. 1980.

[150] J. Obregon and F. Farzaneh, "Definition of nonlinear reflection coefficient of a microwave device using describing function formalism," *IEEE Trans. on Antennas and Propagation*, vol. 32, pp. 452–455, Apr. 1984.

[151] A. I. Kozlov, A. I. Logvin, "Images of radar targets in non-linear scattering of electromagnetic waves," *Combined Optical-Microwave Earth and Atmospher Sensing*, pp. 66–68, April 1995.

[152] R. Luebbers, J. Beggs, K. Chamberlin, "Finite difference time-domain calculation of transients in antennas with nonlinear loads," *IEEE Trans. on Antennas and Propagation*, vol. 41, pp. 566–573, May 1993.

[153] T. M. Zaboronkova, "Electromagnetic wave scattering by loaded dipole antennas," *Proceedings of the Second International Symposium of Trans Black Sea Region on Applied Electromagnetism*, pp. 5–6, June 2000.

[154] C. Charbonniaud, J.P. Teyssier, and R. Quere, "Time-domain pulsed large-signal nonlinear characterization of microwave transistors," 33^{rd} *European Microwave Conference*, vol. 1, pp. 241–244, Oct. 2003.

[155] D. F. Williams, F. Ndagijimana, K. A. Remley, J. A. Dunsmore, and S. Hubert, "Scattering-parameter models and representations for microwave mixers," *IEEE Trans. on Microwave Theory and Techn.*, vol. 53, pp. 314–321, Jan. 2005.

[156] J. Verspecht, D. F. Williams, D. Schreurs, K. A. Remley, and M. D. McKinley, "Linearization of large-signal scattering functions," *IEEE Trans. on Microwave Theory and Techn.*, vol. 53, pp. 1369–1375, April 2005.

[157] A. Soury, E. Ngoya, and J. Rousset, "Behavioral modeling of RF and microwave circuit blocs for hierarchical simulation of modern transceivers," 2005 *IEEE MTT-S Int. Microwave Symposium Digest*, June 2005.

[158] A. Ushida and L. O. Chua, "Frequency-comain analysis of nonlinear circuits driven by multi-tone signals," *IEEE Trans. Circuits Syst.* vol. 9, pp. 766–779, Sept. 1984.

[159] L. DeLocht, G. Vandersteen, P. Wambacq, Y. Rolain, R. Pintelon, J. Schoukens, and S. Donnay, "Identifying the main nonlinear contributions: use of multisine excitations during circuit design," 64^{th} *Automated RF Techniques Group Conf. Digest*, pp. 75–84, Dec. 2004.