IMPACT OF NOISES AND NONLINEARITY ON ANALOG SELF-INTERFERENCE CANCELLATION IN IN-BAND FULL-DUPLEX COMMUNICATIONS

by

Jonathan M. Shilling

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Engineering

Department of Electrical and Computer Engineering

Fort Wayne, Indiana

December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Chao Chen, Chair

Department of Electrical and Computer Engineering

Dr. Todor Cooklev

Department of Electrical and Computer Engineering

Dr. Carlos Pomalaza-Ráez

Department of Electrical and Computer Engineering

Dr. Bin Chen

Department of Electrical and Computer Engineering

Approved by:

Dr. Chao Chen

Dedicated to the billions upon billions of stars that burned too bright, too long and whose eventual culmination would be the amalgamation of those I call parents, friends, and family. Oh, what wonderful stardust!

ACKNOWLEDGMENTS

During my graduate studies I was fortunate enough to have Prof. Chao Chen as both an academic advisor and then thesis advisor. Her willingness to expand and exceed the bounds of her academic interests allowed me to develop the interests in mine. Her intellect, along with a patient and understanding demeanor, were a potent combination that served as guiding light on this project. Additionally, I would like to thank all my previous professors, including serving committee members Dr. Todor Cooklev, Dr. Pomalaz-Ráez, and Dr. Bin Chen. Unbeknownst at the time, their instructive coursework would serve as some of the galvanizing and foundational knowledge forming the backbone of this thesis. Additionally, I want to thank Purdue Fort Wayne for funding and access to their laboratories and equipment. Finally, my deepest gratitude to Valerie. For without her willingness and support, this opportunity and venture may have never been realized. To all those mentioned above, and the ones too multitudinous to explicitly list, a sincere and emphatic thank you for all your support. This project's fruition owes itself to the kindest and assistance bestowed on me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

ADC	Analog-to-Digital Converter
APA	Affine Projection Algorithms
ASC	Analog Self-Interference Cancellation
BGD	Batch Gradient Descent
BPF	Bandpass Filter
CW	Continuous-Wave
DAC	Digital-to-Analog Converter
DSC	Digital Self-Interference Cancellation
DSL	Digital Subscriber Line
DUT	Device Under Test
FD	Full-Duplex
FDD	Frequency Division Duplexing
FDM	Frequency Domain Multiplexing
FDMA	Frequency Domain Multiple Access
FIR	Finite Impulse Response
HD	Half-Duplex
IBFD	In-Band Full-Duplex
IMD	Intermodulation Distortion
IoT	Internet-of-Things
LFSR	Linear Feedback Shift Registers
LMS	Least-Mean-Squares
LNA	Low Noise Amplifier
LTE	Long Term Evolution
MAC	Medium/Media Access Control
MIMO	Multiple-Input Multiple-Output
MSE	Mean-Squared-Error
NLMS	Normalized-Least-Mean-Squares
NSD	Noise Spectral Density

OOB	Out-of-Band
PDF	Probability Distribution Function
PHY	Physical (Layer)
PLL	Phase Lock Loop
PRBS	Pseudo-Random Bit Sequence
PSD	Phase Spectral Density
PSK	Phase Shift Keying
PW	Pulsed-Wave
QAM	Quadrature Amplitude Modulation
QoS	Quality-of-Service
RLS	Recursive Least Squares
RPC	Reflected Power Canceller
SFDR	Spurious-Free Dynamic Range
SGD	Stochastic Gradient Descent
SI	Self-Interference
SIC	Self-Interference Cancellation
SOI	Signal-of-Interest
TDD	Time Division Duplexing
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
UTP	Unshielded Twisted Pair
VCSO	Voltage Controlled Saw Oscillator

ABSTRACT

A wireless revolution has occurred resulting in the formation of a proverbial backbone of wireless devices that our everyday functionality, productivity, and general way of life have become dependent. Consequently, victimizing an already constrained and finite wireless spectrum with further demands for increased bandwidths, greater channel capacities, and an insatiable plea for faster access rates. In-band full-duplexing (IBFD) is an innovative and encouraging technology that aims to answer this tacit mitigation call by bolstering spectral efficiency through simultaneous same frequency band transmission and reception. Conventionally, transceiver-based systems have their respective transmission and reception dictated by occurring in either disparate time slots (half-duplex) or distinct frequencies (out-of-band full-duplex). By achieving simultaneous same band communication, a theoretical doubling in spectral efficiency is rendered feasible. However, transmitter to receiver leakage, or self-interference (SI), remains the most barring frustration to IBFD realization. Being locally generated, SI is considerably stronger (often 50-100dB) than the desired signal-of-interest (SOI). Left unresolved, this unwanted energy saturates the receiver's amplifiers and desensitizes its analog-to-digital converters. Thus, rendering the SOI unintelligible. Therefore, a means of self-interference cancellation (SIC) is necessitated to suppress any polluting SI to levels that of or below the receiver's noise floor.

In this thesis an in-depth history of in-band full duplex technology is first presented, followed by a condensed examination of the SIC domains. Pertinent theory is presented pertaining to noise analysis and estimation relevant to a proposed IBFD transceiver architecture. Finally, a modelled simulation of this transceiver, developed in MATLAB, is presented. Subsequent results detailing an investigative study done on a fully adaptive tapped-branch analog self-interference canceller are shown. Said canceller's variable phase and amplitude weights are set via real-time training using gradient descent algorithms. Evaluation of the results reveal marginal effect on the SIC efficacy due to transmission path nonlinearity and noise distortions alone. However, expansion of model consideration for conceivable cancellation hardware nonlinearities reveals an indirectly proportional degradation of SIC performance by up to 35dB as distortion levels vary from -80 dBm to -10 dBm. These results indicate consideration of such non-idealities should be an integral part of cancellation hardware design for the preclusion of any intrinsic cancellation impediments.

1. INTRODUCTION

Modern life has witnessed the formation of an inextricable link between wireless technology and daily existence. In fact, the ubiquity of cellular networks, Wi-Fi, and an amalgamation of the Internet of Things (IoT) has galvanized efforts to develop novel techniques to augment the capacity of our already limited wireless spectrum. Techniques such as Long-Term Evolution (LTE) were originally introduced in 4G networks as a means of providing high-speed guaranteed Quality-of-Service (QoS) for mobile devices. However, with the advent of 5G networks and development on next generation 6G networks proceeding, it has become unequivocally clear that meticulous and efficient spectrum utilization schemes are necessary to mitigate this impingement if higher capacity, higher data rate access is to be achieved. In-Band Full-Duplex (IBFD) operation is one of these promising techniques recently garnering attention. Particularly alluring, IBFD operation poses the potential of effectively doubling the capacity of any network in which it is implemented [1].

Conventional communication theory has long held the dogmatic axiom that for successful communication, transceivers sharing the same physical medium must transmit and receive in either non-overlapping timeslots or non-overlapping frequency bands. Though, orthogonality through spread-spectrum coding schemes has been another option sometimes employed to avoid interference. IBFD, on the other hand, defiantly challenges these previously accepted "dictums" and strives to establish simultaneous, identical spectrum transmission and reception. By doing so, IBFD theoretically doubles spectral efficiency by removing temporal and frequency constraints.

1.1 Wireless Communication Systems Duplexing

A certain degree of ambiguity has become inherent when discussing the duplexity of a communication system. In particular, it seems that what constitutes a half-duplex (HD), and full-duplex (FD) system has become obscured with a bifurcation apparent between distinct technological industries. However, ostensibly what is agreed upon is that duplexity is a term that connotes the permissible propagation direction between a network's nodes. For the sake of clarity, we will define our terms and how they may be regarded throughout the course of this thesis.

Simplex communication, the most basic, refers to communication in which information is transmitted solely in one direction. That is, communication from sole transmitter to sole receiver (e.g., garage door opener, basic baby monitors, remote controllers, etcetera.). Half-duplex (HD) communication will be referred to networks or systems capable of bi-directional transmission and reception, with the caveat that engaged nodes may not simultaneously transmit and receive(e.g., walkie-talkies, two-way radios, push-to-talk cellular communications). Finally, full-duplex (FD) will be applied to communication systems in which simultaneous reception and transmission is allowed. Figure 1 below illustrates these transmission principles.

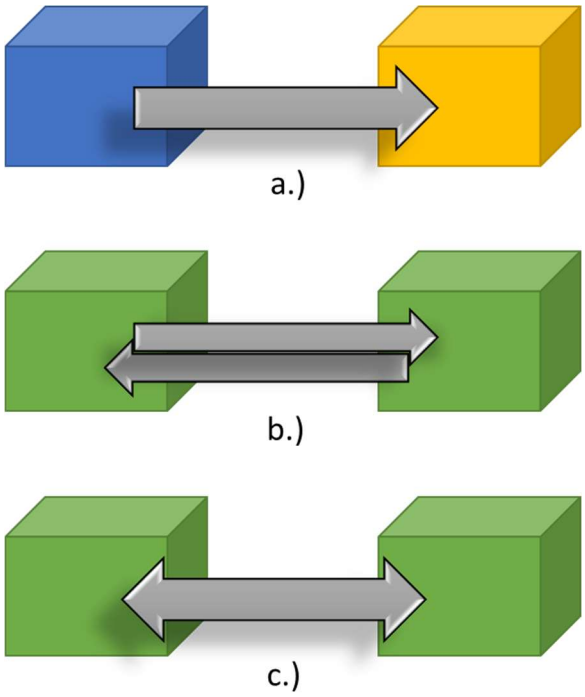

Figure 1 Modes of Duplexity a.) Simplex b.) Half-Duplex c.) Full-Duplex

1.2 Traditional Spectrum Allocation Technologies

Viewing a network's physical transmission medium as a two-dimensional aggregate composed of both finite frequency and time components, one can easily surmise that in order to eschew inter-nodal interference in bidirectional communication, a degree of resource sharing must be incorporated. Traditionally, this has been achieved by means of allocating either unique frequency bands or discrete time slots.

1.2.1 Frequency Division

Communication networks that operate under principles of frequency division attempt to offer unfettered temporal resource access to all constituents by means of frequency allocation. As an older methodology, early telephone systems, television broadcasting, and FM and AM radio were developed under these concepts. Nevertheless, this approach is often still incorporated into modern systems. For example, cellular networks, including 4G and 5G networks, are capable of Frequency-Division Duplexing.

Frequency Division Multiplexing (FDM) refers to the physical layer bandwidth allocation technique in which a network's total useable spectrum is sub-divided into disparate frequency bands [2]. Each band in the communication medium subsequently functions as a distinct channel. Given sufficient out-of-band (OOB) filter rejection and adequate guard band buffering, this allows for non-interfering transmission. Figure 2 below illustrates the fundamental principle of FDM operation.

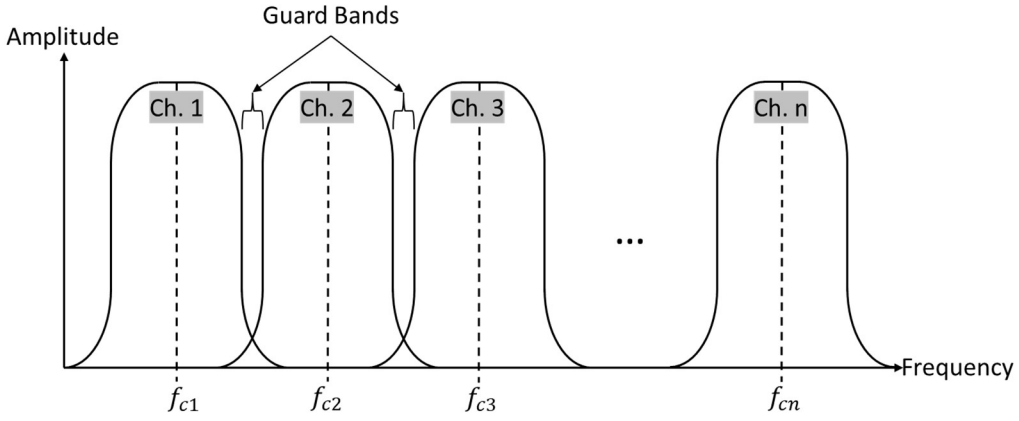

Figure 2 Frequency Division Multiplexing

Frequency Division Duplexing (FDD) refers to the practice of allocating two of these frequency-disjoint channels to a node for simultaneous transmission and reception [3]. An example of a system utilizing this technique is a Digital Subscriber Line (DSL) network. In DSL networks the existing extensive network of landline telephone's unshielded twisted pair (UTP) cable is used to transmit data both to and from the end user; referred to as the downstream and upstream, respectively. In asymmetrical DSL bidirectional simultaneous data transfer is achieved by

allocating for downstream data the band from 138 kHz to 1104 kHz while upstream occupies the region from 25.875 kHz to 138 kHz.

At the expense of potential bandwidth utilization inefficiency, FDM and FDD incorporating networks offer the theoretical potential for zero queueing delay and true simultaneous transmission due to the nodal Tx/Rx frequency perquisite. This of course only holds true given a static model, where the number of nodes is equal to, or less than, the number of allocated transmission or reception channels. For dynamic networks with inadequate channel availability Frequency Division Multiple Access (FDMA) is generally employed. FDMA is a data-layer channel access method that establishes protocol by which multiple users can utilize the same channel [4]. Typically, FDMA achieves this by further subdividing each channel into smaller frequency allocations. Statistical demand assignment is often also used in FDMA to allocate higher demand nodes larger band allocations. These allocations may also be temporary, adjustable, and revokable as beneficially deemed necessary.

1.2.2 Time Division

Contrary to frequency division, time division attempts to offer full spectrum access to network constituents by means of temporal manipulation. Often perceived as easier to implement with digital transmissions, time-division techniques may be more conducive to modern communication which frequently employs digital transmission due to a multitude of apparent benefits such as better noise immunity, heightened security, and ease of signal processing. Salient concepts forming the foundation of time-division schemes include Time-Division Multiplexing (TDM), Time-Division Duplexing (TDD, and Time-Division Multiple Access (TDMA). In fact, 5G New Radio standards proposes TDD to be the sole access technology utilized in the C-, K-, and Ka-bands.

Similar to FDM, TDM is a multiple user access PHY layer technology for a network's transmission medium [2]. TDM delineates periodizing the temporal domain and partitioning it into distinct equal-sized allocable timeslots. Each node is assigned their own distinct timeslot, and during so effectively gains full transmission bandwidth. As period and timeslot duration are somewhat arbitrary in manner chosen based on apparent network benefit, time-sharing schemes often find

increased network user capacity comparatively to frequency counterparts. Figure 3 below illustrates the basic concept in TDM.

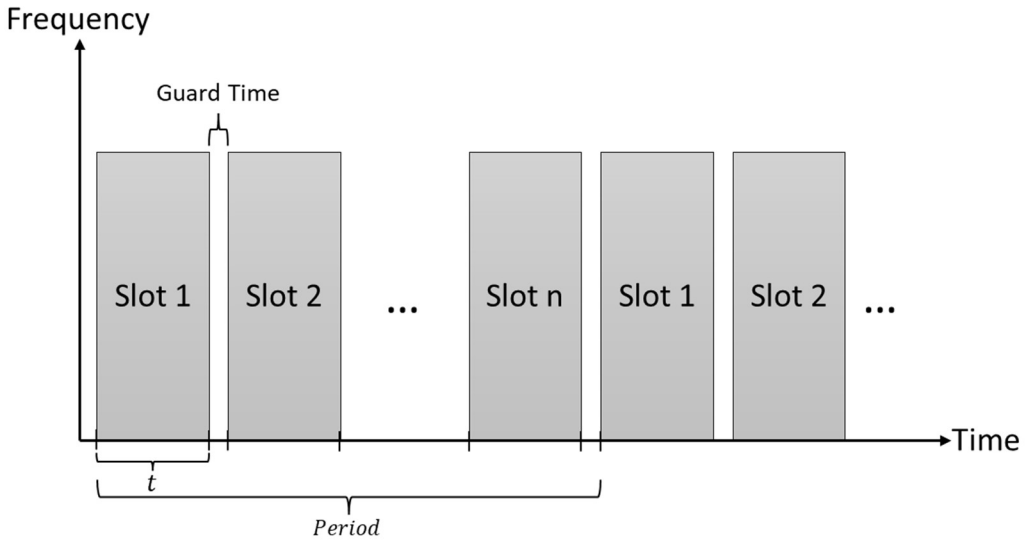

Figure 3 Time Division Multiplexing

Akin to TDM, TDD refers to the PHY layer method of separating the transmission and reception of signals into distinct periods of time [3]. Referring back to our FDD DSL downstream/upstream transmission example spanning 25.875 kHz to 1104 kHz, in TDD both data streams would be allocated the complete 1,078.125 kHz of network bandwidth. However, simultaneous transmission and reception would not be possible and require at least two distinct time slots, with proper timing guard slots. Though, in practice many more than two timeslots would likely be allotted per period. As downstream/upstream traffic often displays inherent bursty and asymmetrical characteristics, multiple timeslots allow proportionate time allocation to the heavier traffic load side, which typically is downstream. By doing so the susceptibility to inefficient network utilization experienced with equal time allocation is alleviated. To further increase a network's user density TDMA is often additionally incorporated.

TDMA, similar to TDM, provides a means for multiple users to access a shared resource by apportioning timeslots [4]. However, TDMA commonly refers to the channel access method in which multiple users attempting to access a single frequency channel are appropriated these timeslots. In this way TDMA is often employed in FDD networks.

1.3 In-Band Full-Duplex Communications

As previously discussed, both frequency division and time division technologies are constrained by either partial bandwidth or time allocation. Hoping to be the next solution to a constrained spectrum that's exacerbated by increased bandwidth demands, In-Band Full-Duplex (IBFD) technology aims to establish simultaneous same-band shared-medium communication.

1.3.1 IBFD Potential Benefits

If achieved, this melding of the strengths of FDD and TDD would present the tremendous opportunity for a theoretical doubling of spectral efficiency. By extension, this would also allow a doubling of channel capacities and physical layer (PHY) throughput gains in single-hop wireless nodes. Several other benefits in IBFD networks have been posited including possible reduced outage probability, reduction of end-to-end delay by allowing partial packet transmission compared to conventional store-and-forward mechanisms, and improvement in detection quality in cognitive radio environments by simultaneous secondary user transmission and scanning [5-7]. Furthermore, it has also been indicated that additional IBFD specific security benefits may be realizable [8,9].

Beyond just these PHY layer potential benefits, additional advantages have been suggested existing within media access control (MAC) layer protocols. With prudent and meticulous FD MAC design it has been suggested that the common pitfalls inherent in HD networks may be able to be alleviated. Specifically, these include enhanced collision avoidance by requiring only one and not both nodes to determine channel state, reducing congestion through scheduling exploits, and mitigating the hidden terminal phenomenon [10]. The hidden terminal phenomenon is known to occur in shared-channel wireless networks when one node begins transmission to another, and a third node, unbeknownst to and not within detectable range of the transmitting one, attempts transmission due to the fallacious sensing of an unoccupied channel. With the assumption that in IBFD networks any node transmitting to another would simultaneously have that node transmitting back to it, albeit even if only with acknowledgements (i.e., ACKs), any node out of range of the former would be able to sense the latter and infer occupation of the channel [7]. However, if the possibility of traffic asymmetry with minimal or no ACK incorporation is considered, then the

hidden node phenomenon persists. One specific novel proposed IBFD MAC protocol that attempts to address traffic asymmetry, and the susceptible inefficiency inherent to it, incorporates a traffic symmetry ratio and transmission threshold precursor [11]. By doing so, it is found that the IBFD throughput can be augmented by an additional 20% when compared to a simple-MAC IBFD network. Though the aforementioned benefits seem to justify and continually galvanize research efforts, it should be noted that the realization of these benefits is highly contingent upon successful amelioration of any IBFD deployment frustrations.

1.3.2 IBFD Challenges and Obstacles

Though the viability of IBFD networks has already been demonstrated in numerous small-scale prototyped systems [12-14], outstanding impediments still warrant further consideration. Some of these prominent obstacles include the design of efficient FD-aware MAC protocols, the addressing of an expected increase in power consumption with consideration to existing mobile communication standards, and effective and adaptive methods of self-interference cancellation [15].

A conducive IBFD MAC protocol requires fastidious care and should, at minimum, consider the following five factors. (i) Backward compatibility with existing HD networks. The reason being that as immediate conformity to IBFD technology from both the user's and the provider's end is unrealistic, any newly proposed protocols should foremost ensure uninterrupted service for existing networks. This is particularly important for IEEE 802.11 networks. (ii) Asymmetric traffic underutilization mitigation. As is witnessed currently in Wi-Fi networks, IBFD systems should expect uneven downstream (or downlink) and upstream (or uplink) traffic loads. In order to maximize overall network efficiency, a means of handling said asymmetry must be devised. (iii) Limitations and performance analysis in proportion to network capacity. In particular, the flexible and dynamic aggregation of nodes to a network should be considered with the hidden node phenomenon at both low and high data rates, and both with and without RTS/CTS-like initiation schemes. (iv) Minimization of energy consumption. As already a prominent confining factor in mobile networks, and with an anticipated increase in power consumption in IBFD nodes, extensive consideration should be given to the pivotal energy-efficient design of proposed protocols. (v)Equality with handling HD and FD nodal traffic. As previously stated, IBFD adoption is likely

to be a progressive and tiered process. Accordingly, to prevent utilization bias, resource dissemination should be indifferent in regard to the legacy or IBFD nature of the node [11,16].

Finally, considered to be the most barring element to IBFD actualization and the prominent challenge investigated in this paper, is the inherent self-interfering environment present in these transceivers. Self-interference (SI) refers to the PHY layer phenomenon in which a transceiver attempting to transmit, will interfere with reception of the desired signal-of-interest (SOI) received on the same band. This interfering signal "leaks" into the reception path, often at a power level tens or even hundreds of times that of the SOI's magnitude. As a result, this SI leads to amplifier saturation and desensitization of the ADC's dynamic range. It has been estimated that self-interference cancellation (SIC) for IBFD transceivers may need to be on the order of 110dB or greater to effectively suppress SI to that of the receiver noise floor [12]. Obtaining such austere levels requires that the SIC be incorporated through several different mechanisms of cancellation. These various techniques often span several different domains of cancellation, including passive suppression, active analog cancellation, and digital cancellation. Of these, active analog cancellation may be considered the primary mode of cancellation. The reason being that the highest levels of cancellation are often achieved here and prove to be efficacious on both the linear and nonlinear constituents. Furthermore, the effectiveness of the subsequent digital domain's cancellation has been shown to be directly dependent on overall analog cancellation performance [17].

1.4 Contributions

This thesis dissects and elucidates a model on a prototypical transceiver devised for the implementation in IBFD networks. It attempts to evaluate and verify the efficacy of an adaptive active analog self-interference canceller in the presence of a nonlinear noise-ridden channel. The overall transceiver and SI model are developed and analyzed programmatically in MATLAB. The SIC hardware architecture focuses on implementing a parallel-branched analog canceller with variable amplitude and phase components that samples the transmission signal post noise inclusion. More specifically, this thesis attempts to make these contributions:

- Component level design of a Quadrature Amplitude Modulation IBFD transceiver.

- Here, the basic components are outlined and proposed in an approach to be able to feasibly render a bit stream into a higher-order phase and amplitude dependent RF signal. Corresponding demodulation hardware is also devised with an included means of analog self-interference cancellation.

- Evaluate SIC efficacy in a stochastic system with various linear and nonlinear noise sources.

  - Explicitly, a multitude of the most egregious noise sources are incorporated via probability distribution functions in the transceiver model. These include, but are not limited to, oscillator phase noise, RF mixer harmonics and intermodulation distortion products, amplifier harmonics, Nyquist-Johnson thermal agitation, and transient power supply perturbations. Additionally, the input binary sequence is generated to exhibit mathematically stochastic properties. In the presence of the aforementioned distortions, analog SIC is employed in an approach to intrinsically include all the induced non-idealities. The level of achieved SIC is evaluated for dependency with the level of distortions.

- Evaluate the algorithmic ability of stochastic gradient descent and mini-batch gradient descent in the determination of the canceller's amplitude and phase weights.

  - Both the convergence time and the overall level of SIC achieved are assessed for stochastic gradient descent and three different batch size mini-batch gradient descent algorithms.

- Assess SIC impairment when nonlinear distortion levels of cancellation hardware are incorporated.

  - With this, the overall canceller's affective ability on SI is evaluated when the noisy transceiver model is expanded to incorporate waxing levels of nonlinear distortion in the cancellation hardware itself.

2. HISTORICAL OVERVIEW OF IBFD

2.1 Early Beginnings With RADAR

Though often heralded as a novel emerging technology, wireless systems demonstrating simultaneous same-band communication have been explored and realized in Radio Detection And Ranging (RADAR) applications since at least the 1940s [18–23]. Traditionally, RADAR systems have been classified into one of two categories; Pulsed-Wave (PW) or Continuous-Wave (CW) [24]. As the nomenclature implies, CW systems operate with a continual transmission and reception period which in turn offers the advantage of more proximal object detection, albeit at the cost of resolution and range [20,21]. However, one of the primary challenges faced by these systems is the intrinsic susceptibility to unintentional transmitter interference due to the underpinning requirement of continual reception. This transmitter-leakage, if not properly ameliorated, leads to receiver saturation, and thereby, effectively obscures proper detection of a conceivable object's reflection [23].

Mitigation strategies for this transmitter-leakage in these early systems often depended on whether the system architecture was bistatic, with separate transmission and receiver antennae, or monostatic, where transmission and reception shared a common antenna. One of the most prominent means of transmission-to-receiver isolation in these early bistatic configurations was achieved by antenna separation which exploited natural shielding and afforded sufficient enough path-loss attenuation [10,18,25]. Monostatic configurations, on the other hand, relied on isolation provided by means of a circulator. These circulators relied on the nonlinear propagation characteristics of magnetic material to render incoming-to-outgoing signal isolation [10,18,25]. Though, at best these circulators often only provided around 30 dB of possible isolation [26]. As many of RADAR applications transmitted in the range of watts of power, while attempting detection of microwatt signals, this isolation alone often proved insufficient of the 50dB requirement set in early CW systems [27]. Consequently, bistatic configurations were often the choice of design of most early systems. Unfortunately, in many cases an overall want for effective isolation resulted in the limitation of transmission power, and by effect, limitation of detection range.

2.2 1960's: Beyond Passive Isolation

In the 1960's a new method of minimizing transmitter-leakage was discovered. Using Feed-Through Nulling, it was asserted that the feed-through, or transmitter-to-receiver leakage, could effectively be eliminated through a syphoned portion of the outgoing transmission signal. This sampled or tapped signal could then be passed through a vector modulator allowing for both amplitude and phase manipulation. Which this manipulation an antiphase equal magnitude "mimicry" of the offending leakage signal could be rendered. From here, this out-of-phase imitation could algebraically be summed with the actual leakage in the receiver to mitigate interference [25,28]. However, the efficacy of this amelioration proved highly contingent on the accuracy of replication. Early attempts with this method were shown to produce narrowband levels of cancellation up to 60dB in monostatic configurations. The first and most significant improvement in leakage mitigation since implementation of the circulator. Unfortunately, limitations in component technology available at the time made this technique prohibitively costly with 60+ kg of precisely machined ferrite rotators required. These early ferrite modulators also often suffered from significant bandwidth limitations and overall lethargic response times [18,27,28]. These limitations would stagnate this approach till advances in microwave component technology and signal processing in the early 1990's made implementation more viable.

2.3 1990's: Radio & Cellular Introduction

Up until the early 1990's PW based systems largely dominated RADAR applications. However, with advancements in microwave component technology and signal processing, a new method emerged of implementing Feed-Through Nulling's underpinning concepts. In the early 90's a method was presented using a Reflected Power Canceller (RPC) which incorporated PIN diode technology in the double balanced mixer of its Vector Modulator. This new tact proved to provide a simple, cheap, and easily produced cancellation circuit. Additionally, larger bandwidths were achieved and were shown to be more conducive to adaptation to changing ambient conditions and aging antennae [27]. Similar to Feed-Through Nulling, the RPC operated upon a small portion of the transmitted signal being extracted after the power amplification in the Tx chain. This diverted signal then entered a double balanced mixer with the received RF signal and was effectively bifurcated into two resulting channel outputs. These outputs ideally possessed a frequency that

would be low and approach zero as the transmitted and received signal approached equivalent frequencies. These rendered low frequency signals were then passed through low-frequency amplification stage, a subsequent phase trim section rendering correct drive voltages, and a low-pass filter for high frequency noise rejection. Finally, the tapped transmitted signal was mixed with the resulting two in an analogue bi-phase modulator before being summed together with the received signal. Using this closed-loop analog cancellation technique, the RPC demonstrated the ability to cancel transmitter-leakage by more than 33dB with a bandwidth of over 400MHz.

In the mid-1980's to mid-90's wireless radio and cellular networks began to proliferate. However, inimical environments unconducive to EM propagation (e.g., tunnels, mountainous terrain, commercial buildings, etc.) hindered communication in these emerging networks. This frustration served as an impetus to find a means to assuage this signal degradation. Relays, devices that receive, amplify, and retransmit signals, proved to be this QoS amelioration. However, the relaying nature of these devices necessitated being able to transmit and receive signals withing the same frequency band without being overwhelmed by the aforementioned transmitter-leakage phenomenon. Again, this leakage potentially led to receiver saturation and even spurious retransmissions via self-feedback. Akin to the early techniques of RADAR, these early relays achieved Tx-to-Rx isolation primarily via antenna separation and the line-of-sight path loss [29, 30]. The overt drawback of this technique was confinement to implementation in large areas that lent themselves high amounts of natural shielding. It therefore became obvious that in order for this technology to progress more efficacious means of leakage cancellation was necessary.

First formally stated in 1999, IBFD was presented as a technology that would offer the benefit of a theoretical doubling of communication channel capacity. This, of course, assumed the transmitter leakage, now referred to as self-interference (SI), could be successfully abated. During this time, theory on an effective means of Self-Interference Cancellation (SIC) was formalized. It was stated that SIC was achievable via subsuming a deconstructive interference of an antiphase but equal magnitude component into the receiver chain. However, the simplicity of this theoretical assertions did not translate into a simplicity of realization. Accordingly, SIC soon saw a demand for a burgeoning of intensive research efforts. These efforts would eventually manifest into classifications of passive suppression, active analog cancellation, and digital cancellation.

Furthermore, as requirements were established it was determined that sufficient cancellation would entail schemes spanning all three domains.

One of the first published attempts at incorporating multiple mitigation domains achieved 72 dB of cancellation for a 200KHz channel with a carrier frequency of 1.8 GHz [31]. To accomplish this, it utilized three different isolation techniques including dual antennae separation, an Rx pre-LNA RF echo canceller, and a baseband digital filter implementing a DSP adaptive algorithm. Through this the plausibility of the hypothesized multi-domain efficacy was demonstrated.

2.4 2000's: First Modern System Demonstrations

Two of the prominent IBFD wireless system implementations that would serve as paradigms for modern attempts, stemmed from Stanford and Rice University in 2010. In the former, a system was designed and tested utilizing off-the-shelf IEEE 802.15.4 hardware with software defined radios to implement IBFD communication with published results reporting SIC of ≥ 60dB and throughput gain of 84% that of comparative half-duplex operation [32]. To achieve this, active analog cancellation was accomplished via a Renesas QHx220 Interference Canceller IC and provided ~20dB of isolation. For digital cancellation, a technique typically reserved to extract packet information post-collision was used called coherent detection. This allowed for isolation of the conflated SI signal. Additionally, a method, dubbed antenna cancellation, was employed in which dual antennae were spaced a distance of d and $d+\lambda/2$ from the receiver antenna. This placement utilized the half wavelength shift between identical waveforms to deconstructively interact and create a null region around the Rx antenna. Overall, the authors concluded that though the methodology proved effective given the system constraints, the rendered isolation was limited to 80dB and was likely insufficient. Additionally, they posited that the antenna cancellation employed would only be effective for narrowband signals.

The system from Rice University utilized off-the-shelf WARPLab 2.4GHz radios to design an IBFD system that again, yielded ~80dB of Tx-to-Rx isolation [33]. Passive suppression was achieved through the use of an orthodox bistatic antenna configuration. Active analog cancellation was achieved via a nondescriptive deconstructive addition in the Rx chain. However, digital

cancellation was handled via offline processing in MATLAB. Nevertheless, utilization measurement results and calculations presented showed significant *Achievable Sum Rate* improvement in the IBFD scenario compared to its HD rendered counterpart.

Another one of the most influential and oft cited modern studies on IBFD networks was published by researchers at Stanford University in 2013 [12]. In this study the authors presented the first functional IEEE 802.11ac compliant full-duplex system that utilized a hybrid cancellation scheme to provide 110 dB of SIC in a noisy indoor environment. Salient in this study, the authors contend that given a 20 dBm transmitter coupled with a -90 dBm receiver noise floor, requirements of their IBFD system would necessitate at least 110dB of linear self-interference cancellation of the main signal. Furthermore, 80 dB of non-linear component self-interference cancellation and 50 dB of transmitter noise cancellation had to be accomplished. Their hardware actualization in the analog domain consisted of a tapped transmission signal fed to a with fixed delay line and variable attenuator canceller. Their digital approach leveraged the foreknowledge of a message's preamble to develop a linear and nonlinear model, and as noted was not unique to their implementation. Rather, the novel elements presented included their problem interpretation and tuning algorithms employed. With the analog domain they posited, and successfully demonstrated, that by straddling the actual delay experienced by the SI signal the sinc interpolation function could be employed and transform attenuation weight estimation into a frequency domain problem of sampling and interpolation. Bolstering this, they accomplished cancellation in the digital domain via modeling the anticipated SI as a non-causal function where foreknowledge of the entire preamble packet allowed the use of samples from future instances to converge on SI in the present. Finally, they manage additional nonlinear cancellation in the digital domain by modelling expected nonlinear components as a simple Taylor Series expansion. To minimize the number of unknown variables needed solving they empirically determined and omitted variables found ineffective. Overall, by implementing the above-mentioned techniques they are able to show necessary cancellation of ≥110 dB at a throughput gain of 1.87x that of HD operation using a SMBV 100A signal generator as a transmitter and a Rohde & Schwarz spectrum analyzer as a stand in for a receiver.

2.5 Current Ventures

IBFD technology over the recent years has witnessed even further proliferation in efforts towards its realization. In the digital domain, extensive efforts have been made to create better and more effective memory inclusive nonlinear models including the parallel Hammerstein model, the Wiener-Hammerstein model, and other Volterra derivative models [34]. Other techniques have also been explored in this domain and include both transmit and receiver beamforming. The latter of which has shown particular interest in Multiple-Input Multiple-Output (MIMO) applications where SI may be suppressed by adaptively adjusting the individual antenna weight in regard to the current SI channel's condition [10]. Beyond just SIC techniques in the PHY layer, efficient and effective implementation at the MAC layer has also been evaluated. Here the aim has typically been divided between developing IBFD conducive protocols that attempt to minimize power consumption, maintain legacy compatibility, and address inefficiencies brought on by anticipated traffic asymmetry [16,35].

3. DOMAINS OF SELF-INTERFERENCE CANCELLATION

As the primary aim of all IBFD systems is to accomplish simultaneous same frequency band communication, transceiver nodes of this type inherently are afflicted by SI in their reception chain. To mitigate this and make communication viable various means of SIC must be integrated into transceiver design. Furthermore, suppressing SI levels to that of or greater than the receiver noise floor often relies on a combination of techniques working in tandem. These techniques are often designated with nomenclature that alludes to their domain of operation and include passive suppression, active analog cancellation, and digital cancellation.

3.1.1 Passive Suppression

The passive suppressive domain, also referred to as the propagation domain, targets the attenuation of the amount of SI actually received by the Rx path. In other words, it attempts to "suppress" the amount of offending signal coupled from the Tx to Rx channel. As one of the first methods of cancellation used, it often focuses on increasing the path loss from Tx to Rx antenna through both physical antenna separation and shielding isolation as shown in Figure 4 [13,36]. Using this technique levels of up to 50 dB of suppression have been achieved with as little as 40 cm of separation distance [12].

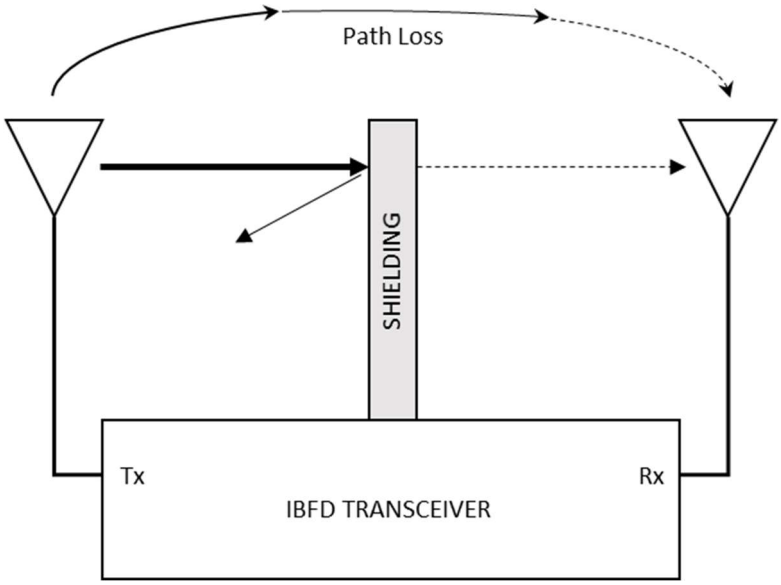

Figure 4 Traditional SI Suppression Techniques

Additional isolation techniques include incorporation of circulators, duplexers, and TR switches which allow bidirectional communication over a single antenna but limit Tx-Rx coupling by exploiting the electromagnetic properties of materials used. More advanced techniques are also often incorporated, especially in MIMO systems that focus on cross polarization and beamforming schemes [37-39].

Overall, this approach of cancellation benefits from its ability to prevent SI intrusion in the first place and reduce linear, nonlinear, and random noise products. However, it suffers from susceptibility to inadvertently reducing transmission power levels, bandwidth constraints in directionality schemes, and difficulty in implementation for small form factor mobile nodes with antenna separation [10].

3.1.2 Analog Cancellation

Bolstering the methods of passive suppression, analog cancellation aims to further reduce the power of SI in the Rx path by emulating an anti-phase imitation of the offending SI signal. This SI replica is then deconstructively added into the receiver chain, before analog-to-digital conversion, and ideally cancels any SI not capable of being suppressed. Two typical approaches exist for this sort of cancellation. These include either utilization of an auxiliary Tx chain or tapping of the transmitted signal for generation of an SIC signal.

In the case of tapping the transmitted signal for the creation of a cancellation signal, a balun (i.e., specialized transformer) typically syphons a small portion of the Tx signal either directly before transmission, for cancellation in the RF domain, or after initial digital-to-analog conversion, for cancellation in the baseband. In the former, as shown in Figure 5, tapping directly prior to transmission allows the sampled signal, and thereby the cancellation signal, to include any noise and nonlinear products introduced in the Tx chain.

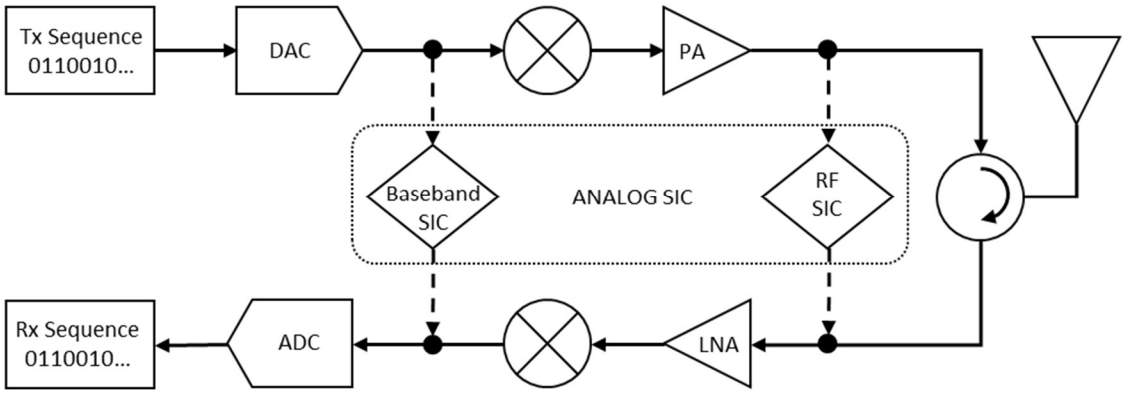

Figure 5 Tapped Locations in Tx Chain

After tapping, the copy is passed into a parallel branch circuitry for SI estimation. Operation of these branches can be viewed in either a time domain centric or frequency domain centric manner. In the time domain, each branch is viewed as having an adjustable attenuation and variable phase and/or delay component. In the frequency domain, each branch is constructed to a have a narrowband filter with variable center frequency and quality factor, as well as an adjustable overall phase response. In either approach, the complex weighting for all elements is attempted to be determined such that their outputs' summation approximates that of the received SI; only 180° out of phase. A depiction of a time domain variable phase and attenuation circuit is shown in the block diagram of Figure 6. Again, this approach of analog cancellation benefits from the fact that the actual nonlinear and noise products introduced in the Tx chain are sampled and included in the SIC process, and thereby, may be superior at eliminating this nonidealities. Levels of cancellation in this domain have been demonstrated achieving up to 45dB to 60dB of SIC [7,12]. However, this approach comes with the caveat that for each antenna included in the system, an SI cancellation circuit of *N* parallel branches is needed [7]. Therefore, for MIMO applications circuit complexity and space requirements may be drastically increased.

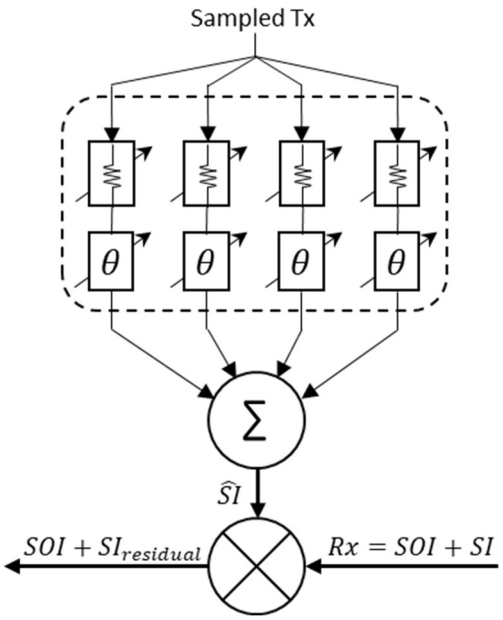

Figure 6 Variable Phase & Attenuation Analog Cancellation Block Diagram

For cancellation using an auxiliary Tx chain, the burden of SI estimation is transferred to the digital domain. In this approach, delaying and weighting of the Tx signal is accomplished digitally before being passed to an auxiliary network for D/A conversion, RF mixing, amplification, etc., and eventual deconstructive introduction in the Rx path. In this way, the number of effective taps can be significantly increased. Allowing for greater control and higher flexibility of the canceller, but at the expense that nonlinear cancellation is often more arduous and ineffective. Additionally, since transmitter noise is, by description, random, this method cannot mitigate this component [37].

3.1.3 Digital Cancellation

One of the latest to be incorporated and final means of SI mitigation is that of digital cancellation. It attempts to remove any remaining, residual SI via channel modeling from the received signal after A/D conversion. This method relies heavily on exploiting the fact that the transceiver has foreknowledge of the intended transmitted signal, and accordingly, can be used to model and generate digital samples to subsequently cancel the SI channel in the digital baseband. However, since this foreknowledge is ideal, it includes no information on imposed distortions and nonlinearities. Therefore, it is inherently susceptible to these system nonidealities. Nevertheless, for successful cancellation, techniques for both linear and nonlinear discrete time baseband models

must be incorporated [10]. Accordingly, digital cancellation approaches are often subdivided and referenced by these main components (i.e., Linear or Nonlinear models).

Linear SI, which constitutes the majority of power existing in the residual SI, is often treated as a noncausal linear function of the transmitted signal. Since it is known in advance, it can be modeled as a linear combination of a selected number of samples both prior to, and after the received signal. Solving for this is typically accomplished by conventional means of minimizing the mean square error utilizing algorithms including Least-Mean-Squares (LMS), Normalized-Least-Mean-Squares (NLMS), Recursive Least Squares (RLS), and Affine Projection Algorithms(APA) [40]. Typically observed with these methods, is an inherent trade-off between computational complexity and convergence time.

Non-linear modeling can be more involved, and less successful, process for the digital domain estimates to conform to. A general approach is to simply model nonlinearities as a polynomial variant, e.g., a low order Taylor series expansion. Still, this approach can be mathematically cumbersome and ineffective [7]. The most recent attempts to improve digital channel modeling and performance have looked at the application of machine learning methods and techniques [41]. In one such case, the application of complex-valued and recurrent neural networks was compared to that of a traditional polynomial model and found that for a comparable 44.5dB of cancellation, they required 33.7% and 26.9% fewer floating-point operations and parameters, respectively [42].

4. NOISE ANALYSIS AND MODELING

Presented in this section is foundational knowledge that is germane to understanding the salient concepts of IBFD transceivers. Consummate understanding of the information in this chapter was utilized in the design and analysis of the architecture presented in this thesis. This chapter begins by defining what is to be considered noise on a signal and explores this concept from a stochastic and probabilistic standpoint. From there it delves into some noise sources found in common transceiver hardware, parameters typically used to characterize them, and the relevance thereof. Next, the type of modulation used in this work's simulation is presented. Finally, an overview of the two variants of gradient descent algorithms evaluated is presented.

4.1 Noisy Signals

A noise free system is one that is to be considered purely deterministic. Stated in other terms, given a set of *n* trial simulations with identical initial conditions, then *n* identical results would be yielded. Alternatively, a noisy system is one regarded as composed of both deterministic and stochastic processes. In these systems, when presented with identical parameters, the random cyclostationary phenomena results in varying trial outcomes composed of both its deterministic and stochastic components. Letting $v(t)$ represent the deterministic noise-free portion and $n(t)$ represent the noisy stochastic component, the composite signal $v_n(t)$ can be expressed as:

$$v_n(t) = v(t) + n(t)$$

Given an ensemble of *N* outcomes from the trials, we can define the expectation, or mean μ, of this signal as:

$$E\{v_n(t)\} = \mu = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} v_n(t)$$

However, upon mathematical evaluation of this expectation with *n(t)* assuming gaussian white noise characteristics, we find that the noise component converges to 0 as *N* approaches infinity and the overall expectation converges to the noise free signal $v(t)$ [43,44]. Consequently, to develop a more illuminating picture of a noisy signal the inclusion of three more germane metrics

is often included. These are variance,σ^2, standard deviation, σ, and the autocorrelation, $R_n(t,\tau)$. Calculated as follows:

$$var\big(v_n(t)\big) = \sigma^2 = E\{(v_n(t) - \mu)^2\}$$
$$std\ dev\big(v_n(t)\big) = \sigma = \sqrt{E\{(v_n(t) - \mu)^2\}}$$
$$R_n(t,\tau) = E\{n(t)n(t-\tau)\}$$

Conceptually, the expectation quantifies the measure of the center of a random variable and variance gives an impression of the spread about this center. When combined, this allows us to describe a noisy process more accurately by developing its Probability Distribution Function (PDF). With this, we are enabled to depict the stochastic perturbations of a signal and system more realistically. Additionally, the calculation of the autocorrelation allows us to determine the correlation, or lack thereof, between the power of two instances separated by τ seconds. Still, many current SPICE noise analysis techniques do not accurately capture valid noise characteristics in many RF devices such as oscillators, mixers, and data converters [45]. To accurately calculate a noise response whose power may vary significantly with time requires incorporation of newer simulation algorithmic approaches.

4.2 Probability Distribution Functions

One specific distribution worth mentioning due the shear ubiquity of observance in nature is the Normal, or Gaussian, Distribution. In electronics processes that follow a normal distribution include noise sources such as Johnson-Nyquist Noise, shot noise, and white noise. Elsewhere this distribution is also observed in black body radiation, the random motion of particles, population intelligent quotients, and many processes in the manufacturing and quality assurance. The general form of the normal distribution's PDF, *f(x),* is expressed in the relationship:

$$f(x) = \frac{e^{\frac{-(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$$

Where *x* represents the random variable, μ the mean, and σ the standard deviation. One particular permutation of importance is found when setting $\mu = 0$ and $\sigma = 1$, and is known as the standard normal distribution. Upon examination of this distribution, we see that approximately 68.2% of outcomes are accounted for within one standard deviation from zero, 95.4% within two, and 99.7%

within three. For random variables following the normal distribution, this relationship is true regardless of the mean.

One further distribution worth mentioning due to its equality characteristics, is the uniform distribution function. Often employed in random sampling or hypotheses evaluation, uniform distribution assigns all possible outcomes an equal likelihood of occurrence. Particular to electronics, uncorrelated or low correlation processes, such as the quantization error in analog-to-digital converters often display uniform distributions.

4.3 Determining a Stochastic Signal: Pseudo-Random Binary Sequence

In order to simulate real-world worst-case stress that may be presented on high-speed serial digital interfaces, a pseudo-random bit sequence (PRBS) is often employed in transceiver evaluation [46]. Though algorithmically generated in a very deterministic manner, PRBS patterns generate a bit sequence that statistically appears stochastic [47,48]. Hardware realization of these patterns is often accomplished via linear feedback shift registers (LFSR) in which specific registers are tapped, XOR'ed together, and subsequently fed back into the input. This is shown below in figure.

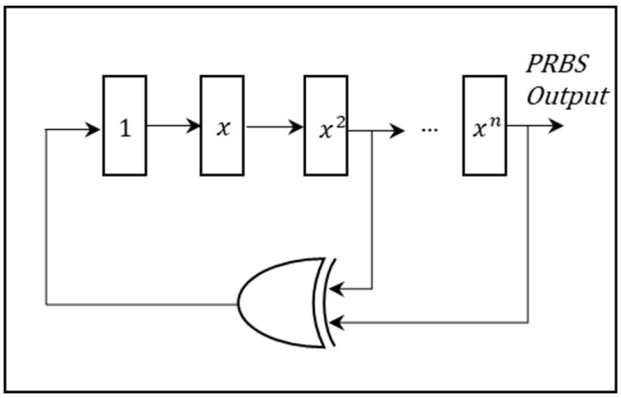

Figure 7 PRBS Generating LFSR

The length of the pattern capable of being generated is directly proportional to the number of registers (i.e., number of bits) used. A common sequence employed in testing is a PRBS31 pattern. Capable of generating $2^{31} - 1$ bits without repetition, this pattern is often denoted by monic polynomial $x^{31} + x^{28} + 1$, where the power of the variable represents the tapped register fed into the XOR gate and '+1' operator represents bit shifting its output back into the LSFR [49].

Useful for analytical calculations, the spectral density of a PRBS sequence displays a special pattern [50]. In particular, the power spectrum of a PRBS sequence of length M can be calculated as:

$$\Phi_u(\omega) = \frac{2\pi\mu^{-2}}{M}\sum_{k=1}^{M-1}\delta\left(\omega - \frac{2\pi k}{M}\right) \quad 0 \leq \omega \leq 2\pi$$

where we let $u(t)$ be representative of the input. If plotted one finds that this power spectrum conforms to a $sinc^2(f)$ envelope with "gullies" appearing in the spectrum at every integer multiple of transmission data rate [51].

4.4 Johnson-Nyquist Noise

Independent of both conductor composition and functionality of electronic circuitry, Johnson-Nyquist noise is an intrinsic noise present in all materials caused by the thermal agitation of charge carriers [52]. For this reason, it is colloquially often referred to as thermal noise. These thermal perturbations in turn lead to minute fluctuations in charge density and current density, and by extension variations in the electric and magnetic fields, which presents itself as a flat bandlimited white noise [53]. Barring low temperature conditions approaching absolute zero or terahertz frequencies, the noise power of this type of noise can be calculated as $N = k_B TB$, where N is the noise power in J/sec (i.e., watts), k_B is the Boltzmann constant of $1.380649 \times 10^{-23} JK^{-1}$, T is the temperature in Kelvin, and B is the bandwidth in Hz [54]. From this, the voltage present in a system of nominal resistance, R_o, due to the thermal agitation can be calculated as

$$v_{thermal} = \sqrt{4k_B TBR_o}$$

4.5 Oscillator Phase Noise

For oscillation to occur with electronic circuits, at a bare minimum, the conditions must be satisfied that are enumerated in the Barkhausen Stability Criterion, which include:

1. Positive feedback must be supplied with unity gain at the frequency of oscillation.
2. The phase shift around this feedback loop must be 0 or an integer multiple of 2π.

In its most reductive conceptual form, an electronic oscillator is one that converts DC and always present circuit white noise into a stable oscillating RF waveform. Ideally, this output oscillation is a single frequency sinusoidal waveform described by the equation:

$$v_{osc}(t) = a_{osc}\cos(2\pi f_o t)$$

where f_o is the frequency of oscillation, a_{osc} is the amplitude, and t is a relative time variable. However, in practice non-linear devices exhibit a spectral density composed of an aggregate of noise sources that result in minute phase perturbations occurring at a range of offset frequencies. If we allow $\vartheta(t)$ to represent this time varying phase perturbations and $a(t)$ to represent time varying amplitude variations, then our oscillation output becomes:

$$v_{osc}(t) = a(t)\cos(2\pi f_o t + \vartheta(t))$$

Classically, phase noise was typically characterized on a spectrum analyzer by measurement of the power, at some oscillation offset frequency, in a 1 Hz window with the power at the ideal oscillation frequency subsequently subtracted [57,58]. However, this method of characterizing phase noise has two significant caveats. First, the accuracy of the measurement is directly proportional to the stability of the spectrum analyzer's internal oscillator, and second, this measurement includes both the amplitude $a(t)$ and phase $\vartheta(t)$ perturbations. As a result, the National Institute of Standards and Technology (NIST) and IEEE have adopted a more formal definition of phase noise, $\mathcal{L}(f)$, as being half of the power spectral density (PSD) of phase fluctuations $S_\vartheta(f)$ (i.e., the single sideband PSD) [59]. Which can be expressed as

$$\mathcal{L}(f) = \frac{S_\vartheta(f)}{2}$$

Figure 8 below displays a typical single sideband phase noise profile with the five characteristic regions commonly observed. Listed from closest to furthest from nominal oscillation, these regions are often denoted as $1/f^4$, $1/f^3$, $1/f^2$, $1/f$, and noise floor, where the nomenclature alludes to the inversely proportional nature of the power verse frequency relationship. Power density in these $1/f^n$ regions display a 40dB, 30dB, 20dB, and 10dB per decade roll off rate, respectively. However, it should be noted that a given phase noise profile may only display a few of the possible regions depending on the frequency range of measurement and the egregiousness of worst offending noise within the range. For instance, though an oscillator may inherently contain noise

sources displaying a 40dB/decade roll off characteristic, this often only becomes apparent at very marginal offset frequencies (e.g., 0.1Hz). Measurement time requirements and complexity therefore often preclude accurate portrayal.

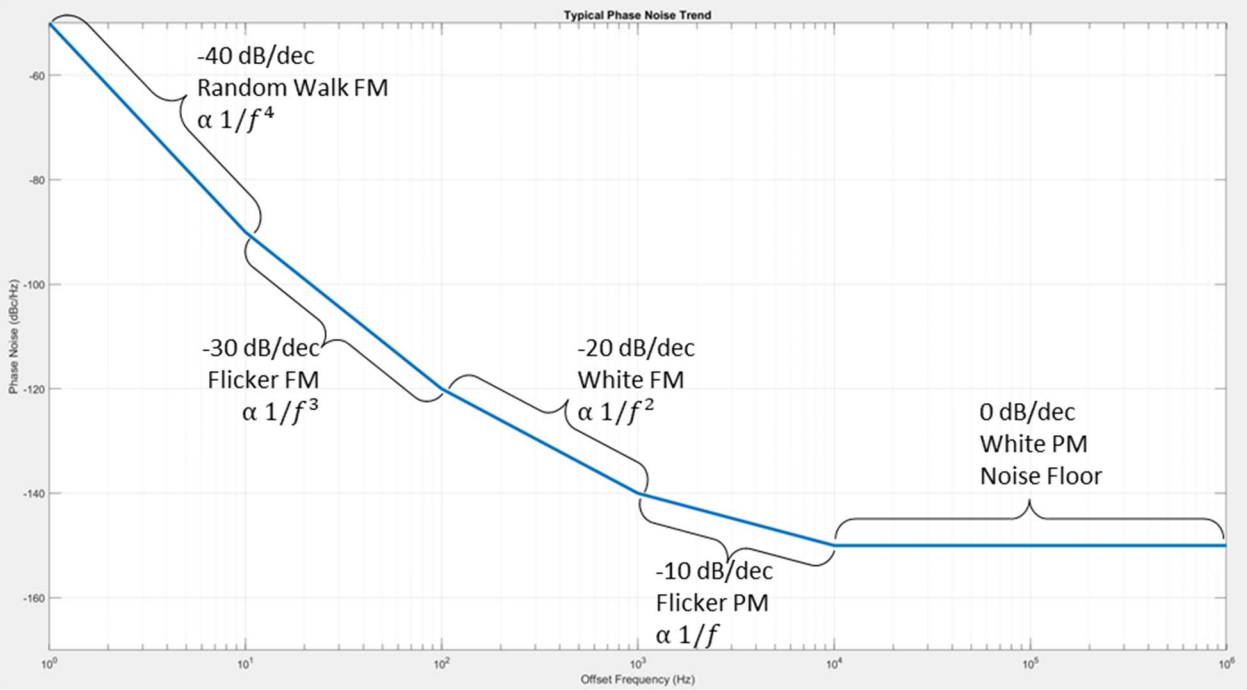

Figure 8 Phase Noise Power Spectrum Density Trends

To precisely characterize phase fluctuations, one common measurement technique outlined by NIST is depicted in Figure 9. With this technique a tunable high stability, low phase noise reference oscillator is tuned to match the oscillation frequency of the device under test (DUT). Using a very narrowband phase lock loop (PLL), these oscillators' phase difference is kept in a state of orthogonality. with each's respective output fed into a double balanced mixer. Each oscillator's respective output is subsequently fed into a double balanced mixer with the mixer's intrinsic nonlinearity behavior rendering the output the multiplicative product of the inputs. After basic trigonometric manipulation we reveal the output of the mixer a function of primarily two components. One at twice the frequency of oscillation and the other solely proportional to the DUT's phase perturbations.

$$v_{DUT}(t) \cdot v_{Ref}(t) = \cos\big(\omega_o t + \vartheta(t)\big) \cdot \sin(\omega_o t) = \frac{1}{2}\big[\sin\big(2\omega_o t + \vartheta(t)\big) - \sin\big(\vartheta(t)\big)\big]$$

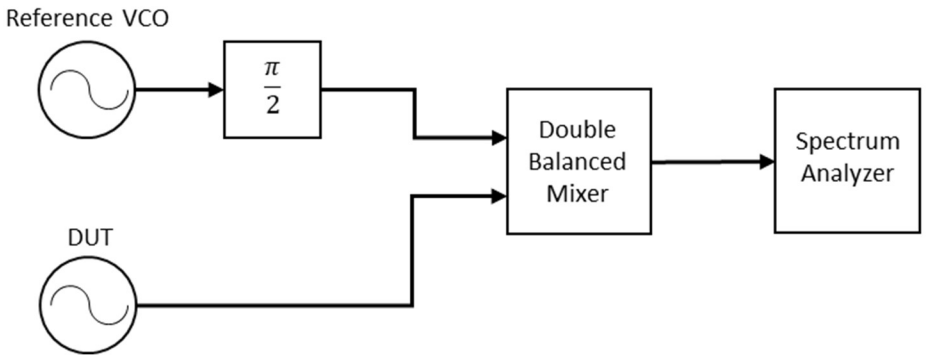

Figure 9 Phase Noise Measurement Block Diagram

The elegance of this output is that by performing simple filtration and amplification the output, solely the product of phase perturbations, can be isolated and measured. Now comprising the form of $asin(\vartheta(t))$, the spectrum analyzer is capable of effectively measuring a voltage directly proportional to the phase deviations of the DUT oscillator and portraying it in terms of offset frequency and power. When quantified with a 1 Hz measurement window and calculated as the decibel unit referenced to the nominal oscillation power, the units of the phase noise's PSD become dBc/Hz.

4.6 Nonlinear Modeling

Many different types of nonlinear models exist each with purported benefits and computational complexities. Of these models, classification is often further subdivided between memory inclusive models (e.g., Volterra, Hammerstein, Wiener, Deep Residual Neural Network, etc.) and memory naïve models (e.g., Polynomial or Taylor Series, Bessel-Fourier, logarithmic, etc.). Furthermore, these memory models can often be categorized as either Volterra derivative, as is the case of the Hammerstein and Wiener model, or neural network based [60,61]. In fact, the Volterra model itself is derivative in nature and just a generalized Taylor series expansion that subsumes a memory effect. Memory in this case, refers to the time discrepancy, or phase perturbations, between a semiconductor device's input and output over time. The simulations and cases considered in this thesis are modeled in transitory discrete-event manner and therefore, do not necessitate memory inclusion. Additionally, the performance of a static polynomial model in RF power amplifiers was compared to that of a parallel Hammerstein, Volterra, and radial basis-function neural network and found to have commensurate or better performance in terms of model

error as the more complex memory models [62]. As a result, devices in which nonlinear distortion was considered (e.g., RF mixers, amplifiers, DAC clock mixing) were chosen to be evaluate by means of Taylor series expansion in this thesis.

For a Taylor series expansion with input, $v_i(t)$, we can model the output, $v_o(t)$ as a polynomial equation of the form:

$$v_o(t) = a_o + a_1 v_i(t) + a_2 v_i^2(t) + a_3 v_i^3(t) + \cdots + a_N v_i^N(t) = \sum_{n=0}^{N} a_n v_i^n(t)$$

Here, a_n is a scalar coefficient and N denotes the degree, or order, of the polynomial. To understand the justification of using a Taylor series expansion as a nonlinear model in semiconductor devices, it is necessary to first examine its basic defining characteristic from the PN junction and corresponding boundary current relationship. Reducing the mechanisms of current flow through a junction to the two basic components of carrier drift and diffusion, then the total current density, J, becomes:

$$J = J_{p|drf} + J_{n|drf} + J_{p|dif} + J_{n|dif}$$

where $J_{p|drf}$ and $J_{n|drf}$ represent the drift current due to hole and electron drift, respectively, and $J_{p|dif}$ and $J_{n|dif}$ represent hole and electron diffusion, respectively. Substituting in carrier mobilities (μ_n & μ_p), concentrations (n & p), diffusion coefficients (D_p & D_n), and basic elementary charge (e), yields the relation

$$J = e\mu_p nE + e\mu_n nE - eD_p \nabla \mathrm{p} + eD_n \nabla n$$

Furthermore, given the conditions of uniform doping and thermal equilibrium being assumed at the boundary, after applying boundary conditions then our overall current density relation reduces to

$$J = \left(\frac{eD_p p_{n0}}{L_p} + \frac{eD_n n_{p0}}{L_n}\right)\left(e^{\frac{eV_a}{k_B T}} - 1\right) = J_s\left(e^{\frac{eV_a}{k_B T}} - 1\right)$$

where V_a is junctions applied voltage and J_s the junctions reverse saturation current. Multiplying through with the PN junctions cross sectional area results in

$$I_{PN} = I_s\left(e^{\frac{eV_a}{k_B T}} - 1\right)$$

which can be identified as the ideal current-voltage relationship of a PN junction [63]. Finally, replacing the exponential term in this equation with the Taylor Series expansion of $I_s e^{\frac{eV_a}{k_B T}}$ produces a decaying infinite series of

$$I_s + I_s \frac{eV_a}{k_B T} + I_s \frac{\left(\frac{eV_a}{k_B T}\right)^2}{2!} + I_s \frac{\left(\frac{eV_a}{k_B T}\right)^3}{3!} + I_s \frac{\left(\frac{eV_a}{k_B T}\right)^4}{4!} + I_s \frac{\left(\frac{eV_a}{k_B T}\right)^5}{5!} + \cdots$$

By juxtaposing this with the nonlinear model given for $v_o(t)$ at the beginning of this section, we can see $v_o(t)$ assumes a more generalized form that can be extrapolated to a nonlinear semiconductor device. Accuracy of the model response in assuming an estimated device depiction then becomes a matter of the order of model employed and determination of correct coefficient matrix for $\boldsymbol{a_i}$.

With the theoretical foundations of our nonlinear model established, let us examine the spectral content for two relevant input functions. For the first input function let us assume it holds the form of a simple ideal harmonic generator (i.e., $v_i(t) = A cos\omega t$). For a 5th order Taylor series, it yields

$$v_o(t) = a_o + a_1 A cos\omega t + a_2 (A cos\omega t)^2 + a_3 (A cos\omega t)^3 + a_4 (A cos\omega t)^4 + a_5 (A cos\omega t)^5$$

Expanding and simplifying this nonlinear polynomial reveals an output sequence composed of the first five harmonics of ω as well as an additional DC element. With this kind of expansion single input nonlinear devices lacking conspicuous intermodulation distortion (e.g., power and low-noise amplifiers) can effectively be modeled. Table 1 lists the corresponding coefficients summarized for the spectral components of this expansion.

Table 1 Coefficients for 5th Order Harmonic Expansion

Spectral Component	Coefficient
DC	$a_0 + \frac{A^2}{2}a_2 + \frac{3A^4}{8}a_4$
ω	$Aa_1 + \frac{3A^3}{4}a_3 + \frac{5A^4}{8}a_5$
2ω	$\frac{A^2}{2}a_2 + \frac{A^4}{4}a_4$
3ω	$\frac{A^3}{2}a_3 + \frac{5A^5}{8}a_5$
4ω	$\frac{A^4}{8}a_4$
5ω	$\frac{A^5}{16}a_5$

Another important model variant worth noting is that of an additive two-element input, passed through a nonlinear device. This variant is often used to derive the output of RF mixers or applied to situations with high likelihood of extraneous signal intrusion in a semiconductor device (e.g., clock mixing with D/A or A/D conversion). In these scenarios, the input adopts the summation of two coupled signals, i.e., $v_i(t) = A_1 cos\omega_1 t + A_2 cos\omega_2 t$, where the frequency of the two signals is represented by ω_1 and ω_2, respectively. For this example, we assume $\omega_2 \geq \omega_1$, and for illustrative simplicity, omission of any relative phase deviations. Additionally, the model has been truncated to 3rd order expansion to avoid the phenomenon of overfitting and due to the fact that intermodulation distortion (IMD) products are typically only specified up to 3rd, or seldom 4th, order products in devices such as power and low-noise amplifiers [64,65]. Accordingly, our nonlinear model becomes

$$v_o(t) = a_o + a_1(A_1 cos\omega_1 t + A_2 cos\omega_2 t) + a_2(A_1 cos\omega_1 t + A_2 cos\omega_2 t)^2 + a_3(A_1 cos\omega_1 t + A_2 cos\omega_2 t)^3$$

Again, after manipulation with applicable trigonometric identities we observe an output spectrum consisting of the first two harmonics for each frequency. However, contrasting the previous model, the nonlinear interaction of a two-element system additionally generates both 2nd and 3rd order IMD products. Table 2 summarizes the coefficients for this resultant spectrum.

Table 2 Coefficients for 3rd Order Expansion with Intermodulation Distortion

Spectral Component	Coefficient
DC	$a_0 + \frac{a_2}{2}(A_1^2 + A_2^2)$
$\boldsymbol{\omega_1}$	$a_1 A_1 + \frac{3a_3}{4}(A_1^3 + 2A_1 A_2^2)$
$\boldsymbol{2\omega_1}$	$\frac{a_2}{2} A_1^2$
$\boldsymbol{3\omega_1}$	$\frac{a_3}{4} A_1^3$
$\boldsymbol{\omega_2 - 2\omega_1}$	$\frac{3a_3}{4}(A_1^2 A_2)$
$\boldsymbol{\omega_2 - \omega_1}$	$a_2 A_1 A_2$
$\boldsymbol{\omega_2}$	$a_1 A_2 + \frac{3a_3}{4}(A_2^3 + 2A_1^2 A_2)$
$\boldsymbol{\omega_2 + \omega_1}$	$a_2 A_1 A_2$
$\boldsymbol{2\omega_2 - \omega_1}$	$\frac{3a_3}{4}(A_1 A_2^2)$
$\boldsymbol{2\omega_2}$	$\frac{a_2}{2} A_2^2$
$\boldsymbol{\omega_2 + 2\omega_1}$	$\frac{3a_3}{4}(A_1^2 A_2)$
$\boldsymbol{2\omega_2 + \omega_1}$	$\frac{3a_3}{4}(A_1 A_2^2)$
$\boldsymbol{3\omega_2}$	$\frac{a_3}{4} A_2^3$

Closer inspection of derived coefficients in the tables reveals an intriguing observation. Our original multiplicative factors ($a_1, a_2, a_3 \ldots$) of the model, when examined from a spectral viewpoint, have become interspersed and disseminated along with the input amplitudes, A_1 and A_2. The specific confluence of which determines the magnitude of each frequency component. By exploiting this relation, it then becomes possible to emulate a desired nonlinear profile by correct assignment of original coefficient values. Therefore, through assignment of different polynomial coefficients (e.g., a_1, a_2, etc.) it becomes possible for one model with equivalent amplitudes (e.g., A_1 and A_2) to be representative of a variety of RF mixing operations (i.e., multiplication, addition, harmonic generation, etc.).

4.7 Intermodulation Distortion, IP3, and Spurious Free Dynamic Range

Intermodulation distortion (IMD) is one of the most popular conventional metrics used to quantify the degree of nonlinearity of a device. As demonstrated in Section 4.6, given the input frequencies f_1 and f_2 into a nonlinear device, harmonics will result which can be represented as mf_1 and nf_2 for the m^{th} and n^{th} harmonic, respectively. Combining the sum and difference of these components results in output spectral content of the form $mf_1 \pm nf_2$. Content assuming this form is what is defined as the IMD products of the device. It should be noted that f_1 and f_2 may in fact be single tone frequencies or actually representative of frequency bands. In either case the relation is still valid. In this relation, the sum $m + n$ is referred to as the order of the IMD product. Two particular IMD products that are often explicitly called out in amplifier and mixer datasheets are the second-order intercept (IP2) and the third-order intercept points (IP3). These points are considered especially pernicious due to the fact that if f_1 and f_2 are relatively close, then $2f_1 - f_2$ and $2f_2 - f_1$ fall spectrally close to ideal amplifier outputs f_1 and f_2, and $2f_1$ and $2f_2$ are vicinal to the ideal mixer summation output $f_1 + f_2$. This relation is illustrated in Figure 10.

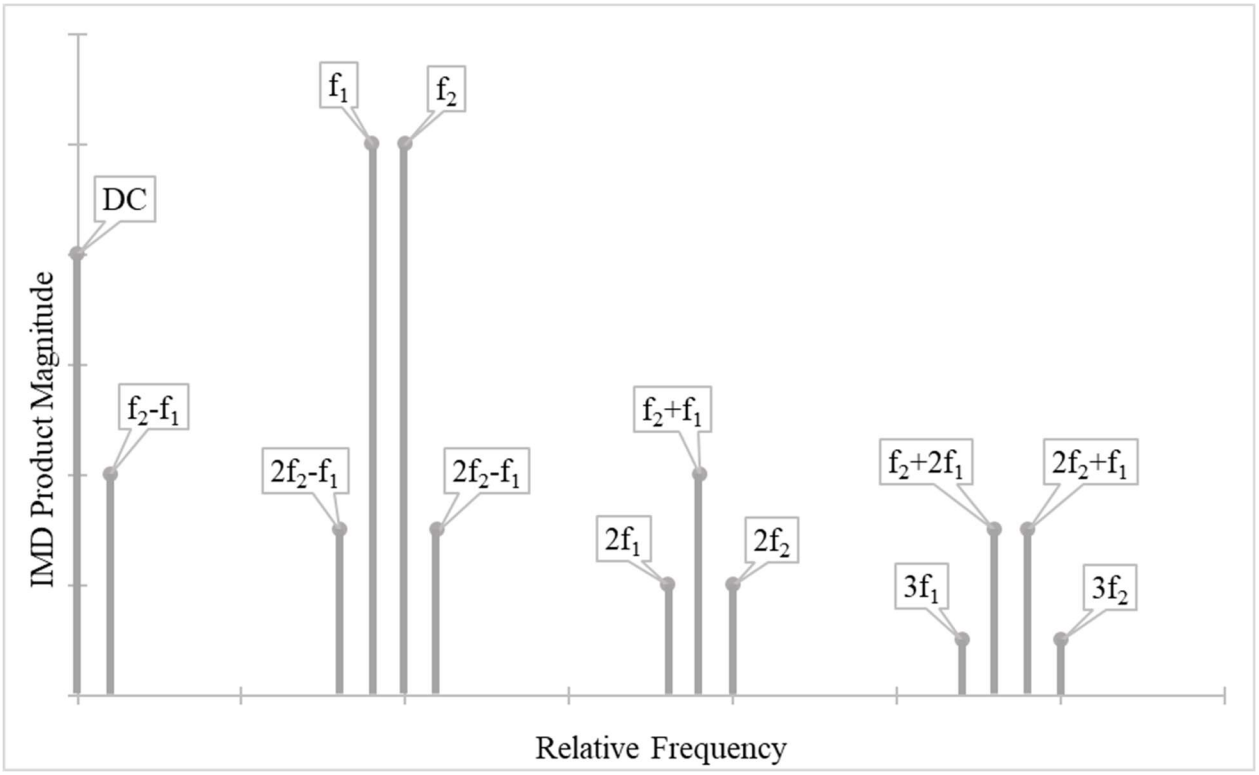

Figure 10 Relative IMD Product Relationships

If the ideal linear output were to be plotted for amplifier against its input power, an interesting relationship would reveal itself. For every 1dB increase in power on the input, a commensurate 1dB of increase would be observed on the output. In other words, the input to output power relation would assume the slope-intercept form $y = mx + b$ where b is the gain of the amplifier and m would be 1. (i.e., 1:1 power in: power out relationship). As the input to output power relations are plotted for other orders of IMD products another interesting pattern manifests. It is found that order of an IMD product corresponds to the slope of the power relationship [66]. Put another way, for a third order product a 1dB increase of power on the input would lead to a 3dB increase on the output. Using this relationship, the IP3 point reveals the theoretical point in which this third order product, with a slope of three, intersects with the ideal linear output, with a slope of one. This is further illustrated in Figure 11.

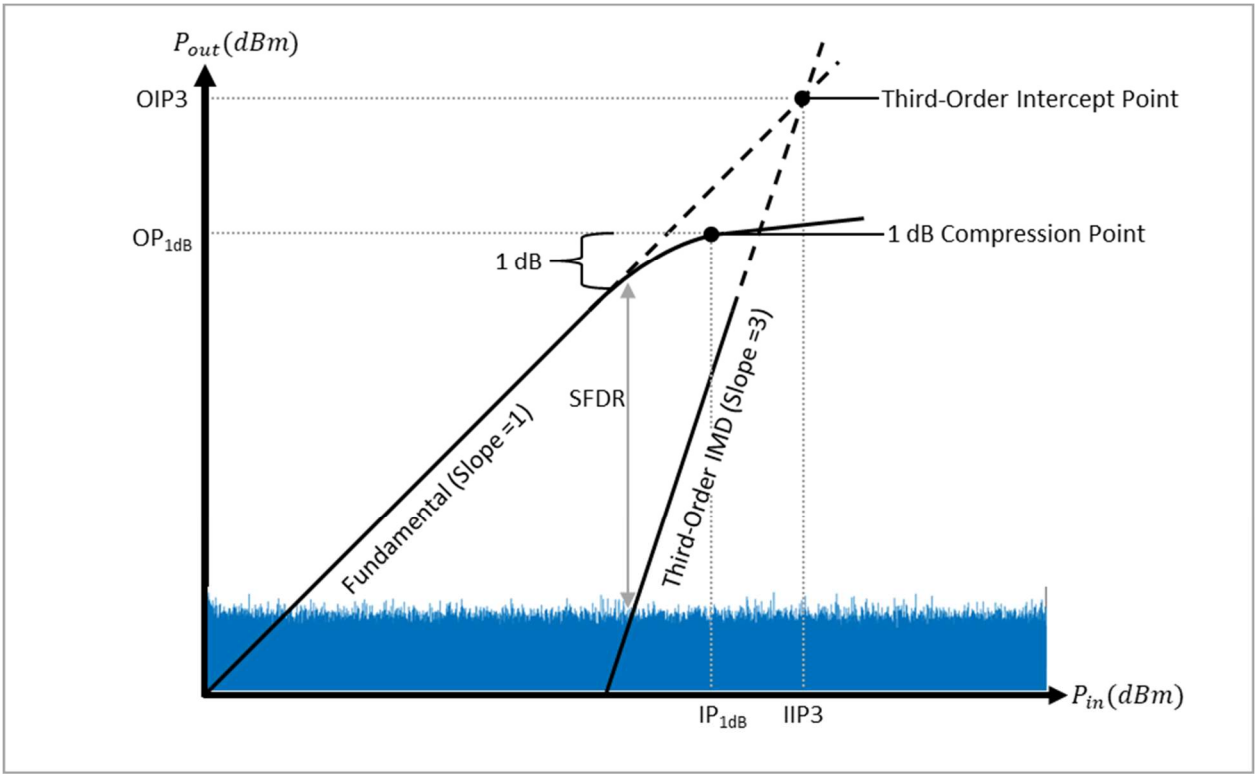

Figure 11 SFDR, Third-Order Intercept, and 1dB Compression Point

As can be seen, IP3 is the third-order intercept point for electrical power, which is a theoretical value as saturation is experienced in the device before the given output levels may be reached. Depicted as OP_{1dB} and IP_{1d} , this saturation level is often characterized by the 1dB compression point and represents an upper operability limit. Complementary, the lower end of operation is often constrained by the device's noise floor, a bandlimited white noise level. Notice that at a certain

point the third order products lie below the noise floor and are thereby nonaffective to the output. This operating range in which spurious signals are unlikely to be observed is effectively known as the spurious-free dynamic range (SFDR). Consequently, by knowing the SFDR, input or output IP3, and 1dB compression point an accurate estimate of a nonlinear device's output can be gleaned given a certain input condition.

4.8 Gradient Descent Algorithm

One of the most fundamental and often employed algorithms in machine learning is a technique known as gradient descent. Gradient descent, classified as a first-order iterative optimization algorithm, employs the partial derivatives, or gradient, of a function in a manner to minimize the error between a set of data and its estimated model [67, 68]. If the set of actual self-interference data received in the IBFD transceiver is denoted SI and the estimation model as $\widehat{SI}(\beta, \theta)$, then the error at any specific instance of time i can be defined as $SI_i - \widehat{SI}_i(\beta, \theta)$. Note β and θ represent the gain and phase used in the estimation model, respectively. Squaring this error precludes the possibility of a negatively rendered result, but more beneficially, allows it to assume the shape of a parabolic function. From here if the error is summed over every point in the data set, and the average, or mean, of this value is found, the overall efficacy of the model can be quantified in a singular value known as the mean-squared-error (MSE). The optimality condition then becomes a function of minimizing the MSE. In this thesis it corresponds to minimizing

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(SI_i - \widehat{SI}_i(\beta, \theta)\right)^2$$

From the parabolic nature of this function, it can be observed that the minimum value, where $\widehat{SI}$ most closely matches SI, would correspond to its vertex. With astute observation one notices that the slope, or gradient in higher dimensions, equates to 0 at this point. Correspondingly, as the point of evaluation diverges from the vertex its gradient increasingly grows, with sign indicative of its relative direction. In gradient descent this relationship is exploited by noting that the opposite direction, or negative gradient, will effectively point back towards the vertex. Finally, estimate parameter update operates in an iterative manner proportional to a fractional multiplier η. Mathematically for the k^{th} parameter update this can be symbolized as

$$\beta^k = \beta^{k-1} - \eta \frac{\partial \widehat{SI}(\beta, \theta)}{\partial \beta}$$

$$\theta^k = \theta^{k-1} - \eta \frac{\partial \widehat{SI}(\beta, \theta)}{\partial \theta}$$

Another piece of information that should be considered relevant is the specific kind of gradient descent implemented in model training. The two types utilized in this work are Stochastic Gradient Descent (SGD) and mini-Batch Gradient Descent (mini-BGD). Reductively, differing these two is just the portion of the data considered in parameter update. In SGD the parameters are updated on a per sample basis (i.e., gradient for one sample is calculated prior to updating). In this manner convergence appears to occur with a certain degree of randomness. Here, the benefit typically is an ease of computation with a relatively slower convergence rate. The other form, mini-BGD, equipartitions the training set into distinct batches. Then the gradient for every element in a batch is computed and concurrently used to update the parameters, with the goal of possibly faster convergence.

5. IBFD TRANSCEIVER SIMULATION

In this section the methodology employed, components comprising, and rationale behind the simulation is presented. In order to evaluate the theoretical performance of analog self-interference cancellation (ASC) given variation in noise and component performance, a systematic model is developed in MATLAB of a prototypical IBFD transceiver. This model was developed to be noise cognizant for the most egregious noise sources. Though designed around IEEE802.11ax's lower transmission frequency of 2.4GHz, the intent was not to confine results or design to a particular wireless networking standard. Hence, an attempted generality in component and system emulation is taken into consideration to allow for extrapolation to other standards.

5.1 Transceiver Hardware Description

An overview of the guiding transceiver hardware topology highlighting the domains of cancellation and sources of SI is depicted in Figure 12. As illustrated, the transmission channel is assumed to take a digital baseband signal, convert to its analog equivalent, and modulate on to a higher frequency carrier. Additionally, as discussed in Section 4.6 component nonlinearities are accounted for. Accordingly, the subsequent inclusion of a RF power amplifier pre-transmission is incorporated. In a corresponding manner, the reception channel was designed to mirror functionality of the Tx path. A lower power signal-of-interest (SOI) is assumed to be received, coupled with a much greater SI component. The adopted mechanisms of SI impingement are considered to be; (i)direct transmission through finite duplexer or circulator isolation, (ii)signal reflection from system to antenna impedance mismatch, and (iii)ambient reflections caused by electromagnetic environmental interactions. Before amplification, the corrupting SI is attempted to be minimized by means of deconstructive interference with a post-PA sampled Tx signal. Sampling at this point effectively allows for the analog self-interference canceller to inherently consider all transmission nonlinearities, distortions, and noise without necessitating extraneous modeling or auxiliary transmission paths. Albeit, at the expense of any incurred difficulties of cancellation in the RF band and probability of increased circuit complexity for MIMO applications [69]. After ASC, the SOI and any residual SI passes through a Low Noise Amplifier (LNA), ideally with a higher dynamic range to ameliorate the effects of imperfect cancellation. From here the

signal goes through a demodulation process for down conversion to its analog baseband form. This baseband analog signal is then sampled and converted by the ADC into its equivalent digital bitstream. Further cancellation via digital domain techniques could be accomplished at this point. As alluded to in the discussion on in Section 3, imperfect ASC coupled with a finite dynamic range of the ADC effectively bounds the efficacy achievable with the DSC. Therefore, overall transceiver SIC is paramount to satisfactory prior ASC performance. Given the oft quoted total cancellation goal of 110dB, this would set an objective of 60dB of cancellation for the ASC.

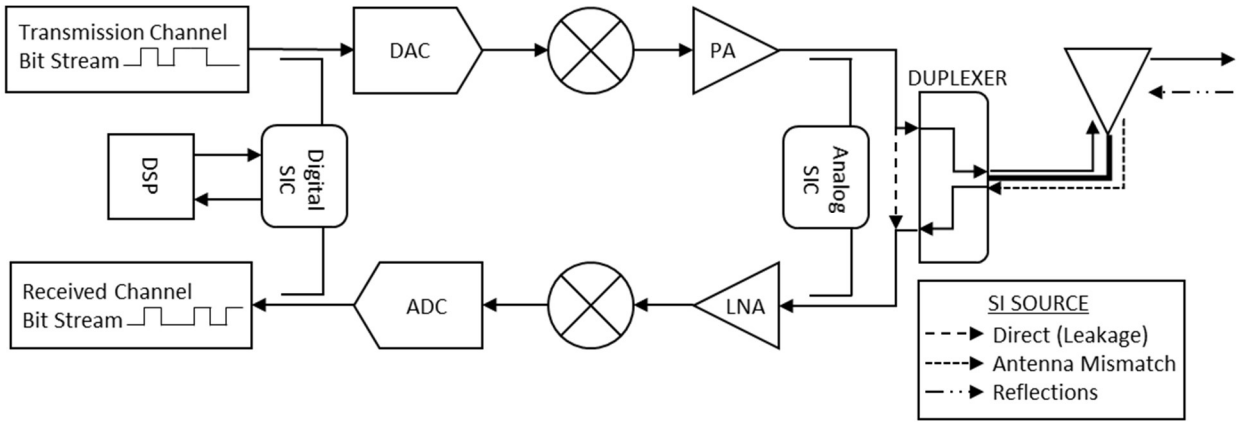

Figure 12 Basic Transceiver Topology

Stepping further from this generalized architecture, the modulation scheme and transmission bit pattern is considered. In order to stress the system by the full range of conceivable real-world scenarios, the input sequence should be able to emulate both upper and lower frequency bounds from anticipated bit rates. Furthermore, to avoid any unintentional pattern-awareness system dependencies, this emulation should be done in as stochastic a manner as possible. In accordance with these considerations, a PRBS pattern of order 31 was chosen to serve as the input sequence generating algorithm. Discussed in Section 4.3, a PRBS pattern is a commonly employed means to test high speed digital serial interfaces that, though produced in a deterministic manner, displays mathematically stochastic properties. For example, an approximate 1:1 '0' to '1' generation correspondence is found when averaged over the entire sequence. Additionally, a power spectral density with apparent frequency independent characteristics (i.e., white noise-esque) is achieved. Furthermore, by choosing an order of 31 for the PRBS effectively allows a sequence of $2^{31} - 1$ possible bits without sequence repetition.

As the primary benefit sought from IBFD development is to augment network information capacity given a constrained spectrum, the modulation scheme employed is probable to be one that attempts to maximizes channel capacity and is akin to something already deployed. With this in mind, two conventional higher order digital modulation techniques appear to fit the bill; Multiple-Phase Shift Keying (M-PSK) and Quadrature Amplitude Modulation (QAM) [2,3]. In fact, these two methods represent the most traditionally employed modulation methods for fixed broadband wireless networks with QAM holding a per-bit spectral efficiency over PSK [76]. In M-PSK, the basic underlying principle is that bits can be encoded onto a constant magnitude symbol and the relative phase of the symbol denotes the encoded bit sequence. The more distinctions in phase, the larger the number of bits able to be encoded. Albeit the harder to accurately discern the symbol in a noisy channel. QAM takes a similar approach. However, with QAM both phase and amplitude are used to create symbol distinction. In QAM, two signals, one designated the in-phase or *I* component, and the other, the quadrature or *Q* component, are kept 90° out of phase with one another. The respective amplitude of each component is determined by a designated number of bits, and when this phase-shifted amplitude-adjusted set of components is summed, the result is a distinct phase-amplitude combination representative of the input bit sequence. For example, a 16QAM system would use two bits on its *I* component to set four possible amplitudes and two bits on its *Q* component in a similar manner. When summed, due to the quadrature relationship, results in a symbol space of 16 distinct outcomes. Implicit in this amplitude inclusive modulation, is an expansion of the overall effective symbol discernment area to that of M-PSK whose amplitude exclusion confines the total area of discernment regardless of distinctive phase resolution. Furthermore, established technical standards for broadband digital transmission already specify operation for 256-QAM, and current deployments of Wi-Fi 6 (802.11ax) use 1024-QAM operation [70,71]. As a result, QAM is chosen as the default modulation mechanism in this transceiver model. Figure 13 depicts a more detailed transceiver model depicting these input decisions.

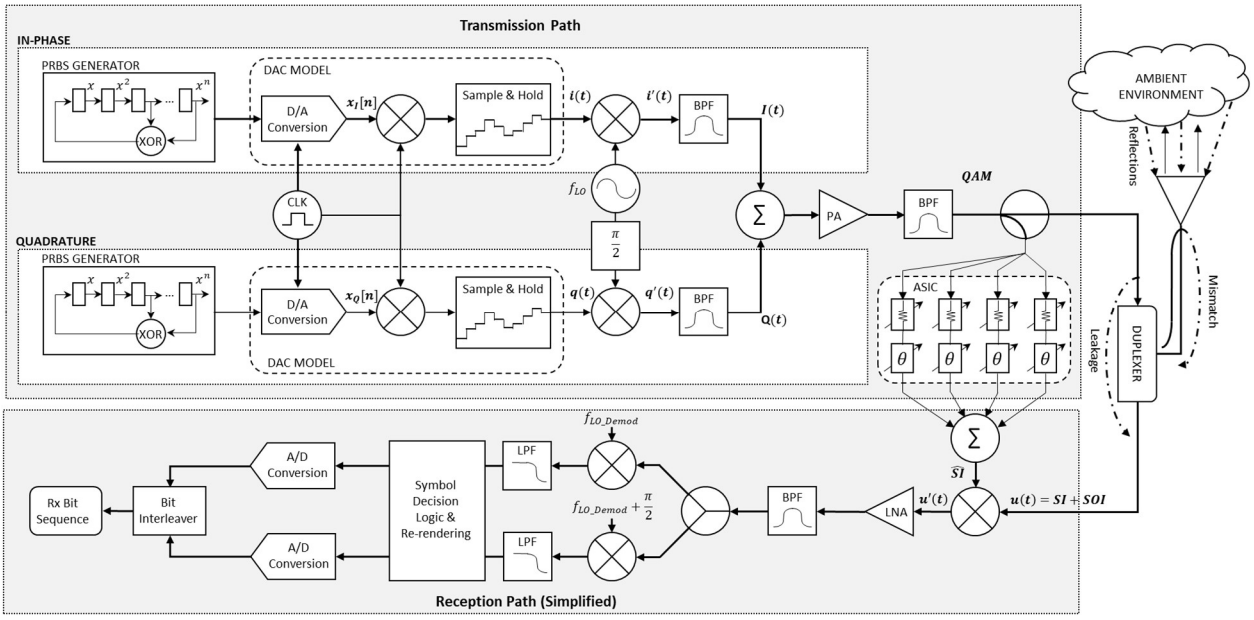

Figure 13 IBFD QAM Transceiver Architecture

In this more comprehensive model, the Tx Channel can now be observed as being bifurcated into two branches; in-phase and quadrature. With each segment, the DAC is now considered to be a nonlinearly modeled device with a non-ideal output rendered by a fluctuating reference voltage. An imperfect clock to output isolation results in an intrusive clock component consequently mixing with the DAC's output. In addition to these nonidealities, system thermal noise and DAC noise spectral density profiles are assimilated into the signal conversion as well as a non-ideal bandlimited interpolation achieved via an emulated sample-and-hold circuit. From there, each baseband analog signal undergoes up-conversion via RF mixing with a phase noise affected local oscillator. A RF mixer is ideally, in principle, a linear multiplier. However, as all RF mixers include some form of nonlinear element (e.g., Schottky Diode, FET, or other transistors) all mixer's in the transceivers topology are modeled as third-order polynomial nonlinear devices. By allowing the one branch to be fed a $\pi/2$ phase shifted replica of the oscillations, the complementary signals are effectively rendered in quadrature. After mixing, bandpass filtering is employed to attempt to isolate and extract the desired spectral components from the second and third-order IMD adulterated spectrum. These filtered signals, denoted $I(t)$ and $Q(t)$, are now composed primarily of components from the desired linear products of the LO and baseband mixing. Next, the output's quadrature amplitude modulated signal is produced by a further subsequent nonlinear RF mixing stage. It's at this stage where susceptibility to I/Q imbalance is

implicitly incorporated as any noisy stochastic element from proceeding operations (e.g., amplitude of phase variation) will effectively render a non-ideal symbol level. This tertiary nonlinear product is then passed through a fourth and final nonlinear amplification stage. The PA's fifth-order nonlinear profile primarily results in imparting harmonic distortion up to and including its fifth element. Additionally, post-amplification, the effect of the total system noise figure is observed as an overall degradation in the signal's noise floor (i.e., increase in the bandlimited white noise spectral content). Finally, a second stage of bandpass filtering is employed for further nonlinear amelioration before transmission. However, this time extraction is done to provide a mathematical summation, not multiplication, of spectral components $I(t) + Q(t)$. Thus, this modulated output becomes a function of both the amplitude (i.e., multiplicative mixing) and relative 90° phase relationship (i.e., additive mixing) effectively rendering the QAM signal.

Parallel in approach, the Rx channel is shown in its equivalent dual configuration for proper demodulation of a received QAM signal. Similar to what was shown in the generalized IBFD Transceiver Topology, after reception as much of the offending SI signal is attempted to be cancelled in the analog domain before subsequent amplification. However, in this depiction, the ASC is shown as a specific cancellation architecture which features a network of parallel branches, or taps, each consisting of a variable amplitude and phase component. In this way each tap attempts to find optimal weight settings, via real-time gradient descent optimization, for their respective phase and amplitude so that their aggregate output, $\widehat{SI}$, when added with the SI corrupted signal $u(t)$, leaves only the desired signal-of-interest (SOI) for amplification. However, in practice, their resultant output, $u'(t)$, likely consists of both the SOI and a portion of the remaining SI, termed residual SI. To attempt further reduction of the residual SI, subsequent digital domain cancellation could take place after demodulation and analog to digital conversion. Though, for the evaluation of analog domain SIC efficacy these additional steps are considered superfluous. In the sense that anything done unto the signal past $u'(t)$ would not affect the ASC's cancellation ability. Accordingly, in this thesis all measurements and calculations are performed prior to LNA amplification.

5.2 Simulation Delineation

This section attempts to step the reader through each of the simulated stages so that they may understand the processes involved and how the calculations are performed. Figure 14 depicts a sequential overview of these steps in the modeled transceiver. To give a more cogent elucidation, the overall program has been enumerated into six distinctive segments including:

1. Initialization
2. Digital to Analog Conversion
3. Phase Noise Affected Oscillation
4. RF Mixing for $I(t)$ and $Q(t)$ Components
5. RF Mixing & QAM Amplification
6. ASC Training & SIC Performance

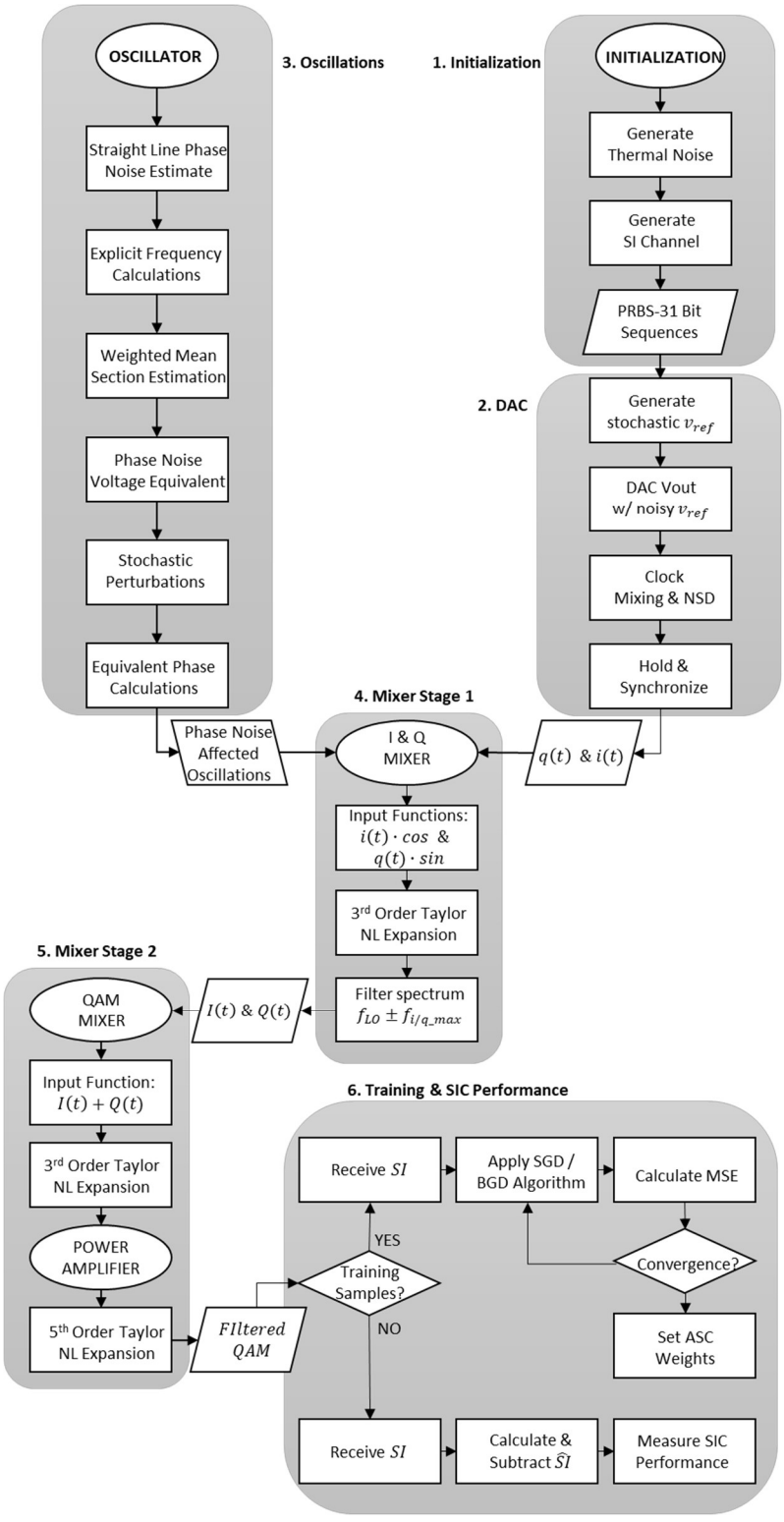

Figure 14 Simulation Flowchart

5.2.1 Initialization

During the initialization segment, user-set parameters are defined and randomized variables are created. Additionally, the theoretical Nyquist-Johnson (i.e., thermal) system noise is calculated, and input PRBS bit streams are generated. For an overview of program variables utilized and their respective definitions, please refer to Section 6.1. Of these variables, the following are worth specifically emphasizing due to their randomized assignment due this stage:

nBranches – Represents the number of branches (or taps) used in the ASC during the run. $nBranches \sim U(x), x \in \{4,5, \dots ,30\}$.

nPath – Represents the total number of paths that SI is coupled into the Rx channel through. $nPath \sim U(x), x \in \{20,21, \dots ,50\}$.

modelBeta – Randomly assigned initial attenuations for each branch of the ASC. $modelBeta \sim U(0,0.1)$.

modelPhase – Randomly assigned initial phase setting for each ASC branch. $modelPhase \sim U(0,2\pi)$.

pathLoss – Path attenuation array for total number of SI paths. Generated in an algorithmic manner where the first loss if selected from the uniform distribution of 0 to 0.25, and every subsequent loss is selected uniformly selected from 0.5 minus the sum of all preceding losses. In this way, regardless of the number of randomly generated SI paths, the limit of the sum of all paths converges to 0.5 as the number of paths tends to infinity. i.e., $\lim_{nPath \to \infty} \sum_{\forall i} \alpha_i = 0.5$, where α_i is the attenuation for the *ith* path. Note that the pathLoss and pathPhase assume that the paths with least loss, or greatest Rx coupling, and the least phase shift are correlated. This was an assumption made in lieu of having actual environmental data to model the SI channel on. Logic behind this assumption assumes a confined environment in which the path length elongates in proportion to the number of reflective boundary interactions a signal traverses before recoupling into the transceiver.

pathPhase - Represents the phase shift the transmitted signal will undergo when traversing a specific SI path. $pathPhase \sim U(0,2\pi)$.

convergence – Number used to calculate the percentage decrease of initial mean-square-error in the gradient descent algorithm and conclude that the algorithm has converged. $convergence \sim U(x), x \in \{0.0001\%\ 0.001\%\ 0.01\%\ 0.05\%\ 0.1\%\}$.

DACCoeff – Coefficients of Taylor Series Expansion used for DAC output and DAC clock mixing. Third order expansion assumes DC component, a_o, is zero and other three components are distributed $DACCoeff \sim U(0,1)$.

IQCoeff – Coefficients of Taylor Series Expansion used for $i(t)$ and $q(t)$ mixing with their respective oscillation components. Third order expansion assumes DC component, a_o, is zero and other three components are $IQCoeff \sim U(0,1)$.

QAMCoeff – Coefficients of Taylor Series Expansion used for $\mathrm{I}(t)$ and $\mathrm{Q}(t)$ mixing to produce the overall output QAM signal. Third order expansion assumes DC component, a_o, is zero and other three components are $QAMCoeff \sim U(0,1)$.

PACoeff – Coefficients of Taylor Series Expansion used for generating harmonics components as the QAM signal is passed through the power amplifier. Fifth order expansion assumes DC component, a_o, is zero and other four components are $IQCoeff \sim U(0,1)$.

vThermal – Variable the contains the systems calculated Johnson-Nyquist noise due to thermal charge carrier perturbations. It is calculated for the standard ambient temperature of 298.15 Kelvin (25°C) and then has a stochastic element incorporated by setting the final thermal voltage to a randomly generated number with a normal distribution. I.e., $vThermal \sim N(ideal, 0.1)$, where $ideal$ is the mean and 0.1 is the variance of the distribution.

Finally, in the initialization phase the bit stream for both the I and Q components are generated with a PRBS pattern of order 31. The length of the PRBS pattern generated is set to the number of desired analog samples multiplied by the number of conversion bits of the DAC. In this way the program ensures that pre- or post- '0' padding is rendered superfluous.

5.2.2 Digital-To-Analog Conversion

As alluded to by the eponymous titling of this section, its primary purpose is to handle the conversion of input bit streams to equivalent analog voltage based on the number of conversion bits set, reference voltage supplied, and internal voltage drop assumed by the DAC. Additionally, during this portion the noise contribution by the DAC is determined and incorporated based on common datasheet parameters including clock to output intrusion nonlinearities, SFDR, and noise spectral density [72,73].

The program's ideal input to output relationship for the modeled DAC can mathematically be described as

$$v_{DAC} = \frac{D(v_{ref} - v_d)}{2^b - 1}$$

Where D is the decimal value of the input binary word, v_{ref} and v_d represent the reference voltage supplied and internal voltage drop, respectively, and b is the number of bits used in conversion. To accomplish this ideal conversion, the script *DAC.m* is called. It commences by generating a base-2 array that lists the decimal value for each word's respective bit, from MSB to LSB, and repeats this array for the total number of symbols. From here it uses the input bit stream to compute an element-wise multiplication with the base-2 array producing an output array only retaining the equivalent decimal values for bits holding a '1' value. Computing each word's equivalent decimal value is then accomplished by the summation of each word's corresponding array elements. Effectively this produces an array the length of the specified number of samples with each element holding a word's decimal equivalent value. This process is illustrated for a single byte in Figure 15.

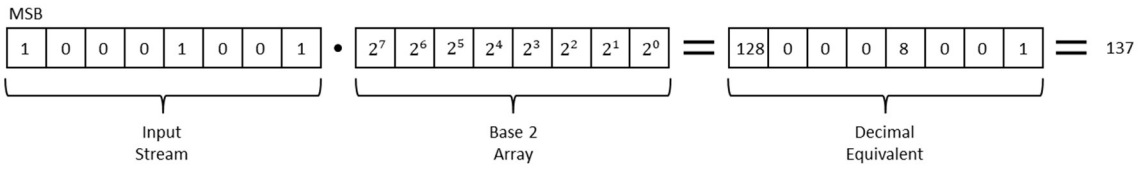

Figure 15 DAC via Array Multiplication

With this computed decimal equivalent array, the ideal output could then be obtained by simple multiplication of the effective conversion voltage (i.e., $v_{ref} - v_d$). However, incorporation of transitory power supply fluctuations during conversion is performed by generating a voltage array with a distinct v_{ref} for each sample. Each v_{ref} is randomly selected from a normal distribution with a mean of the ideal reference voltage, and a variance set by the user. By doing so the DAC's output voltage is rendered stochastic at each simulation even in the event of identical input bit sequences.

After this initial analog conversion, the rendered samples are passed to script *RF_Mixer.m* with the DAC's output timing clock array. The amplitude of the timing clock is calculated as $10^{\frac{CLK_{iso}}{20}}$, where CLK_{iso} is the specified clock-to-output decibel isolation. The actual magnitude of clock

mixing nonlinear distortions is able to be empirically determined from a DAC's datasheet specification for Spurious Free Dynamic Range. As described in 4.7, stated simply, SFDR represents the range between the desired output components in the DAC's spectrum and the next highest harmonic, or spur, present. In this way, SFDR only alludes to a conceivable nonlinear spectrum.

One final noise taken into consideration in this stage, is the noise spectral density (NSD). The NSD of a DAC is primarily a representation of the noise power profile, relative to the power of the desired spectral element(s), measured within a specified window. For both simulation simplicity and taking note that many DACs' display a NSD profile that often display only a marginal few dB variation over the characterized spectrum, the decision was made to incorporate NSD as a bandlimited white noise source [72,73]. Accordingly, the equivalent additive gaussian white noise voltage incorporated for NSD was estimated as $10\log_{10} 10^{\frac{NSD}{10}} \Big/ \frac{(v_{ref}-v_d)^2}{R}$, where NSD is the specified noise spectral density in dBc/Hz. For a typical simulative run, Figure 16 depicts the initial baseband rendered analog samples for PRBS pattern of order 31. This signal in particular had a

bandwidth of 100 MHz and shows an extraction of the first 1000 samples produced.

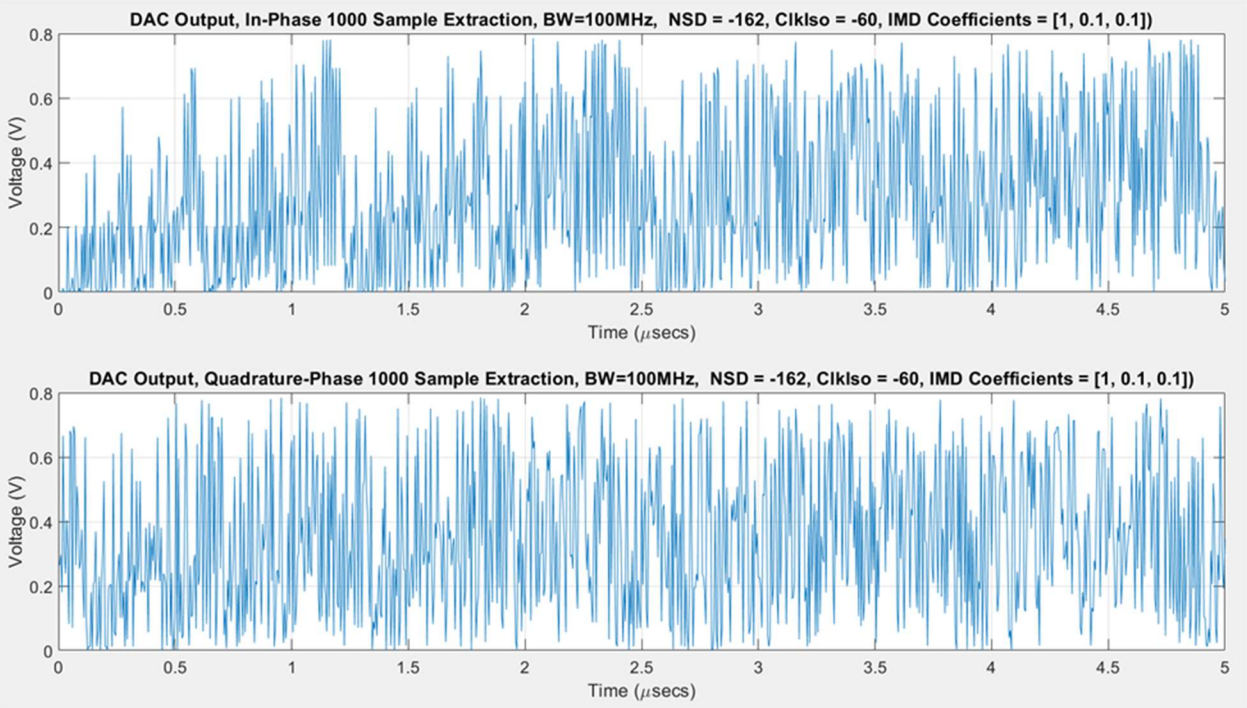

Figure 16 DAC 1000 Sample Extraction

After non-ideal interpolation is accounted for by integration of a generic sample-and-hold circuit, a signal is rendered characteristic of the one observed in Figure 17. Here each plateau represents the analog voltage of one sample initially rendered in the conversion process.

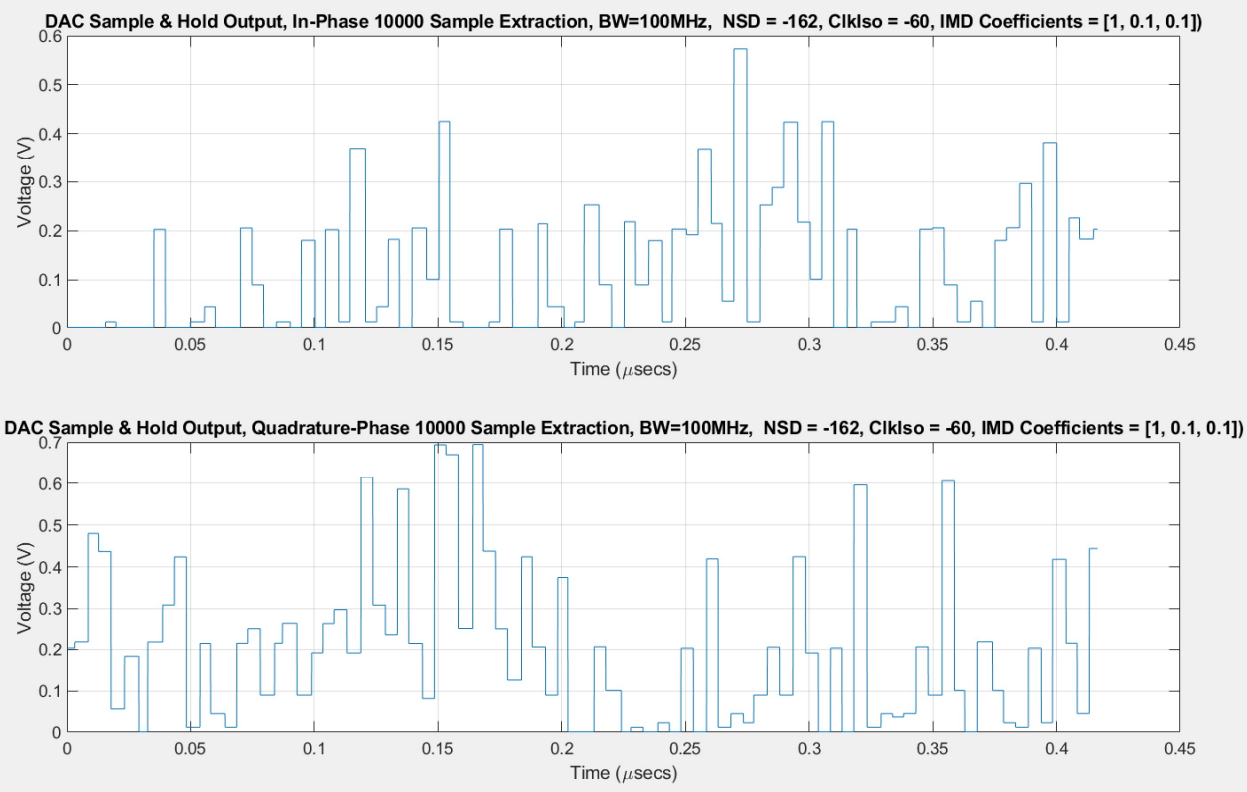

Figure 17 DAC Sample & Hold Output

5.2.3 Phase Noise Affected Oscillator

Working in a converse manner from the theory developed in section 4.5, this next section aims to compute the analogous voltage fluctuations for a rendered discrete time series when given a phase noise profile's end points specification. If the phase noise voltages, $v_{PN}(t)$, are assumed to be of the form $v_{PN}(t) = a \cdot sin(\vartheta(t))$, then when equivalent phase noise voltages are able to be discerned the actual phase perturbations can simply be found by taking the inverse sine of the series (I.e., $\vartheta(t) = sin^{-1}(v_{PN}(t)) \quad (Letting\ a = 1)$). Calculation of the voltage, given a one-sided PSD profile turns out to be a relatively easy feat. For a discrete-time series, $x(n)$, or discrete-frequency series, $X(f)$, we have the general PSD relation, $P_{xx}(f)$, as:

$$P_{xx}(f) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j2\pi f}\right|^2 = \frac{1}{N}|X(f)|^2$$

To establish a realistic phase noise PSD profile for the simulated oscillator, the point-wise specifications from Vectron's OX-305 Ultra Low Phase Noise Oven Controlled Crystal Oscillator

(ULPN-OCXO) and their VS-701 Voltage Controlled Saw Oscillator (VCSO) were analyzed [74,75]. The point-wise profile given in Table 3.

Table 3 Simulated Phase Noise Profile

Frequency Offset (Hz)	Phase Noise	Unit
10	-70	dBc/Hz
100	-100	dBc/Hz
1,000	-125	dBc/Hz
10,000	-145	dBc/Hz
100,000	-160	dBc/Hz
1,000,000	-169	dBc/Hz
10,000,000	-170	dBc/Hz

Using these specifications, a simple ideal straight-line profile is able to be produced when frequency is plotted on a logarithmic scale. This profile conforms to the general function $dBc = m\log_{10} f + b$. Where each section has its own distinct slope and theoretical intercept. Slope may be calculated by $m = \frac{dBc_{i+1} - dBc_i}{\log_{10} f_{i+1} - \log_{10} f_i}$ and intercept $b = dBc_i - mlog_{10}f_i$. Calculated values for these regions are displayed in Table 4. From this, the phase noise values are able to be calculated for any frequency of interest. Stochastic variation may be accomplished when the calculated value is set to the mean of the normal distribution and standard deviation is user specified. One simulation's rendering of this interpolated straight-line phase noise profile and its stochastic counterpart are displayed in Figure 18.

Table 4 Calculate Phase Noise Slopes & Intercepts

Region Span (Hz)	Slope	Intercept	Unit
10 - 100	-30	-40	dBc/Hz
101 - 1,000	-25	-50	dBc/Hz
1,001 - 10,000	-20	-65	dBc/Hz
10,001 - 100,000	-15	-85	dBc/Hz
100,001 - 1,000,000	-9	-115	dBc/Hz
1,000,001 - 10,000,000	-1	-163	dBc/Hz

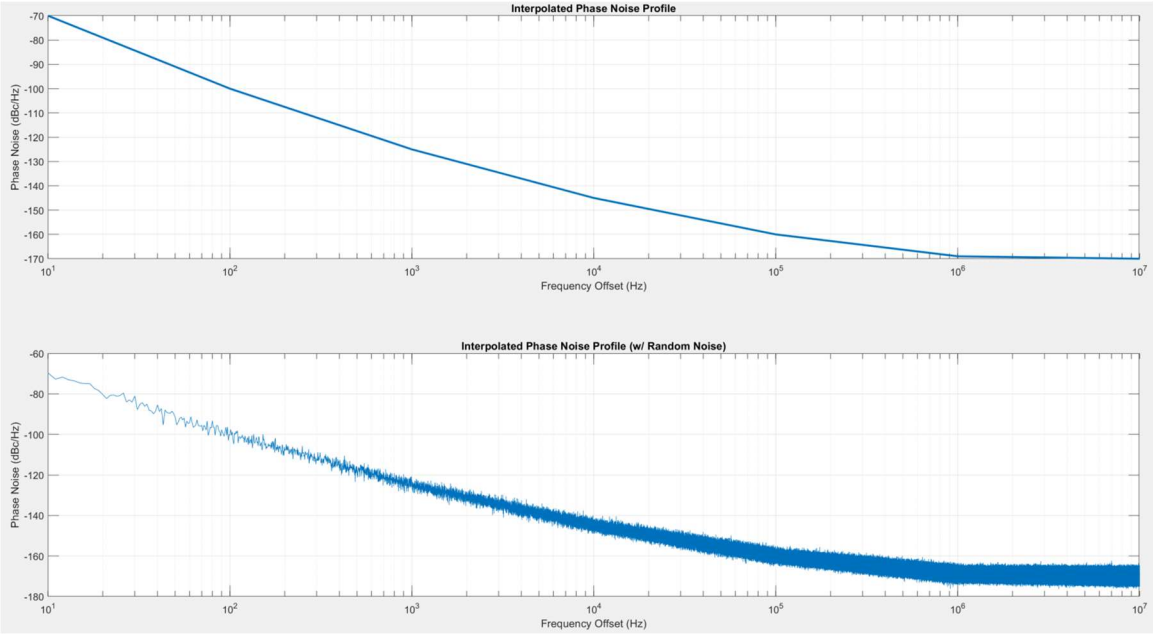

Figure 18 Phase Noise Profiles: Straight-line (Top) & Noisy (Bottom) Interpolation

After rendering the stochastic phase noise estimate, the task becomes a matter of translation into a voltage discrete time series for transmission signal incorporation. To convert from units of dBc into voltage of a particular offset frequency, $v_P(f)$, the following relation was used

$$v_P(f) = \sqrt{2P_c Z_o \cdot 10^{\frac{dBc(f)}{10}}}$$

Here P_c is the carrier power in watts and Z_o is the nominal characteristic impedance of the IBFD transceiver. With this, a discrete estimation of phase noise, $\hat{\vartheta}(t)$ can be calculated by regarding each offset frequency as the summation of individual sinusoidal sources. This gives

$$\sin\left(\hat{\vartheta}(t)\right) = \sum_{i=f_{start}}^{f_{end}} v_{pi}(f)\sin\left(2\pi f t\right)$$

Then

$$\hat{\vartheta}(t) = sin^{-1}\left(\sum_{i=f_{start}}^{f_{end}} v_{pi}(f)\sin(2\pi f t)\right)$$

However, as phase noise profiles typically range with frequencies from 10Hz to 10MHz carrier offset frequency, this method of inclusion is computationally intensive, requiring N number of multiplications and N-1 summations. Where N is the number of offset frequencies for each time index t rendered. To ease the burden and reduce computational time, it was decided that only the most egregious phase noise offenders, corresponding to offset frequencies from 10Hz to 1KHz, would be directly calculated. It was discovered that other bands of phase noise could have their overall cumulative effect considered by a weighted average at a median frequency of a specific section. Derivation of this can be seen in Appendix A. Therefore, for phase noise sections greater than 1KHz, the phase noise section $\hat{\vartheta}_i(t)$ can be estimated as

$$\hat{\vartheta}_i(t) = sin^{-1}\big(w \cdot v_{pavg}(f_{median})\sin(2\pi f_{median} t)\big)$$

The weighting factor, w, is set as the number of discrete integer frequencies within the band. Summing the phase deviations of each section at every time index t will then have an overall effect on the signal equivalent to that of the given profile. This phase noise inclusive output of then becomes

$$v_{OSC}(t) = v_c \cos\big(\omega_o t + \vartheta(t)\big)$$

A juxtaposition of a 1000 sample extract and it's corresponding phase noise profile are depicted in the top and bottom images of Figure 19, respectively.

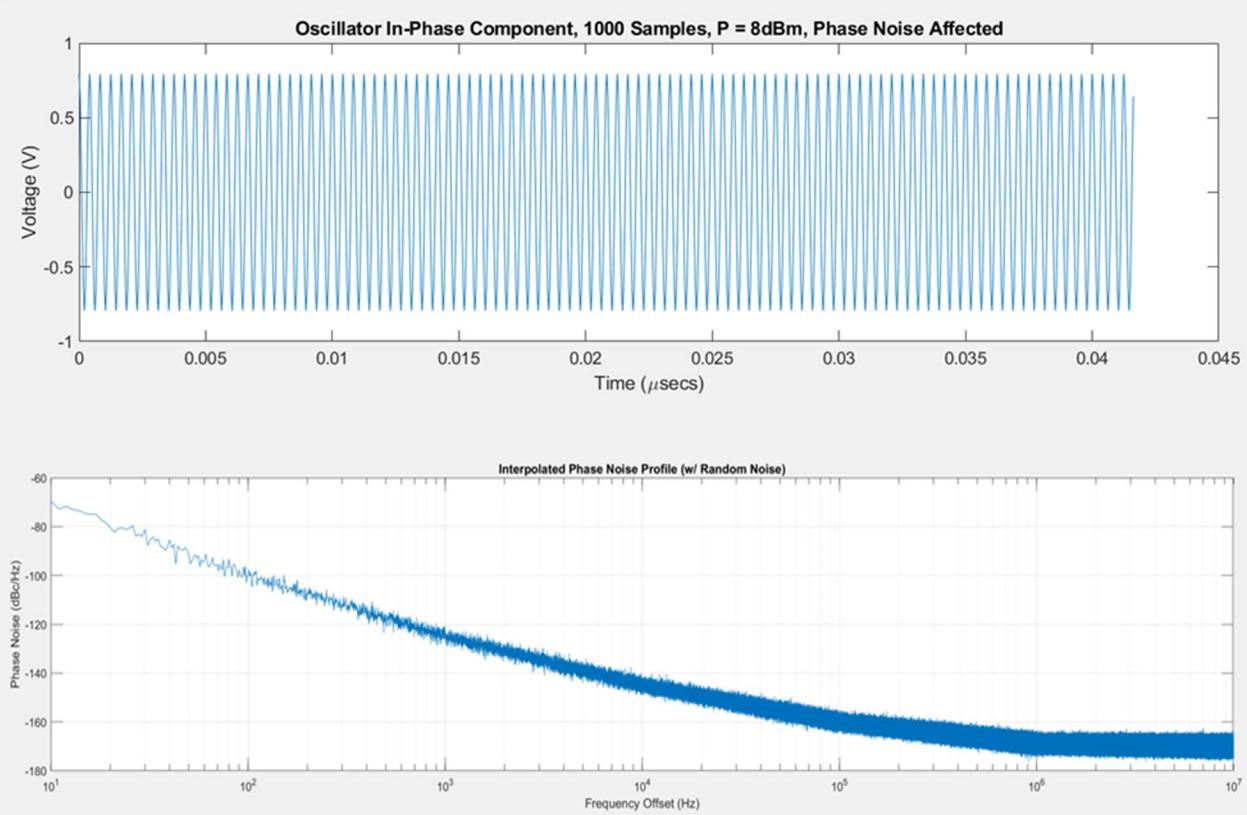

Figure 19 2.4GHz Oscillator Outputs Cosine (Top) & Sine (Bottom)

5.2.4 RF Mixing for *I(t)* and *Q(t)*

After computing both the phase noise affected oscillations and the temporally synchronized DAC samples, these resultant components are combined through an RF mixer to produce the in-phase and quadrature phase components. The in-phase component, *I(t),* ideally assumes the form of $I(t) = i(t)\cos(\omega_{LO}t)$ and *Q(t)* the form $Q(t) = q(t)\sin(\omega_{LO}t)$. The explicit sinusoidal components represent those of the oscillator, and $i(t)$ and $q(t)$ that of the DAC's output sampled-and-held baseband components. To achieve the desired outputs the baseband and carrier components are passed into the script *IQ_Mixer.m* where a third-order Taylor series expansion allows for the estimation of mixer nonlinearities. As described in section 4.6, the magnitude of the nonlinearities is controlled via a given three element coefficient array. The DC component, a_0, is set to its ideal value 0 and the remaining three components are either set to [1,0,0] for noise excluding runs, or randomized from the uniform distribution $U(0,1)$.

Post mixing, each component now subsumes all spectral content from the corresponding harmonic and IMD products the mixing process imparts. Mirroring likely "real-world" mitigation strategies, the produced spectrum is then filtered to attempt isolation of the upconverted desired portion of the spectrum and reject higher and lower frequency distortion. This is done through the incorporation of a Finite Impulse Response (FIR) bandpass filter (BPF) with a roll-off factor of 0.8 and target stopband attenuation of -60 dB. The passband frequencies, set according to the trigonometric relation $\cos(\omega_{BBMax}t) \cdot \cos(\omega_{LO}t)$, correspond to spectral content spanning from from $f_{LO} \pm f_{BBMax}$. Note, f_{BBMax} denotes the maximum expected baseband frequency of 100MHz and 2.4GHz was used as the frequency of oscillation. Figure 20 below illustrates an eighty-nanosecond portion of in-phase and quadrature components produced during one of the preliminary simulations.

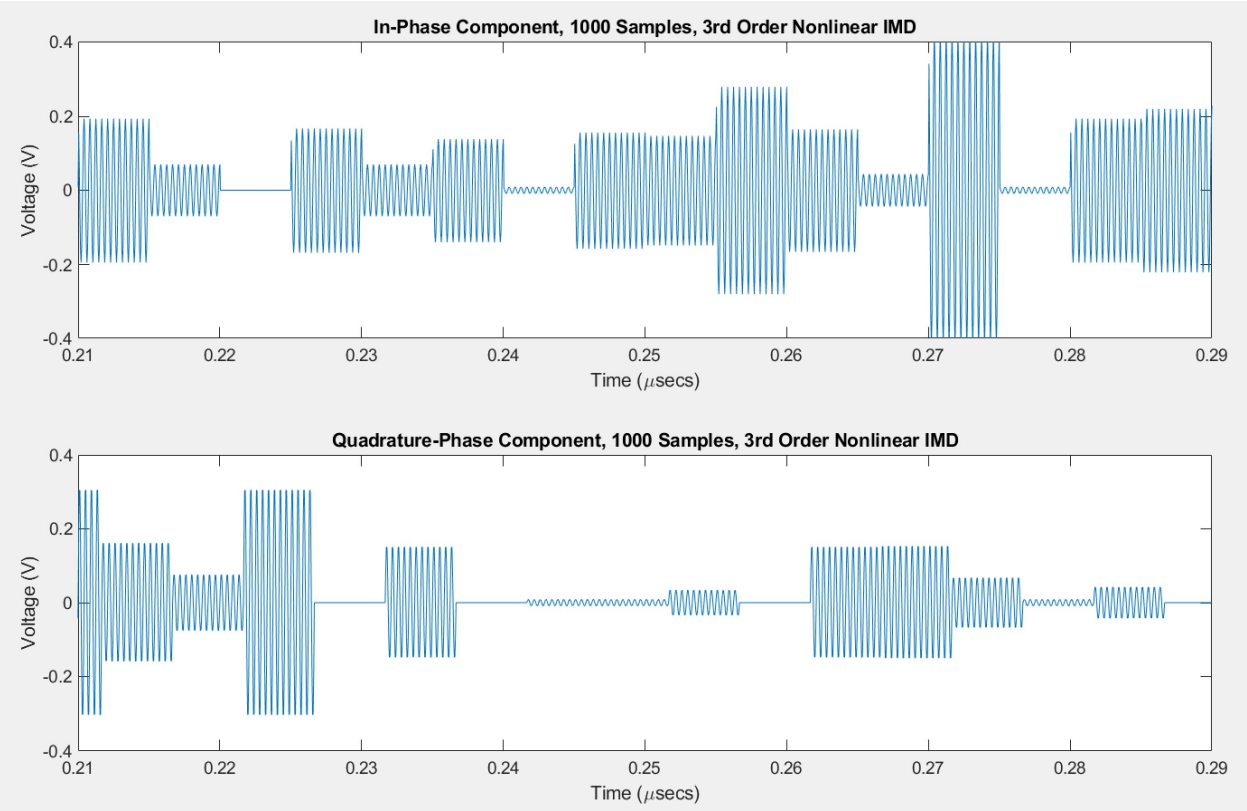

Figure 20 *I(t)* & *Q(t)* 1000 Sample Extraction

5.2.5 RF Mixing & QAM Amplification

An idealized quadrature amplitude modulated signal assumes the form of $i(t)\cos(\omega_{LO}t) + q(t)\sin(\omega_{LO}t)$. It can be observed that this relation corresponds to simply the addition of $I(t)$ and $Q(t)$ which were discussed in the previous section. To effectively obtain the addition of these two components both signals are again fed into a subsequent RF mixer. Again, the RF mixer's nonlinearities are estimated using a third order Taylor expansion with DC component set to zero and other nonlinearity coefficients randomly assigned via the distribution $U(0,1)$ on noise affected runs.

Prior to filtering for the desired summed components, the signal is fed through the transmission chain's power amplification stage, where a fifth-order Taylor expansion is used to estimate imparted harmonics. Note that as both transmission line loss and device insertion loss were not accounted for in this simulation, gain of the PA was left at one and inclusion served to solely replicate expected harmonic nonlinearities. Using common datasheet specifications for gain and OIP3 from devices including Maxim's MAX2242 and Mini-Circuits PMA2-252LN+ [64,65], the following relationship was derived to estimate the anticipated most egregious third order IMD product voltage, v_{IMD}^3, from a given input level.

$$v_{IMD}^3 = \sqrt{10^{0.006 \quad ^2 - 2.42374} \cdot R}$$

Here R denotes the ideal system characteristic impedance and Vin the input voltage. Using this relationship and the coefficient relationship delineated in Section 4.6, appropriate coefficients could be assigned based on a given input. However, many of the investigative simulations were left to randomly generate corresponding coefficient from the distribution $U(0,1)$ to explore the affective severity given a theoretical device performance.

The final stage of amplification follows the PA, where a FIR BPF is modeled with a roll-off factor of 0.8 and a target stopband attenuation of -60dB. As is used in the previous mixing stage, passband frequencies once again span between $\omega_{LO} \pm \omega_{BBMax}$. However, the nuanced implication in the filtration frequencies at this stage are representative of an additive, not multiplicative process as was previously done. After filtering the signal is now fully modulated into its desired transmittable

512 QAM output. A portion of a rendered 512 QAM signal with a 2.4GHz carrier and 100MHz baseband bandwidth signal is depicted in Figure 21.

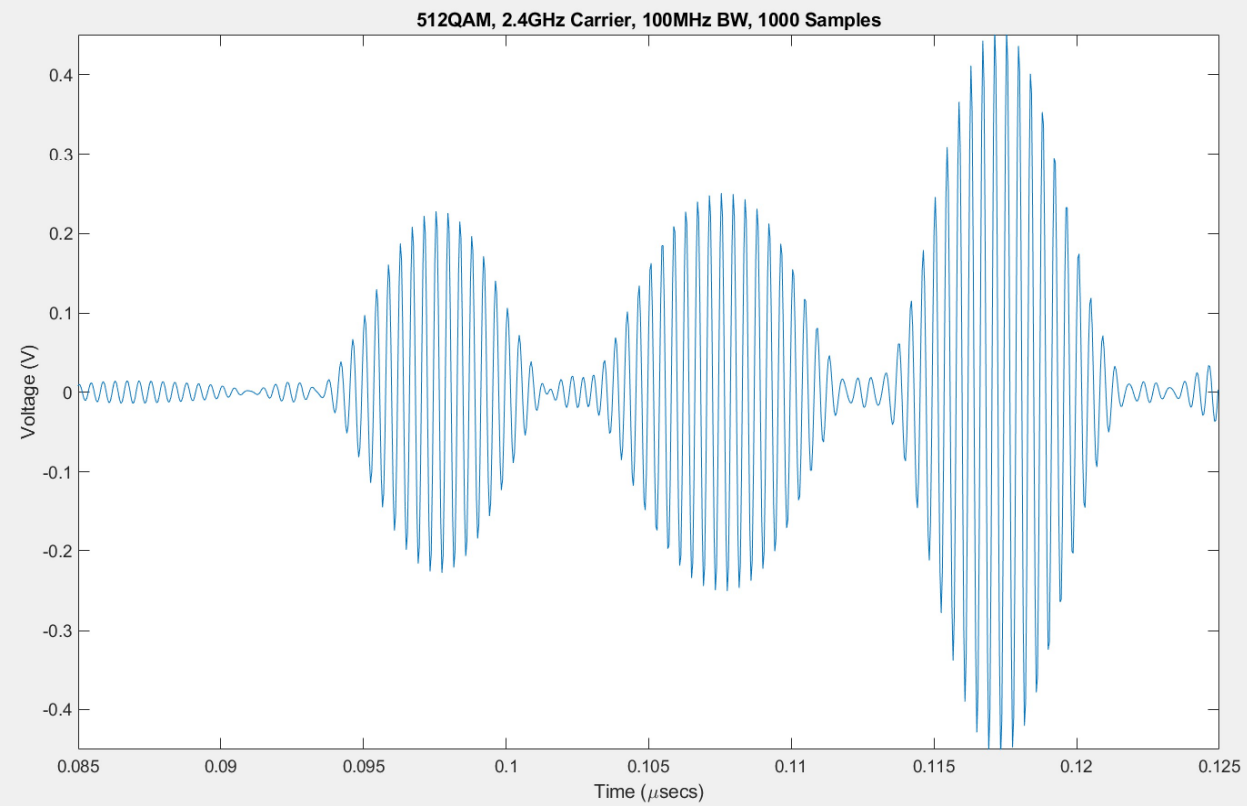

Figure 21 512QAM, 2.4GHz, 100MHz BW 1000 Sample Extraction

5.2.6 ASC Parameter Determination & SIC Performance Evaluation

During this final portion of the simulation determination of appropriate amplitude and phase weights for the ASC is accomplished through the use of a gradient descent algorithm. Additionally, the various measurements and calculations used to characterize transceiver performance are described.

Contrary to techniques employed in digital domain cancellation, analog cancellation typically assumes no channel foreknowledge. Therefore, for the tapped parallel branch architecture employed in this simulation to be effective, some mechanism of feedback had to be incorporated in order to allow for the SI channel estimation. As discussed in Subsection 3.1.2, traditionally this has been accomplished through use of a transmitted training pilot-tone or preamble sequence. Following suit, before transmission of the intended signal a relatively small number of samples

are extracted and used for training purposes. As the overall content of the baseband signal subsumed a mathematically stochastic PRBS pattern, the first *N* samples syphoned off could be considered ergodic of the overall transmitted signal.

As the output samples extracted for training are now representative of a noisy multi-tonal discrete time series, incorporation of ASC amplitude and phase weights is done via the implementation of a Hilbert transform. This amplitude and phase incorporation is summarized by the following mathematical relationship

$$\alpha x[n+\theta] \cong \alpha(x[n]\cdot\cos(\theta) - H\{x[n]\}\cdot\sin(\theta))$$

Here, the ideal time sequence affected by loss α and phase shift θ, $\alpha x[t+\theta]$, is estimated by using the actual given discrete series $x[n]$ and its corresponding Hilbert transform, $H\{x[n]\}$. Multiplying the two by the cosine and sine of the desired phase shift, respectively, and then subtracting the two products effectively allows for phase shift incorporation for the entire spectral content of the discrete series. Subsequently, the multiplication by the amplitude weights renders overall signal attenuation by the specified amount. The derivation of this relationship can be found in Appendix A of this paper. By using this relationship, the modeled ASC self-interference, $\widehat{SI}$, is determined for every branch, tap attenuation weights, β, and phase shifts, ϑ. Therefore, every branch of the ASC effectively adopts the form of $\widehat{SI}_i = \beta_i(x[n]\cdot\cos(\vartheta_i) - H\{x[n]\}\cdot\sin(\vartheta_i))$. Accordingly, the overall modeled SI is calculated as

$$\widehat{SI} = \sum_{i=1}^{nBranches} \widehat{SI}_i$$

To derive proper weighting for the estimated SI a gradient descent algorithm is employed. Both a pure stochastic gradient descent (SGD) and mini-batch gradient descent (mini-BGD) algorithm are incorporated for possible usage in the program. Selection of which occurred during initialization by setting parameter *GDType* to '1' or '2' for either SGD or BGD, respectively. Per the underlying tenets described in Section 4.8, the loss function to minimize for this algorithm is defined as

$$argmin\ L\left(SI, \widehat{SI}\right) = \sum (SI - \widehat{SI})^2$$

$$= \sum_{k=1}^{nSamples} \left(\sum_{j=1}^{nPaths} \alpha_j\left(x_k[n] \cdot \cos\left(\theta_j\right) - H\{x_k[n]\} \cdot \sin\left(\theta_j\right)\right) \right.$$

$$\left. - \sum_{i=1}^{nBranches} \beta_i(x_k[n] \cdot \cos(\vartheta_i) - H\{x_k[n]\} \cdot \sin(\vartheta_i)) \right)^2$$

Instead of a typical iterative approach with a designated number of epochs, or complete training set passes, a variable dynamically determined mechanism of convergence was integrated. With this approach, the initial loss was calculated and then multiplied by a specified convergence factor denoting the percent reduction from initial loss needed to be achieved before adequate convergence is concluded.

As is imposed by gradient descent type algorithms, all the model weights are updated according to loss function partial derivative, $\frac{\partial L}{\partial}$, and learning rate, η. η was empirically determined and appeared to produce the fastest convergence without overshoot and accompanying divergence given a value of 0.01. As the number partial derivatives necessary is a function of the overall number of branches in the ASC, these too are selected during initialization. For this reason, a complete list of calculated partial derivative cannot be obtained. However, a generalized form for each branch *i*'s attenuation and phase partial derivatives is

$$\frac{\partial L}{\partial \beta_i} = \left(SI - \widehat{SI}\right) \cdot (x[n] \cos(\vartheta_i) - H\{x[n]\} \sin(\vartheta_i))$$

$$\frac{\partial L}{\partial \vartheta_i} = \left(SI - \widehat{SI}\right) \cdot (\beta_i x[n] \sin(\vartheta_i) - \beta_i H\{x[n]\} \cos(\vartheta_i))$$

Now, branch weights are updated according to:

$$\beta_i^{n+1} = \beta_i^n + \eta \frac{\partial L}{\partial \beta_i^n}$$

$$\vartheta_i^{n+1} = \vartheta_i^n + \eta \frac{\partial L}{\partial \vartheta_i^n}$$

At the conclusion of training, the converged values are passed to their respective ASC branch to be set and provide future SI channel emulation.

At this point in the simulation the remaining non-training samples are transmitted through the actual prescribed SI channel and received in the Rx path. In a simultaneous manner, these non-training samples are "tapped" off of their transmission path and passed through the ASC branches where the anticipated SI is determined. The actual SI and estimated SI are then deconstructively added together, with their difference reflective of any residual SI remaining in the system post analog cancellation.

After cancellation the primary metrics used to quantify overall system performance are the actual decibel ratio of SIC and the maximum power from the most egregious IMD envelope. Figure 22 displays a juxtaposition of two rendered PSDs from a 2.4GHz 512 QAM signal. On the top, a spectrum is displayed with overtly discernable third-order distortion products resulting in a spurious band protruding from the noise floor. Conversely, on the bottom a well-defined QAM envelope is shown free off any apparent spurs or distortions. Note that the IMD affected PSD on the top was rendered with a purposely reduced thermal noise to help emphasis the characteristics of the distortion bands.

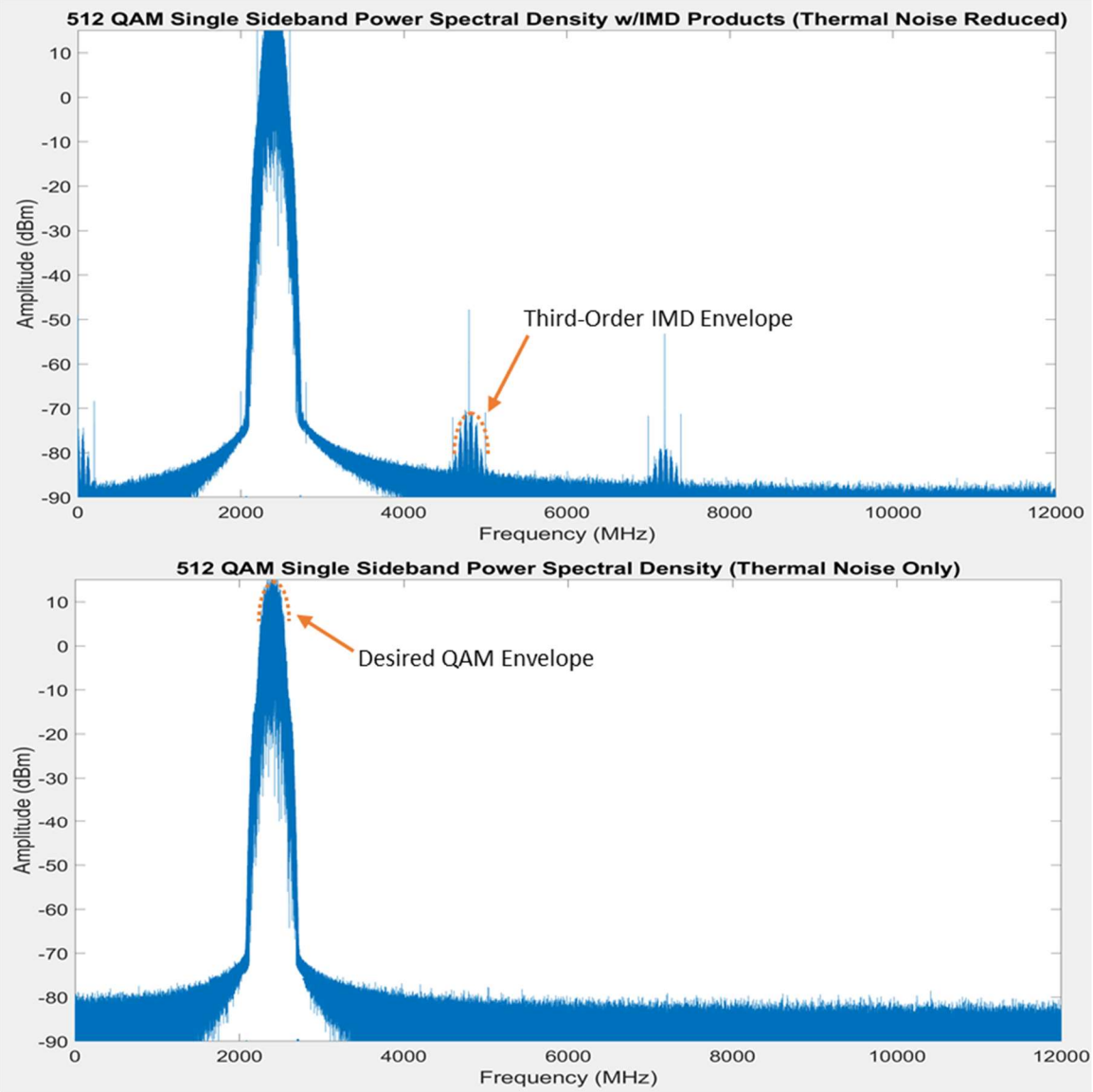

Figure 22 2.4GHz 512QAM PSD Profile with Nonlinear Distortion (Reduced AWGN)

An averaged moving-mean decibel ratio of the achieved SIC level is adopted as a means to provide for a single comprehensive measure the transceiver's SIC performance. This value is calculated by first subtracting the sample-by-sample level of estimated SI from the ASC from the initial received SI. Effectively, this gives an impression of the residual SI remaining in the transceiver post ASC. To render this level relative to the initially received SI's level, it is then divided by the received SI and converted to decibels. Mathematically this is summarized as

$$SIC_{dBm} = 10\log_{10}\left(\frac{SI - \widehat{SI}}{0.001 \cdot SI}\right)$$

From there the calculated windowed moving mean is determined to smooth outlying performance and render a better overall system SIC performance. Figure 23 depicts this quantity from a typical simulation. Finally, the average of this moving mean series is calculated, and this value is recorded as the overall SIC level achieved. Encapsulation of performance in this manner helps to connote a more comprehensive reflective picture of performance. Now the overall simulative run is considered completed, pertinent information is logged, and any specified subsequent runs are started back at parameter initialization.

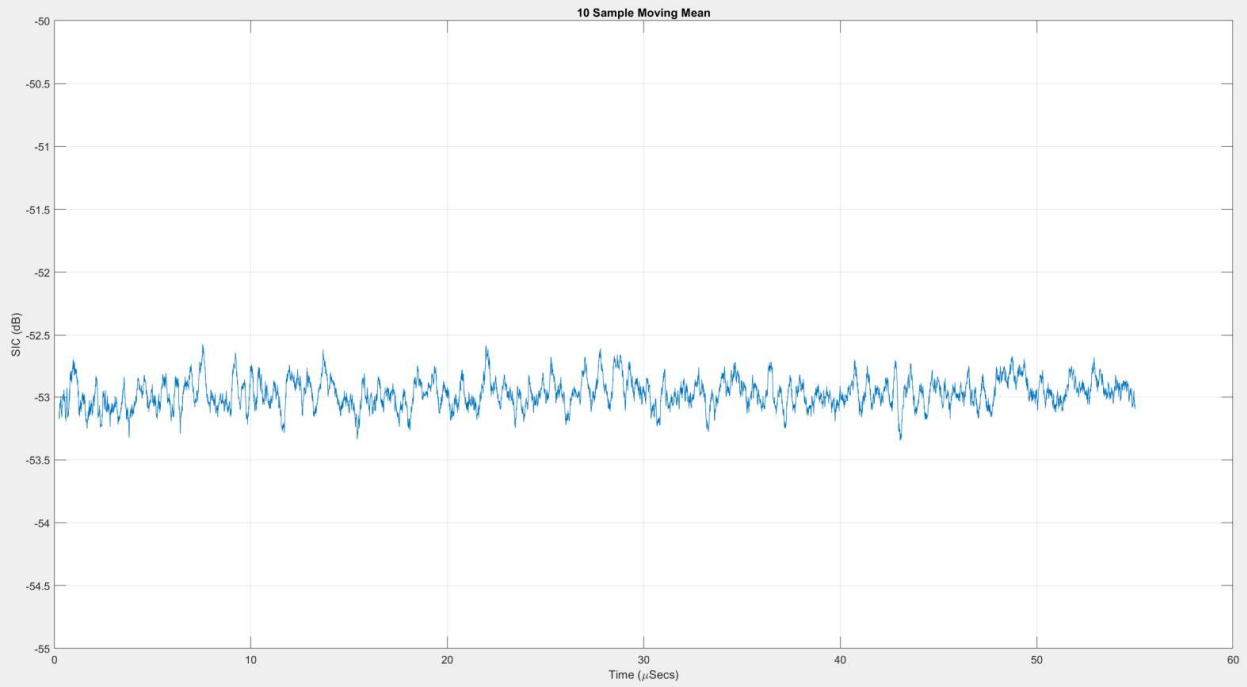

Figure 23 Moving Mean SIC Performance

6. RESULTS

This chapter intends to illustrate the overall simulation results based on methodology outlined in Chapter 5. The section is broken down into two main constituents. In the first, experimental conditions are established, delineating both runtime variant and invariant parameters and a brief description of representation value is given. In the latter, specific simulations will be presented. These include initial preliminary and SIC efficaciousness investigations with both with and without transmission noise incursion. Finally, simulative runs exploring the ramifications of nonlinearities in the ASC cancellation path is presented. Analysis and relevant implications will be given alongside each presentation.

6.1 Variables & Parameters

The simulation parameters outlined in Table 5 denote the conditions under which every simulation was performed. The parameter column dictates the variable name used during simulation with assigned value to their right. A brief description of what each parameter represents is given in the rightmost column. These values represent the default values in the simulation with selection done to represent standard testing conditions and reflection of typical off-the-shelf component parameters.

Table 5 Static Simulation Parameters

PARAMETER	VALUE	UNITS	DESCRIPTION
R	50	Ω	System Characteristic Resistance/Impedance
kB	1.38E-23	Joules/Kelvin	Boltzmann Constant
T	298.15	Kelvin	System Temperature
df	1	Hz	Measurement Bandwidth for Thermal Noise
passBW	100	MHz	Bandwidth of baseband
eta	0.01	-	Gradient Descent Learning Rate
bADC	8	bits	ADC bits rendered per symbol
bDAC	8	bits	DAC bits per symbol rendered
vref	1	V	DAC/ADC Supplied Reference Voltage
vdrop	0.3	V	DAC/ADC Internal Volt Drop
NSD	-162	dBc/Hz	DAC Noise Spectral Density
fCLK	200	MHz	DAC Output Symbol Rate
clkISO	-120	dB	DAC Clock to Output Isolation
fLO	2400	MHz	Oscillator Nominal Frequency
posc	8	dBm	Oscillator Nominal Output Power
pnVar	1.1	-	Phase Noise Variance
Fs	24000	MHz	Sampling Frequency (Determines Time Resolution)
fmin	2300	MHz	BPF 3dB Lower Edge Frequency
fmax	2500	MHz	BPF 3dB Upper Edge Frequency
Av	1		Power Amplifier's Voltage Gain
nsamples	11000	Samples	Pre-Sample & Hold DAC Rendered Samples
convergence	0.0001		Factor Dictates GD Training Extent

The parameters listed in Table 6 represent the variables that either differed throughout a run or did not retain fixed values between simulations. Many of these parameters were chosen at random with a uniform distribution.

Table 6 Dynamic Simulation Parameters

PARAMETER	VALUE	DESCRIPTION
GDType	[1,2]	1 = SGD, 2 = BGD
nLoops	Varies	Number of Times Simulation Commences
nBranches	Random~U(4,30)	Number of Branches in ASC
nPath	Random~U(20,50)	Number of Reflection Paths for SI Channel
modelBeta	Random~U(0,0.1)	Container for Gains of ASC
modelPhase	Random~U(0,2π)	Container for Phases of ASC
nTrain	Varies	Number of Samples Used In Training ASC
batchSize	[10,100,1000]	Mini-BGD Batch Size
NF_Mix	Random~U(2,14)	Noise Figure of Mixer
NF_PA	Random~U(2,5)	Noise Figure of Power Amplifier
DACCoeff	Random~U(0,1)	Parameters Control DAC Nonlinearities
IQCoeff	Random~U(0,1)	Parameters Control I & Q RF Mixer Nonlinearities
QAMCoeff	Random~U(0,1)	Parameters Control QAM RF Mixer Nonlinearities
PACoeff	Random~U(0,1)	Parameters Control Power Amplifier Nonlinearities

6.2 Simulation Runs

6.2.1 Stochastic verses Mini-Batch

During this first set of simulations a total of 200 simulations were run to evaluate the SIC performance of the SGD verses mini-BGD algorithms. Of these 200 runs, 50 were dedicated to SGD with the remaining 150 equipartitioned amongst mini-BGD with batch sizes of 10, 100, and 1000. The number of ASC branches and SI channel paths were held fixed at 4 and 30, respectively. 5000 samples from a total 1,320,001 rendered per simulation were extracted for training phase and amplitude weights of the analog canceller. The convergence factor was held fixed at 0.001%. The overall results from 50 runs of each method are shown in Figure 24 with SGD achieving the highest level of SI cancellation at an average of -60.1 dB. Mini-BGD with a batch size of 1000 (denoted BGD-1000) achieved the worst performance with an average of -29.6 dB, or 50.7% worse than SGD. Additionally, the number of iterations of weight updates to achieve convergence only differed marginally for SGD, BGD-10, and BGD-100. The fastest convergence, albeit the overall worst performer, was BGD-1000 with 3.64% less weight updates required when compared to SGD. Summarized SIC mean and standard deviation, as well as weight update requirements, for all methods can be found in Table 7.

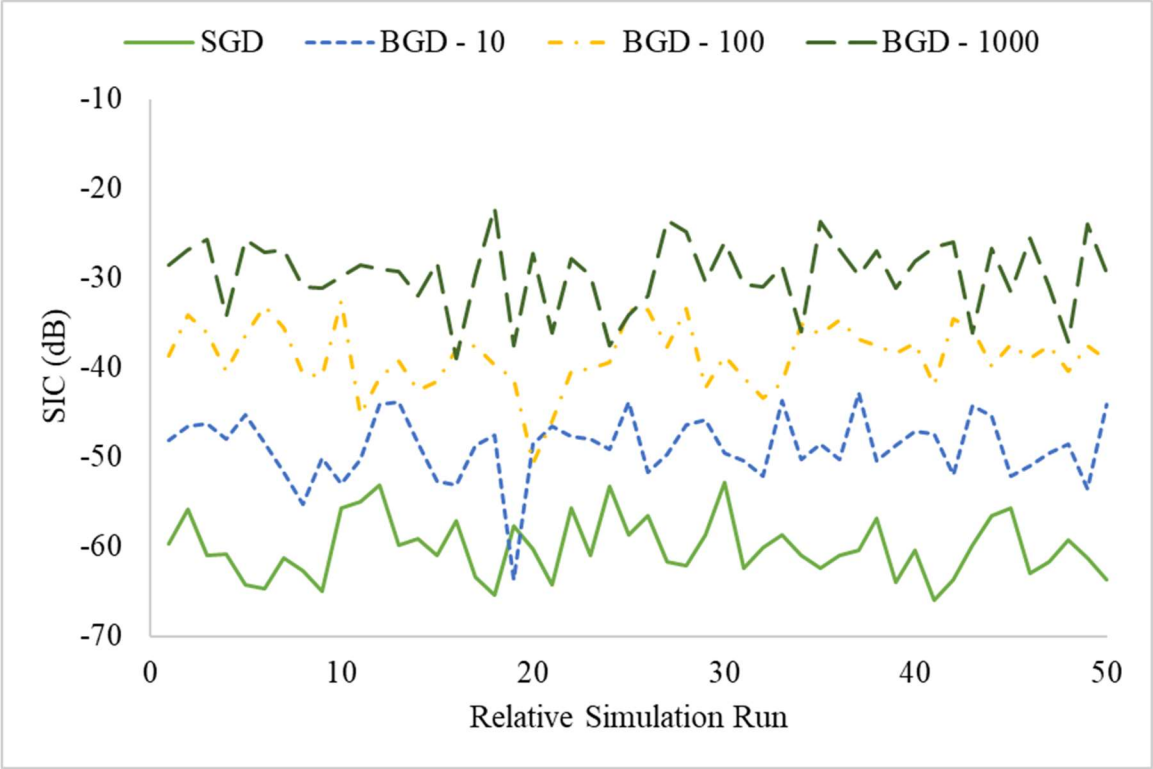

Figure 24 Stochastic verse Mini-Batch Gradient Descent Performance

Table 7 SGD vs. BGD Mean and Standard Deviation

SIC Performance			Iterations of Weight Updates	
Method	Mean (μ)	Deviation (σ)	Mean (μ)	Relative %
SGD	-60.1	3.31	6575	Ref.
BGD-10	-48.9	3.60	6601	0.39%
BGD-100	-38.7	3.53	6749	2.24%
BGD-1000	-29.6	3.97	6503	-3.64%

A subsequent 100 simulation sweep was run with SGD to explore the sensitivity of SIC to the convergence factor. Convergence factor was allowed to vary uniformly from {0.1%, 0.05%, 0.01%, 0.001, 0.0001%, 0.00001%}, while all other parameters were kept equivalent to the prior simulations. Figure 25 shows that an exponential degradation in performance was experienced as the convergence factor increased. As convergence factor was found to directly relate to the achieved mean-squared-error, this was used to give a more absolute impression of performance degradation.

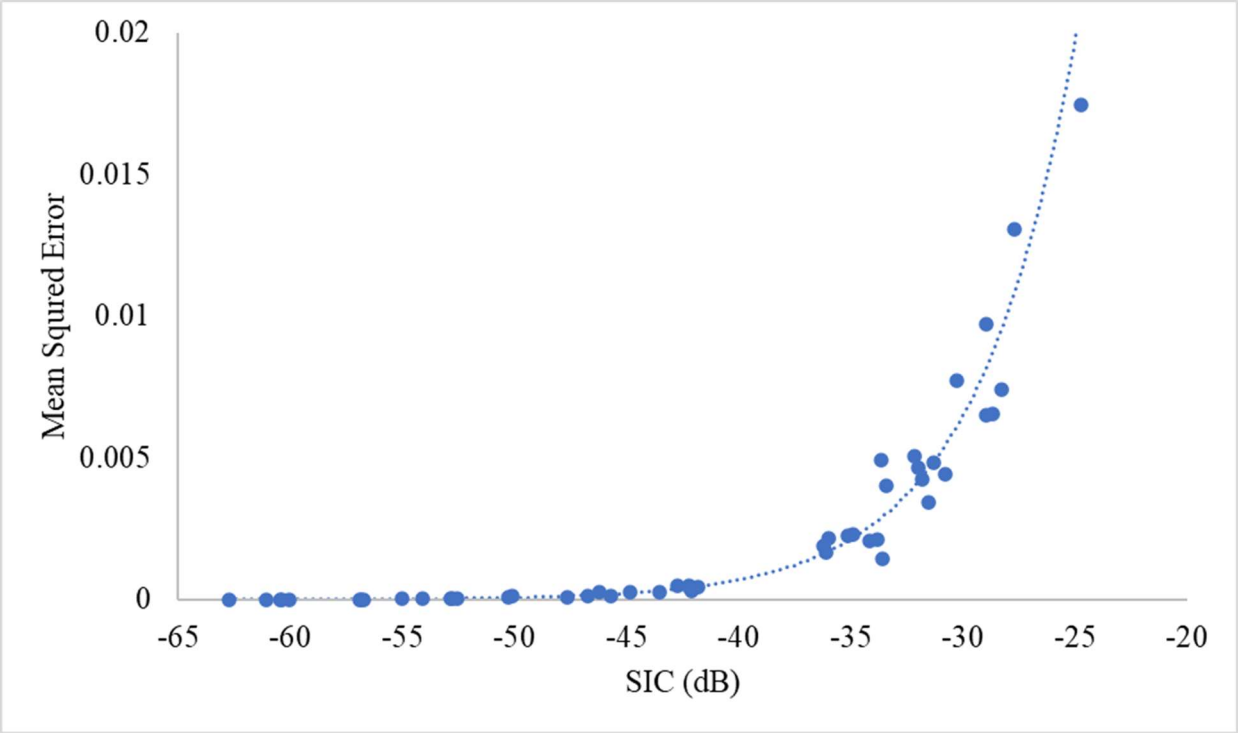

Figure 25 Effect of Convergence Factor (MSE) on SIC Performance

6.2.2 Noise and Nonlinearity

The ability of the analog self-interference canceller to effectively cancel SI corrupted with intermodulation distortion products, harmonics, phase noise, and other systematic gaussian noises was verified over 100 different simulations. Bifurcating these simulations into two 50 run sets allowed for one partition to serve as baseline, with only thermal and a marginal amount of phase noise affecting the system, and the other with varying severities of distortion. All nonlinear distortion coefficients were selected on a uniform distribution U(0,1) and chosen randomly at each simulations initialization. For both runs an SGD algorithm with a convergence factor of 0.0001% was used. Overall sample and training size was held fixed at 1,320,001 and 5,000 samples, respectively, and the number of ASC branches and SI paths at 4 and 30, respectively. Results depicted in Figure 26 show that though the most egregious IMD products ranged in average power from approximately -86 dBm up to -42 dBm, the overall SIC achieved showed no correlation. Baseline simulations produced an IMD product power level with an average of -83 dBm and achieved a mean SIC level of -60.5 dB with a standard deviation of 3.1. The noise varying runs ranged in value from approximately -78 dBm to -42 dBm and achieved a mean SIC level of -63.6

dB with a standard deviation of 3.2. Note that both means were within one standard deviation and least squares linear trendline (dashed grey line) displayed a slope of only 0.12dB.

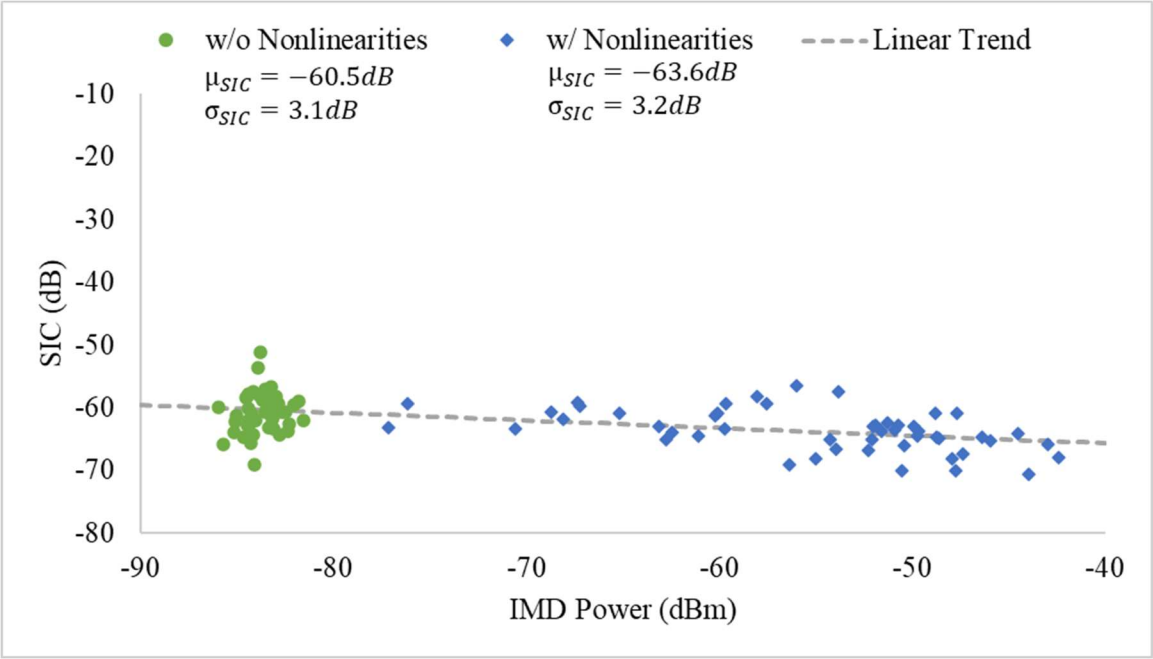

Figure 26 SIC Performance vs. Transmission Chain Distortion

6.2.3 Nonlinearities in Cancellation Hardware

Evaluation of conceivable nonlinearities specific to the analog cancellation hardware or its distinctive electrical path were evaluated by incorporation of a generic 5th order polynomial nonlinear model in-series with its input. 500 distinct runs were performed using SGD with a convergence factor of 0.0001%. Of this collection, the first 100 simulations only thermal and phase noise were generated with a preclusion in variance. The remaining 400 samples allowed all nonlinear coefficients to be chosen randomly from U(0,1). Noise figure for the mixers was chosen randomly from U(4,14) and for the PA from U(2, 6). Again, 5000 training samples were extracted for training out of a used out of a 1,320,001-sample rendered signal. As illustrated in Figure 27, without in-series nonlinearity introduced, the transceiver attained a mean SIC level of -59.2 dB with the power of IMD products measuring a mean and standard deviation of -84.4 dBm and 2.1 dB, respectively. After introduction, SIC displayed a linear degradation in performance that corresponded proportionally with measured IMD powers. SIC performance ranged from

approximately -50 dB to -25 dB as the IMD product power varied from -80 dBm to -10 dBm. Similarly, the mean decreased to -39.3 dB with a standard deviation of 4.4dB

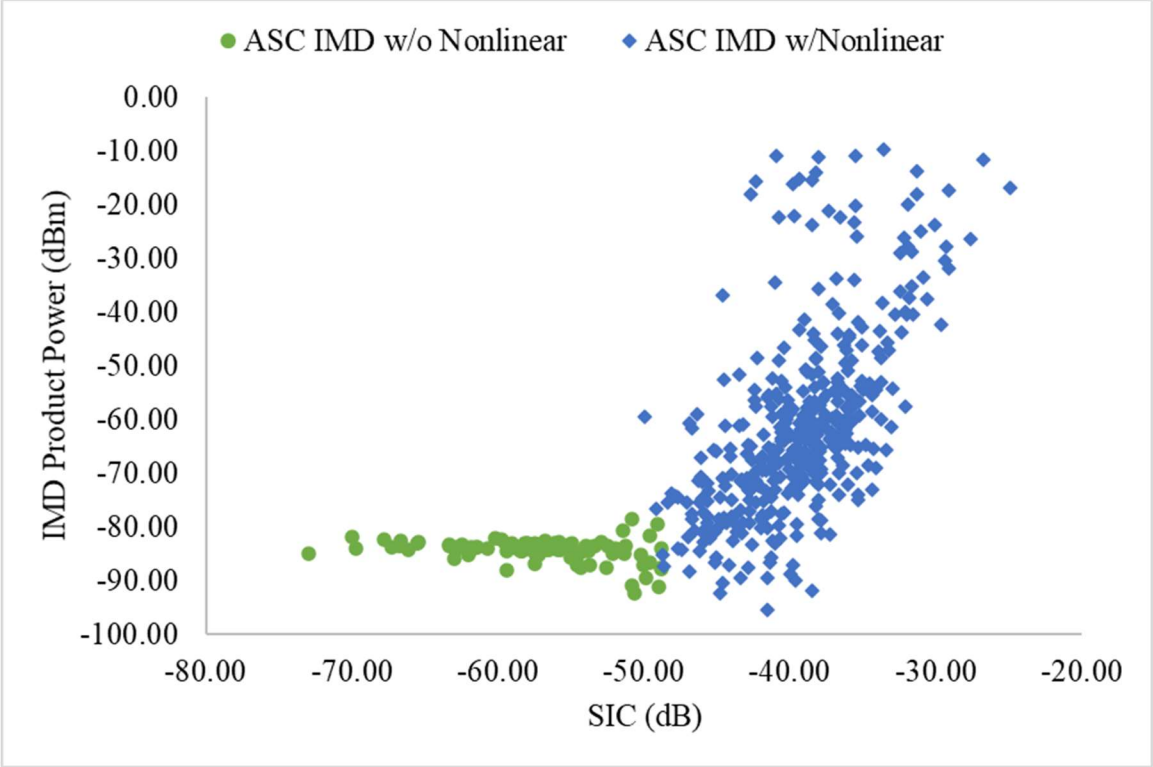

Figure 27 SIC Performance vs ASC Hardware IMD Introduction

The results showing a degradation in overall achieved SIC performance seem to indicate an inclusion of spectral content that, though not acute enough to prevent a relative convergence of ASC's weights, are unable to be ameliorated via mere amplitude or phase adjustment alone. In this sense, even if the phase and amplitude of the received SI is able to be emulated precisely, extraneous content would be imparted on the received signal post deconstructive addition. Therefore, if this were true, as the nonlinearities of the cancellation hardware become substantial enough, an inadvertent auxiliary path for SI introduction is formed. Then, it would be expected that as the spurious content grew, which is in proportion to the ASC's nonlinearities, the overall level of residual SI and reduction in SIC would do so commensurately. In truth, this may be the phenomenon alluded to in the IMD cost-balance relationship of Figure 27.

Mathematically to illustrate this notion consider the reductive case of a simple transmitted sinusoid of the form $a_1 \cos\omega_1 t$. Reflective of the behavior of the simulation, assume our ASC is fed a portion of the energy of this transmitted signal, which can be denoted $a_2 \cos\omega_1 t$. Now if the ASC is assumed to be even a basic second-order polynomial model, then the spectral content of the cancellation signal becomes $a_2 \cos\omega_1 t + a_2^2 \cos^2 \omega_1 t = a_2 \cos\omega_1 t + \frac{a_2^2}{2} \cos 2\omega_1 t + \frac{a_2^2}{2}$. The incoming SI is an amplitude and phase permutation of the original transmitted signal which can be expressed as $a_{SI} \cos(\omega_1 t + \theta_{SI})$. Where a_{SI} and θ_{SI} denote the respective amplitude and phase shift of single path SI component. Similarly, allow our ASC's nonlinear signal to be manipulated by any amplitude and phase weighting a_w and θ_w to render a cancellation signal of the form $a_w a_2 \cos(\omega_1 t + \theta_w) + \frac{a_w a_2^2}{2} \cos(2\omega_1 t + \theta_w) + \frac{a_w a_2^2}{2}$. Then the achievable cancellation of the two signals effectively becomes

$$a_{SI} \cos(\omega_1 t + \theta_{SI}) - \left[a_w a_2 \cos(\omega_1 t + \theta_w) + \frac{a_w a_2^2}{2} \cos(2\omega_1 t + \theta_w) + \frac{a_w a_2^2}{2} \right]$$

To further simplify this for illustrative purposes, suppose that the phases are exactly matched and equal to 0 (i.e., $\theta_{SI} = \theta_w = 0$). As the input signal a_1 is unspecified and could theoretically subsume any value, then arbitrarily a_2 can be selected as 1 for further simplification. Now our possible cancellation becomes

$$a_{SI} \cos(\omega_1 t) - \left[a_w \cos(\omega_1 t) + \frac{a_w}{2} \cos(2\omega_1 t) + \frac{a_w}{2} \right]$$

From this, it can be observed that perfect cancellation only occurs when the bracketed components equal the SI component. Graphically this may be solved for as the intersections of the two components when $\omega_1 t$ subsumes the values from 0 to 2π, and a_w and a_{SI} are allowed to hold various conceivable values. For all but the trivial case of $a_w = a_{SI} = 0$ it can be observed that at most, four specific points occur where perfect cancellation could be achieved. At these very distinct instances of time SIC degradation would not be expected to occur. However, for every other instance SIC performance experiences degradation due to imparted nonlinear components.

7. DISCUSSION AND FUTURE WORK

Though the premise of in-band full duplexing has been demonstrated in numerous theoretical simulations and one-off ad-hoc prototypes, to the best of my knowledge, a production viable in-band full duplex transceiver has never been achieved. Conventionally, a means to achieve sufficient transmitter-to-receiver isolation from a disproportionately larger offender has barred any such developments. Nevertheless, recent advances in self-interference cancellation have shown that techniques, when properly implemented, do provide means to counter this egregious offender. Initial efforts must focus on prevention and suppression on interference in the propagation domain by high isolation duplexers, circulators, and antenna-nulling methods. As demonstrated in this paper, the next and first line of defense against Tx-Rx coupled interference becomes the responsibility of analog domain cancellation. Overall transceiver performance is paramount to its effectiveness. Subsequent digital domain techniques exploiting the transceiver's foreknowledge may aid, but not supplant this cancellation, and is best left for residual self-interference cleanup.

In this paper it was shown through a modeled IBFD transceiver architecture, that by the inclusion of a variable amplitude and phase branched canceller, sampling the signal after primary distortions were incurred, up to and beyond 60dB of aggregate cancellation could be achieved. Furthermore, no auxiliary circuitry or additional channel modeling had to be done in order to account for nonlinearities. However as discovered, this ability may very well be dependent on the actual nonlinearities of the cancellation hardware itself. Significant incursion of canceller nonlinearities, if accounted for with prudent consideration, may very well limit the efficacy of the overall cancellation. Furthermore, as this technique dictates a signal be sampled or tapped as close to final transmission as possible, the likelihood of encountering higher power signals is significant. Primary approaches to sampling these high-power signals are to use baluns or directional couplers. Devices of which are notorious for IMD production at higher powers. Beyond that, nonlinearities are almost certain to become a function of any device that aims to manipulate signal phase, delay, and magnitude. Though the absolute degree to which, and affecting severity, are still yet to be determined.

Conceivably future work could focus on attempts to build an IBFD transceiver and characterizing nonlinearities in the cancellation path by attempting in-series inclusion of nonlinear devices (e.g., RF amplifiers, diode bridges, isolation transformers, etc.). During this thesis, significant time was spent attempting build an IBFD transceiver testbed around Ettus's USRP-B200mini software defined radio with Kumu Networks KU10405 four branch interference cancellation board. Despite a separate antenna structure, it was discovered that substantial leakage, much of the kind discussed in this paper, existed on the B200mini even prior to transmission which precluded any attempts at same frequency reception. An additional module was not able to be procured and with funding depleted efforts had to be abandoned.

Additionally, discretion should be exercised when considering the absoluteness of reported value veracity of this model. By definition a model is an estimated depiction of an observed occurrence. Therefore, inherently any model will have assumptions and compromises that affect its overall accuracy. For example, though this model did allow for any possible phase or magnitude variation in a stochastic fashion when generating the SI channel, it did not account for an actual propagation time delay. Limited to direct leakage and antenna mismatch self-interference, then this time delay may very well be negligible given the signal's propagation velocity. However, as reflection paths become more substantial (e.g., outdoor environments, larger arenas, etc.) the overall estimated SI channel may fail to accurate reflect conditions. Future work on characterizing propagation patterns of 2.4GHz and 5GHz signals in a variety of settings would likely bolster the ability of the model to estimate and incorporate delayed channel characteristics.

Other future work that may prove beneficial and advantageous includes expanding this model to incorporate techniques of digital cancellation. As it is widely known the effectiveness of digital cancellation is directly dependent on the overall analog cancellation, this would allow a better understanding of the two's coupled interaction. Possibly revealing which nonlinear or noise sources were most difficult to mitigate, and any posing constraints to digital domain efficacy. Additionally, this model's underlying assumption was that of a single-input and single-output antenna. As the purported benefits of multiple-input multiple-output systems are gaining wind, expanding the study to involve these scenarios may shed light on questions of the viability of the tapped-branch architecture relative to anticipated increased circuit complexity. One final

suggestion for future work is the development and incorporation of a dynamic SI channel model. This model assumed that once the simulation began, no changes would occur to the channel model till subsequent executions. As this is unrealistic for any mobile wireless device, developing a dynamic model would allow for evaluation of the overall deleterious effect this renders, leading to possibility of allusions to amelioration methods.

8. CONCLUSION

The theoretical doubling of spectral efficiency has placated a historical hesitancy to develop in-band full duplexing technology. A shift in perspective has modern communication systems viewing this once untenable notion as both theoretically and physically viable. Self-interference, being the apparent frustrating element to IBFD realization, is now considered a tractable phenomenon with various mitigation means across the passive suppression, active analog, and more contemporary digital domains.

In this thesis the historical background of IBFD was established and presented beginning from its earliest *fons et origo* in 1940's RADAR applications, to the current efforts and attempts at full-duplex realization. Following this historical overview, the domains of self-interference cancellation were discussed with select techniques emphasized and highlighted. Bolstering this information, a foundational knowledge was presented delineating some of the most salient and germane concepts for the design of IBFD transceiver architecture. A subsequent exemplar transceiver model was then outlined. Allowing for an augmented information capacity density, it was designed around quadrature amplitude modulation. To theoretically evaluate this model and efficacy of analog self-interference cancellation, a noise inclusive simulative model was developed and written in MATLAB. Initial model exploration showed the superiority of stochastic gradient descent to mini-batch gradient descent in a 5000-sample training of canceller weights. On average, SGD attained -60.1 dB of SIC over a 50-simulation span. Additionally, it experienced on average approximately 12 dB, 22 dB, and 30 dB more cancellation than mini-BGD with batch sizes of 10, 100, and 1000, respectively. When computation time was quantified by the total number of required ASC weight adjustments to meet convergence, the time required was comparable between SGD, BGD-10, and BGD-100. BGD-1000 proved to be marginally faster in convergence by 3.64%, albeit at a significant degradation in SIC performance. The overall ability in which the canceller could mimic the received interference was found to correlate to the training's achieved mean-square-error in an exponential manner. Initially, the degree of achievable SIC was found not to be influenced by degree of nonlinear distortion and noise introduction in the transmission chain. In effect, this verified that by sampling the transmission signal post Tx noise incursion, the aforementioned canceller inherently subsumed nonlinearities and distortions in its cancellation

effect. However, as the model was expanded to include possible nonlinearities conceivably introduced in the cancellation hardware itself, the overall efficacy degraded in a commensurate manner. By allowing the nonlinear distortion in the cancellation hardware itself to vary from a baseline -80 dBm to -10dBm, the average provided cancellation experienced a degradation from an initial mean of 59.2 dB to a reduction, at worse of 25 dB.

These preliminary findings allude to the necessity to incorporate careful consideration into the hardware implementation of analog self-interference cancellation. Failure to do so may inherently lead to underperformance and inimical self-interference cancellation due to an intrinsic nonlinearity in initial IBFD transceiver designs. Given these results, further investigation is prudent and warranted on these matters.

APPENDIX A. MATHEMATICAL DERIVATIONS

Phase Noise Weighted Average Derivation

Given a section of a phase noise profile that assumes the relationship

$$dBc = m\log_{10} f + b$$

Where at a frequency f the magnitude is dBc, slope m, and theoretical intercept is b. If the end-point specifications are given as points (f_1, dBc_1) and (f_2, dBc_2) then

$$m = \frac{dBc_2 - dBc_1}{\log f_2 - \log f_1}$$

and

$$b = dBc_2 - m\log f_2$$

The average value of this region is found from

$$dBc_{avg} = \frac{1}{f_2 - f_1}\int_{f1}^{f2} m\log_{10} f + b\, df$$

$$= \frac{m/ln10}{f_2 - f_1}\int_{f1}^{f2} lnf\, df + \frac{b}{f_2 - f_1}\int_{f1}^{f2} df$$

$$= \frac{m/ln10}{f_2 - f_1}(flnf - f)|_{f1}^{f2} + \frac{b}{f_2 - f_1}(f)|_{f1}^{f2}$$

$$dBc_{avg} = \frac{\frac{m}{ln10}}{f_2 - f_1}(f_2 lnf_2 - f_1 lnf_1 - f_2 + f_1) + b$$

Which when multiplied by the bandwidth (i.e., $f_2 - f_1$) results in

$$\frac{m}{ln10}(f_2 lnf_2 - f_1 lnf_1 - f_2 + f_1) + b(f_2 - f_1) \equiv y_1$$

Now suppose a function exists holding a constant value c between frequency endpoints f_2 and f_1. The total area under this curve becomes

$$y_2 = \int_{f1}^{f2} c\, df$$

$$= cf|_{f1}^{f2} = c(f_2 - f_1)$$

Then if c is set to the average value of the phase noise section y_2 becomes

$$y_2 = \left(\frac{\frac{m}{ln10}}{f_2 - f_1} (f_2 lnf_2 - f_1 lnf_1 - f_2 + f_1) + b \right) \cdot (f_2 - f_1)$$

$$= \frac{m}{ln10} (f_2 lnf_2 - f_1 lnf_1 - f_2 + f_1) + b(f_2 - f_1)$$

$$\therefore y_1 = y_2$$

Therefore, the equivalence has been shown between our original straight line phase noise approximation and our bandwidth scaled average constant value section.

Hilbert Transform Phase Shift Incorporation

Suppose an original signal composed of an infinite series of sinusoidal harmonics

$$x_1(t) = \sum_{i=1}^{\infty} a_i cos\omega_i t$$

Now suppose that some desired amplitude scaling β and phase shift θ is imparted to every component in this signal.

$$x_2(t) = \beta \sum_{i=1}^{\infty} a_i \cos(\omega_i t + \theta)$$

Using trigonometric identity $\cos(\alpha + \gamma) = \cos\alpha cos\gamma - sin\alpha sin\gamma$ renders the above as

$$x_2(t) = \beta \sum_{i=1}^{\infty} a_i \cos\omega_i t cos\theta - a_i \sin\omega_i t sin\theta$$

Expanding and factoring this series results in

$$x_2(t) = \beta cos\theta \sum_{i=1}^{\infty} a_i \cos\omega_i t - \beta sin\theta \sum_{i=1}^{\infty} a_i sin\omega_i t$$

$$\therefore x_2(t) = \beta cos\theta \cdot x_1(t) - \beta sin\theta \cdot H\{x_1(t)\}$$

When compared with our original aggregate signal $x_1(t)$ and its Hilbert transform $H\{x_1(t)\}$ the equivalence for amplitude and phase inclusion by means of the Hilbert transform is seen.

APPENDIX B. SIMULATION CODE

```
%% SIMULATION PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nLoops = 50; %# of Simulations to Run
outputData = zeros(nLoops,30);
outputData2 = zeros(nLoops,14);
for loop = 1:nLoops
    GDType = 1;
    ASCCoeff =  [0,0,0,0];
    if loop<51  %Sets # Runs Excluding Noise
        DACCoeff = [1,0,0];
%Coefficient Matrix for DAC RF Mixing
        IQCoeff = [1,0,0];
%Coefficient Matrix for I/Q with OSC Mixing
        QAMCoeff = [1,0,0];
%Coefficient Matrix for QAM(I+Q) Mixing
        PACoeff = [1,0,0,0];
    else
        DACCoeff = [1,rand(),rand()];
%Coefficient Matrix for DAC RF Mixing
        IQCoeff = [1,rand(),rand()];
%Coefficient Matrix for I/Q with OSC Mixing
        QAMCoeff = [1,rand(),rand()];
%Coefficient Matrix for QAM(I+Q) Mixing
        PACoeff = [1,rand(),rand(),rand()];
        %ASCCoeff =  [rand(),rand(),rand(),rand()]/10;
    end
    %%Thermal Noise Parameters
    R = 50;                 %System Characteristic Impedance
    kB = 1.380649*10^-23;   %Boltzmann Constant (J/K)
    T = 298.15;             %System Temperature (Kelvin); 25C = 77F = 298.15K
    df = 1;                 %Noise/Measurement BW (Hz)
    %%Self-Interference Parameters
    nBranches = 4;%randi([4 50]);  %# ASC branches/taps Randomized Unf. Dist.
    nPath = 30;%randi([4 50]);     %# SI Paths - Randomized Unf. Dist.
    modelBeta = rand(1,nBranches)/10; %Initial SIC Gain Weights
    modelPhase = rand(1,nBranches);   %Initial Phase Weights
    %%Training Parameters
    %GDType = 1;            %SGD = 1, BGD=2
    %batchSize = 10;        %Mini Batch Size (BGD Only)
    passBW = 100;           %Baseband Bandwidth
    nTrain = 5000;          %# of Training Samples
    eta = 0.01;             %Learning Rate
    convergence = randsample([1 10 100 500 1000],1)/1000000;
    %%DAC/ADC Parameters
    bADC = 8;               %ADC Bits
    bDAC = 8;               %DAC Bits
    vref = 1;               %DAC/ADC Reference Voltage
    vdrop = 0.3;            %DAC/ADC Internal Voltage Drop
    NSD = -162;             %DAC Noise Spectral Density
    fCLK = 2*passBW;        %Symbol output Rate (MHz)
    clkIso = -60;           %Set Clock Isolation from Output (dB)
    %%Oscillator & Filtration Parameters
```

```
fLO = 2400;             %Oscillator Frequency (MHz)
Fs = 10*fLO;            %Sampling Frequency (MHz)
fmin = fLO-passBW;      %Minimum frequency to filter (MHz)
fmax = fLO+passBW;      %Maximum frequency to filter (MHz)
posc = 8;               %Oscillator power (dBm)
pnVar = 2.1;            %Norm Rand Dist Sigma (dBc)
%%Noise Figures
NF_Mix = 4.4;           %Noise Figure
NF_PA = 3.4;            %Power Amplifier NF
Av = 1;                 %Power Amplifier Gain, Max = 5.62
%%Sequence  Parameters
nsamples = 11000;       %DAC Rendered #Samples
testTime = [0:1/fCLK:1/fCLK*(nsamples-1)];     %Time Array of Simulation
testSeq = false;        %True = Test Sequence Input, False = PRBS Input


%% JOHNSON-NYQUIST (THERMAL) NOISE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calculate thermal noise for at given bandwidth % system temperature
vThermal = sqrt(4*kB*R*T*df); %Thermal Voltage with df measurement window

%% GENERATE # OF SI PATHS, LOSS, & PHASE SHIFT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pathLoss = zeros(1, nPath);
pathLoss(1,1) = rand(1)/4;              %Initial path loss U(0,0.25)
pathPhase = rand([1,nPath])*2*pi;       %Path Phase Shift U(0,2*pi)
%Path Attenuations - Derived from Randomized Uniform Distribution where
%Sum of all SI paths is < 0.5
pathSum=0;
for p = 2:nPath
    pathSum = pathSum + pathLoss(1,p-1);
    pathLoss(1, p) = (0.5-pathSum)*rand()/2;
end
%Ensures largest delay is associated with largest loss.
pathLoss = sort(pathLoss, 'descend');
pathPhase = sort(pathPhase, 'ascend');

%% GENERATE I & Q BITSTREAMS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if testSeq
    %TEST SEQUENCE PATTERN
    testI = 1*cos(2*pi*1.*testTime)+1;
    testQ = 1*cos(2*pi*1.*testTime)+1;
    streamI = ADC(testI, 1/(2*bADC), vref, vdrop, bADC, 0, 0, 0, 0);
    streamQ = ADC(testQ, 1/(2*bADC), vref, vdrop, bADC, 0, 0, 0, 0);
else
    %PRBS SEQUENCE PATTERN
    streamI = prbs(31, nsamples*bDAC);
    streamQ = prbs(31, nsamples*bDAC);
end

%% DAC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Idac = DAC(streamI(1,:), fCLK, vref, vdrop, bDAC, 0.001);
Qdac = DAC(streamQ(1,:), fCLK, vref, vdrop, bDAC, 0.001);

%IMD & SFDR Incorporation
```

```
sqr = ones(size(Idac(1,:)))*10^(clkIso/20);   %Clock rising edge
DAC_clk = [sqr;sqr;Idac(3,:);Idac(4,:)];            %Required format
IdacTemp = RF_Mixer(DAC_clk, Idac, DACCoeff, -1, -1);
QdacTemp = RF_Mixer(DAC_clk, Qdac, DACCoeff, -1, -1);

%NSD Incorporation
Idac(2,:) = awgn(IdacTemp(2,:), abs(10*log10(10^(NSD/10)/((vref-
vdrop)^2/R))), 'measured');
Qdac(2,:)     =     awgn(QdacTemp(2,:),     abs(10*log10(10^(NSD/10)/((vref-
vdrop)^2/R))), 'measured');
ttl_time = Idac(3,size(Idac,2));      %Total time of symbols (usecs)

%% OSCILLATOR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
osc = OSCILLATOR(fLO, Fs, ttl_time, R, posc, pnVar);

%% SYNCHRONIZER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sample-&- Hold Circuit Synchronizes to match oscillators timing
%Necessary as all signals are treated as discrete time events
Isynch = Synchronizer(Idac, osc);
Qsynch = Synchronizer(Qdac, osc);

%% IQ MIXER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Ideal Mixing = x[n]*cos(wLO*t)
%   x[n] freq = [wLow, wHigh]
%   LO freq = wLO
%Ideal IQ mixer freq components:
%   band1 <= f <= band2
%       wLO-wHigh <= band1 <= wLO-wLow
%       wLO+wLow <= band2 <= wLO+wHigh
Imix = IQ_Mixer(Isynch, osc, 0, IQCoeff, NF_Mix);   %Mixes I with LO Cos
Qmix = IQ_Mixer(Qsynch, osc, 90, IQCoeff, NF_Mix);  %Mixes Q with LO Sin

%Filter for upconverted frequency spectrum
Iflt = BPF(Imix,'fir', Fs, fmin, fmax, 0.9, 60);
Qflt = BPF(Qmix,'fir', Fs, fmin, fmax, 0.9, 60);

%% QAM MIXER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
QAMunf = RF_Mixer(Iflt, Qflt, QAMCoeff,-1,-1);

%% POWER AMPLIFIER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
QAM_PA = PA(QAMunf, Av, PACoeff, NF_PA);

%Filter from fLO-BW/2 to fLO+BW/2
QAM = BPF(QAMunf,'fir', Fs, fmin, fmax, 0.9, 60);
%% NOISE FIGURE & THERMAL VOLTAGE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
vTout = sqrt(10^(NF_Mix/10)*10^(NF_PA/10)*10^(NF_Mix/10)*vThermal^2);
vTx = sqrt(mean(QAM(1,:).^2));
snrOut = 10*log10(vTx/vTout);
QAM(2,:) = awgn(QAM(2,:),snrOut,'measured');

%% SNR, Tx, IMD PWR CALCULATION
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[pwrTx, pwrIMD, SNRTx] = SNRCalc(QAM(2,:), Fs);

%Extract QAM Symbols for Training
trainTx = QAM(2,200:200+nTrain);
trainTime = QAM(3,200:200+nTrain);

%Extract Remaining QAM Symbols for Transmission
QAMTx = [QAM(1,nTrain+1:length(QAM(1,:)));...
    QAM(2,nTrain+1:length(QAM(2,:)));...
    QAM(3,nTrain+1:length(QAM(3,:)));...
    QAM(4,nTrain+1:length(QAM(4,:)))];

%% GENERATE SELF-INTERFERENCE SIGNALS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hnTrain = imag(hilbert(trainTx));       %Training Hilbert Transform
hnReceive = imag(hilbert(QAMTx(2,:)));  %Transmission Hilbert Transform

trainPath = zeros([nPath,length(trainTx)]);
receivePath = zeros([nPath,length(QAMTx(2,:))]);
for p = 1:nPath
        trainPath(p,:) = pathLoss(p)*( trainTx*cos(pathPhase(p))-
              hnTrain*sin(pathPhase(p)) );
        receivePath(p,:) = pathLoss(p)*( QAMTx(2,:)*cos(pathPhase(p))-
              hnReceive*sin(pathPhase(p)) );
end
trainSI = sum(trainPath);       %Training Through Actual SI Channel
actualSI = sum(receivePath);    %Transmitted Through Actual SI Channel

%% ASC ML TRAINING
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
trainTx = ASCNL(trainTx,ASCCoeff);
if GDType == 1
    [modelBeta, modelPhase, MSE, epochs] = MLSGD(trainTx, trainSI,
        trainTime, eta, convergence, modelBeta, modelPhase);
elseif GDType == 2
    [modelBeta, modelPhase, MSE, epochs] = MLBGD(trainTx, trainSI,
        trainTime, eta, batchSize, convergence, modelBeta, modelPhase);
end

%% ASC SI ESTIMATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SI_model = zeros([4,length(QAMTx(2,:))]);
for p = 1:nBranches
        SI_model(p,:) = modelBeta(p)*( QAMTx(2,:)*cos(modelPhase(p))-
              hnReceive*sin(modelPhase(p)) );
end
modelSI = sum(SI_model);    %Modeled SI through ASC
modelSI = ASCNL(modelSI,ASCCoeff);

%% MEASUREMENTS/CALCULATIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[pwrSI, pwrSIIMD, SNRTx3] = SNRCalc(actualSI, Fs);
[pwrASC, pwrASCNL, SNRTx2] = SNRCalc(modelSI, Fs);
SIC = abs((actualSI-modelSI)./actualSI);
SICdB = movmean(20*log10(SIC),nTrain);
SICavg = mean(SICdB);
```

```
    %% RECORD RUN DATA
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    outputData(loop,:) = [nPath, nBranches, GDType, length(QAM(2,:)), nTrain,
      convergence*100, epochs, MSE(1), MSE(2), SICavg, pwrTx, pwrIMD, SNRTx,
      DACCoeff, IQCoeff, QAMCoeff, PACoeff, ASCCoeff];
    outputData2(loop,:) = [nPath, nBranches, GDType, length(QAM(2,:)),
      nTrain, convergence*100, epochs, SICavg, MSE(2), pwrTx,
      pwrIMD,pwrSI,pwrSIIMD, pwrASCNL];
end
%%  WRITE DATA TO FILE
%%%%%%%%%%%%%%%%%%%%%%
%fileID = 'SimData.xlsx';
%writematrix(outputData,fileID,'WriteMode','append');
fileID = 'ASCNLData.xlsx';
writematrix(outputData2,fileID,'WriteMode','append');

%Input sequence assumed as positive voltage
%Output reads MSB to LSB
%Output is 4 Row Matrix with:
%   Row 1 = Ideal bit sequence
%   Row 2 = Noisy bit sequence
%   Row 3 = Ideal bit start time
%   Row 4 = Jittery bit start time
function adc_out = ADC(xn, clk, vref, vdrop, ADCbits, jtr, jtr_std, v_mu,
v_sigma)
    vmax = vref-vdrop;
    xnp = xn/vmax;              %Signal now scaled from [0,1]
    xnd = xnp*((2^ADCbits)-1);  %Equivalent decimal value based on # bits
    xnr = round(xnd);           %Quantization error, round to nearest int
    xn_b = de2bi(xnr,ADCbits);  %MATLAB Binary conversion
    xn_b = flip(xn_b, 2);       %Flips elements wrt to dim 2(rows)
    ibit = reshape(xn_b.',1,ADCbits*length(xn)); %Binary array,MSB->LSB
    nbit = ibit;    %Place holder for noisy bit sequence
    ti = [0:length(ibit)-1]*clk;   %Ideal start time of bit out
    tn = ti; %Place holder for jitter inclusive timing
    adc_out = [ibit;nbit;ti;tn];
end

function ASCOut = ASCNL(vi, COEFF)
    %5th Order Harmonic Generator
    vn_secd = COEFF(1,1)*vi.^2;
    vn_thrd = COEFF(1,2)*vi.^3;
    vn_frth = COEFF(1,3)*vi.^4;
    vn_ffth = COEFF(1,4)*vi.^5;
    von = vi + vn_secd + vn_thrd + vn_frth + vn_ffth; %Nonlinear Output
    ASCOut = von;
End

%Input Matrix:
%   Row 1 = Ideal Voltage
%   Row 2 = Noise Voltage
%   Row 3 = Ideal Time
%   Row 4 = Noise Time
function bpf_out = BPF(input,IR, Fs, f_low, f_high, steep, stopbandatt)
    ideal = input(1,:);
    noise = input(2,:);
```

```
    %Catch for NaN
    ideal(isnan(ideal))=0;
    noise(isnan(noise))=0;
    videal            =           bandpass(ideal,[f_low            f_high],Fs,
'ImpulseResponse',IR,'Steepness',steep,'StopbandAttenuation', stopbandatt);
    vnoise            =           bandpass(noise,[f_low            f_high],Fs,
'ImpulseResponse',IR,'Steepness',steep,'StopbandAttenuation', stopbandatt);
    bpf_out = [videal; vnoise; input(3,:); input(4,:)];
end

% MATRIX INPUT:
%   Row 1 = Binary Sequence of 0's & 1's
% MATRIX OUTPUT:
%   Row 1 = Ideal Voltage
%   Row 2 = Noisy Voltage
%   Row 3 = Ideal Time
%   Row 4 = Jittery Time
% PARAMETERS:
%   bin_seq [bits]= Input Binary Sequence
%   clk [Hz] = Sampling clock, dictates output rate of DACs
%   vref [volts] =  Reference voltage
%   vint [volts] = [volts] Internal voltage drop
%   nbits [bits] = # of bits used in conversion, sets resolution(step size)
%   sigma_prc [%] = Standard Deviation of Power Supply Vref Variation as %
%   of nominal

function dac_out = DAC(bin_seq, clk, vref, vdrop, nbits, sigma_prc)
    %% BIN TO DECIMAL CONVERSION
    %Generate a Base 2 Binary Value Array; MSB -> LSB
    n = 0:(nbits-1);
    n = flip(n,2);
    base2 = (2*ones(1,nbits)).^n;
    %Calculate Decimal Value of Each Symbol
    nwords = fix(length(bin_seq)/nbits); %# of words/symbols out of DAC
    base2 = repmat(base2,1,nwords); %Repeat Base 2 for each symbol/word
    dec_seq = bin_seq .* base2; %If bit=1 value remains, if 0 then = 0
    dec_seq = reshape(dec_seq, [nbits,nwords]); %One word per row
    dec_seq = dec_seq.';
    dec_value = sum(dec_seq,2); %Calculate decimal equivalent of each word

    %% SYMBOL VOLTAGE CALCULATION
    %Calculate Noisy Vref
    %Mu = vref, sigma_percent = % of vref as std dev
    vref_noise = normrnd(vref,sigma_prc*vref,[1,nwords]);

    %Calculate Ideal Symbol Voltage
    v_ideal = dec_value * (vref-vdrop)/(2^nbits-1);
    v_ideal = transpose(v_ideal);

    %Calculate Noisy Symbol Voltage
    resolution = (vref_noise-vdrop)/(2^nbits-1);
    v_noise = dec_value.*resolution.';
    v_noise = transpose(v_noise);

    %% SYMBOL TIMING
    %Calculate Ideal Output Times
    t_ideal = (1:nwords)*(1/clk);
```

```
    %Generate Jitter with mu of 0 & sigma = 1/(12clk) then ideal to ideal
    %time for jittery pertubations
    jtr = normrnd(0, 1/(12*clk), [1,length(t_ideal)]);
    t_jtr = t_ideal + jtr;

    %% OUTPUT MATRIX CREATION
    dac_out = [v_ideal;v_noise;t_ideal;t_jtr];
end

%Weakly nonlinear RF Mixer Model
%Model focuses on multiplicative output without additional filtering and
%amplification needed.
%Assumes times are already synchronized between base and carrier
function mixer_out = IQ_Mixer(base, osc, phase, coeffMatrix, NF)

    %Base Voltages
    vbi = base(1,:);
    vbn = base(2,:);

    %Carrier Cos Voltages
    if phase == 0
       vci = osc(1,:);
       vcn = osc(3,:);
    elseif phase == 90
       vci = osc(2,:);
       vcn = osc(4,:);
    else
        return
    end
   t = base(3,:);

    vi = vbi.*vci;    %Ideal input
    vn = vbn.*vcn;    %Noisy input

    %Calculates Ideal & Noisy Cos Outputs with full spectrum
    voi = vi;
    von = coeffMatrix(1,1)*vn + coeffMatrix(1,2)*vn.^2 + coeffMatrix(1,3)*vn.^3;
    mixer_out = [voi; von; t; t];
end

function lpf_out = LPF(input, Fs, f_high, steep, stopbandatt)
    ideal = input(1,:);
    noise = input(2,:);
    videal                                                                  =
lowpass(ideal,f_high,Fs,'ImpulseResponse','fir','Steepness',steep,'StopbandAt
tenuation',stopbandatt);
    vnoise                                                                  =
lowpass(noise,f_high,Fs,'ImpulseResponse','fir','Steepness',steep,'StopbandAt
tenuation',stopbandatt);
    lpf_out = [videal;vnoise;input(3,:);input(4,:)];
end

%Input Form:
%   input = [2xN]:
%       Row 1 = Data
%       Row 2 = Times
```

```
%   eta = (scalar) Learning Rate
%   epochs = (scalar) Complete Iterations through all training data
%   inBeta = [B1, B2, B3, B4] Starting Attenuation Weights
%   inPhases = [P1, P2, P3, P4] Starting Phase Weights
%Output Form:
%   outBeta = [B1, B2, B3, B4] Final Attenuation Weights
%   outPhases = [P1, P2, P3, P4] Final Phase Weights
%   MSE = [Start, Finish] Mean square error
%
% Model's Underpinning Theory
% Given:    y = Bcos(wt+ theta)
%             = Bcos(wt)cos(theta) - Bsin(wt)sin(theta)]
% I.e., Bi = Amplitude Weight of Branch 'i'
%       di = Phase Weight of Branch 'i'
% Each branch will have one beta and one phase
function [outBeta, outPhase, MSE, epochs] = MLBGD(Tx,SI,time, eta, batchSize,
convergePercent, inBeta, inPhase)
    %MSE = zeros(epochs,length(Tx));%Mean Square Error Matrix
    pdB = zeros(1,4);                   %Beta Partial Derivatives
    pdP = zeros(1,4);                   %Phase Partial Derivatives

    hn = imag(hilbert(Tx));   %Hilbert Transformation

    %4 Branch SI Model
    SI_model = inBeta(1)*( (Tx)*cos(inPhase(1))-hn*sin(inPhase(1)) )...
                + inBeta(2)*( (Tx)*cos(inPhase(2))-hn*sin(inPhase(2)) )...
                + inBeta(3)*( (Tx)*cos(inPhase(3))-hn*sin(inPhase(3)) )...
                + inBeta(4)*( (Tx)*cos(inPhase(4))-hn*sin(inPhase(4)) );

    nBatches = fix(length(Tx)/batchSize);

    %Calculate inital MSE and assign to current & previous
    initMSE = sum(( SI - SI_model ).^2);
    curMSE = initMSE;
    prevMSE = 0;
    epochs = 0;
    tempMSE = zeros(1,nBatches);
    %for e = 1:epochs %e = epoch
    %Keep looping till less than % of original MSE
    while( abs(prevMSE-curMSE) > convergePercent*initMSE)
    %inputTx = normrnd(inputTx, 0.1);    %Incorporates Stochasticness in each
iteration
        for b = 1:nBatches %k = current sample
            %Set Partial Derivatives for Gain to 0 at batch start
            pdB(1) = 0;
            pdB(2) = 0;
            pdB(3) = 0;
            pdB(4) = 0;
            %Set Partial Derivatives for Phase to 0 at batch start
            pdP(1) = 0;
            pdP(2) = 0;
            pdP(3) = 0;
            pdP(4) = 0;

            for k = 1+(b-1)*batchSize:((b-1)*batchSize)+batchSize
                %Partial Derivatives for Attenuations
                pdB(1) = pdB(1) + ( SI(1,k) - SI_model(1,k) ) * (
```

```
            (Tx(1,k))*cos(inPhase(1)) - hn(1,k)*sin(inPhase(1)) );
          pdB(2) = pdB(2) + ( SI(1,k) - SI_model(1,k) ) * (
            (Tx(1,k))*cos(inPhase(2)) - hn(1,k)*sin(inPhase(2)) );
          pdB(3) = pdB(3) + ( SI(1,k) - SI_model(1,k) ) * (
            (Tx(1,k))*cos(inPhase(3)) - hn(1,k)*sin(inPhase(3)) );
          pdB(4) = pdB(4) + ( SI(1,k) - SI_model(1,k) ) * (
            (Tx(1,k))*cos(inPhase(4)) - hn(1,k)*sin(inPhase(4)) );
          %Partial Derivatives for Delays
          pdP(1) = pdP(1) + ( SI(1,k) - SI_model(1,k) ) * ( -
            inBeta(1)*(Tx(1,k))*sin(inPhase(1)) -
            inBeta(1)*hn(1,k)*cos(inPhase(1)) );
          pdP(2) = pdP(2) + ( SI(1,k) - SI_model(1,k) ) * ( -
            inBeta(2)*(Tx(1,k))*sin(inPhase(2))                          -
            inBeta(2)*hn(1,k)*cos(inPhase(2)) );
          pdP(3) = pdP(3) + ( SI(1,k) - SI_model(1,k) ) * ( -
            inBeta(3)*(Tx(1,k))*sin(inPhase(3))                          -
            inBeta(3)*hn(1,k)*cos(inPhase(3)) );
          pdP(4) = pdP(4) + ( SI(1,k) - SI_model(1,k) ) * ( -
            inBeta(4)*(Tx(1,k))*sin(inPhase(4))                          -
            inBeta(4)*hn(1,k)*cos(inPhase(4)) );

      end
      % Find Average Gain of Mini-batch for weight update
      pdB(1) = pdB(1)/batchSize;
      pdB(2) = pdB(2)/batchSize;
      pdB(3) = pdB(3)/batchSize;
      pdB(4) = pdB(4)/batchSize;
      % Find Average Phase of Mini-batch for weight update
      pdP(1) = pdP(1)/batchSize;
      pdP(2) = pdP(2)/batchSize;
      pdP(3) = pdP(3)/batchSize;
      pdP(4) = pdP(4)/batchSize;

      %Update Attenuation Weights
      inBeta(1) = inBeta(1) + eta*pdB(1);
      inBeta(2) = inBeta(2) + eta*pdB(2);
      inBeta(3) = inBeta(3) + eta*pdB(3);
      inBeta(4) = inBeta(4) + eta*pdB(4);

      %Update Phase Weights
      inPhase(1) = inPhase(1) + eta*pdP(1);
      inPhase(2) = inPhase(2) + eta*pdP(2);
      inPhase(3) = inPhase(3) + eta*pdP(3);
      inPhase(4) = inPhase(4) + eta*pdP(4);

      %Update Model From New Weights
      SI_model = inBeta(1)*( (Tx)*cos(inPhase(1))-hn*sin(inPhase(1))
      )...
          + inBeta(2)*( (Tx)*cos(inPhase(2))-hn*sin(inPhase(2)) )...
          + inBeta(3)*( (Tx)*cos(inPhase(3))-hn*sin(inPhase(3)) )...
          + inBeta(4)*( (Tx)*cos(inPhase(4))-hn*sin(inPhase(4)) );
      tempMSE(1,b) = sum(( SI - SI_model ).^2);
  end
  %Recalculate Model
  SI_model = inBeta(1)*( (Tx)*cos(inPhase(1))-hn*sin(inPhase(1)) )...
          + inBeta(2)*( (Tx)*cos(inPhase(2))-hn*sin(inPhase(2)) )...
          + inBeta(3)*( (Tx)*cos(inPhase(3))-hn*sin(inPhase(3)) )...
```

```
                + inBeta(4)*( (Tx)*cos(inPhase(4))-hn*sin(inPhase(4)) );

        %Calculate new MSE
        prevMSE = curMSE;
        curMSE = sum(( SI - SI_model ).^2);
        epochs = epochs + 1;    %Track the number of epochs it takes
    end

    outBeta = inBeta;
    outPhase = inPhase;
    MSE = [initMSE, curMSE];
end

%Input Form:
%   input = [2xN]:
%       Row 1 = Data
%       Row 2 = Times
%   eta = (scalar) Learning Rate
%   epochs = (scalar) Complete Iterations through all training data
%   inBeta = [B1, B2, B3, B4] Starting Attenuation Weights
%   inPhases = [P1, P2, P3, P4] Starting Phase Weights
%Output Form:
%   outBeta = [B1, B2, B3, B4] Final Attenuation Weights
%   outPhases = [P1, P2, P3, P4] Final Phase Weights
%   MSE = [Start, Finish] Mean square error
%
% Model's Underpinning Theory
% Given:    y = Bcos(wt+ theta)
%             = Bcos(wt)cos(theta) - Bsin(wt)sin(theta)]
% I.e., Bi = Amplitude Weight of Branch 'i'
%       di = Phase Weight of Branch 'i'
% Each branch will have one beta and one phase
function [outBeta, outPhase, MSE, epochs] = MLSGD(Tx, SI, time, eta,
convergePercent, inBeta, inPhase)

    nBranches = length(inBeta); %# of Branches in ASC
    pdB = zeros(1,nBranches);   %Beta Partial Derivatives
    pdP = zeros(1,nBranches);   %Phase Partial Derivatives

    hn = imag(hilbert(Tx));     %Hilbert Transformation

    %Branch SI Model
    model = zeros(nBranches, length(Tx));
    for b = 1:nBranches
        model(b,:) = inBeta(b)*( (Tx)*cos(inPhase(b))-hn*sin(inPhase(b)) );
    end
    SI_model = sum(model);

    %Calculate inital MSE and assign to current & previous
    initMSE = sum(( SI - SI_model ).^2);
    curMSE = initMSE;
    prevMSE = 0;
    epochs = 0;

    %for e = 1:epochs %e = epoch
    %Keep looping till less than % of original MSE
    while( abs(prevMSE-curMSE) > convergePercent*initMSE)
```

```
        for k = 1:length(Tx) %k = current sample
            %SI_model = inBeta(1)*( (Tx)*cos(inPhase(1))-hn*sin(inPhase(1))
            )...
            + inBeta(2)*( (Tx)*cos(inPhase(2))-hn*sin(inPhase(2)) )...
            + inBeta(3)*( (Tx)*cos(inPhase(3))-hn*sin(inPhase(3)) )...
            + inBeta(4)*( (Tx)*cos(inPhase(4))-hn*sin(inPhase(4)) );

            %Each           branch's           partial           derivatives
            %Update weights, and update model
            for b = 1:nBranches
                pdB(b) = ( SI(1,k) - SI_model(1,k) ) * (
                  (Tx(1,k))*cos(inPhase(b)) - hn(1,k)*sin(inPhase(b)) );
                pdP(b) = ( SI(1,k) - SI_model(1,k) ) * ( -
                  inBeta(b)*(Tx(1,k))*sin(inPhase(b)) -
                  inBeta(b)*hn(1,k)*cos(inPhase(b)) );
                inBeta(b) = inBeta(b) + eta*pdB(b);
                inPhase(b) = inPhase(b) + eta*pdP(b);
                model(b,:) = inBeta(b)*( (Tx)*cos(inPhase(b))-
                  hn*sin(inPhase(b)) );
            end
            SI_model = sum(model); %Aggregate ASC output

        end
        %Recalculate Model
        for b = 1:nBranches
            model(b,:) = inBeta(b)*( (Tx)*cos(inPhase(b))-
                  hn*sin(inPhase(b)));
        end
        SI_model = sum(model);

        %Calculate new MSE
        prevMSE = curMSE;
        curMSE = sum(( SI - SI_model ).^2);
        epochs = epochs + 1;    %Track the number of epochs it takes
    end

    outBeta = inBeta;
    outPhase = inPhase;
    MSE = [initMSE, curMSE];
end


% OUTPUT MATRIX:
%   Row 1 = Ideal cos voltages
%   Row 2 = Ideal sin voltages
%   Row 3 = Phase Noise cos voltages
%   Row 4 = Phase Noise sin voltages
%   Row 5 = Time
% PARAMETERS:
%   fosc = ideal frequency of oscillation
%   Fs = samples per second to render
%   ttlTime = total time of oscillations to calculate
%   Ro = Nominal system resistance
%   pcar_dBm = Oscillation power in dBm
%   sigmadBc = Standard deviation for dBc values

function osc_out = OSCILLATOR(fosc, Fs, ttlTime, Ro, pcar_dBm, sigmadBc)
```

```
time = [0:1/Fs:ttlTime];  %Time array given oscillators timing

%% OSCILLATOR PARAMETERS
%Adapted From Vectron's (Microsemi) OX-305 % VS-701 Phase Noise Specs
dBc = [-70, -100, -125, -145, -160, -169, -170];

f_offset = [10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7];
pcar_W = 10^(pcar_dBm/10)*0.001;
vcar_pk = sqrt(2*pcar_W*Ro);


%% COMPUTATIONALLY SPARING METHOD OF PHASE NOISE CALCULATION
% Shown to produce final time series voltage that differs on average by
% only 0.5uV
f1 = [f_offset(1):f_offset(2)];
f2 = [f_offset(2)+1:f_offset(3)];
fn = zeros(1, length(f_offset)-3); %One frequency for each section except 1st
& 2nd

%Calculates median frequency of sections after 2
for i = 3:length(f_offset)-1
    fn(i-2) = (f_offset(i)+f_offset(i+1))/2;
end

m = zeros(1, length(f_offset)-1); %One slope for each section
b = zeros(1, length(f_offset)-1); %One intercept for each section
%Calculate function for each section
for i = 1:length(f_offset)-1
    m(i) = ( dBc(i+1)-dBc(i) ) /...
        ( log10(f_offset(i+1))- log10(f_offset(i)) );
    b(i) = dBc(i) - m(i)*log10(f_offset(i));
end

%Calculate dBc value for each frequency from straight line approximation
%and incorporate randomness
dBc1 = m(1)*log10(f1)+b(1);                 %Ideal Straight Line Approx
dBc1 = normrnd(dBc1, sigmadBc, [1,length(dBc1)]);
dBc2 = m(2)*log10(f2)+b(2);                 %Ideal Straight Line Approx
dBc2 = normrnd(dBc2, sigmadBc, [1,length(dBc2)]);

%dBcAvg takes average integral function of phase noise band between endpts
%Includes stochastic-ness using normal distribution of mu = avg(dBc) &
%sigma as user set
dBcAvg = zeros(1,length(f_offset)-3);
for i = 3:length(f_offset)-1
    dBcAvg(i-2) = normrnd(...
        m(i)/log(10) *...
        (  f_offset(i+1)*log(f_offset(i+1))  -  f_offset(i)*log(f_offset(i))  +
f_offset(i) - f_offset(i+1) )...
        /( f_offset(i+1)-f_offset(i) )+b(i), sigmadBc, [1,1] );
end

%Calculate Watts from dBc
pnW1 = pcar_W * 10.^(dBc1/10);
pnW2 = pcar_W * 10.^(dBc2/10);
pnWn = pcar_W * 10.^(dBcAvg/10);
```

```
%Calculate Peak Voltage from Watts
pnVpk1 = sqrt(2*pnW1*Ro);
pnVpk2 = sqrt(2*pnW2*Ro);
pnVpkn = sqrt(2*pnWn*Ro);


%Calculate sin(theta) from peak voltage of phase noise
pn_sin1 = zeros(length(f1), length(time));
pn_sin2 = zeros(length(f2), length(time));
pn_sinN = zeros(length(fn), length(time));
% For f1
for i = 1:length(f1)
    pn_sin1(i,:) = pnVpk1(1,i)*sin(2*pi*f1(1,i).*time);
end
% For f2
for i = 1:length(f2)
    pn_sin2(i,:) = pnVpk2(1,i)*sin(2*pi*f2(1,i).*time);
end
% For fn,
for i = 3:length(f_offset)-1
    bwCoeff = (f_offset(i+1)-f_offset(i)); %Weight each voltage by # samples in
range (i.e. bandwidth of section)
    pn_sinN(i-2,:) =  bwCoeff * pnVpkn(1,i-2) * sin(2*pi*fn(1,i-2).*time);
end

%Combine all phase noise voltage sinusoids
pn = sum(pn_sin1);
pn2 = sum(pn_sin2);
pnN = sum(pn_sinN);
%Take arccsin of phase noise voltage to find actual theta(t) phase
%deviations
theta = asin(pn+pn2+pnN);

%% OSCILLATION CALCULATIONS
% Ideal Oscillations
v_cos = vcar_pk * cos(2*pi*fosc.*time);
v_sin = vcar_pk * sin(2*pi*fosc.*time);
% Phase Noise Inclusive Oscillations
vpn_cos = vcar_pk * cos(2*pi*fosc.*time+theta);
vpn_sin = vcar_pk * sin(2*pi*fosc.*time+theta);

%% OUTPUT MATRIX
osc_out = [v_cos;v_sin;vpn_cos;vpn_sin;time];

%% DIRECT METHOD OF PHASE NOISE CALCULATION (VERY COMPUTATIONALLY INTENSIVE!!!)
% f1 = [f_offset(1):f_offset(2)];     %1/f^3
% f2 = [f_offset(2)+1:f_offset(3)];
% f3 = [f_offset(3)+1:f_offset(4)];   %1/f^2
% f4 = [f_offset(4)+1:f_offset(5)];
% f5 = [f_offset(5)+1:f_offset(6)];   %1/f
% f6 = [f_offset(6)+1:f_offset(7)];
% freq = [f1, f2, f3, f4, f5, f6];%+fosc;
%
% % Calculate Straight line approximation for each section of the form:
% %   dBc = m*log(f)+b
% %   m = slope = (dBc2 - dBc1) / ( log10(f2) - log10(f1) )
```

```
% %   b = intercept = dBc2 - m*log10(f2)
% m = zeros(1, length(f_offset)-1);
% b = zeros(1, length(f_offset)-1);
%
% for i = 1:length(f_offset)-1
%     m(i) = ( dBc(i+1)-dBc(i) ) /...
%         ( log10(f_offset(i+1))- log10(f_offset(i)) );
%     b(i) = dBc(i) - m(i)*log10(f_offset(i));
% end
%
% dBc1 = m(1)*log10(f1)+b(1);                 %Ideal Straight Line Approx
% dBc1n = normrnd(dBc1, 1.7, [1,length(dBc1)]);  %With Stochastic Noise w/Normal
Dist with Std Dev = 1
% dBc2 = m(2)*log10(f2)+b(2);
% dBc2n = normrnd(dBc2, 1.7, [1,length(dBc2)]);
% dBc3 = m(3)*log10(f3)+b(3);
% dBc3n = normrnd(dBc3, 1.6, [1,length(dBc3)]);
% dBc4 = m(4)*log10(f4)+b(4);
% dBc4n = normrnd(dBc4, 1.5, [1,length(dBc4)]);
% dBc5 = m(5)*log10(f5)+b(5);
% dBc5n = normrnd(dBc5, 1.4, [1,length(dBc5)]);
% dBc6 = m(6)*log10(f6)+b(6);
% dBc6n = normrnd(dBc6, 1.3, [1,length(dBc6)]);
% dBc = [dBc1, dBc2, dBc3, dBc4, dBc5, dBc6];
% dBcN = [dBc1n, dBc2n, dBc3n, dBc4n, dBc5n, dBc6n];
%
%
% %Convert from Phase Noise from dBc to Watts given that:
% %   dBc = 10*log10(Psig/Pcar)
% %We see that:
% %   Psig = Pcar * 10^(dBc/10)
% pnW = pcar_W * 10.^(dBc/10);
%
% %Convert Phase Noise watts to peak voltage for each frequency:
% %   P = Vrms^2/R
% %   Vrms = sqrt(P*R)
% %   Vpk = Vrms * sqrt(2) = sqrt(2*P*R)
% pnVpk = sqrt(2*pnW*Ro); %Amplitude (Voltage Spectrum)
%
% % Since Discrete PN can be thought of as multiidonous summed sinusoids
% % We can treat each offset frequency as a distinct sinusoidal source
% % with each's respective amplitude rendered above.
% % From the sheer quantity of sources, more efficient matrix multiplication
% % is not possible. Therefore, computationally intensive, but memory sparing,
% % recycling for loops must be used.
% %vpn_cos1 = zeros(1,length(time));  %Time array storing PN equivalent voltage
at each time
% vpn_sin = zeros(1,length(time));
%
% %Very Time Intensive Computation requiring on average >1000secs
% %(>16minutes)
% for t=1:length(time)
%         sinArray = pnVpk.*(sin(2*pi*time(t).*freq));
%         vpn_sin(1,t) = sum(sinArray);
% end
%
% theta1 = asin(vpn_sin);
```

```
%
% % PHASE NOISE INCLUSIVE OSCILLATIONS
% vpn_cos1 = vcar_pk*cos(2*pi*fosc.*time+theta1);
% vpn_sin1 = vcar_pk*sin(2*pi*fosc.*time+theta1);
%
% % IDEAL OSCILLATIONS
% vosc_cos = vcar_pk*cos(2*pi*fosc.*time);
% vosc_sin = vcar_pk*sin(2*pi*fosc.*time);
%
% figure()
% subplot(2,1,1)
% semilogx(freq,dBc, 'LineWidth', 2)
% title('Interpolated Phase Noise Profile')
% xlabel('Frequency Offset (Hz)');
% ylabel('Phase Noise (dBc/Hz)');
% grid on
% subplot(2,1,2)
% semilogx(freq,dBcN)
% title('Interpolated Phase Noise Profile (w/ Random Noise)')
% xlabel('Frequency Offset (Hz)');
% ylabel('Phase Noise (dBc/Hz)');
% grid on

end

%This function computes the nonlinear output of a 5th order polynomial
%derived by Taylor series expansion through a PN junction.
%It is a memoryless model with a coefficient matrix BCDE which may be
%altered to match specific datasheet specifications.
%Primary purpose is modeling for Power Amplifiers
%
%Assumes input matrix in the form of:
%   Row 1 = Ideal voltages (Noise-free)
%   Row 2 = Noisy voltages
%   Row 3 = Ideal Timing (Noise-free)
%   Row 4 = Noisy Timing

function paOut = PA(in, Av, COEFF, NF)

    %Input Voltages
    vi = in(1,:);   %Ideal
    vn = in(2,:);   %Noisy

    %Harmonics
    vn_secd = COEFF(1,1)*in(2,:).^2;
    vn_thrd = COEFF(1,2)*in(2,:).^3;
    vn_frth = COEFF(1,3)*in(2,:).^4;
    vn_ffth = COEFF(1,4)*in(2,:).^5;

    voi = Av*vi;   %Ideal Output
    von = Av*vn + vn_secd + vn_thrd + vn_frth + vn_ffth; %Nonlinear Output

    paOut = [voi;von;in(3,:);in(4,:)];
end

function [dBmTx, dBmIMD, SNRTx] = SNRCalc(timeSeries, Fs)
    %Convert To Frequency Domain via FFT
```

```
    timeFreq = fft(timeSeries);

    %Extract & Scale Voltage Magnitude SSB
    voltMag = abs(timeFreq)/length(timeFreq)*2;
    voltMag(1)=voltMag(1)/2;    %Correct incidental DC doubling
    magdBm = 20*log10(voltMag/0.001);

    %Convert to SSB
    voltMag = voltMag(1:round(length(voltMag)/2,0));
    magdBm = magdBm(1:round(length(magdBm)/2,0));

    %Above code transforms input time series into equivalent
    %frequency domain voltage magnitude
    %Here, logic may be inserted on which portion of spectrum to
    %make power measurements (*Remember volt to power relationship)
    %and if it should be measured based on peak, average,
    %envelope, entire spectrum, or other components.
    %For broadband signals, determining an envelope and basing
    %measurement off of this may be most germane.
end


%Weakly nonlinear RF Mixer Model
%Assumes times are already synchronized between base and carrier
%INPUT FORM:
%   Row 1 = Ideal voltages (Noise-free)
%   Row 2 = Noisy voltages
%   Row 3 = Synchronized Timing
%   Row 4 = Synchronized Timing

function mixer_out = RF_Mixer(in1, in2, coeffMatrix, phase, NF)
%CURRENTLY IDEAL MIXER WITH B=0 C=0
%coeffMatrix = [1,0,0];  %Polynomial coefficient Array
            ti  =  in1(3,:);  %  in1  %  in2  timing's  must  match  for  accurate
calculation
            tn = in1(4,:);
        if phase == 0
            vi1 = in1(1,:); %Base
            vn1 = in1(2,:); %Base Noise
            vi2 = in2(1,:); %Osc Cos
            vn2 = in2(3,:); %Osc Cos + PN
        elseif phase == 90
            vi1 = in1(1,:); %Base
            vn1 = in1(2,:); %Base Noise
            vi2 = in2(2,:); %Osc Sin
            vn2 = in2(4,:); %Osc Sin + PN
        else
            vi1 = in1(1,:); %I or Q
            vn1 = in1(2,:);
            vi2 = in2(1,:); %I or Q
            vn2 = in2(2,:);
        end

    vi = vi1 + vi2;     %Ideal input
    vn = vn1 + vn2;     %Noisy input

    %Calculates Ideal & Noisy 3rd order nonlinear polynomial outputs
```

```
    %Always filter after mixing for best SNR
    voi = vi;
    von = coeffMatrix(1,1)*vn + coeffMatrix(1,2)*vn.^2 + coeffMatrix(1,3)*vn.^3;

    mixer_out = [voi; von; ti; tn];
end

%In1 is synchronized to in2
%Base
    %Row 1 = signal volt
    %Row 2 = signal+noise volt
    %Row 3 = timing
    %Row 4 = timing + jitter
%Carrier
    %Row1 = Ideal cos volt
    %Row2 = Ideal sin volt
    %Row3 = cos+PN
    %Row4 = sin+PN, Row5 = Time
function baseout = Synchronizer(basein, carrierin)
    baseout = zeros(size(basein,1),size(carrierin,2));
    baseout(3,:) = carrierin(5,:);  %Combined signal assumes time of carrier
    baseout(4,:) = carrierin(5,:);  %Now ideal and jitter timing are same
    %For Ideal Times & Voltages
    j = 1;
    n = size(basein,2);
    carriertime = carrierin(5,j);
    for i = 1:n
        basetime = basein(3,i);
        while basetime >= carriertime
            baseout(1,j) = basein(1,i); %Defining Ideal Voltages
            j = j+1;
            if j <= size(carrierin,2)
                carriertime = carrierin(5,j);
            else
                carriertime = basetime+1; %Ensures carrier > base
            end
        end
    end
    %For Noisy Times & Voltages
    j = 1;
    n = size(basein,2);
    carriertime = carrierin(5,j);
    for i = 1:n
        basetime = basein(4,i);
        while basetime >= carriertime
            baseout(2,j) = basein(2,i); %Defining Noisy Voltages
            j = j+1;
            if j <= size(carrierin,2)
                carriertime = carrierin(5,j);
            else
                carriertime = basetime+1; %Ensures carrier > base
            end
        end
    end
end
```

REFERENCES

[1] E. V. Rogozhnikov, A. S. Koldomov, and V. A. Vorobyov, "Full duplex wireless communication system, analog cancellation: Review of methods and experimental research," *2016 International Siberian Conference on Control and Communications (SIBCON)*, Jun. 2016.

[2] W. Tomasi, *Electronic Communications Systems: Fundamentals through advanced*. Upper Saddle River, NJ: Pearson-Prentice Hall, 2004.

[3] B. P. Lathi and Z. Ding, *Modern digital and Analog Communication Systems*. New York, NY: Oxford University Press, 2019.

[4] J. F. Kurose and K. W. Ross, *Computer networking: A top-down approach*. Harlow, United Kingdom: Pearson Education Limited, 2013.

[5] A. Sabharwal, P. Schniter, D. Guo, D. W. Bliss, S. Rangarajan, and R. Wichman, "In-band full-duplex wireless: Challenges and opportunities," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 9, pp. 1637–1652, May 2014.

[6] D. Kim, H. Lee, and D. Hong, "A survey of in-band full-duplex transmission: From the perspective of PHY and Mac Layers," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2017–2046, 2015.

[7] Z. Zhang, K. Long, A. V. Vasilakos, and L. Hanzo, "Full-duplex wireless communications: Challenges, Solutions, and future research directions," *Proceedings of the IEEE*, vol. 104, no. 7, pp. 1369–1409, Feb. 2016.

[8] H. Bany Salameh, Z. Khader, and A. Al Ajlouni, "Intelligent secure networking in in-band full-duplex dynamic access networks: Spectrum Management and Routing Protocol," *Journal of Network and Systems Management*, vol. 29, no. 2, Feb. 2021.

[9] K. Pärlin and T. Riihonen, "Full-duplex transceivers for defense and security applications," *Full-Duplex Communications for Future Wireless Networks*, pp. 249–274, 2020.

[10] L. Song, R. Wichman, Y. Li, and Z. Han, *Full-duplex Communications and Networks*. Cambridge, United Kingdom: Cambridge University Press, 2017.

[11]M.R.Q. Murad, "Performance Analysis and Enhancements for In-Band Full-Duplex Wireless Networks," Ph.D. dissertation, Dept. of Elect. Eng. and Comp. Sci., Univ. of California, Irvine, 2019. https://escholarship.org/uc/item/57b8v1rp

[12] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013.

[13] M. Duarte, C. Dick, and A. Sabharwal, “Experiment-driven characterization of full-duplex wireless systems,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 12, pp. 4296–4307, 2012.

[14] E. Everett, “Full-duplex infrastructure nodes: Achieving long range with half-duplex mobiles,” thesis, Rice University, Houston, Tx, 2012.

[15] T. Vial, A. Lefevre, M. Le Penven, and Q. Bodinier, “A short review of current challenges and potential applications of full duplex in wireless networks,” *2017 XXXIInd General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)*, 2017.

[16] N. Singh, D. Gunawardena, A. Proutiere, B. Radunovi, H. V. Balan, and P. Key, “Efficient and fair MAC for wireless networks with self-interference cancellation,” *2011 International Symposium of Modeling and Optimization of Mobile, Ad Hoc, and Wireless Networks*, 2011.

[17] C. Despina-Stoian, A. Digulescu-Popescu, S. Alexandra, R. Youssef, and E. Radoi, “Comparison of adaptive filtering strategies for self-interference cancellation in LTE Communication Systems,” *2020 13th International Conference on Communications (COMM)*, 2020.

[18] C. D. Nwankwo, L. Zhang, A. Quddus, M. A. Imran, and R. Tafazolli, “A survey of self-interference management techniques for single frequency full duplex systems,” *IEEE Access*, vol. 6, pp. 30242–30268, Jun. 2018.

[19] R. J. Kircher, “Two-Way Signal Transmission System,” 09-Dec-1947.

[20] W. V. B. Roberts, “Short-Range Continuous Wave Radar System,” 25-Jul-1950.

[21] W. V. B. Roberts, “Continuous Wave Radar System,” 14-Jul-1953.

[22] D. H. Ring, “Signal Wave Duplexing System,” 03-Apr-1951.

[23] R. B. Colton, “Radar in the United States Army history and early development at the Signal Corps Laboratories, Fort Monmouth, N.J.,” *Proceedings of the IRE*, vol. 33, no. 11, pp. 740–753, 1945.

[24] M. G. R. Bozzoli, "An introduction to radio-location," in *Transactions of the South African Institute of Electrical Engineers*, vol. 35, no. 4, pp. 77-83, July 1944.

[25] W. R. Hutchins, “Feed-Through Nulling Systems,” 13-Feb-1962.

[26] A. Clavin and E. A. Ohm, "A Broad-Band Microwave Circulator (Correspondence and Authors' Reply)," in IRE Transactions on Microwave Theory and Techniques, vol. 5, no. 2, pp. 164-165, April 1957, doi: 10.1109/TMTT.1957.1125125.

[27] P. D. L. Beasley, A. G. Stove, B. J. Reits and B. As, "Solving the problems of a single antenna frequency modulated CW radar," IEEE International Conference on Radar, 1990, pp. 391-395, doi: 10.1109/RADAR.1990.201197.

[28] F. O'Hara and G. Moore, "A high performance CW receiver using feedthrough nulling," *Microwave Journal*, vol. 6, pp. 63-71, Sep. 1963.

[29] W. T. Slingsby and J. P. McGeehan, "Antenna isolation measurements for on-frequency radio repeaters," 1995 Ninth International Conference on Antennas and Propagation, ICAP '95 (Conf. Publ. No. 407), 1995, pp. 239-243 vol.1, doi: 10.1049/cp:19950300.

[30] C. R. Anderson et al., "Antenna Isolation, Wideband Multipath Propagation Measurements, and Interference Mitigation for On-frequency Repeaters," IEEE SoutheastCon, 2004. Proceedings., 2004, pp. 110-114, doi: 10.1109/SECON.2004.1287906.

[31] J. D. Gibson, *Mobile Communications Handbook*. Boca Raton ; London ; New York: CRC Press, Taylor et Francis Group, 2017.

[32] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving Single Channel, Full Duplex Wireless Communication," *Proceedings of the sixteenth annual international conference on Mobile computing and networking - MobiCom '10*, Sep. 2010.

[33] M. Duarte and A. Sabharwal, "Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results," *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, 2010.

[34] K. E. Kolodziej, B. T. Perry, and J. S. Herd, "In-band full-duplex technology: Techniques and Systems survey," *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 7, pp. 3025–3041, 2019.

[35] M. A. Alim, M. Kobayashi, S. Saruwatari, and T. Watanabe, "In-band full-duplex medium access control design for heterogeneous wireless LAN," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, 2017.

[36] J.-G. Kim, S. Ko, S. Jeon, J.-W. Park, and S. Hong, ``Balanced topology to cancel Tx leakage in CW radar," IEEE Microw. Wireless Compon. Lett., vol. 14, no. 9, pp. 443445, Sep. 2004.

[37] D. Senaratne and C. Tellambura, ``Beamforming for space division duplexing," in Proc. IEEE ICC, Jun. 2011, pp. 15.

[38] B. P. Day, A. R. Margetts, D. W. Bliss, and P. Schniter, "Full-duplex bidirectional MIMO: Achievable rates under limited dynamic range," IEEE Trans. Signal Process., vol. 60, no. 7, pp. 3702-3713, Jul. 2012.

[39] E. Everett, A. Sahai, and A. Sabharwal, "Passive self-interference suppressionfor full-duplex infrastructure nodes," IEEE Trans. Wireless Commun., vol. 13, no. 2, pp. 680694, Jan. 2014.

[40] S. Sadjina, C. Motz, T. Paireder, M. Huemer, and H. Pretl, “A survey of self-interference in LTE-advanced and 5G New Radio Wireless transceivers,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 3, pp. 1118–1131, 2020.

[41] H. Guo, J. Xu, S. Zhu, and S. Wu, “Realtime Software defined self-interference cancellation based on machine learning for in-band Full Duplex Wireless Communications,” *2018 International Conference on Computing, Networking and Communications (ICNC)*, 2018

[42] A. T. Kristensen, A. Burg, and A. Balatsoukas-Stimming, “Advanced machine learning techniques for self-interference cancellation in full-duplex radios,” *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 2019.

[43] D. Brandt and M. Hoffman, “Measurements, statistics, and errors,” in *CAS CERN accelerator school: Beam diagnostics: Dourdan, France , 28 may - 6 june 2008: Proceedings*, Geneva: CERN, 2009, pp. 157–185.

[44]“CHAPTER 9 Noise - MIT,” 05-Mar-2012. [Online]. Available: http://web.mit.edu/6.02/www/s2012/handouts/9.pdf. [Accessed: 11-May-2021].

[45] J. Phillips and K. Kundert, "Noise in mixers, oscillators, samplers, and logic an introduction to cyclostationary noise," Proceedings of the IEEE 2000 Custom Integrated Circuits Conference (Cat. No.00CH37044), 2000, pp. 431-438, doi: 10.1109/CICC.2000.852702.

[46] B. Querbach, S. Puligundla, D. Becerra, Z. T. Schoenborn, and P. Chiang, “Comparison of hardware based and software based stress testing of memory IO interface,” *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Dec. 2013.

[47] H. Okawara, “Hideo Okawara’s Mixed Signal ... - advantest corporation,” *Advantest*, Jun-2013. [Online]. Available: https://www3.advantest.com/documents/11348/3e95df23-22f5-441e-8598-f1d99c2382cb. [Accessed: 16-Oct-2021].

[48] P. Daponte, L. De Vito, G. Iadarola, and S. Rapuano, “Effects of PRBS jitter on random demodulation analog-to-information converters,” *2016 IEEE Metrology for Aerospace (MetroAeroSpace)*, 2016.

[49] Phabrix, *Using Pseudo-Random Binary Sequences to Stress Test Serial Digitial Interfaces*. Phabrix, 2019.

[50] H. Asada, “2.160 System Identification, Estimation, and Learning Lecture Notes No. 19,” *MIT OpenCourseWare, Massachusetts Institute of Technology*, 01-May-2006. [Online]. Available: https://ocw.mit.edu/courses/mechanical-engineering/2-160-identification-estimation-and-learning-spring-2006/lecture-notes/. [Accessed: 16-Oct-2021].

[51] Redd, J. & Lyon, C.. (2004). *Spectral content of NRZ test patterns*. Edn -Boston then Denver then Highlands Ranch Co-. 49. 67-72.

[52] D. V. Perepelitsa, “Johnson Noise and Shot Noise,” *MIT*, 27-Nov-2006. [Online]. Available: https://web.mit.edu/dvp/Public/noise-paper.pdf. [Accessed: 16-Oct-2021].

[53] M. E. Dieckmann, A. Ynnerman, S. C. Chapman, G. Rowlands, and N. Andersson, “Simulating thermal noise,” *Physica Scripta*, vol. 69, no. 6, pp. 456–460, 2004.

[54] Nyquist, H. (1928). "Thermal Agitation of Electric Charge in Conductors". Physical Review. 32 (110): 110–113.

[55] H. Okawara, “Hideo Okawara’s Mixed Signal ... - advantest corporation,” *Advantest*, Jun-2013. [Online]. Available: https://www3.advantest.com/documents/11348/3e95df23-22f5-441e-8598-f1d99c2382cb. [Accessed: 16-Oct-2021].

[56] P. Daponte, L. De Vito, G. Iadarola, and S. Rapuano, “Effects of PRBS jitter on random demodulation analog-to-information converters,” *2016 IEEE Metrology for Aerospace (MetroAeroSpace)*, 2016.

[57] U. L. Rohde, A. K. Poddar, and A. M. Apte, “Getting its measure: Oscillator phase noise measurement techniques and limitations,” *IEEE Microwave Magazine*, vol. 14, no. 6, pp. 73–86, 2013.

[58] “AN10062 Phase Noise Measurement Guide For Oscillators,” *SITime*, Aug-2019. [Online]. Available: https://www.sitime.com/sites/default/files/gated/AN10062-Phase-Noise-Measurement-Guide-for-Oscillators.pdf . [Accessed: 07-Apr-2021].

[59] *Characterization of Clocks and Oscillators,* NIST TN-1337, 1990.

[60] R. Raich and G. T. Zhou, “On the modeling of memory nonlinear effects of power amplifiers for communication applications,” *Proceedings of 2002 IEEE 10th Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop.*, Oct. 2002.

[61] P. Schrangl and L. Giarré, “On optimal design of experiments for static polynomial approximation of nonlinear systems,” *Systems & Control Letters*, vol. 143, p. 104758, 2020.

[62] M. Isaksson, D. Wisell, and D. Ronnow, “A comparative analysis of behavioral models for RF Power Amplifiers,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 1, pp. 348–359, 2006.

[63] D. A. Neamen, “Chapter 8 The pn Junction Diode,” in *Semiconductor physics and devices: Basic principles*, New York, NY: McGraw-Hill, 2012, pp. 277–322.

[64] Mini-Circuits, “Ultra High Dynamic Range Monolithic Amplifier,” PHA-202+ datasheet.

[65] MAXIM, “2.4GHz to 2.5GHz Linear Power Amplifier,” MAX2242 datasheet, [Revised Nov. 2003].

[66] D. M. Pozar, *Microwave and RF design of Wireless Systems*. New York, New York: Wiley, 2001.

[67] J. Watt, R. Borhani, and A. K. Katsaggelos, *Machine learning refined: Foundations, algorithms, and applications*. Cambridge: Cambridge University Press, 2020.

[68] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA: The MIT Press, 2017.

[69] T. Fukui, K. Komatsu, Y. Miyaji, and H. Uehara, "Analog self-interference cancellation using auxiliary transmitter considering IQ imbalance and amplifier nonlinearity," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7439–7452, Nov. 2020.

[70] *Digital Transmission Standard for Cable Television,* ANSI/SCTE 07, 2018.

[71] D. Huang, "Wi-Fi 6 fundamentals: What is 1024-QAM?," *CommScope*. [Online]. Available: https://www.commscope.com/blog/2018/wi-fi-6-fundamentals-what-is-1024-qam/. [Accessed: 20-Sept-2021].

[72] Analog Devices, "Dual, 16-Bit, 1000 MSPS, TxDAC+ Digital-to-Analog Converter," AD9125 datasheet, 2010.

[73] Texas Instrument, "Single-Channel, 14-, 12-, and 10-Bit, 500-MSPS, Digital-to-Analog Converters," DAC3151/3162 datasheet, Aug. 2013 [Revised Feb. 2018].

[74] Vectron, "OX-305 Ultra Low Phase Noise Oven Controlled Crystal Oscillator," OX-305, [Revised May 2019].

[75] Vectron, "VS-701 High Shock Discrete Voltage Controlled SAW Oscillator," VS-701, [Revised June 2019].

[76] A. V. Menon, A. Gunjegai, A. Kurup, and D. G. Kurup, "Combined amplitude and phase noise effects in QAM direct conversion receivers," *2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, Dec. 2015.