UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

A LEAST MEAN-SQUARE APPROACH TO NONLINEAR EQUALIZATION OF DIGITAL ARRAY RECEIVERS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

NICHOLAS LOUIS PECCARELLI

Norman, Oklahoma

2018

A LEAST MEAN-SQUARE APPROACH TO NONLINEAR EQUALIZATION
OF DIGITAL ARRAY RECEIVERS

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Caleb Fulton, Chair

Dr. Nathan Goodman

Dr. Mark Yeary

Dedicated to my father, who taught me the value of hard work and education.

Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under grant no. D15A00090. I would also like to thank Blake James for designing the NASM software suite.

Table of Contents

List of Tables

List of Figures

Abstract

With an increased demand for fully-digital arrays for radar and communications systems by making use of low-cost components with relaxed linearity requirements, nonlinear equalization (NLEQ) is needed to increase the linearity and dynamic range. An iterative solution is proposed in the least mean-square algorithm (LMS) and is shown to be very effective at mitigating intermodulation distortion (IMD) in digital array channels. Temperature and frequency changes in the system also cause the nonlinear characteristics of the system to change, requiring an adaptive NLEQ solution such as LMS. Odd-order IMD spurs correlate to predictable directions at the array level, but, with the use of NLEQ at the channel level, can be decorrelated. Decorrelation is made more difficult to achieve when the array channels are not identical, due to the temperature and process variations that make up all electronics, requiring the use of some type of coefficient averaging.

1 Background, Motivation, and Previous Work

In recent years there has been an increased desire for low-cost, reconfigurable, multi-purpose fully-digital phased array and multiple-input miltiple-output (MIMO) systems for radar, electronic warfare (EW), as well as wireless communications [1]–[7]. Fully-digital arrays, compared to their analog and subarray-digital counter parts, have a transceiver at each element, leading to element-level digital control of the array. This level of control, on both transmit and receive, gives the array much greater flexibility and makes multi-mission systems much more feasable. This flexibility has lead to programs such as the FAA and NOAA's multifunction phased array radar (MPAR) program, which plans to combine weather survaliance, air traffic control, as well as other missions into a single dual-polarized fully-digital phased array radar system [7] and the DARPA Arrays at Commercial Timescales (ACT) program, which focuses on the development of modular common tiles to be used to increase scalability, reduce cost, and reduce development time of future systems [1]. The Army Digital Array Radar (DAR), a 16-element S-band radar, made use of panel-level integration and commercial off-the-shelf (COTS) components to demonstrate a low-cost, scalable system [8]. Currently, The University of Oklahoma is working on Horus, a project which seeks to develop an 8x8 element dual-polarization line-replaceable unit (LRU) for MPAR-type applications, making use of the Analog Devices AD9371 highly integrated transceiver chip [6], [9]. The AD9371 is one of the most recent examples of system-on-chip (SoC) integration, lead by improved design techniques for radio-frequency integrated circuits (RFICs) and advancements in silicon complementary metal-oxide-semiconductors (CMOS) processes [5], featuring dual-differential transmitters (Tx) and receivers (Rx), a tunable range from 300MHz to 6000MHz, Tx synthesis bandwidth (BW) of up to 250MHz and Rx BW of 8MHz to 100MHz. Another major advantage of

fully-digital arrays is their ability to form multiple, simultaneous beams on receive, due to the use of digital beamforming (DBF), allowing digital array radars to simultaneously track multiple targets [5], [6]. System-level dynamic range (DR) also scales with the number of transceivers in an array, and in the case of fully-digital arrays with the number of elements [6]. The gain in DR and typically high cost of digital arrays has lead to a push for the use of low cost, low complexity and COTS components, often times with relaxed linearity requirements.

Though digital arrays provide many advantages, they also have several notable drawbacks. With element level digitization, and forming multiple, simultaneous beams, digital arrays are extremely computationally expensive and with the increasing desire for ultra wideband (UWB) systems, requiring high bandwidth analog-to-digital converters (ADC), computational costs will only increase. Precise clock synchronization of each element's transceiver is also very difficult to achieve, and any timing differences between elements will cause errors on both transmit and receive [5]. Clock phase drift is often caused by temperature variation, and it is important to note that the aforementioned low cost, low complexity components may not provide compensation for temperature variation. Temperature variation also affects the performance of other active components such as low noise amplifiers (LNA), mixers, ADCs, and even tunable filters. The effects of temperature variation on component performance can be compensated for digitally, but this then adds to the calibration complexity of digital arrays. As previously mentioned, digital arrays, compared to analog arrays, provide spatial filtering after digitization, through DBF. This, coupled with the typically wide angle response of the individual antenna elements, leaves digital arrays open to strong interferers. Such interferers can distort the desired signal, introduce nonlinearities into the system, or even push the system into saturation.

This thesis will focus on using nonlinear equalization (NLEQ) to extend the linearity and DR of digital array receivers and adapt to changes in the system, such as temperature and frequency variation, ultimatlly decorrelating the spurious products at the array level. Chapter two will provide an overview of digital array receivers and demonstrate the effects of temperature and frequency dependence on system performance. Next, chapter three will demonstrate the algorithms used for adaptive NLEQ at the channel level. Chapter four will then demonstrate the correlation of spurious products at the array level and derive a method for decorrelating them. Lastly, chapter five will demonstrate the effectiveness of the NLEQ methods shown in the previous chapters on measured data of an digital array channel and a simulated digital array.

1.1 Digital Array Overview

Digital array systems have seen increased popularity for both radar and wireless communications applications and have many benefits over analog array and single-antenna systems. Analog arrays have been around for multiple decades, used for their ability to electronically scan, but were typically much more expensive than mechanically steered systems. Analog arrays made use of phase shifters and complex feeding networks to electronically steer the antenna beam, but suffered from beam-squint when higher bandwidths were used in large arrays and are limited to forming a single beam. After the introduction of solid state amplifiers and more digitized back-ends, subarray-digital arrays were introduced, allowing for more beams to be formed and greater flexibility. Such systems consist of multiple smaller arrays, each having their own transmitter and receiver, being digitized and sharing the same back-end. The digitized subarrays were then beamformed using DBF techniques, which provided more flexibility and adaptability. Leveraging Moore's law

and advancements in RFIC design and a desire for mission flexibility has lead to a recent push for fully-digital arrays. Fully-digital arrays consist of a transmitter and receiver at each element lending control of each individual element to the digital back-end. Digitized control of each element removes the need for phase shifters and creative feeding networks, with the Tx/Rx chip being placed as close to the antenna as possible.

Analog systems with a single transmitter and receiver tend to require tube-based transmit amplifiers to produce enough power to transmit the signals to the desired range. Tube-based transmitters are physically much larger than modern solid-state amplifiers, can be extremely expensive, and often have a short life-span. Solid-state amplifiers are much smaller, being able to be integrated in a chip, and have a much longer lifespan, but they are extremely limited in power, typically only being able to transmit a maximum power of less than one thousand Watts. Digital arrays use solid state amplifiers, with each element having its own, making use of the amplifier's small size, low cost, and long lifespan. The low transmit power of the solid state amplifiers is not a problem for digital arrays since each element has its own transmitter and the power transmitted from each element is summed at the array level.

Even though fully-digital arrays require many times the number of digital components than subarray-digital and analog arrays, the element-level dynamic range can be lowered, contributing to lower device linearity unless the overall system dynamic range is preserved through the decorrelation of undesired signals amongst the elements. Element-level digital arrays suffer from spurious products caused lower device linearity, but these spurs can be corrected in the back-end because each element is digitized. Fully-digital arrays also benefit from graceful degradation, which means that the system can still function fully even when some of the

elements have failed and stopped working. Lastly, with different types of components, such as LNAs, filters, mixers, etc., making up a digital array channel, modeling the nonlinearities of the channel can be complex, including both power and frequency dependence.

1.2 System Nonlinearities

Low-cost radio frequency (RF) active components tend to be nonlinear [2], causing intermodulation distortion (IMD), harmonic distortion, and compression; all of which need to be corrected. The simplest way to represent the nonlinearities of a system is through a simple power-series polynomial, but this lacks any representation of frequency dependence, meaning that it only represents a memory-less nonlinear system. Conversely, the Volterra series provides a robust representation of nonlinear systems, including all memory and cross-terms, but can have an extremely large number of terms, making it difficult to implement in a realizable system. The memory polynomial (MP), on the other hand, can provide a very good representation of a nonlinear system with memory, but, since it doesn't include any cross terms, is much simpler to implement than the Volterra series. Table 1 shows the different series with which to represent nonlinear systems; the one used throughout this thesis will be the MP. Receive-side NLEQ requires more memory terms than digital pre-distortion (DPD) due to the inclusion of a BPF in the front-end. The filter is naturally a very frequency dependent component, which is what makes it useful, but this then produces nonlinear memory effects in-band.

Series Type	Equation
Power Series	$y[k] = \sum_{p=1}^{P} = a_p x^p$
Volterra Series	$y[k] = \sum_{p=1}^{P} (\sum_{m_1=0}^{M-1} \cdot \sum_{m_p=0}^{M-1} h_p(m_1, ..., m_p) \prod_{q=1}^{p} x[k-m_q])$
Memory Polynomial	$y[k] = \sum_{m=0}^{M-1} \sum_{p=1}^{P} a_{pm} x^p[k-m]$
Cross-Memory Polynomial	$y[k] = \sum_{m=0}^{M-1} \sum_{p=1}^{P} a_{pm} x[k-m]\|x[k]\|^{p-1}$
Generalized Memory Polynomial	$y[k] = \sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} \sum_{p=1}^{P} a_{pm} x[k-m_1] x[k-m_2]^{p-1}$

Table 1: A summary of the different types of series/polynomials to describe nonlinear systems.

1.2.1 Changes in System Characteristics

The characteristics of a nonlinear system change with power, frequency, and temperature, requiring the use of an adaptive correction method. Figure 1 shows the input power vs output power of an amplifier with the 1dB compression point and the third-order intercept (IP3) shown. Both the 1dB compression point and IP3 are common figures of merit for linearity; typically, the higher these figures are, the more linear the device/system is. This figure shows the power dependence of a system, which is considered linear until around the 1dB compression point, when the output power stops following the linear relationship, and when the third-order spurious products begin to show up. Frequency dependence can also effect the performance of arrays, especially wideband and tunable systems. This phenomenon is most easily seen in the frequency response of a filter; there is a passband and a stopband, which completely depend on frequency, there can also be ripple in each of the bands. Ripple in the passband of a digital array channel creates frequency dependence in the system on top of the power dependence previously discusses.

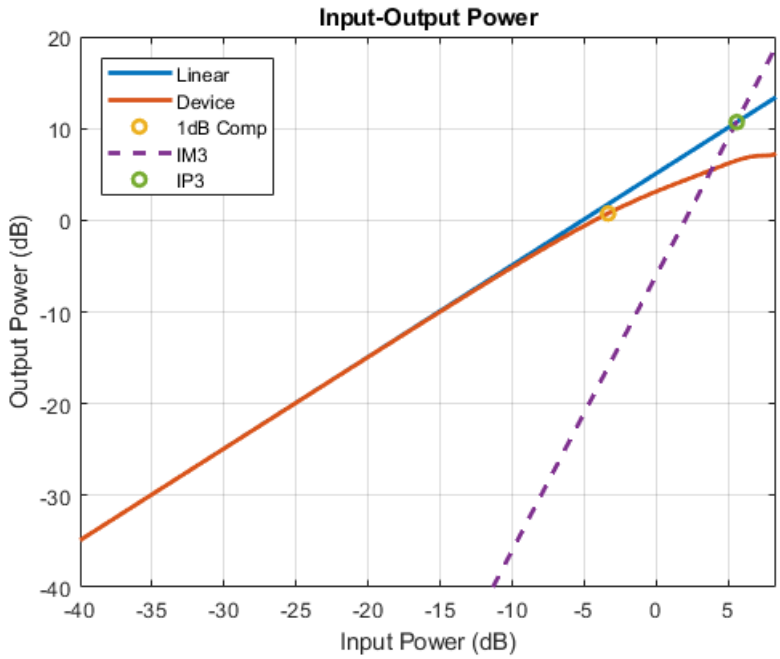


Figure 1: Input-Output power of a generic nonlinear system

Lastly, device, and therefore system, performance depends on temperature. The power and frequency dependencies of devices change with temperature due to the physics of electronic devices.

Power dependence can be accounted for with the use of a power-diverse training waveform, such as white Gaussian noise (WGN). A wide-bandwidth waveform and the use of memory terms in the nonlinear correction coefficients can correct the frequency dependence in the system. Finally, the use of an adaptive calibration method will help to account for changes in temperature and large changes in frequency, due to the use of tunable components.

1.3 Nonlinear Equalization Techniques

Typically, on transmit, as much power as possible is pushed through the system, meaning that the front-end power amplifiers (PAs) operate in the nonlinear region.

Even though operating in the nonlinear region on transmit is extremely power efficient, IMD causes spectral spreading, which can be transmitted out-of-band and with modern communication channels neighboring each other, the transmitted IMD can pollute adjacent channels [10]. Much development has been done to correct this spectrum spreading through the use of digital pre-distortion. Digital pre-distortion was done in [11], where the authors employed a cross-memory polynomial model (CMPM) using recursive least squares (RLS) in a field-programmable gate array (FPGA). They then compared these results with the results gathered from a memory-less polynomial and a memory polynomial (MP) correction. They showed similar IMD mitigation for both the CMPM and MP correction, which was a much greater improvement when compared to the memory-less model. The authors of [12] utilized an adaptive filtered-x least mean-square (LMS) algorithm to implement a Volterra series correction. LMS is a commonly used adaptive algorithm, but, when used in pre-distortion, can have serious stability problems due to the time delays of the applied Volterra series. On the other hand, filtered-x LMS, which is commonly used for pre-distortion filtering, provides an adaptive algorithm without introducing instability.

Compared to transmit, on receive the user does not have control over what goes through the system, or the power of the desired signal and of potential interferers. Digital post-distortion, used to correct nonlinearities on receive, has been done in [2], [10], [13]–[17]. The authors of [2] used weighted least squares (WLS) to correct odd-order products in a digital array radar (DAR). Least squares was also used, with a memory-less polynomial, by the authors of [18] to correct the nonlinearities produced by a cost-efficient CMOS ADC. They used an auxiliary ADC, with a reduced bandwidth to ensure good linearity, to provide continuous correction updates. The effectiveness of the updating model was then shown by varying the temperature

of the ADC, and thus its nonlinear characteristics.

The LMS algorithm, a stochastic solution compared to the deterministic least-squares solution, was used in [14] to apply a memory-less correction to a GSM-based desired signal with two strong intereferers, and in [15] to correct both RF and baseband nonlinearities. The authors of [19] proposed a CMOS implementation of LMS, using an auxiliary ADC, for tuning a channel select filter to mitigate interferer IMD. The normalized LMS (NLMS) algorithm was applied in [13] and [10], where the authors in both used a simple power series based correction, and in [16] to recover the desired signal covered by intereferer IMD. Lastly, in [17] the effect of interferers from neighboring communication channels in UWB systems and the need to correct IMD products created by interferers, especially when they fall on top of a weak desired signal, is shown. The traditional technique of nonlinear receiver calibration consists of two tones [2], [17], but the narrow-band assumption behind the use of two tones is not valid for UWB reconfigurable systems due to the presence of frequency dependence in wide-bandwidth signals [11].

Fully-digital arrays are especially sensitive to interferers, compared to analog arrays, because the spacial filtering of interferers is provided after digitization, meaning that IMD from the interferers will occur before the interferer is mitigated through the use of adaptive digital beamforming (ADBF), or through analog beamforming, in general. IMD products caused by interferers can fall on top of weaker desired signals, degrading the data. This leads to a need to be able to mitigate the nonlinear distortion caused by interferers before digital beamforming. Nonlinear equalization (NLEQ) performed immediately after quantization can help to correct data and mitigate distortion. This technique was used in [2] to decorrelate third-order IMD in the Army DAR, which were previously shown to correlate to specific beam angles in [20].

1.4 Present Work

The work presented in this thesis provides an iterative, adaptive, NLEQ solution that is capable of, and necessary for, dealing with both temperature and frequency changes. This solution is similar to the solution proposed in [15], but a memory polynomial is used instead of a static power series in order to also correct the frequency dependence of the wideband system. Data is gathered from a two different digital array channels, making use of the AD9371 previously mentioned, one with a nonlinear amplifier and the other with a nonlinear varactor-tuned bandpass filter. Temperature changes are applied to the channel that makes use of the nonlinear amplifier and the importance and effectiveness of an adaptive NLEQ method is shown. Then, the channel with the varactor-tuned bandpass filter is tuned to different S-band center frequencies and, again, the effectiveness and necessity of an adaptive NLEQ solution is shown.

Lastly, the channel with the nonlinear amplifier is modeled, instead of corrected, using the proposed NLEQ method, calculating the nonlinear coefficients that describe the channel. These coefficients are then used in an in-house MATLAB toolbox to simulate a linear array made up of these nonlinear channels. The proposed NLEQ method is then applied in the simulation to see its effectiveness on the array level. The simulation is then run for different array sizes to see the how NLEQ performance changes with the number of elements. The goal of the work presented here is to extend the linearity of low-cost digital array channels, first on the channel level, and then at then at the array level.

2 Digital Array Receiver

Fully-digital arrays, compared to their analog counterparts, have many times more components - such as transceivers, mixers, and filters - and thus can incur higher costs. In order to lower the cost of the system, low-cost components are often used, along with relaxed linearity requirements. Applying NLEQ at the channel level, before digital beamforming, can extend the linearity, and thus the dynamic range, of the digital array. An example of a single channel and its nonlinearities is presented here. A fully-digital array containing nonlinear elements is then modeled. Lastly, the nonlinear characteristics of the devices contained in each channel can change with temperature, frequency, and time, therefore, it is very important to also account for these possible changes when considering NLEQ solutions and correction methods. Lastly, examples of how temperature and frequency changes effect device nonlinearities, and thus system nonlinearities.

2.1 Channel Model

Digital array channels tend to make use of the direct conversion receiver, such as the AD9371 used in this thesis and the Hours project mentioned in Chapter 1 The direct conversion receiver, compared to the popular superheterodyne receiver, has no intermediate frequency (IF) stange, converting the signals at the carrier frequency directly down to complex baseband. It should be mentioned, though, that even though the results presented here are for a direct conversion receiver, the same methods can be used to apply NLEQ to other receiver types.

The receiver channel of one element of a digital array, with the nonlinear components in red, is shown in Figure 2. The bandpass filter placed immediately after the LNA for UWB and tunable systems usually has a large passband and very low selectivity [15], allowing out-of-band interferers to enter the receiver [16], or is

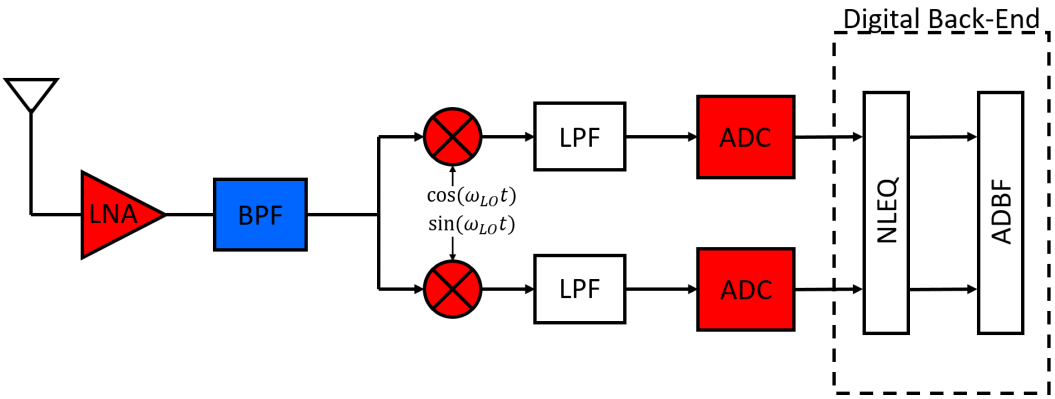


Figure 2: Example of a single channel of a digital array

tunable in nature, which is typically achieved with the use of active, nonlinear components, thus adding to the nonlinearities of the channel. The LNA nonlinearities are generated at RF, with the even-order and odd-order-sum IMD and harmonics falling far outside of the band-of-interest being filtered out. Odd-order difference nonlinearities fall in band, next to their first-order counterparts, distorting the received signal [21]. The nonlinearities caused by the mixer and ADC generate baseband spurs, with both even and odd-order IMD and harmonics generally falling in band [2]. This paper focuses solely on the RF nonlinearities caused by the LNA and bandpass filter, assuming the LNA or filter reach compression before the baseband nonlinearities from the mixers and ADCs show up above the noise floor.

The first component in the digital receiver channel, shown in figure 2, is the antenna. Antennas used for phased arrays typically have minimal spatial selectivity to give the array wide scanning angles, but at the element level it allows for interferers from all directions. This can allow interferers to cause in-band spurs, push the channels into compression, or even into saturation because digital arrays, unlike analog arrays, spatially filter after digitization with DBF. The first component after the antenna is typically an LNA in order to minimize system noise figure and maximize gain. LNAs are nonlinear devices, often limiting the spurious-free dynamic

Main Tone	f_1, f_2
Second Order Harmonics	$2f_1, 2f_2$
Second Order Sum and Difference	$f_1 + f_2, f_1 - f_2, f_2 - f_1$
Third Order Harmonics	$3f_1, 3f_2$
Third Order Sum and Difference	$2f_1 + f_2, 2f_2 + f_1, 2f_1 - f_2, 2f_2 - f_1$

Table 2: A summary of common intermodulation distortion products

range (SFDR) of a system, and much of this thesis will be focused on correcting system nonlinearities caused by an LNA in a digital array front-end. The output of the LNA can be represented, in the static/memoryless case, by

$$\tilde{x}_{LNA}(t) = \sum_{p=1}^{P} a_p d^p(t). \tag{1}$$

The output of the LNA is then passed through the BPF, which can be a passive or active device, but with the increasing demand for reconfigurable, multi-mission systems, tunable BPFs are often required and thus require the use of active components such as varactor diodes or radio frequency microelectromechanical systems (RF-MEMS) capacitors. It is also important to note that many of the distortion products caused by the LNA will be filtered out by the BPF, passing only the linear and the odd-order difference terms.

$$\tilde{x}_{BPF}(t) = \tilde{x}_{LNA}(t) \circledast h_{BPF}(t) \tag{2}$$

After the filter, the signal is then split between two mixers, one in-phase (I) and the other quadrature (Q). The mixer is a nonlinear device that produces the sum and difference of two frequencies, one of the frequencies being the output from the filter and the other being the center frequency of the system. The difference of the two frequencies shifts the signal from RF to baseband where it can more easily be digitized.

$$x_I(t) = \tilde{x}_{BPF}(t)\cos(\omega_0 t) \tag{3}$$

$$x_Q(t) = \tilde{x}_{BPF}(t)\sin(\omega_0 t) \tag{4}$$

The mixers are then followed by an active LPF to remove the sum signals and any other higher order products, including any remaining even-order and odd-order-sum RF nonlinearities caused by the LNA or BPF. The mixer can also introduce other nonlinearities into the system and, compared to the RF nonlinearities caused by the LNA and BPF, these nonlinearities are at baseband and aren't filtered out by the LPF, meaning that both even-order and odd-order spurs generated by the mixer remain in-band.

$$x_{I_{LPF}}(t) = x_I(t) \circledast h_{LPF}(t) \tag{5}$$

$$x_{Q_{LPF}}(t) = x_Q(t) \circledast h_{LPF}(t) \tag{6}$$

Lastly, after the mixer and LPF, the I and Q signals are digitized by their respective ADCs. As will all other active electronics, ADCs are also inherently nonlinear and can produce more baseband spurs, but as previously mentioned, in this thesis we will assume that the baseband components are much more linear than the RF components, and thus there are no baseband nonlinearities in the system. After being digitized, the I and Q samples are combined as described by equation 7, assuming there are no I/Q imbalances in the channel.

$$x[n] = x_I[n] + jx_Q[n] \tag{7}$$

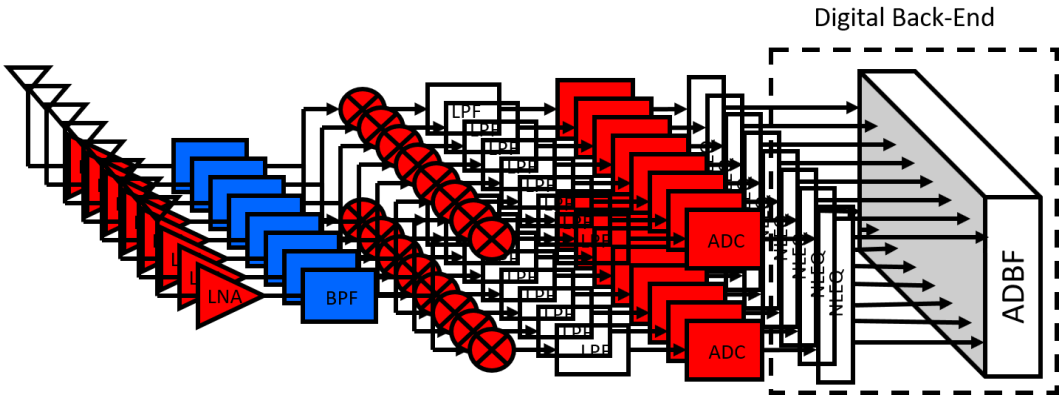


Figure 3: Example of a digital array

2.2 Digital Array Model

A digital array, shown in Figure 3, consists of multiple digitized channels. Since each element is digitized, DBF can be implemented and can be described as a summation of each element's received signals. Extending equation 7 to reflect the received signal for a specific array element we write

$$x_m[n] = x_{I_m}[n] + jx_{Q_m}[n] \tag{8}$$

Equation 9 denotes the DBF of M elements, where $\mathbf{r_m}$ is the location of the m^{th} element, and is equivalent to taking the spatial-domain Fourier transform. It is through this Fourier transform that sources sum coherently (correlate) and noise sums incoherently (decorrelates).

$$E(\theta, \phi, n) = \sum_{m=1}^{M} x_m[n] e^{j\mathbf{k}\cdot\mathbf{r_m}} \tag{9}$$

Equation 10 denotes DBF in the frequency domain, where $X_m[\omega]$ is the Fourier transform of $x_m[n]$. This thesis will consider a linear array, scanning only in azimuth, with $\frac{\lambda}{2}$ element spacing, therefore we can simplify equation 10 in equation 11, where k is the wavenumber.

Description	Frequency	Angle
Main Tone 1	f_1	θ_1
Main Tone 2	f_2	θ_2
Third Order Difference	$2f_1 - f_2$	$\theta = \sin^{-1}(\frac{2f_1 \sin\theta_1 - f_2 \sin\theta_2}{2f_1 - f_2})$
	$2f_2 - f_1$	$\theta = \sin^{-1}(\frac{2f_2 \sin\theta_2 - f_1 \sin\theta_1}{2f_2 - f_1})$

Table 3: A description of the correlation of third-order spurs in an array.

$$E(\theta, \phi, \omega) = \sum_{m=1}^{M} X_m[\omega] e^{j\mathbf{k}\cdot\mathbf{r_m}} \tag{10}$$

$$\begin{aligned} E(\phi, \omega) &= \sum_{m=1}^{M} X_m[\omega] e^{jknd\sin\phi} \\ &= \sum_{m=1}^{M} X_m[\omega] e^{j\pi n sin\phi} \end{aligned} \tag{11}$$

Digital beamforming is equivalent to taking the spatial-domain Fourier transform. It is through this Fourier transform that sources sum coherently (correlate) and noise sums incoherently (decorrelates). Sources correlate to their respective direction of arrival with beamforming, as frequencies correlate with the frequency-domain Fourier transform. The IMD spurs produced by nonlinearities in the array channels also correlate to specific, predictable, directions [20], shown in Table 3.

Figure 4 shows the frequency-domain DBF of a 12-element array with nonlinear channels. The array received signals from two sources, one at -11° with a baseband frequency of 17MHz, and the other from 15° with a baseband frequency of 11MHz. The nonlinear channels of the array also introduced two third-order IMD spurs at 5MHz and 23MHz with correlated directions of 45° and -40°, respectively. Figure 5 shows the frequency-domain slice of each tone to show its respective spatial beam.

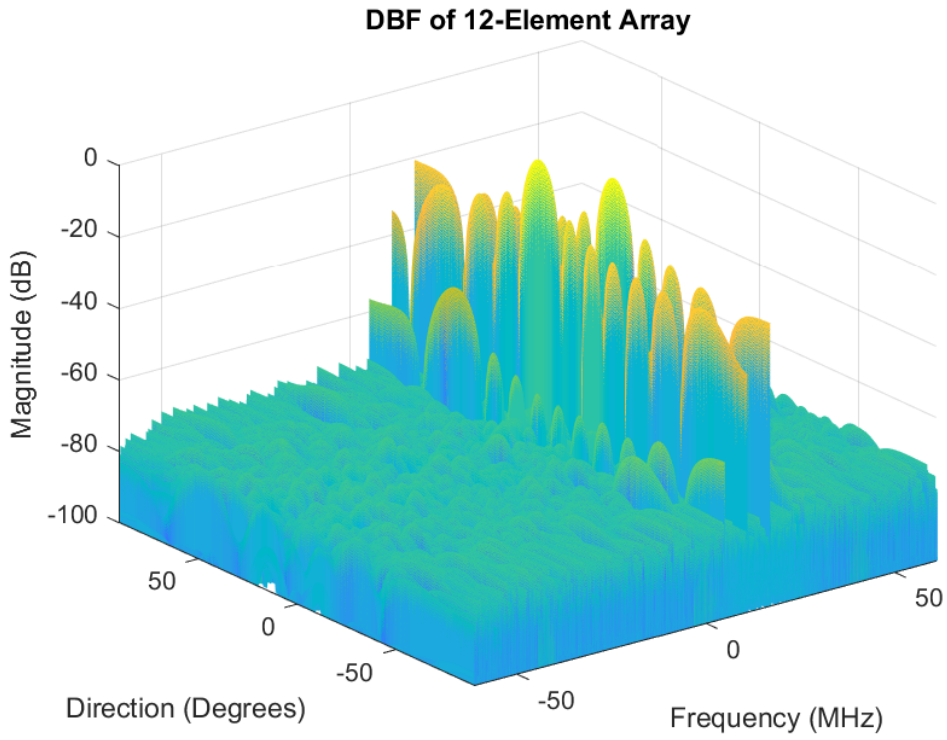


Figure 4: Beamforming of a 12 element digital array (11 nonlinear elements and 1 linear-auxiliary element) for all directions ($-90°$ to $90°$) and the entire bandwidth. There are two received tones, one at 11MHz from $15°$ and the other at 17MHz from $-11°$, and their two third-order spurs, which correlate to 5MHz from an apparent direction of $45°$ and 23MHz from an apparent direction of $-40°$.

It is with this view that the spatial correlation of the IMD spurs, and the importance of not only mitigating them in the frequency domain, but also decorrelating them in the spatial domain becomes more apparent.

2.3 Array Element Differences

Ideally each element of an array would be identical and perform the same, but there are many factors that influence each element differently, making each element perform in a different way. Mutual coupling and and edge-effects, caused by the array antenna design, are two of the things that can effect how each element of the

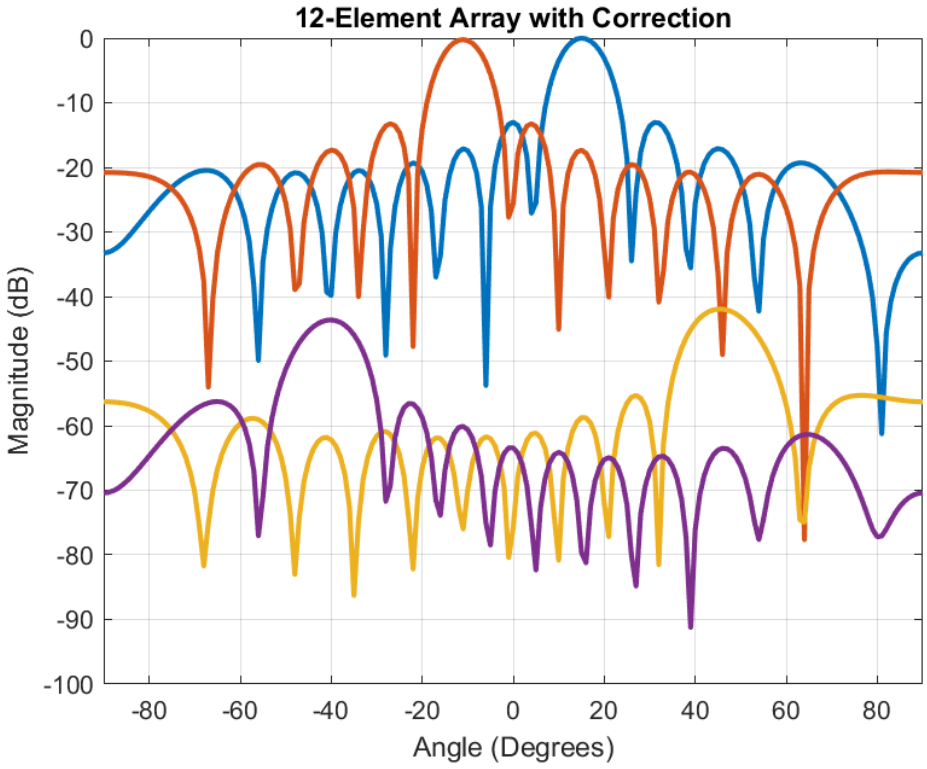


Figure 5: Beamforming of a 12 element digital array (11 nonlinear elements and 1 linear-auxiliary element) with two received tones, one at 11MHz from 15° (blue) and the other at 17MHz from -11° (orange), and their two third-order spurs, which correlate to 5MHz from an apparent direction of 45° (yellow) and 23MHz from an apparent direction of -40° (purple).

array will perform differently than other elements of the array. For this thesis, we will ignore these array element discrepancies and assume that the antenna elements are homogeneous, focusing on the differences between each element's respective channel.

It is well known that electronics are not perfect devices and are usually classified by fabrication/performance tolerances, and, generally, the higher the tolerance the higher the price. For example, semiconductor chip yields have decreased drastically with decreased node size due to process variation, going from a yield of 90% for 350nm processes, to 50% for 90nm, to 30% for 45nm nodes [22]. Fully-digital arrays require solid-state, often on-chip, low-cost amplifiers, meaning that the gain and RF nonlinearities of each element will be a victim of the heterogeneity of semiconductors. It is not uncommon to have gain tolerances of ± 1.5dB, and with gain errors come changes in the transfer function of the amplifier and its nonlinearities.

With the knowledge that the characteristics of each channel are different, we can rewrite the polynomial in equation 1 as having different coefficients for each element, as described in equation 12. We can then represent the coefficients as having some underlying Gaussian distribution, with mean μ_p. This means that, given a large enough array, the performance of the array should converge to the mean of the underlying distributions of the channels, through the DBF summation.

$$x_{LNA_n}(t) = \sum_{p=1}^{P} a_{np} d_n^p(t). \tag{12}$$

2.3.1 Temperature Effects on Nonlinear Systems

Along with process variations, temperature variations have become very important integrated circuit (IC) design issues. These effects are especially of concern to RF front-end circuitry, specifically for the LNA, but also for mixers and even passive

components [23]. MOSFET threshold voltage (V_T) and mobility (μ_n) are both effected by temperature, decreasing with increased temperature, given by equations 13 and 14.

$$V_T(T) = V_T(T_0) = \alpha_{V_T}\Delta T \tag{13}$$

$$\mu_n(T) = \mu_n(T_0)(\frac{T}{T_0}) \tag{14}$$

Techniques for process and temperature compensation were studied in [23], for LNAs and mixers, by implementing a biasing circuit. This correction was acheived by adjusting the size of two NMOS resistors, reducing temperature-based variation of the gain of the LNA, for a range of -40° C to 80° C, from 4.85dB to a mere 0.14dB. This solution, though elegent, adds cost and complexity to the LNA, which, for a large fully-digital array, can turn into a significant amount. And, if the bias circuit begins to suffer from either process or temperature variations, they can compound with the variations of the LNA. Also, adding more components to the LNA, not only increases the cost and complexity, but also increases the failure rate.

Temperature and process variation compensation was also studied in [24] for a power amplifier (PA) on transmit, but correction was implemented digitally, through digital pre-distortion (DPD). Since DPD is generally used to extend the linearity of the PA, allowing it to operate in the nonlinear region, where it is most efficient, the 1-dB compression point was of extreme concern to the authors. They found that, with their device under test (DUT), with temperature variation from -25° C to 100° C, the 1-dB compression point varied from 36.7dBm to 19.4dBm. The authors then implemented a digital correction for these performance variations, by comparing the expected output of the PA with the actual output of the PA, with some threshold error. Whenever the error would rise above the threshold, the coefficient training

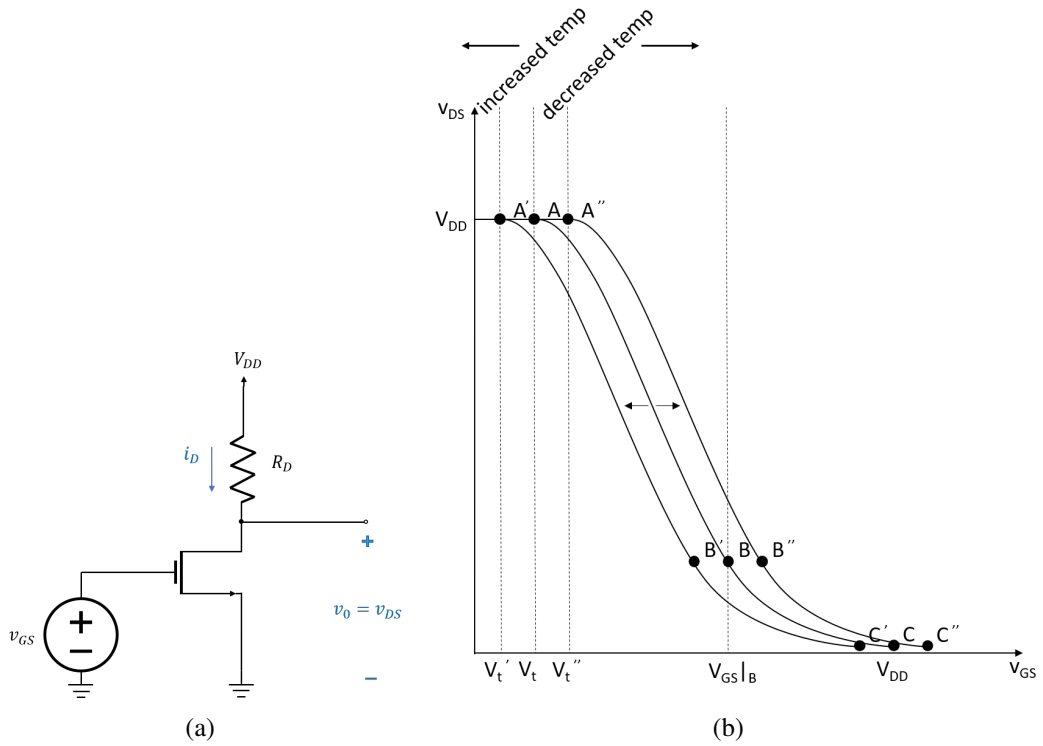
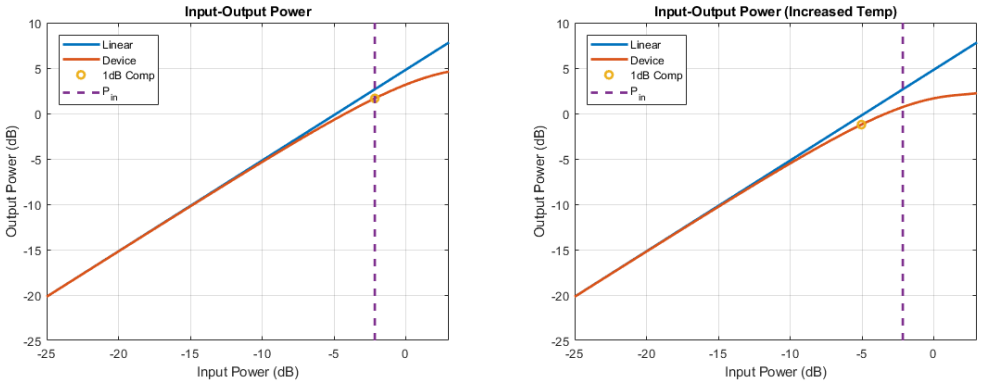


Figure 6: (a) A MOSFET amplifier with input v_{GS} and output v_{DS}. (b) The voltage transfer characteristic (VTC) of the amplifier in (a). When the temperature of the MOSFET amplifier increases, V_t decreases, shifting the entire VTC curve; for example operating at point B, which represents the voltage between the saturation and triode reigions of the MOSFET, appears linear while operating at the temperature that provides V_t and V_t'', but for the temperature that shifts V_t to V_t' the same input voltage is now operating in the triode region, which behaves nonlinearly.



(a) (b)

Figure 7: An example of how amplifier gain can change with temperature; (a) shows the 1dB compression point intersects with the input power (purple dashed line), but when the temperature of the amplifier is increased (b) shows that the same input power now exceeds the 1dB compression point, resulting in stronger nonlinearities.

was implemented until the error converged to some value below the threshold. They used a GMP, from Table 1, to model the nonlinearities of the PA. An adaptive digital solution, such as the one implemented by these authors, is easy to implement, especially when DPD was already being used. A digital solution, compared to a hardware-based solution as proposed in [23], is more general and can be applied to any digital front-end; it also keeps the hardware cost and complexity to a minimum.

Lastly, it is important to point out that every element on an array is not operating at the same temperature. The temperature of each element is also constantly changing due to factors such as power consumption, the relative location of each element on the array (interior elements will naturally retain more thermal energy than exterior elements, as seen in Figure 8, and even things such as weather and location of the sun relative to the array as seen in Figure 9.

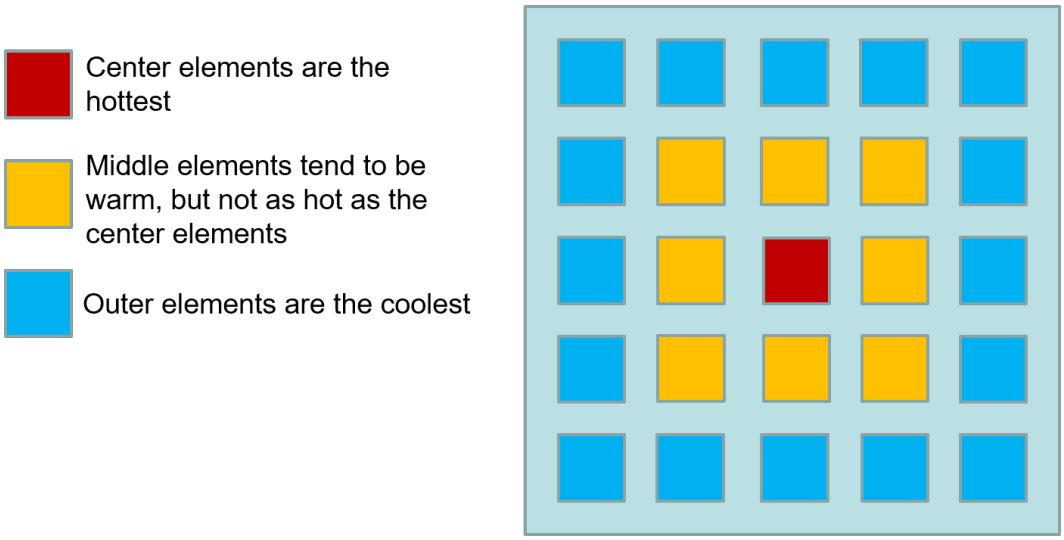


Figure 8: An example of how the array elements operate at different temperatures based on their location in the array.

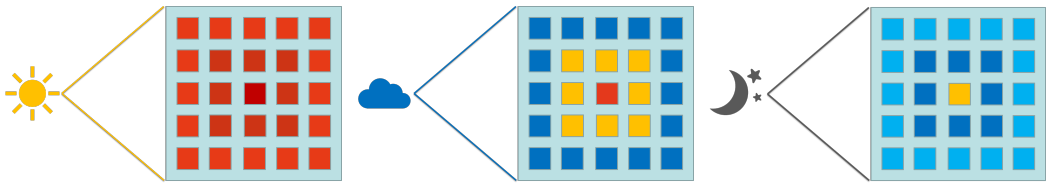


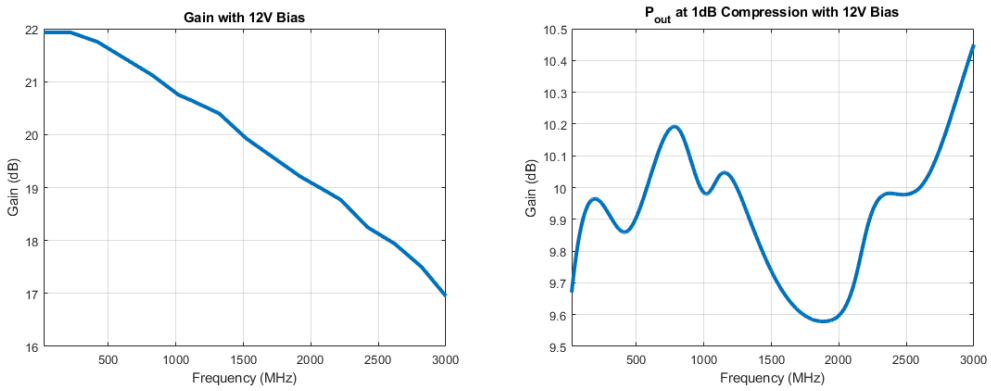
Figure 9: An example of how the temperature of the array/array elements changes throughout a day based on weather and the location of the sun.

2.3.2 Frequency Dependence of Nonlinear Systems

Modern and future fully-digital arrays are expected to be UWB and reconfigurable to a large range of frequencies. One of the struggles with this demand is that even with many devices and components claiming to be wideband, device performance and characterization varies greatly with frequency, as it did with temperature. The previous section, which discussed the effect of temperature variation on nonlinear systems and devices, focused mostly on how the gain and 1dB compression point vary with temperature, which are mostly the result of a static, memory-less, power series. Frequency dependence is purely the results of memory in the transfer function, or polynomial, that describes the system or device.

The simplest way to model frequency dependence due to memory is through a finite impulse response (FIR) filter. Each tap of the N^{th} order filter represents the coefficient for that time delay; for example, the first coefficient is for a delay of zero ($x[n]$), the second coefficient is for a delay of one ($x[n-1]$), and this process continues until the N^{th} coefficient, which is for a delay of N ($x[n-(N-1)]$).

These frequency dependencies are also seen in all electronic devices, whether it's the antenna, LNA, filter, mixer, or ADC; they all have some amount of frequency dependence. This is most obviously seen with antennas and filters, where they have passbands, where frequencies are passed through, and stopbands, where frequencies are rejected. But, frequency dependence is also present inside of the passband, most obviously seen through ripple. Frequency dependence is also a huge factor for amplifiers; for example, the Mini Circuits ZJL-3G+, which is used in a nonlinear channel in Section 5, amplifier's gain and 1dB compression point output power versus frequency are shown in Figure 10. The amplifier's gain has a linear relationship with frequency, decreasing about 5dB over the 3GHz passband. The 1dB compression point also changes with frequency, but its relationship is very



(a) (b)

Figure 10: Plots of the performance of the Mini Circuits ZJL-3G+ amplifier with respect to frequency; (a) shows the gain in dB with frequency and (b) shows how the 1dB point is dependent on frequency.

nonlinear. From this, we can see that the nonlinear characteristics of the amplifier change greatly with frequency.

For narrowband systems, frequency dependence is not an issue as long as the frequencies of interest are inside the operating band of the system. This is because if the bandwidth of interest is much smaller than the ripple, there will be little to no difference between the first frequency and the last frequency. But, when dealing with wideband systems, and especially UWB systems, it is imperative to account for frequency dependence in the system. Applying NLEQ can correct both the power and frequency dependence in a system, and can adapt to changes in the system when an iterative solution, such as LMS, is used.

3 Nonlinear Equalization with Least Mean-Square

With all of the variations in channel characteristics that can occur from operating frequency and temperature, an adaptive solution to NLEQ is needed. Previously, in [2], weighted least-squares (WLS), given in equation 15, was employed as an NLEQ solution for direct conversion digital array channels. WLS, though an effective NLEQ solution to static systems, is not efficient enough at updating due to the need for a matrix inversion. Least mean-square (LMS) is an iterative solution to NLEQ, used in [14], [15], allowing the algorithm to adapt the weights to changes in system characteristics.

$$\begin{gathered} \mathbf{WX}b = \mathbf{WY} \longrightarrow \\ b = (\mathbf{X}^H\mathbf{WX})^{-1}\mathbf{X}^H\mathbf{WY} \end{gathered} \tag{15}$$

The LMS algorithm is a stochastic descent method that attempts to minimize the mean-square error (MSE) of the desired signal $d(n)$ and the corrected signal $y(n)$.

$$MSE = \frac{1}{N}\sum_{n=1}^{N}[d(n) - y(n)]^2 \tag{16}$$

Figure 11 shows the iterative LMS algorithm with a desired signal $d(n)$, an unknown nonlinear system $\mathbf{h}(n)$, a distorted signal $\mathbf{x}(n)$, adaptive weights $\mathbf{w}(n)$, corrected signal $y(n)$, and error $e(n)$.

The cost function, which is derived from the mean square error, is given by

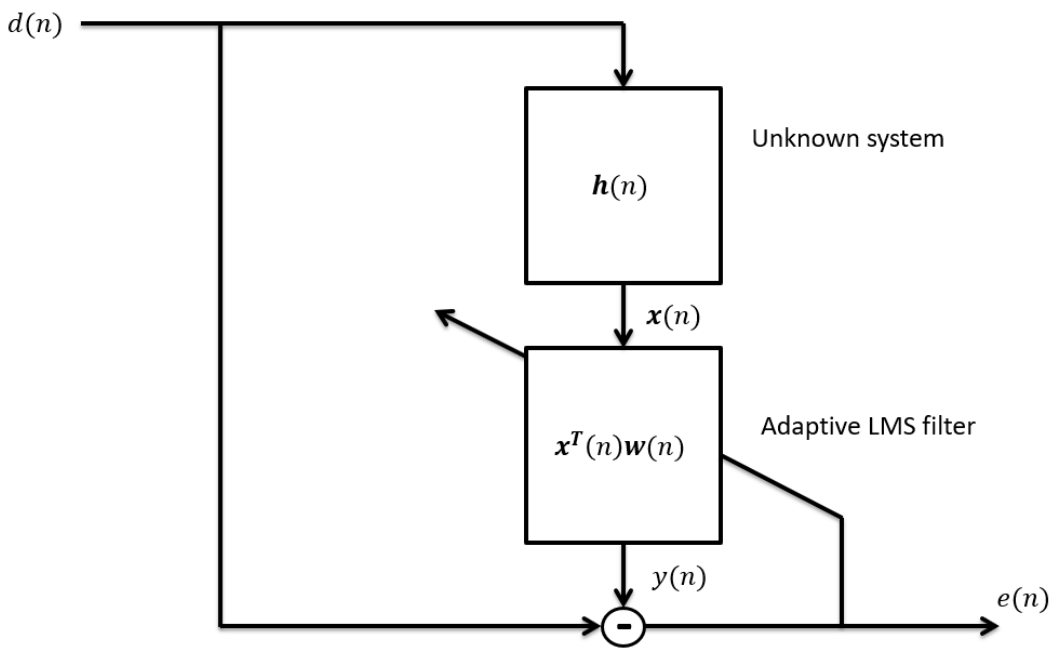


Figure 11: A block diagram of the LMS algorithm.

$$\begin{aligned} J =& E\{[d(n) - y(n)]^2\} \\ =& E\{e^2(n)\}. \end{aligned} \tag{17}$$

The corrected signal $y(n)$ is given by

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \tag{18}$$

where $\mathbf{x}(n)$ is the distorted signal and $\mathbf{w}(n)$ are the adaptive weights.

Combining equations 17 and 18 we get the cost function in terms of the adaptive weights.

$$J[\mathbf{w}(n)] = E\{[d(n) - \mathbf{w}^T(n)\mathbf{x}(n)]^2\} \tag{19}$$

We can expand equation 19 to be

$$J[\mathbf{w}(n)] = \sigma_d^2 - 2\mathbf{p}_{dx} + 2\mathbf{R}_x\mathbf{w}(n) \tag{20}$$

where σ_d^2 is the variance of the desired signal, $\mathbf{p}_{dx}$ is the cross-correlation vector of $d(n)$ and $\mathbf{x}(n)$, and $\mathbf{R}_x$ is the correlation matrix of the distorted signal.

The minimum of the cost function is when its first and second derivatives with respect to $\mathbf{w}(n)$ are zero and positive, respectively. The first derivative is given by the gradient

$$\begin{aligned} \nabla_w J[\mathbf{w}(n)] &= -2\mathbf{p}_{dx} + 2\mathbf{R}_x\mathbf{w}(n) \\ &= -2d(n)\mathbf{x}(n) + 2\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n) \\ &= -2e(n)\mathbf{x}(n) \end{aligned} \tag{21}$$

Finally, the next weight $\mathbf{w}(n+1)$ is given by

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \mu\nabla_w J[\mathbf{w}(n)] \\ &= \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n) \end{aligned} \tag{22}$$

where μ is the step-size. The step-size determines by how much the weights will be adjusted for each iteration; a larger step-size leads to faster convergence, but also less stability, while a small step-size take longer to converge, but also provides a much more stable solution.

3.1 LMS Algorithm for Nonlinear Correction

Receiver-side NLEQ is different from simply modeling a nonlinear system, as the signals have already been passed through the system, with the nonlinearities ap-

plied. Applying NLEQ after the signals have passed through the nonlinear systems is essentially the same as cascading two nonlinear systems; the second is used to remove the nonlinearities added by the first. It is very important to point out, though, that many nonlinearities fall outside of the band-of-interest, such as even-order products, and also some odd-order products, and, of course, there is also the possibly that some of these tones will alias in. There is also the issue of strong interferers. Such interferers could fall out-of-band, but produce spurs that fall in-band, making it nearly impossible to correct.

For example, we will look two tones and their output from a third-order nonlinear system. Let ω_1 and ω_2 be the two baseband frequencies, and let ω_0 be the LO frequency. Then

$$d = d_1 + d_2 = A_1 \cos((\omega_1 + \omega_0)t) + A_2 cos((\omega_2 + \omega_0)t) \tag{23}$$

is the input signal into the array, where

$$d_1 = A_1 \cos((\omega_1 + \omega_0)t) \tag{24}$$

is from a direction of ϕ_1, and

$$d_2 = A_2 \cos((\omega_2 + \omega_0)t) \tag{25}$$

is from a direction of ϕ_2.

The nonlinear channel is characterized by

$$y_{out} = k_1 y_{in} + k_2 y_{in}^3 \tag{26}$$

Then, given the input signals and the channel, we get

$$
\begin{aligned}
x =&(k_1A_1 + \frac{3}{4}k_2A_1^3 + \frac{3}{2}k_2A_1A_2^2)\cos((\omega_1+\omega_0)t) \\
&+ (k_1A_2 + \frac{3}{4}k_2A_2^3 + \frac{3}{2}k_2A_1^2A_2)\cos((\omega_2+\omega_0)t) \\
&+ \frac{1}{4}k_2A_1^3\cos(3(\omega_1+\omega_0)t) \\
&+ \frac{1}{4}k_2A_2^3\cos(3(\omega_2+\omega_0)t) \\
&+ \frac{3}{4}k_2A_1^2A_2\cos((2\omega_1-\omega_2+\omega_0))t) \\
&+ \frac{3}{4}k_2A_1^2A_2\cos((2\omega_1+\omega_2+\omega_0))t) \\
&+ \frac{3}{4}k_2A_1A_2^2\cos((2\omega_2-\omega_1+\omega_0))t) \\
&+ \frac{3}{4}k_2A_1A_2^2\cos((2\omega_2+\omega_1+\omega_0))t)
\end{aligned}
\tag{27}
$$

After filtering, we only care about

$$
\begin{aligned}
x =&(k_1A_1 + \frac{3}{4}k_2A_1^3 + \frac{3}{2}k_2A_1A_2^2)\cos(\omega_1 t) \\
&+ (k_1A_2 + \frac{3}{4}k_2A_2^3 + \frac{3}{2}k_2A_1^2A_2)\cos(\omega_2 t) \\
&+ \frac{3}{4}k_2A_1^2A_2\cos((2\omega_1-\omega_2)t) \\
&+ \frac{3}{4}k_2A_1A_2^2\cos((2\omega_2-\omega_1)t)
\end{aligned}
\tag{28}
$$

Then, converting the signals down to baseband into I and Q channels by multiplying x by $\cos(\omega_0 t)$ and $-\sin(\omega_0 t)$, we get

$$
\begin{aligned}
x =& (k_1 A_1 + \frac{3}{4} k_2 A_1^3 + \frac{3}{2} k_2 A_1 A_2^2) e^{j(\omega_1 t)} \\
&+ (k_1 A_2 + \frac{3}{4} k_2 A_2^3 + \frac{3}{2} k_2 A_1^2 A_2) e^{j(\omega_2 t)} \\
&+ \frac{3}{4} k_2 A_1^2 A_2 e^{j((2\omega_1 - \omega_2) t)} \\
&+ \frac{3}{4} k_2 A_1 A_2^2 e^{j((2\omega_2 - \omega_1) t)}
\end{aligned}
\tag{29}
$$

where $e^{-j\theta} = \cos(\theta) - j\sin(\theta)$.

The question then becomes, how do you remove the third-order spurs without having *a priori* knowledge of which tones are your desired tones? The first step is to have an extremely linear auxilliary channel as well, whose baseband received signal is shown in equation 30.

$$
d(t) = A_1 e^{(\omega_1 t)} + A_2 e^{(\omega_2 t)} \tag{30}
$$

The third-order nonlinearities were applied by cubing the two cosines, but now the receive data is in complex baseband and wont reproduce the same nonlinearities simply by cubing the signals. Though, it can be shown that by taking the absolute value of the complex exponential and squaring that, then multiplying the result by the complex signals produces the same received nonlinearities. Equations 31-33 show how this method produces the same nonlinearities that fell in-band, thus, giving us the ability to correct the undesired tones.

$$
\begin{aligned}
z =& e^{j\theta} + e^{j\phi} \\
&= \cos\theta + \cos\phi + j\sin\theta + j\sin\phi
\end{aligned}
\tag{31}
$$

$$\begin{aligned} |z| =& \sqrt{(\cos\theta + \cos\phi)^2 + (\sin\theta + \sin\phi)^2} \\ =& \sqrt{2 + 2\cos(\theta - \phi)} \end{aligned} \tag{32}$$

$$\begin{aligned} |z|^2 z =& (2 + 2\cos(\theta - \phi))(\cos\theta + \cos\phi + j\sin\theta + j\sin\phi) \\ =& 3e^{j\theta} + 3e^{j\phi} + e^{j(2\theta - \phi)} + e^{j(2\phi - \theta)} \end{aligned} \tag{33}$$

The auxiliary channel is helpful for training the coefficients, but having an auxiliary channel for each nonlinear channel in the array defeats the cost-effective purpose of using nonlinear channels. Therefore, it is essential that the correction is applied to the received nonlinear channels, only using the auxiliary channel for training.

3.1.1 Memory Polynomial Model

There are many different ways to model nonlinear systems, the most general, and complex being the Volterra series. A more simplified version of the Volterra series can be found in the memory polynomial (MP), which can make a good estimation of the nonlinear characteristics of a system.

$$y(n) = \sum_{p=0}^{P-1} \sum_{m=0}^{M-1} w_{pm} x(n-m)|x(n-m)|^{2p} \tag{34}$$

The memory polynomial, given in equation 34, can have P power terms and M memory terms. Only the odd power terms are included, for example if we have $P = 3$ then the MP will be made up of the first, third, and fifth orders of x. The M memory terms, on the other hand, can be thought of as taps on an FIR filter, and help to remove frequency dependence, when they are applied to NLEQ. Lastly,

when p and m are both zero we have the linear term of the MP.

In order to apply the LMS algorithm to the MP model we must set up the weights $\mathbf{w}(n)$ and the basis of the distorted signal $\mathbf{x}(n)$ in a specific way. Since the basis is based on the MP, we let

$$\mathbf{W}, \mathbf{X} \in \mathbb{C}^{P \times M} \tag{35}$$

where the p^{th} by m^{th} entry in the weight matrix $\mathbf{W}$ corresponds to the MP weight, w_{pm}; with the distorted signal matrix $\mathbf{X}$ corresponding in the same way. The initial weight matrix must be conditioned in such a way as to give a low enough initial error to allow the algorithm to step toward convergence of the weights, rather than cause the weights to diverge to extremely large numbers. It was determined that the best starting point was the assumption of linearity; that is, setting the linear weight to one and the rest of the weights to zero.

$$\begin{aligned} &w_{1,1} = 1 \\ &w_{i,j} = 0, \qquad\qquad i = 2, 3, ..., P; \quad j = 2, 3, ..., M \end{aligned} \tag{36}$$

Finally, the step size μ had to be determined to allow the weights to converge efficiently. The larger the step size is, the quicker it will converge, but it also becomes more likely that the weights diverge, or oscillate around the minimum. It was determined that the absolute value of the largest value of the nonlinear signal created a good basis for which to create the step size matrix.

$$\alpha = max(\mid \mathbf{x} \mid) \tag{37}$$

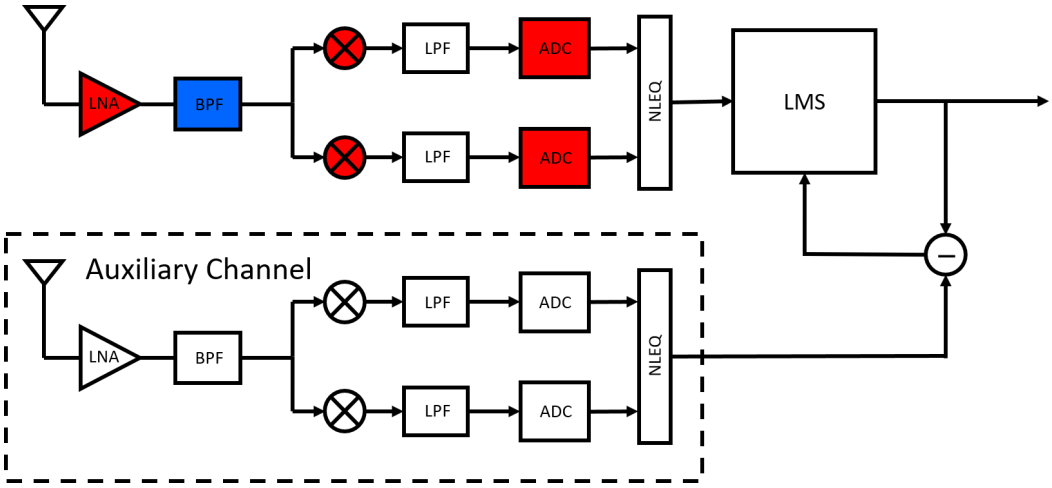


Figure 12: A block diagram of the implementation of LMS for a nonlinear channel and an auxiliary channel.

$$\mu_{i,j} = a_{i,j}\frac{1}{\alpha^{2i}}, \qquad i = 1, 2, ..., P; \quad j = 1, 2, ..., M \tag{38}$$

where $a_{i,j}$ are predetermined values that are used to meet convergence requirements.

Due to the iterative nature of LMS, the entire signal is not corrected at once, rather k samples are used for each iteration, allowing the weights to converge, and adapt to potential changes in the system. This adaptability is a very advantageous characteristic of LMS, as will be shown later in this paper.

3.2 LMS Training

An example of the nonlinear-auxiliary channel pair is shown in Figure 12, where the output of the nonlinear channel goes through the LMS filter for NLEQ, and the output is then compared with the output of the auxiliary channel, assuming the two channels are phase-aligned, giving an error that is then used to update the weights.

3.2.1 Convergence Considerations

The choice of the step-size μ is very critical for achieving fast convergence. If the step-size is too large, the algorithm will never converge to a solution and if it does, it will take an extremely large amount of computational time. If the step-size is too small, the algorithm could diverge because it oversteps the solution and then continues to do so until it diverges. Therefore, it is very important to choose the step-size that is best for the specific system.

3.2.2 Adaptability of LMS

A very important characteristic of LMS, especially compared to previous NLEQ methods such as WLS, is that it is an iterative solution with a very computationally simple update. This inherent iterative nature makes LMS the ideal solution for receive-side NLEQ because it is able to adapt to any changes in the system and its characteristics, and to the received data, in general. Some of these system changes, as discussed in Chapter 2, consist of device temperature and operating frequency. Changes in frequency consist not only of center frequency of the array, but also received frequency, in general, which is very common for modern UWB systems, especially in situations with many interferers, as is often the case in modern communications systems and EW.

4 Nonlinear Equalization for Digital Arrays

Using NLEQ for a single channel was discussed in the previous chapter, where the nonlinear channel was corrected using an LMS filter and a linear auxiliary channel for coefficient training. In the end, though, NLEQ will be used for an entire digital array, and there will be many fewer auxiliary channels than nonlinear channels. This expansion from the correction of a single channel to an entire digital array brings many complications and trade-offs. It is not computationally efficient to train each individual element of the array for NLEQ coefficients, but element differences such as the device variations discussed in Chapter 2 make it difficult to use coefficients derived from one element to correct another element.

It is well known, and was shown in Chapter 2, that each element in an array doesn't operate at the same temperature, and this variation in temperature is due to multiple factors, such as the element's location on the array, the amount of power being pushed through the specific element, the cooling system, etc. This requires channels of the same temperature to be corrected with the same NLEQ coefficients. There is also the issue of device process variation, which was shown to easily be 3dB or greater. Obviously, there is a huge trade-off to be made between number of elements used to train the coefficients, more of which can greatly increase computational costs, and the accuracy of the coefficients to every element in the array.

One solution to this trade-off is to simply use one element for coefficient training and use the coefficients from that single element to correct every element in the array. Obviously, this solution is the most computationally efficient solution, but with poor decorrelation effectiveness for arrays with large temperature and device variation, especially for larger arrays. Another solution is to use a sufficient number of elements for coefficient training and then to average those coefficients and apply those averaged coefficients to the entire array. The effectiveness of each of these

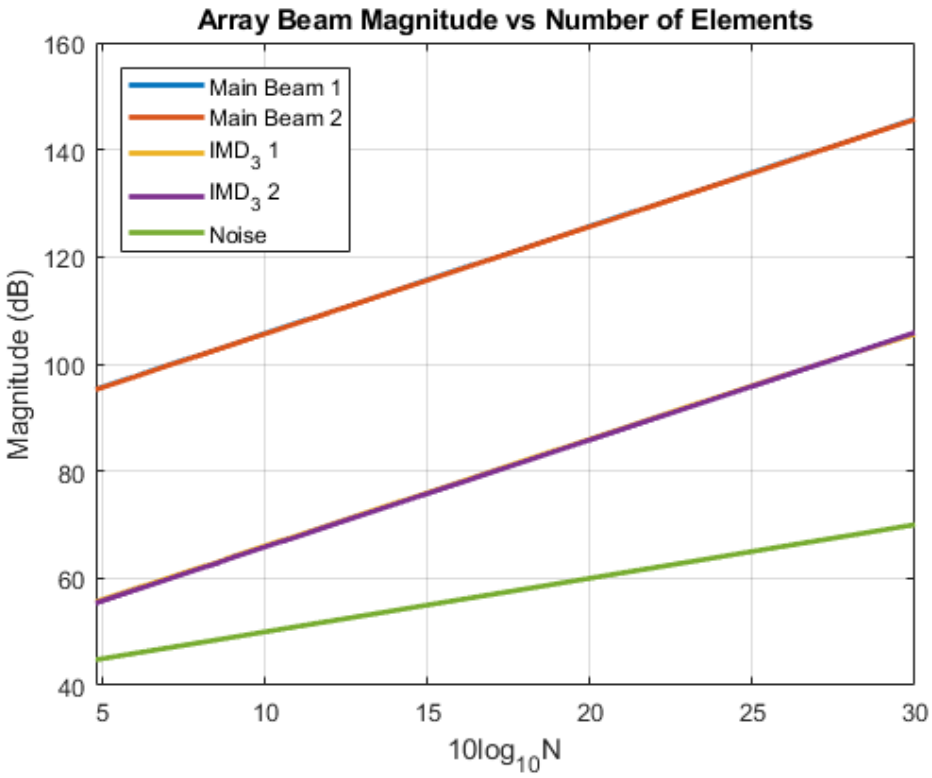


Figure 13: An example of what is correlated and decorrelated in an array.

methods depends heavily on the size of the array (number of elements), and the channel standard deviation.

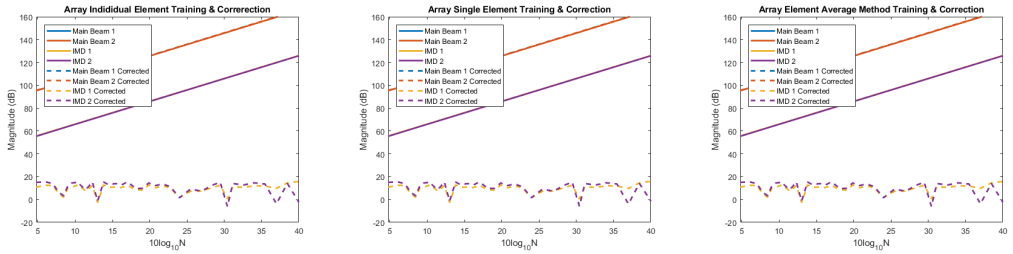
4.1 Correlation of IMD

It was shown in [2], [20] that third-order IMD, and thus all nonlinear spurs, correlate to specific DoA in an array. This correlation was shown in Section 2.2, and the ultimate goal of NLEQ for an array is to mitigate the spurs at the channel level and then decorrelate them at the array level. It is also known that received signals correlate at the array level, on a scale of $20log_{10}N$, where N is the number of elements in the array. Noise, on the other hand, is decorrelated at the array level, scaling by $10log_{10}N$, due to its random nature. Even-order spurious products can

also be decorrelated at the array level, as was deomonstrated in [2], with methods such as random phasing of the local oscillator (LO) on each channel. Figure 13 shows the correlation of the two received tones and their third-order IMD products, each of which scale by $20log_{10}N$, compared to noise, which is shown to only scale by $10log_{10}N$. One conclusion that can be drawn from this figure is that while the signal-to-noise ration (SNR) of the array increases with more elements, the SFDR does not change.

4.2 Decorrelation of IMD

Decorrelation of spurs in an array is simple when NLEQ can mitigate the spurs completely, pushing them into noise, but for a real array with channel variations, the NLEQ coefficients used on each element would also need to be trained on each respective element. But, as previously stated, this is not very ideal as it required a great amount of computational resources, which not only add time and power needs, but also cost. Ideally, training would be done on a small group of elements so that the mean coefficients could be derived and then applied to the entire array. Unfortunately, this is much more complicated than it appears because of the aforementioned channel variations. The simplest way to model these variations is when the errors are independent and identically distributed (iid) complex Gaussian random variables, which makes finding the mean very simple, but when the NLEQ methods discussed in Chapter 3 are used, these coefficients are then cubed for third-order correction, and further powers are applied for correcting higher order nonlinearities. It was shown in [25] that the cube of a Gaussian distribution is indeterminate, shown in equation 41, making it difficult to find the coefficients that would correct the average element.

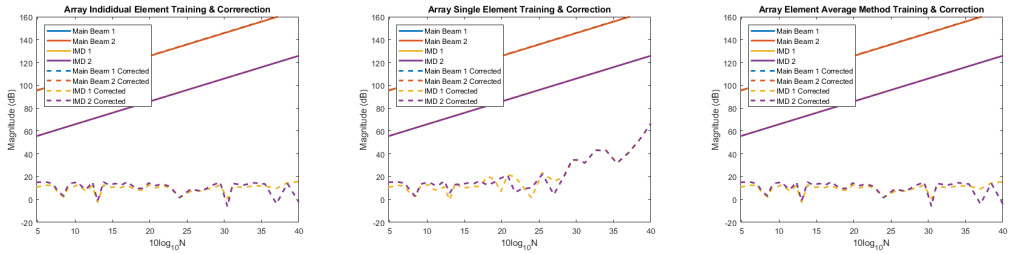


(a) (b) (c)

Figure 14: Simulated examples of array-level decorrelation of odd-order spurs from arrays of different sizes, with identical channels. (a) shows the results when each individual element is trained and corrected, (b) shows the results of training the coefficients on one element and using those results to correct all of the elements in the array, and (c) shows the results from averaging the NLEQ coefficients from the array elements and using the averaged coefficients to correct the entire array. Since the elements are identical, each of the methods achieves perfect decorrelation of the spurs.

4.2.1 Identical Array Elements

As previously discussed, in a real system there will be performance differences between elements in an array, but ideally the elements would be identical. For an array with identical elements, the decorrelation of odd-order spurs can be achieved using the NLEQ coefficients from a single channel, as shown in Figure 14 (b). The simulated channels had a simple third-order power series nonlinearity, with a linear coefficient of 1 and a third-order coefficient of -0.01. Since no noise was added to the simulation, WLS was used to find the NLEQ coefficients. As Figure 14 shows, individual correction of each channel, training on a single element and using those coefficients to correct the entire array, and finding the average coefficients from the array and using those to correct all of the elements have the exact same performance. Each method is able to completely decorrelate the third-order spurs when there are not channel variations in the array.



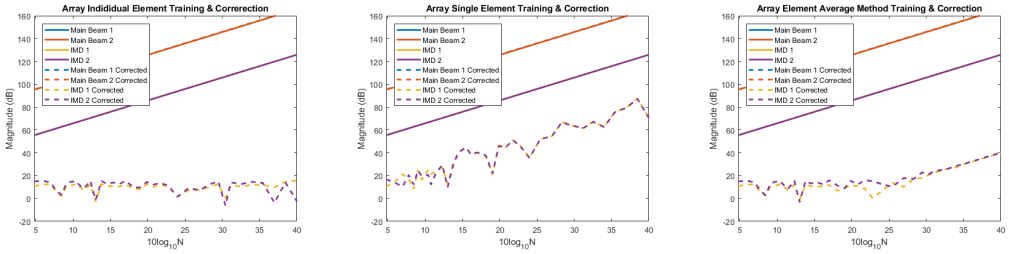
(a) (b) (c)

Figure 15: Simulated examples of array-level decorrelation of odd-order spurs from arrays of different sizes, with Gaussian distributed channels with a standard deviation of 0.01. (a) shows the results when each individual element is trained and corrected, (b) shows the results of training the coefficients on one element and using those results to correct all of the elements in the array, which decorrelated the spurs for array sizes less than 600 elements, and (c) shows the results from averaging the NLEQ coefficients from the array elements and using the averaged coefficients to correct the entire array, which decorrelated the spurs for all simulated array sizes.

4.2.2 Gaussian Distributed Array Elements

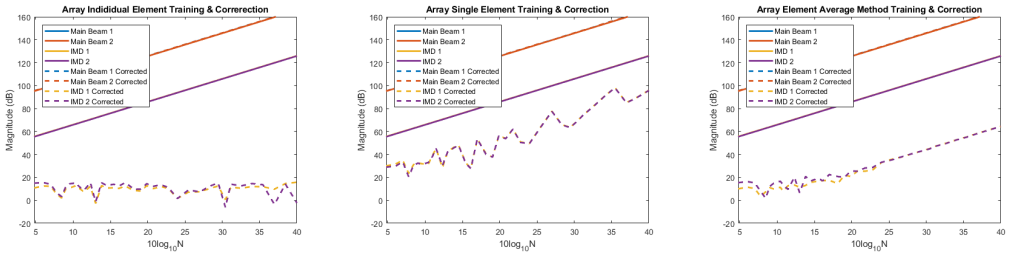
It has been mentioned many times during this thesis that each channel in a real digital array system will suffer from performance variations. It is a very important fact that needs to be taken into account when designing an array and when attempting to apply receive-side NLEQ. The previous section of this chapter showed that the odd-order spurs of an array with identical channels are decorrelated with the NLEQ coefficients of a single channel from the array, but that is not the case when channel variations are introduced into the array. The performance variations in digital array channels can simply be modeled as iid Gaussian random variables. This section will go over the performance limitations of array-level NLEQ when the channel nonlinearities are Gaussian distributed. A mathematical model and proposed solution are also derived and shown.

Figures 15 - 18 show the amount of decorrelation that can be achieved for digital arrays with Gaussian nonlinear channel characteristics, with NLEQ being applied to each individual channel, using a single channel for training and applying those



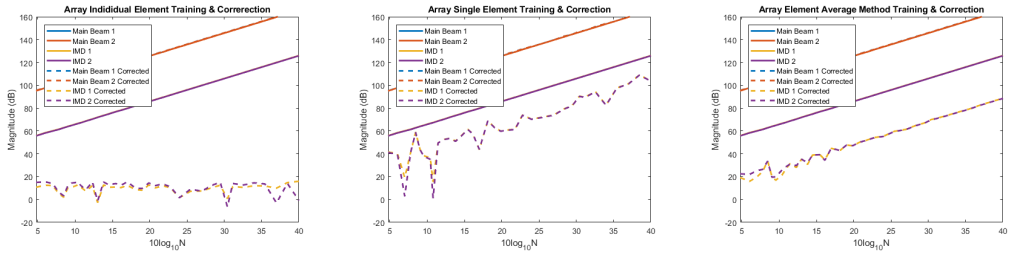
(a) (b) (c)

Figure 16: Simulated examples of array-level decorrelation of odd-order spurs from arrays of different sizes, with Gaussian distributed channels with a standard deviation of 0.05. (a) shows the results when each individual element is trained and corrected, (b) shows the results of training the coefficients on one element and using those results to correct all of the elements in the array, which only decorrelated the spurs for arrays with less than 10 elements, and (c) shows the results from averaging the NLEQ coefficients from the array elements and using the averaged coefficients to correct the entire array, which decorrelated the spurs for arrays with less than 600 elements.



(a) (b) (c)

Figure 17: Simulated examples of array-level decorrelation of odd-order spurs from arrays of different sizes, with Gaussian distributed channels with a standard deviation of 0.1. (a) shows the results when each individual element is trained and corrected, (b) shows the results of training the coefficients on one element and using those results to correct all of the elements in the array, which never decorrelated the spurs for any of the simulated array sizes, and (c) shows the results from averaging the NLEQ coefficients from the array elements and using the averaged coefficients to correct the entire array, which decorrelated the spurs for arrays with less than 30 elements.



(a) (b) (c)

Figure 18: Simulated examples of array-level decorrelation of odd-order spurs from arrays of different sizes, with Gaussian distributed channels with a standard deviation of 0.2. (a) shows the results when each individual element is trained and corrected, (b) shows the results of training the coefficients on one element and using those results to correct all of the elements in the array, and (c) shows the results from averaging the NLEQ coefficients from the array elements and using the averaged coefficients to correct the entire array. Only the individual element training decorrelated the spurs for the simulated arrays; the other two methods failed to decorrelate the spurs for even the arrays with the fewest elements.

coefficients to all of the other channels, and taking the mean NLEQ coefficients and applying those to the array, for different standard deviations. For all standard deviations, individual training and correction achieve the same amount of decorrelation as each other and also the same as the identical element array shown in Figure 14. Single channel training is effective for systems with few elements and very low standard deviation, but once the standard deviation is increased to any realistic values, the corrected spurs scale with $20log_{10}N$, which implies correlation. That's because only the parts of the channel coefficients that are correlated with the trained element are corrected, while the remaining errors correlate from all of the non-trained channels. Finally, using the average NLEQ coefficients appears to work well for arrays with less than 10 elements, but as the number of elements is increased the corrected spurs begin to follow a $20log_{10}N$ path. Thus, using the coefficients from a single element isn't an effective solution, as was predicted, but using the average coefficients also fails to decorrelate the odd-order spurs. It is

apparent that averaging nonlinear Gaussian random variables does not lead to the true mean of the original distribution. This is due to the nonlinearities introduced during correction, which leads to higher odd-order powers of the Gaussians, which are indeterminant as shown in [25].

To determine the solution for array level decorrelation of odd-order spurs, we must first derive a model for the Gaussian distributed channel coefficients and the coefficients and frequencies that are produced when equation 29 is cubed (shown again below in equation 39).

$$\begin{aligned} x =& (k_1 A_1 + \frac{3}{4} k_2 A_1^3 + \frac{3}{2} k_2 A_1 A_2^2) e^{j(\omega_1 t)} \\ &+ (k_1 A_2 + \frac{3}{4} k_2 A_2^3 + \frac{3}{2} k_2 A_1^2 A_2) e^{j(\omega_2 t)} \\ &+ \frac{3}{4} k_2 A_1^2 A_2 e^{j((2\omega_1 - \omega_2) t)} \\ &+ \frac{3}{4} k_2 A_1 A_2^2 e^{j((2\omega_2 - \omega_1) t)} \end{aligned} \tag{39}$$

Its obvious that cubing equation 39 creates an extremely large number of terms. For example, cubing a four-term-polynomial will produce twenty terms, but in this case, since we are cubing exponentials/sinusoids it can lead to an even greater number of terms. Furthermore, the first two terms of equation 39 have a coefficient made up of a tri-nomial, which will have ten terms when cubed. Lastly, since these are exponentials and not cosines, we must first take the absolute value of x and square it, and then multiply that result by x, as shown in Chapter 3. The result of this is given in Table 4.

Tone	Coefficient	Angle
Main Tone 1	$[[k_1^3A_1^3+\frac{27}{64}k_2^3A_1^9+\frac{27}{8}k_2^3A_1^3A_2^6+\frac{9}{4}k_1^2k_2A_1^5+\frac{9}{2}k_1^2k_2A_1^3A_2^2+$ $\frac{27}{16}k_1k_2^2A_1^7+\frac{27}{4}k_1k_2^2A_1^3A_2^4+\frac{81}{32}k_2^3A_1^7A_2^2+\frac{81}{16}k_2^3A_1^5A_2^4+$ $\frac{27}{8}k_1k_2^2A_1^5A_2^2]+3[k_1^3A_1A_2^2+\frac{9}{16}k_1k_2^2A_1A_2^6+\frac{9}{4}k_1k_2^2A_1^5A_2^2+$ $\frac{3}{2}k_1^2k_2A_1A_2^4+3k_1^2k_2A_1^3A_2^2+\frac{3}{2}k_1k_2A_1A_2^3+\frac{3}{4}k_1^2k_2A_1^3A_2^2+$ $\frac{27}{64}k_2^3A_1^3A_2^6+\frac{27}{16}k_2^3A_1^7A_2^2+\frac{9}{8}k_1k_2^2A_1^3A_2^4+\frac{9}{4}k_1k_2^2A_1^5A_2^2+$ $\frac{9}{8}k_2^2A_1^3A_2^3+\frac{3}{2}k_1^2k_2A_1A_2^4+\frac{27}{32}k_2^3A_1A_2^8+\frac{27}{8}k_2^3A_1^5A_2^4+$ $\frac{9}{4}k_1k_2^2A_1^3A_2^4+\frac{9}{4}k_2^2A_1A_2^5]+3[\frac{9}{16}k_1k_2^2A_1^5A_2^2+\frac{27}{64}k_2^3A_1^7A_2^2+$ $\frac{27}{32}k_2^3A_1^5A_2^4]+3[\frac{9}{16}k_1k_2^2A_1^3A_2^4+\frac{27}{64}k_2^3A_1^5A_2^4+\frac{27}{32}k_2^3A_1^3A_2^6]+$ $2[\frac{3}{4}k_1^2k_2A_1A_2^4+\frac{27}{64}k_2^3A_1A_2^8+\frac{27}{16}k_2^3A_1^5A_2^3+\frac{9}{8}k_1k_2^2A_1A_2^6+$ $\frac{9}{4}k_1k_2^2A_1^3A_2^4+\frac{27}{16}k_2^3A_1^3A_2^6]+4[\frac{3}{4}k_1^2k_2A_1^3A_2^2+\frac{27}{16}k_1k_2^2A_1^3A_2^4+$ $\frac{27}{16}k_1k_2^2A_1^5A_2^2+\frac{27}{32}k_2^3A_1^7A_2^2+\frac{27}{32}k_2^3A_1^3A_2^6+\frac{135}{64}k_2^3A_1^5A_2^4]+$ $4[\frac{9}{16}k_1k_2^2A_1^3A_2^4+\frac{27}{63}k_2^3A_1^3A_2^6+\frac{27}{32}k_2^2A_1^5A_2^4]]$	$e^{j\omega_1 t}$
Main Tone 2	$[[k_1^3A_2^3+\frac{27}{64}k_2^3A_2^9+\frac{27}{8}k_2^3A_1^6A_2^3+\frac{9}{4}k_1^2k_2A_2^5+\frac{9}{2}k_1^2k_2A_1^2A_2^3+$ $\frac{27}{16}k_1k_2^2A_2^7+\frac{27}{4}k_1k_2^2A_1^4A_2^3+\frac{81}{32}k_2^3A_1^2A_2^7+\frac{81}{16}k_2^3A_1^4A_2^5+$ $\frac{27}{8}k_1k_2^2A_1^2A_2^5]+3[k_1^3A_1^2A_2+\frac{9}{16}k_1k_2^2A_1^6A_2+\frac{9}{4}k_1k_2^2A_1^2A_2^5+$ $\frac{3}{2}k_1^2k_2A_1^4A_2+3k_1^2k_2A_1^2A_2^3+\frac{3}{2}k_1k_2A_1^3A_2+\frac{3}{4}k_1^2k_2A_1^2A_2^3+$ $\frac{27}{64}k_2^3A_1^6A_2^3+\frac{27}{16}k_2^3A_1^2A_2^7+\frac{9}{8}k_1k_2^2A_1^4A_2^3+\frac{9}{4}k_1k_2^2A_1^2A_2^5+$ $\frac{9}{8}k_2^2A_1^3A_2^3+\frac{3}{2}k_1^2k_2A_1^4A_2+\frac{27}{32}k_2^3A_1^8A_2+\frac{27}{8}k_2^3A_1^4A_2^5+$ $\frac{9}{4}k_1k_2^2A_1^4A_2^3+\frac{9}{4}k_2^2A_1^5A_2]+2[\frac{9}{16}k_1k_2^2A_1^5A_2^2+\frac{27}{64}k_2^3A_1^7A_2^2+$ $\frac{27}{32}k_2^3A_1^5A_2^4]+3[\frac{9}{16}k_1k_2^2A_1^4A_2^3+\frac{27}{64}k_2^3A_1^4A_2^5+\frac{27}{32}k_2^3A_1^6A_2^3]+$ $3[\frac{9}{16}k_1k_2^2A_1^2A_2^5+\frac{27}{64}k_2^3A_1^2A_2^7+\frac{27}{32}k_2^3A_1^4A_2^5]+4[\frac{3}{4}k_1^2k_2A_1^2A_2^3+$ $\frac{27}{16}k_1k_2^2A_1^2A_2^5+\frac{27}{16}k_1k_2^2A_1^4A_2^3+\frac{27}{32}k_2^3A_1^6A_2^3+\frac{27}{32}k_2^3A_1^2A_2^7+$ $\frac{135}{64}k_2^3A_1^4A_2^5]+4[\frac{9}{16}k_1k_2^2A_1^4A_2^3+\frac{27}{63}k_2^3A_1^6A_2^3+\frac{27}{32}k_2^2A_1^4A_2^5]]$	$e^{j\omega_2 t}$

Third Order Difference	$+[[\frac{9}{64}k_2^3A_1^6A_2^3]+2[k_1^3A_1^2A_2+\frac{9}{16}k_1k_2^2A_1^6A_2+\frac{9}{4}k_1k_2^2A_1^2A_2^5+\frac{3}{2}k_1^2k_2A_1^4A_2+3k_1^2k_2A_1^2A_2^3+\frac{3}{2}k_1k_2A_1^3A_2+\frac{3}{4}k_1^2k_2A_1^2A_2^3+\frac{27}{64}k_2^3A_1^6A_2^3+\frac{27}{16}k_2^3A_1^2A_2^7+\frac{9}{8}k_1k_2^2A_1^4A_2^3+\frac{9}{4}k_1k_2^2A_1^2A_2^5+\frac{9}{8}k_2^2A_1^3A_2^3+\frac{3}{2}k_1^2k_2A_1^4A_2+\frac{27}{32}k_2^3A_1^8A_2+\frac{27}{8}k_2^3A_1^4A_2^5+\frac{9}{4}k_1k_2^2A_1^4A_2^3+\frac{9}{4}k_2^2A_1^5A_2]+3[\frac{9}{16}k_1k_2^2A_1^5A_2^2+\frac{27}{64}k_2^3A_1^7A_2^2+\frac{27}{32}k_2^3A_1^5A_2^4]+3[\frac{3}{4}k_1^2k_2A_1^2A_2^3+\frac{27}{64}k_2^3A_1^2A_2^7+\frac{27}{16}k_2^3A_1^6A_2^2+\frac{9}{8}k_1k_2^2A_1^2A_2^5+\frac{9}{4}k_1k_2^2A_1^4A_2^3+\frac{27}{16}k_2^3A_1^4A_2^5]+3[\frac{27}{64}k_2^3A_1^3A_2^6]+4[\frac{3}{4}k_1^2k_2A_1^2A_2^3+\frac{27}{16}k_1k_2^2A_1^2A_2^5+\frac{27}{16}k_1k_2^2A_1^4A_2^3+\frac{27}{32}k_2^3A_1^6A_2^3+\frac{27}{32}k_2^3A_1^2A_2^7+\frac{135}{64}k_2^3A_1^4A_2^5]]$	$e^{j(2\omega_1-\omega_2)t}$
4	$+[[\frac{9}{64}k_2^3A_1^3A_2^6]+3[\frac{3}{4}k_1^2k_2A_1A_2^4+\frac{27}{64}k_2^3A_1A_2^8+\frac{27}{16}k_2^3A_1^5A_2^3+\frac{9}{8}k_1k_2^2A_1A_2^6+\frac{9}{4}k_1k_2^2A_1^3A_2^4+\frac{27}{16}k_2^3A_1^3A_2^6]+[k_1^3A_1A_2^2+\frac{9}{16}k_1k_2^2A_1A_2^6+\frac{9}{4}k_1k_2^2A_1^5A_2^2+\frac{3}{2}k_1^2k_2A_1A_2^4+3k_1^2k_2A_1^3A_2^2+\frac{3}{2}k_1k_2A_1A_2^3+\frac{3}{4}k_1^2k_2A_1^3A_2^2+\frac{27}{64}k_2^3A_1^3A_2^6+\frac{27}{16}k_2^3A_1^7A_2^2+\frac{9}{8}k_1k_2^2A_1^3A_2^4+\frac{9}{4}k_1k_2^2A_1^5A_2^2+\frac{9}{8}k_2^2A_1^3A_2^3+\frac{3}{2}k_1^2k_2A_1A_2^4+\frac{27}{32}k_2^3A_1A_2^8+\frac{27}{8}k_2^3A_1^5A_2^4+\frac{9}{4}k_1k_2^2A_1^3A_2^4+\frac{9}{4}k_2^2A_1A_2^5]+3[\frac{3}{4}k_1^2k_2A_1A_2^4+\frac{27}{64}k_2^3A_1A_2^8+\frac{27}{16}k_2^3A_1^5A_2^3+\frac{9}{8}k_1k_2^2A_1A_2^6+\frac{9}{4}k_1k_2^2A_1^3A_2^4+\frac{27}{16}k_2^3A_1^3A_2^6]+3[\frac{27}{64}k_2^3A_1^5A_2^4]+4[\frac{3}{4}k_1^2k_2A_1^3A_2^2+\frac{27}{16}k_1k_2^2A_1^3A_2^4+\frac{27}{16}k_1k_2^2A_1^5A_2^2+\frac{27}{32}k_2^3A_1^7A_2^2+\frac{27}{32}k_2^3A_1^3A_2^6+\frac{135}{64}k_2^3A_1^5A_2^4]]$	$e^{j(2\omega_2-\omega_1)t}$
5	$+[2[\frac{9}{16}k_1k_2^2A_1^5A_2^2+\frac{27}{64}k_2^3A_1^7A_2^2+\frac{27}{32}k_2^3A_1^5A_2^4]+2[\frac{3}{4}k_1^2k_2A_1A_2^4+\frac{27}{64}k_2^3A_1A_2^8+\frac{27}{16}k_2^3A_1^5A_2^3+\frac{9}{8}k_1k_2^2A_1A_2^6+\frac{9}{4}k_1k_2^2A_1^3A_2^4+\frac{27}{16}k_2^3A_1^3A_2^6]+4[\frac{3}{4}k_1^2k_2A_1^3A_2^2+\frac{27}{16}k_1k_2^2A_1^3A_2^4+\frac{27}{16}k_1k_2^2A_1^5A_2^2+\frac{27}{32}k_2^3A_1^7A_2^2+\frac{27}{32}k_2^3A_1^3A_2^6+\frac{135}{64}k_2^3A_1^5A_2^4]+4[\frac{9}{16}k_1k_2^2A_1^3A_2^4+\frac{27}{63}k_2^3A_1^3A_2^6+\frac{27}{32}k_2^2A_1^5A_2^4]]$	$e^{j(3\omega_1-2\omega_2)t}$

6	$[2[\frac{3}{4}k_1^2k_2A_1^2A_2^3+\frac{27}{64}k_2^3A_1^2A_2^7+\frac{27}{16}k_2^3A_1^6A_2^2+\frac{9}{8}k_1k_2^2A_1^2A_2^5+\frac{9}{4}k_1k_2^2A_1^4A_2^3+\frac{27}{16}k_2^3A_1^4A_2^5]+2[\frac{9}{16}k_1k_2^2A_1^2A_2^5+\frac{27}{64}k_2^3A_1^2A_2^7+\frac{27}{32}k_2^3A_1^4A_2^5]+4[\frac{9}{16}k_1k_2^2A_1^4A_2^3+\frac{27}{63}k_2^3A_1^6A_2^3+\frac{27}{32}k_2^2A_1^4A_2^5]+4[\frac{3}{4}k_1^2k_2A_1^2A_2^3+\frac{27}{16}k_1k_2^2A_1^2A_2^5+\frac{27}{16}k_1k_2^2A_1^4A_2^3+\frac{27}{32}k_2^3A_1^6A_2^3+\frac{27}{32}k_2^3A_1^2A_2^7+\frac{135}{64}k_2^3A_1^4A_2^5]]$	$e^{j(3\omega_2-2\omega_1)t}$
7	$[2[\frac{9}{16}k_1k_2^2A_1^4A_2^3+\frac{27}{64}k_2^3A_1^4A_2^5+\frac{27}{32}k_2^3A_1^6A_2^3]+4[\frac{9}{16}k_1k_2^2A_1^4A_2^3+\frac{27}{63}k_2^3A_1^6A_2^3+\frac{27}{32}k_2^2A_1^4A_2^5]]$	$e^{j(4\omega_1-3\omega_2)t}$
8	$[2[\frac{9}{16}k_1k_2^2A_1^3A_2^4+\frac{27}{64}k_2^3A_1^5A_2^4+\frac{27}{32}k_2^3A_1^3A_2^6]+4[\frac{9}{16}k_1k_2^2A_1^3A_2^4+\frac{27}{63}k_2^3A_1^3A_2^6+\frac{27}{32}k_2^2A_1^5A_2^4]]$	$e^{j(4\omega_2-3\omega_1)t}$
9	$2[\frac{27}{64}k_2^3A_1^5A_2^4]$	$e^{j(5\omega_1-4\omega_2)t}$
10	$2[\frac{27}{64}k_2^3A_1^3A_2^6]$	$e^{j(5\omega_2-4\omega_1)t}$

Table 4: The different terms, and their respective coefficients, that show up when applying a third-order correction to a third-order nonlinearity.

Now let k_1 and k_2 from equation 39 be two iid Gaussian random variables with means of μ_1 and μ_2 and standard deviations of σ_1 and σ_2, given by

$$f(k_n|\mu_n,\sigma_n)=\frac{1}{\sqrt{2\pi\sigma_n^2}}e^{-\frac{(k_n-\mu_1)^2}{2\sigma_n^2}} \tag{40}$$

We can see from table 4 that there are four different cubes of the two Gaussian channel coefficients, which are k_1^3, k_2^3, $k_1^2k_2$, and $k_1k_2^2$, and can then be represented as

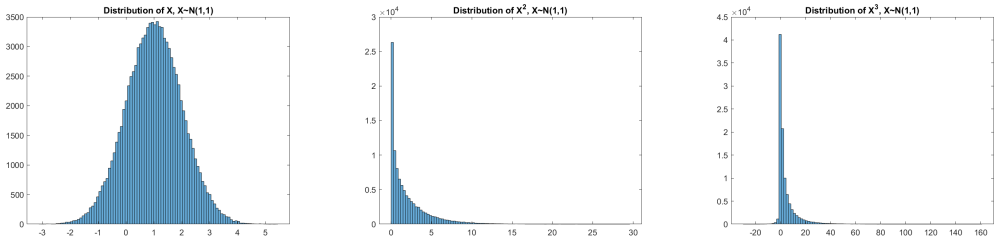
$$f(k_n^3|\mu_n,\sigma_n) = \frac{1}{3\sqrt{2\pi\sigma_n^2}}|k_n|^{-\frac{2}{3}}e^{-\frac{(k_n^{\frac{1}{3}}-\mu_n)^2}{2\sigma_n^2}} \tag{41}$$

and

$$f(k_n^2k_m|\mu_n,\sigma_n,\mu_m,\sigma_m) = \frac{1}{4\pi\sigma_n^2\sigma_m}|k_n|^{-\frac{1}{2}}e^{-\frac{1}{2}(\frac{(k_n^{\frac{1}{2}}-\mu_n)^2}{\sigma_n^2}+\frac{(k_m-\mu_m)^2}{\sigma_m^2})} \tag{42}$$

To demonstrate why these probability density functions (PDFs) are important, we will use an example of a random variable $X \sim N(1,1)$, which can represent the approximate linear term of a device with unit gain. Figure 19 shows the histograms of the outcomes of 10001 trials; the first figure shows a Gaussian distribution with a mean of one. The second figure shows the square of the random variable, which is all positive and has a mean of about two. Lastly, the third figure shows the cube of the random variable and its distribution with a mean of about four. The mean of the random variable represents the mean of a device or system, or the expected performance of the device; in this case the mean is one. The square and the cube of the value one are both one, which is different than the mean of the square and the cube of the Gaussian distribution with a mean of one. Therefore, we can say that for a Gaussian distributed random variable with a non-zero mean, the means of the square and the cube of the distribution do not follow the square and the cube of the mean of the distribution. Thus, it is very important to realize that the coefficients are calculated using the cube of the random variable coefficients that define the channels. Thus, taking the mean of the trained NLEQ coefficients of elements of an array is not the same as the true NLEQ coefficients of the mean element.

One proposed solution is to take the nth-root of the coefficients, taking the mean of those values, and then raising those coefficients to their respective powers. This

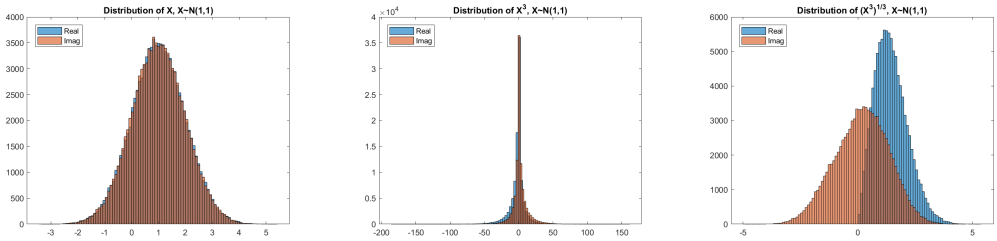


(a) (b) (c)

Figure 19: Histograms of (a) a Gaussian distribution with a mean and variance of one, (b) the square of the previous distribution, and (c) the cube of the same distribution.

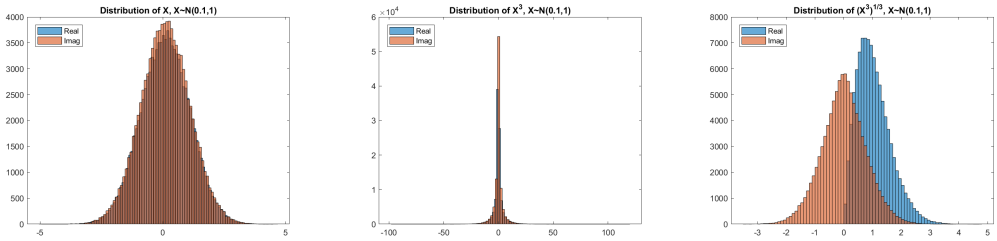
is a valid solution for real-valued Gaussian random variables, but digital array receivers deal with complex numbers. Figures 20 to 22 show the distributions of complex-Gaussian random variables with variance of one and means of one, one-tenth, and ten, respectively. It is quite apparent that taking the cube-root of the cube of the complex-Guassian random variable does not reproduce the original distribution, at least for random variables with means closer to zero. Figure 22, with a mean of 10+j10, shows that the distribution is almost completely reconstructed the original distribution. In this case, the original random variable had a mean of 9.999 + j10.0052 and the reconstructed random variable had a mean of 9.9991 + j10.0010; the means are nearly exactly the same. This is a stark contrast to the random variable with a mean of one, shown in figure 20, which had an original mean of 1.0050 + j0.9983 and a reconstructed mean of 1.4429 + j0.1718, and the random variable with a mean of one-tenth, which had an original mean of 0.1005 + j0.0987 and a reconstructed mean of 1.0413 - j0.0048. Therefore, we can conclude that this method is most effective for random variables which have means greater than one, but we can also see that taking the cube-root can bring the mean of the random variable closer to the original mean in the case of the random variable with a mean of one, but further separates the mean of the cube-rooted random variable from the true

mean in the case of the random variable with a mean of one-tenth. This proposed solution could be an effective way to determine the mean NLEQ coefficients for an array, depending on the distributions of the channel coefficients.



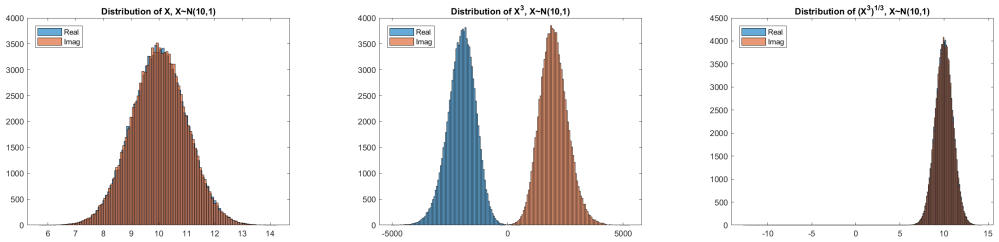
(a) (b) (c)

Figure 20: Histograms of (a) a complex Gaussian distribution with a mean and variance of one, (b) the cube of the previous distribution, and (c) the cube-root of the cubed distribution.



(a) (b) (c)

Figure 21: Histograms of (a) a complex Gaussian distribution with a mean of 0.1 and variance of one, (b) the cube of the previous distribution, and (c) the cube-root of the cubed distribution.



(a) (b) (c)

Figure 22: Histograms of (a) a complex Gaussian distribution with a mean of ten and variance of one, (b) the cube of the previous distribution, and (c) the cube-root of the cubed distribution.

5 Results

The methods presented in the previous chapters gave a proposed solution to NLEQ and its application for digital arrays. Here we gathered data from two different nonlinear digital array channels to determine the robustness of the LMS adaptive solution, with respect to changes in device temperature and frequency, and to determine how necessary adaptability is. Finally, simulations of an entire digital array with nonlinear channels were done to determine the extent to which decorrelation of spurs could be achieved.

5.1 LNA Testbed Setup

In this section, data gathered from an Analog Devices AD9371 transceiver, using one of its transmitters (Tx1) and both receive channels (Rx1, Rx2) to represent the auxiliary and nonlinear channel, respectively; shown in Figure 23. The Tx channel fed a band-limited additive white Gaussian noise (AWGN) 20MHz signal, centered in the band, at 2.7GHz, into a power divider. One output of the power divider was fed into the auxiliary channel and the other output was fed into the pre-amplifier, a MiniCircuits XRL-3500+, the nonlinear-amplifier, a MiniCircuits ZJL-3G+, and then into the nonlinear channel. Attenuators were placed before both Rx channels so that their respective signals would have the same amplitude and to not cause any nonlinearities in the receivers. The lengths of the cables of both Rx channels were also adjusted as to remove any potential time-delay between the two channels.

The IQ data collected from each channel contained 2^{20} samples (8533μs of data). The Fourier transform of the collected data from both channels is shown in Figure 24. The nonlinear channel has third-order IMD products of about 20dB above the noise floor; the goal of the correction will be to mitigate these distortion effects. The initial error of the nonlinear data was -4.4670 dB, as determined by an

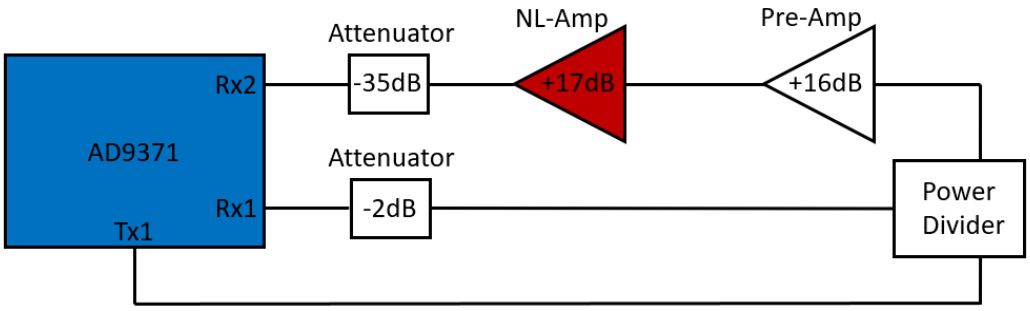


Figure 23: A block-diagram of the test setup for gathering data of a digital array channel with its front-end amplifier operating in the nonlinear region.

equation used in [12].

$$error = 10\log_{10}\left(\frac{E\{e^2\}}{E\{d^2\}}\right) \quad dB \tag{43}$$

5.1.1 NLEQ on LNA Nonlinearities

The LMS algorithm for NLEQ was then run on the collected data set, with 80 samples used per iteration. Simulations were done to determine that 80 samples per iteration was the ideal number, creating quicker convergence as fewer iterations were needed to be able to iterate through the entire data set, but the error was not skewed by using too many samples. Five memory terms (taps on an FIR filter) and six power terms (up to the 11th order) were used for the correction coefficients. The algorithm took 9.8930 seconds to iterate through the entire data set; the waveform corrected with the calibrated coefficients has an error of -36.2664 dB, an improvement of 31.7994 dB, shown in Figure 25 (a).

Figure 25 (b) is the power-in vs power-out plot of the distorted and corrected data. The nonlinear channel has a lot of smearing, especially at the lower power levels, that is corrected by LMS. This smearing is caused by frequency dependence in the nonlinear channel. The curve at high power levels in the nonlinear channel is

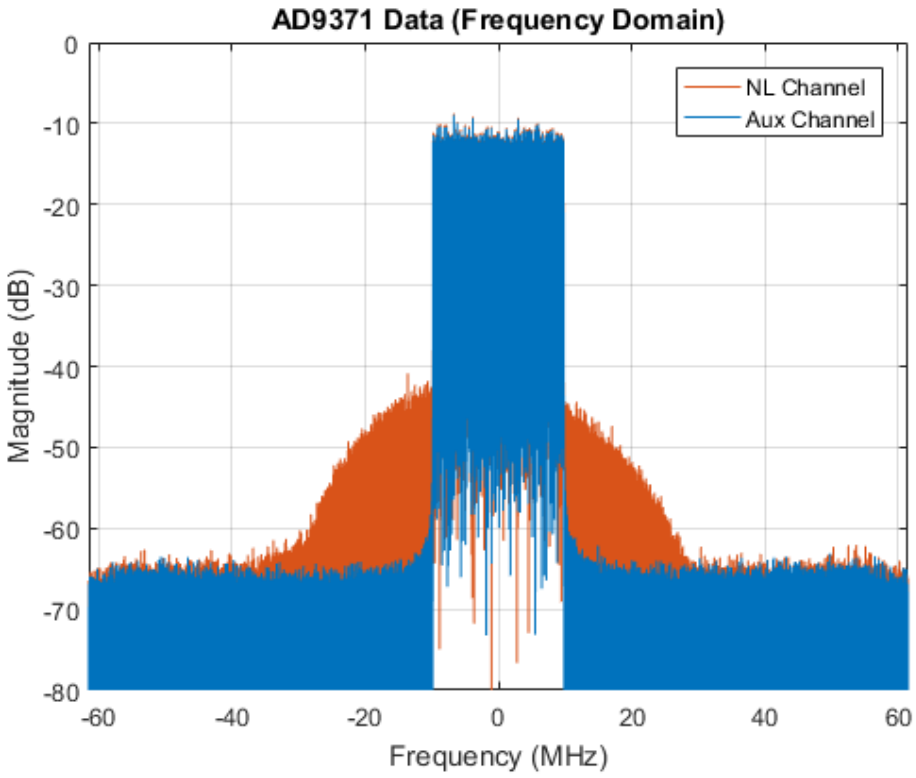
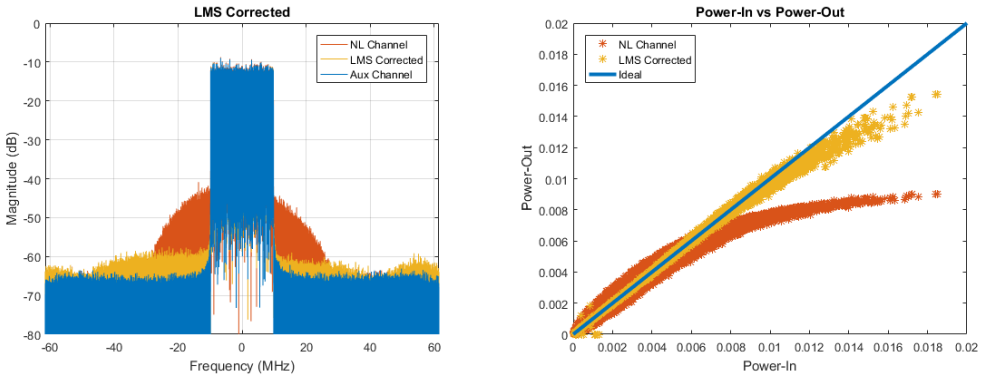
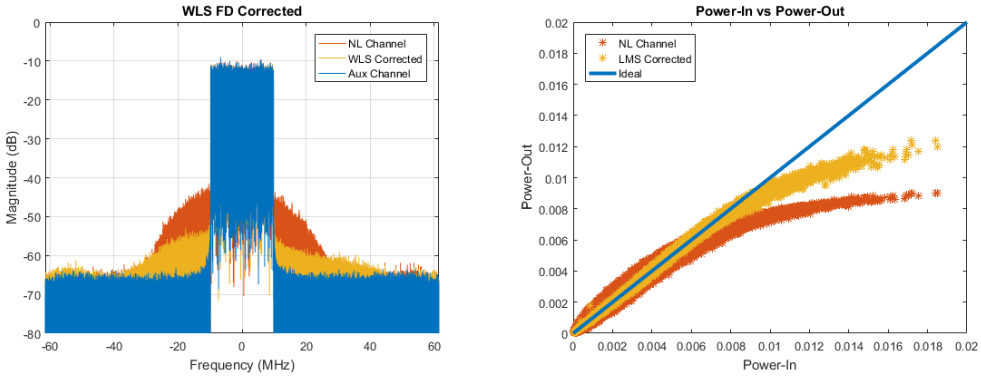


Figure 24: The frequency domain of the data gathered from the AD9371 with its front-end amplifier operating in the nonlinear region, for the nonlinear channel, and with the auxiliary amplifier operating linearly.



(a) (b)

Figure 25: Results from using LMS as a solution for NLEQ.



(a) (b)

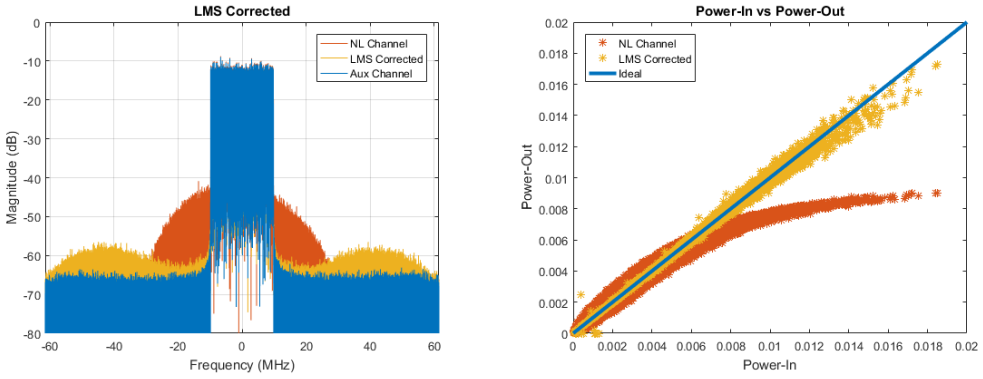
Figure 26: Results from using WLS as a solution for NLEQ.

straightened out with the LMS correction. This reduction in the curve is due to the mitigation of the IMD products of the nonlinear channel.

5.2 LMS Performance Compared to WLS

In order to gain a more accurate view of the performance of this correction, we now compare the results to those produced by a previously used WLS NLEQ method. WLS can be extremely computationally costly due to the need for a matrix inverse. This means that WLS does not have same level of adaptability as an iterative solution like LMS. Due to RAM restrictions, we are only able to calculate WLS weights for part of the collected data set. We randomly selected $1.5(2^{15})$ consecutive samples from the 2^{20} samples that were collected. Coefficients for nonlinear correction were gathered from WLS, which took 11.160776 seconds to run. The weights were then applied to the entire data set to produce a corrected data set, seen in Figure 26, with an error of -30.7098 dB.

From these results, we can see that the LMS correction achieved another 5.5566 dB of correction in 1.2678 fewer seconds. Figure 25 (b) shows a greater increase in linearity of the LMS correction compared to the WLS correction power plot in



(a) (b)

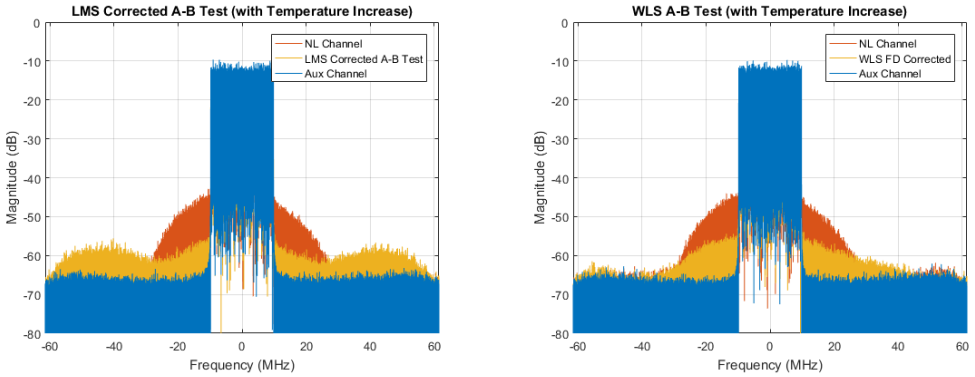
Figure 27: Results from using LMS as a solution for NLEQ, after further iterating over the data set.

Figure 26 (b). Figure 25 (a) also shows a much higher level of mitigation of the IMD products compared to the WLS correction in Figure 26 (a).

5.3 Complete Convergence of LMS

Running the LMS algorithm for another 19 iterations though the entire data set produces further refinement of the correction, and complete convergence of the weights. The error achieved was -39.2190 dB, which is a 2.9526 dB improvement over a single iteration through the entire data set. Figure 27 (a) shows nearly complete mitigation of the IMD products, but an increase in the noise floor on the outer edges of the band. These two shoulders are caused by the memory terms included in the correction, which is the same as the ripple caused by an FIR filter.

Figure 27 (b) shows a further increase in linearity produced by the LMS correction. Next, we will demonstrate the adaptability of LMS to changes in the system.



(a) (b)

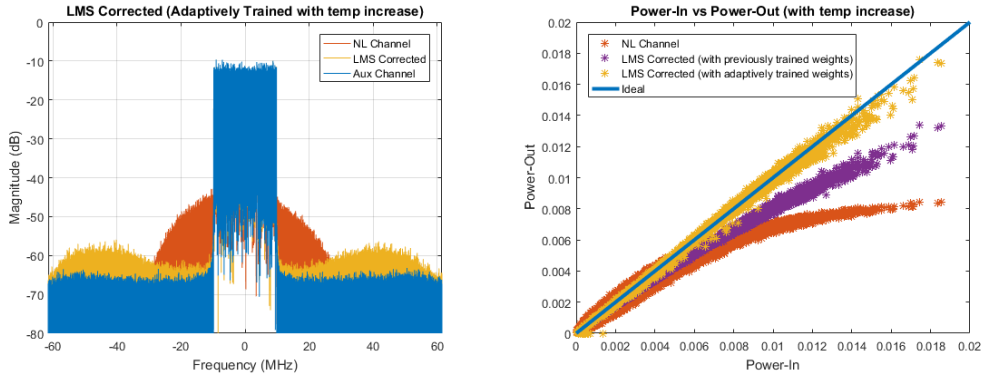
Figure 28: Comparison of the effectivness of the LMS and WLS NLEQ coefficients after the temperature of the system was increased.

5.4 Adaptability of LMS to Temperature Changes

The temperature of the nonlinear amplifier was increased from 30C to 60C, changing the nonlinear characteristics of the system. The previously calculated weights were then applied to the data, showing a degradation in the effectiveness of the weights to mitigate nonlinear effects and IMD products. The corrected data had an error of -18.8954dB, which is 20.3236dB less than was previously achieved with those weights. The WLS weights were then applied to the distorted data and produced corrected data with an error of -18.5941dB. Figure 28 (a) shows a slight change in the mitigation of the IMD products.

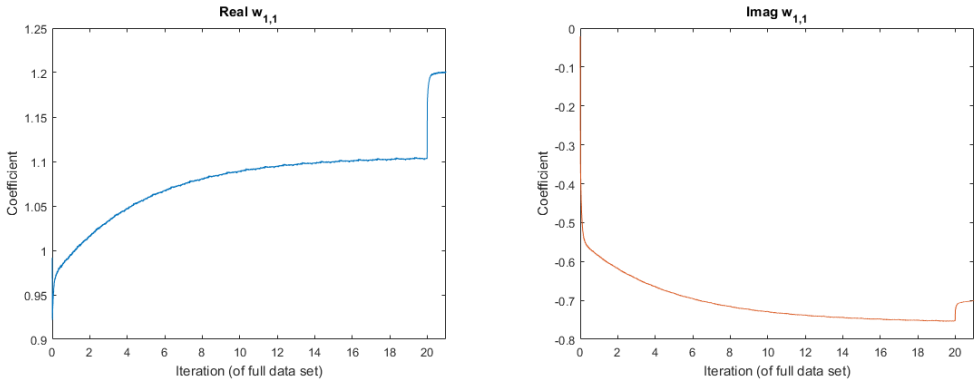
Since LMS is an iterative solution to NLEQ, it is able to adapt the weights to any changes in the system's characteristics. Running LMS on the new data set, with the initial weights set to the values of the previously calculated wights, produces new weights that, when applied to the data set, have an error of -39.1141dB.

Figure 29 (b) shows the improvement from the correction achieved on the temperature effected data with the previously calculated weights (in purple) and the new, adaptively trained correction.



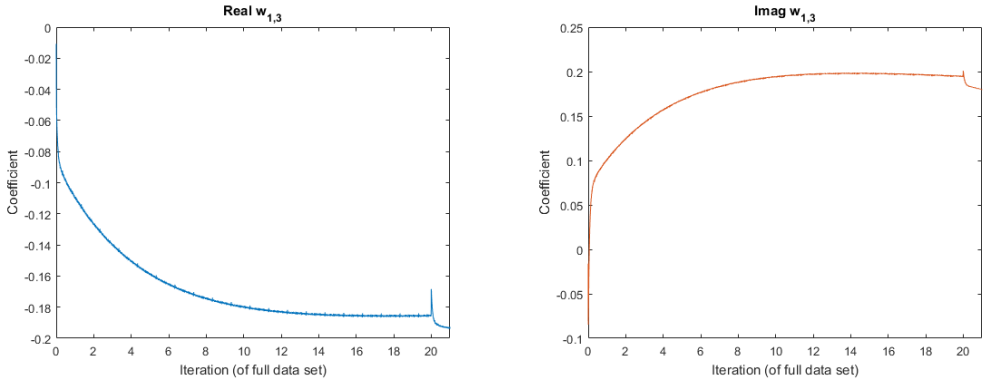
(a) (b)

Figure 29: The results of LMS's ability to adapt the coefficients to changes in system temperature.



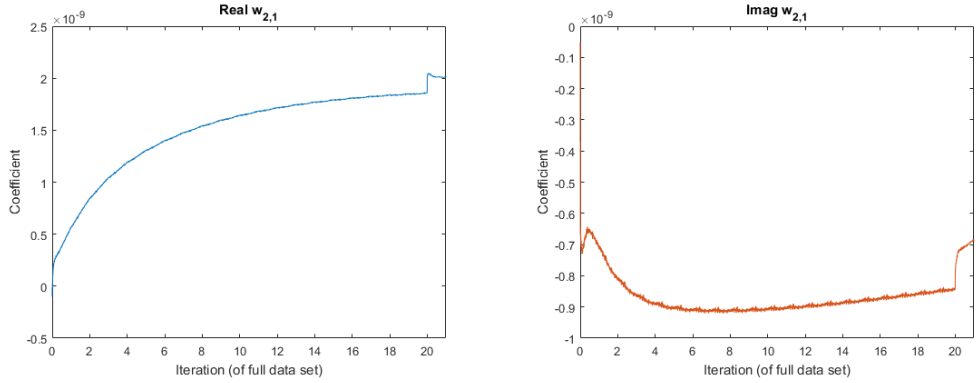
(a) (b)

Figure 30: An example of the LMS NLEQ linear coefficient; (a) real and (b) imag.



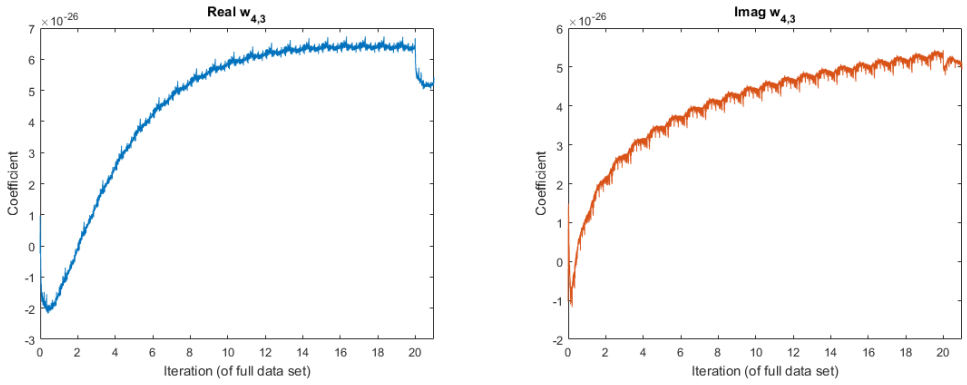
(a) (b)

Figure 31: An example of the LMS NLEQ linear coefficient with a delay of 2; (a) real and (b) imag.



(a) (b)

Figure 32: An example of the LMS NLEQ third-order coefficient; (a) real and (b) imag.



(a) (b)

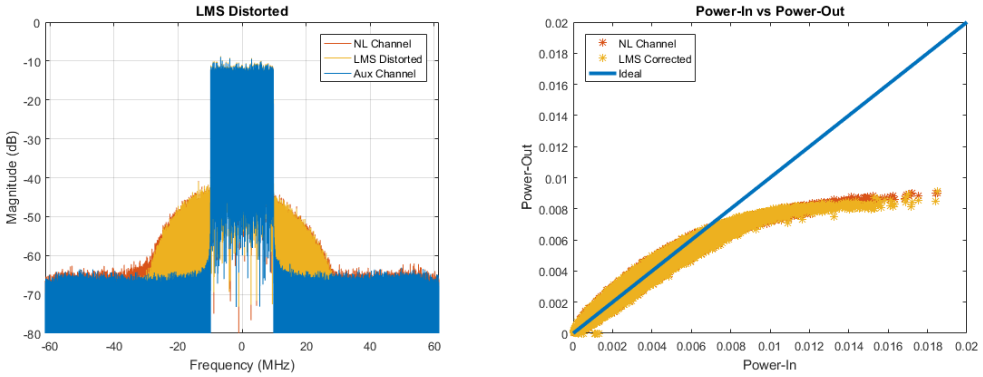
Figure 33: An example of the LMS NLEQ seventh-order coefficient with a delay of two; (a) real and (b) imag.

Figures 30-33 show some of the coefficients, where the first 20 iterations were done on the original data set and the final iteration was done on the temperature effected data set. The coefficients quickly adapted to the changes in the system and converged to the new values in only a few sub-iterations.

5.4.1 Nonlinear System Characterization

The saturation of amplifiers in a receive channel are not usually caused by the desired receive signal, rather, they are typically caused by unwanted in-band interferers. These interferers can cause IMD products that fall on top of the typically low power desired signal. With the use of the nonlinear correction shown in the previous section, we are able to remove the distortion and recover the desired data. Since it is difficult to demonstrate multiple interferers in a laboratory setting, it is ideal to create a well defined model of a specific system to be used for simulations.

The Nonlinear Array System Modeler (NASM) creates a good basis from which to form these simulations. NASM is an in-house MATLAB toolbox with objects that represent each part of a receive chain, including the ability to simulate an entire nonlinear digital phased array. The channel object can be given a set of nonlienar



(a) (b)

Figure 34: Results of nonlinear system characterization using LMS.

coefficients to produce the nonlinearities and distort the signal. These coefficients can represent a real system, and can be produced in the same fashion that the nonlinear correction was done in the previous section.

For gathering the nonlinear distortion coefficients, compared to the coefficients for NLEQ, the desired signal is replaced with the nonlinear signal. Using the same LMS algorithm as in the previous section, nonlinear coefficients that represent the system were computed. Figures 34 (a) and 34 (b) show the resulting distortion produced from the coefficients. Figure 34 (b) shows particularly promising results that verify that the nonlinear coefficients well-represent the system's response to different frequencies and power levels. These coefficients are used in Section 5.6 to simulate the IMD caused by two interferers and the recovery of the desired signal through LMS nonlinear correction.

5.5 Bandpass Filter Testbed Setup

Next, we explored the variations in nonlinear device characteristics with changes in frequency. A varactor tuned bandpass filter was used, instead of an amplifier, to represent the nonlinear component in the system. As the varactors had a much

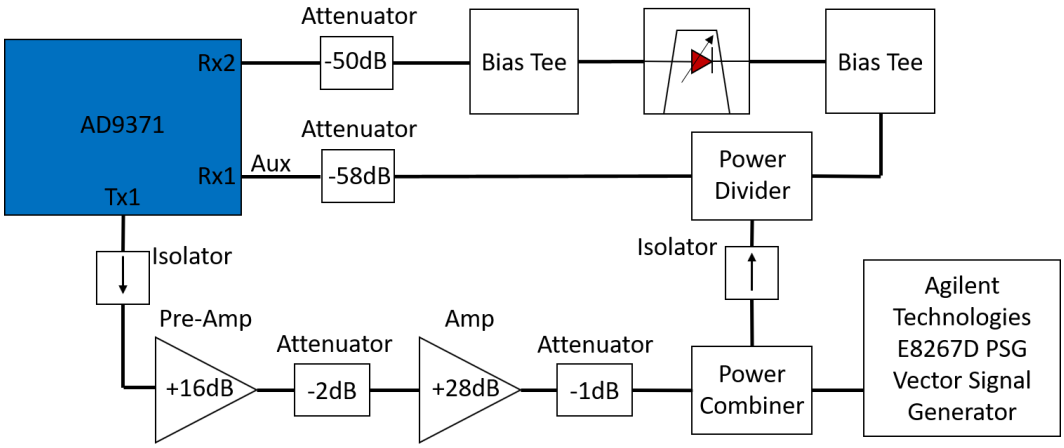
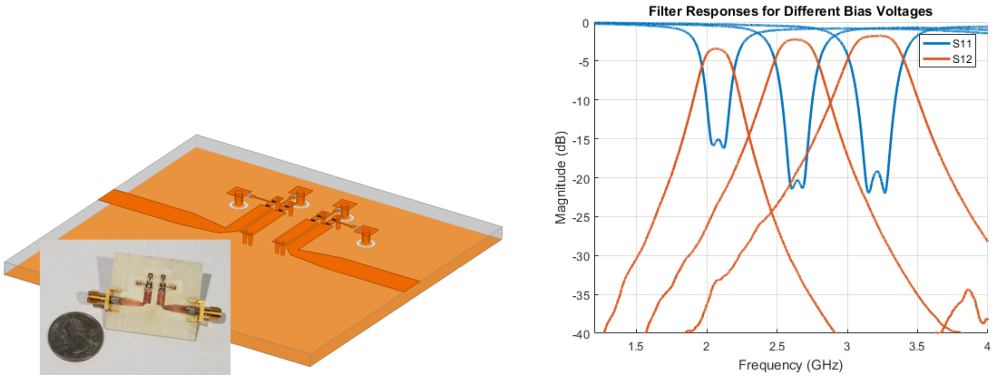


Figure 35: Block-diagram of the test setup with a nonlinear varactor-loaded bandpass filter.

higher IIP3 than the previously used amplifier, a larger pre-amp was needed to provide enough power to push the filter into the nonlinear region. Also, due to this need of more power, two tones were used in order achieve a lower peak-to-average power ratio, compared to the high peak-to-average power provided by the WGN. Data was gathered from the Analog Devices AD9371 transceiver, using one of the Tx channels to feed a sinusoid through two amplifiers, a MiniCircuits ZRL-3500+ followed by a MiniCircuits ZVE-8G+, combining with another sinusoid produced by an Agilent Technologies E8267 PSG Vector Signal Generator. The combined output was then fed into a power divider with one output going through 58dB of attenuators into the auxilliary receiver and the other output being fed into the varactor loaded bandpass filter, seen in Fig. 36. Both channels were attenuated to ensure linear behavior of the receivers themselves. The center frequency of the filter was tested at 2.1, 2.4, and 2.7 GHz, with the two input tones being offset 8 and 13 MHz from the center frequency.



(a) (b)

Figure 36: Varactor-loaded bandpass filter

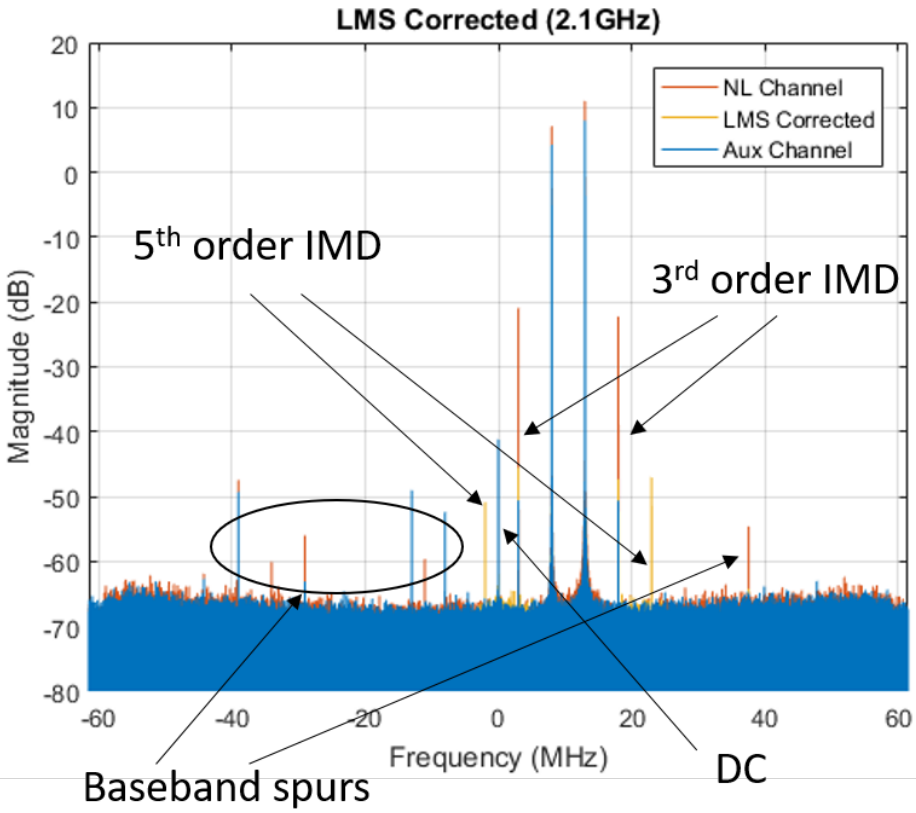


Figure 37: The results of the LMS correction on two tones at a center frequency of 2.1GHz.

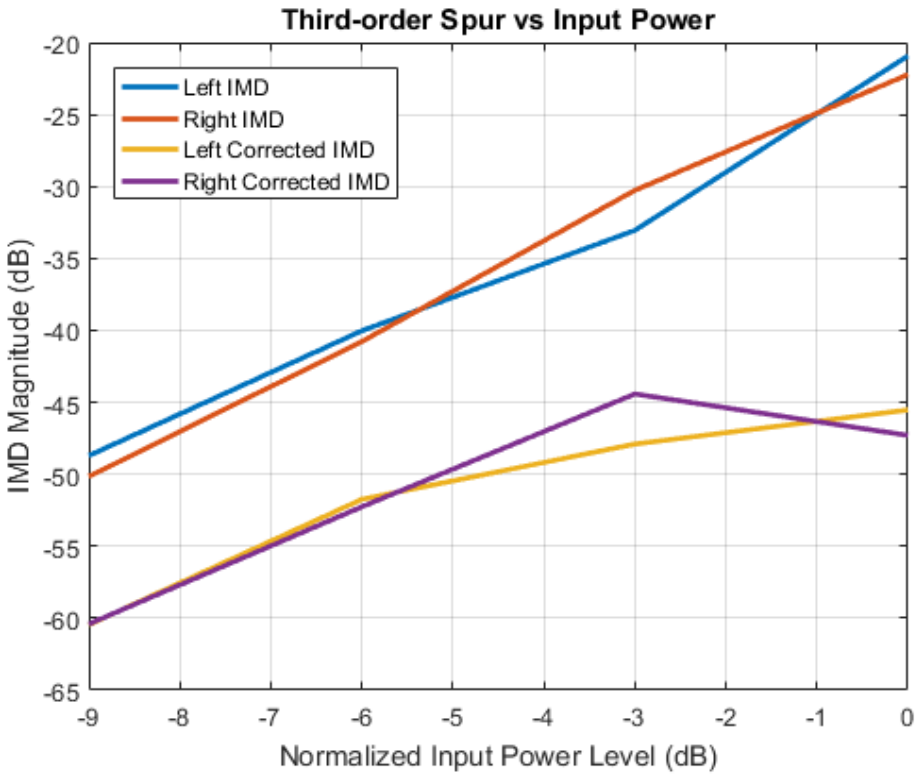


Figure 38: Using the correction coefficients from the normalized power level of 0dB on data sets of -3,-6, and -9dB. The correction continues to be effective with changes in power.

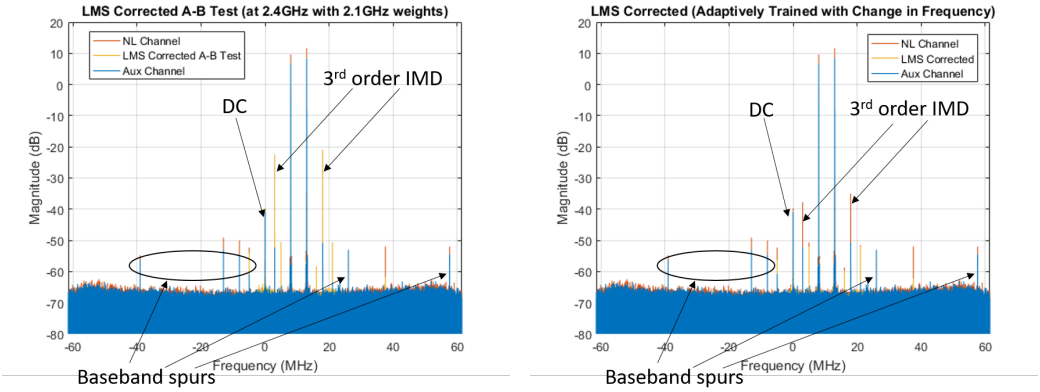
5.5.1 NLEQ on Bandpass Filter Nonlinearities

The data gathered at the highest tested power level for 2.1GHz was then trained and corrected using the proposed LMS method. The applied memory polynomial contained two power terms and eight memory coefficients. The third-order IMD was suppressed by 25dB, as shown in Fig. 37. The same coefficients were then used to correct the third-order IMD at lower power levels. Figure 38 shows that the coefficients continue to mitigate the IMD produced by the filter over a dynamic range, but distort the natural 3:1 slope of these products; this makes it difficult to continue using IIP3 as a metric for systems that make use of these techniques.

The same coefficients were then used on data gathered at 2.4GHz after tuning to this new frequency. Figure 39 (a) shows that not only did the coefficients not mitigate the third-order IMD, but it exacerbated the interference by increasing the unwanted spurs. Then, allowing LMS to run some iterations through the new data set, with the initial weights being the previously trained coefficients, the third-order IMD caused by the filter was again mitigated, as shown in Fig. 39 (b). Depending on the application, it may be prudent to use a closed-loop version of this scheme (with constant adaptation), rather than periodic, iterative corrections.

5.6 Simulation of Nonlinear System in NASM

In this section the distortion coefficients calculated in Section 5.4.1 are used to produce nonlinearities in a receive channel with two in-band interferers. The desired signal is a 10MHz QPSK waveform centered in the band at 2.7GHz. The two interferers are a 20MHz AWGN waveform offset by -30MHz and a high power tone offset by -15MHz. The IMD products produced by the intermodulation of the wideband and single-tone interferers will fall directly on top of the desired signal and will distort the data. As in the previous section, an auxiliary channel is used to train



(a) (b)

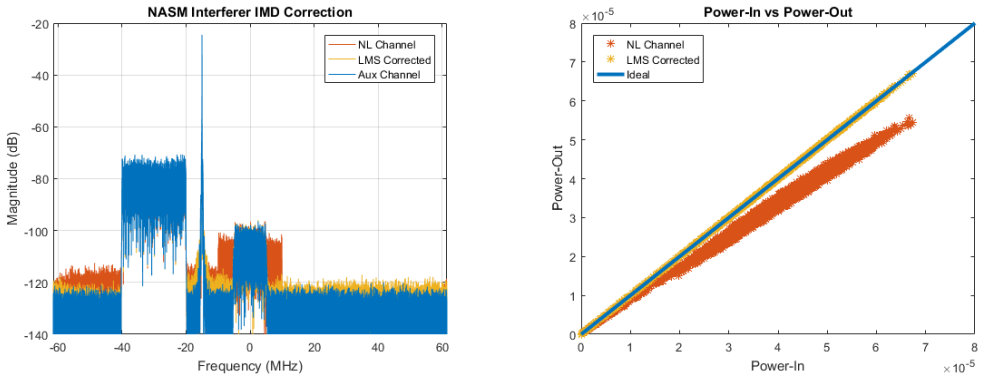
Figure 39: Corrected two-tone input data; (a) Applying the coefficients from 2.1GHz on data gathered from 2.4GHz. The third-order spurs, instead of being suppressed, are amplified, requiring new coefficients for the change in frequency. (b) LMS iteratively adapts the coefficients to the system's changes and is able to mitigate the third-order IMD.

the LMS correction coefficients and correct the nonlinear channel.

Figure 40 shows that the nonlinear correction removes the IMD products that fell on top of the desired signal. Figure 41 (a) show QPSK data from the auxiliary channel and Figure 41 (b) shows the data from the nonlinear channel. The data from the nonlinear channel is completely distorted and is impossible to receive any of the desired data. Figure 41 (c) shows the QPSK data from the corrected signal and it very closely matches the data from the auxiliary channel. The nonlinear corrected was able to remove the distortion cause by the two interferers and completely recover the desired data.

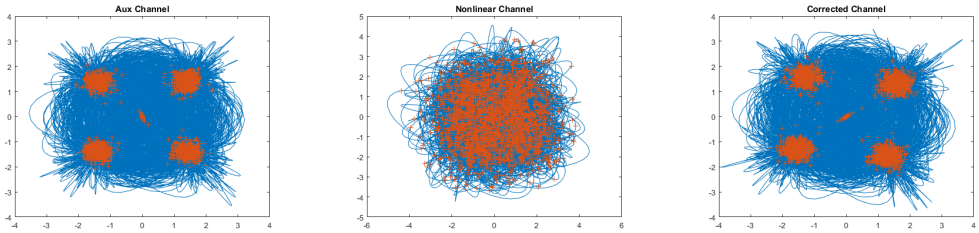
5.7 Simulation of Nonlinear Digital Array

Having demonstrated the performance of the LMS-based correction on a single channel, we then want to evaluate it's effectiveness on an array. Using NASM and the calculated nonlinear coefficients, we simulated a 12 element linear array



(a) (b)

Figure 40: Results of LMS NLEQ being used in an NASM simulation.



(a) (b) (c)

Figure 41: Results of LMS NLEQ being used in an NASM simulation showing that the QPSK desired data can be perfectly recovered after employing NLEQ to mitigate the odd-order spurs that had distorted the waveform.

with two-tone input, each tone from a different direction. The two third-order IMD correlate to predictable angles given by equations in [20], used by the authors of [2]. Then the array was simulated over a large range of sizes (number of elements) and the magnitudes of the main tones and third-order spurs were compared to determine decorrelation effectivness with different array sizes. This was done for both arrays with identical elements and arrays with elements with channel characteristics that are Gaussian distributed. Furthermore, the different averaging techniques described in Chapter 1.1 were applied and evaluated.

5.7.1 Decorrelation of Identical Array Elements with Temperature Change

The array was simulated with at a frequency of 2.7GHz with the two tones, at baseband, having frequencies of 11MHz and 17MHz and directions of 15° and -11°, respectively. The third-order IMD were at baseband frequencies of 5MHz and 23MHz with correlated directions of 45.27° and -39.68°, respectively.

The first array simulation, shown in Figure 42, used coefficients from the 30°C channel. The channels were corrected using the LMS-based correction method, decorrelating the third-order IMD by 23.47dB and 24.24dB. The second and third arrays were simulated using the coefficients from the 60°C channel. The first of these two simulations was corrected using the correction coefficients from the first array simulation, shown in Figure 43. The decorrelation of the third-order IMD in this simulation were only 10.61dB and 11.73dB, less than half of the correction when the coefficients were used on the channels they were calibrated on.

The correction coefficients were then iteratively calibrated by using LMS allowing it to adapt the values to the new channel characteristics. Figure 44 shows that the adaptive weights decorrelated the IMD by 19.94dB and 19.68dB. Using the iterative calibration solution to correct the nonlinear IMD of an array provided 9dB

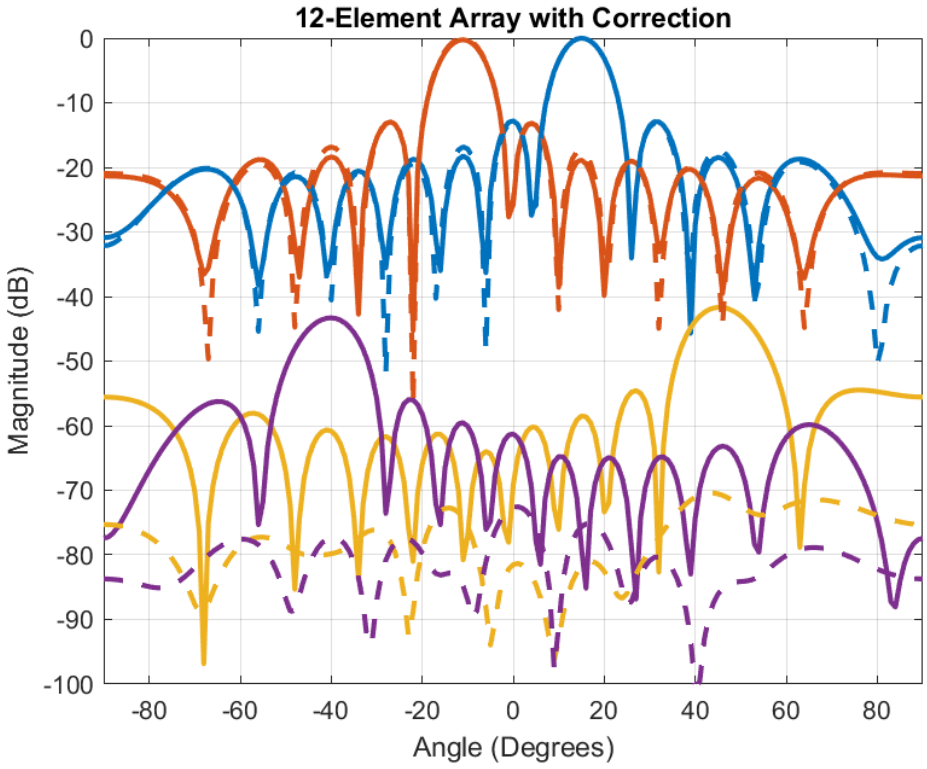


Figure 42: Digital beamforming of a simulated 12-element array with nonlinear receive channels, solid lines, and its nonlinear correction, dashed lines. The two input tones are at baseband frequencies of 11MHz (blue) and 17MHz (orange) with third-order IMD at 5MHz (yellow) and 23MHz (purple). The correction shows the decorrelation of the third-order spurs.

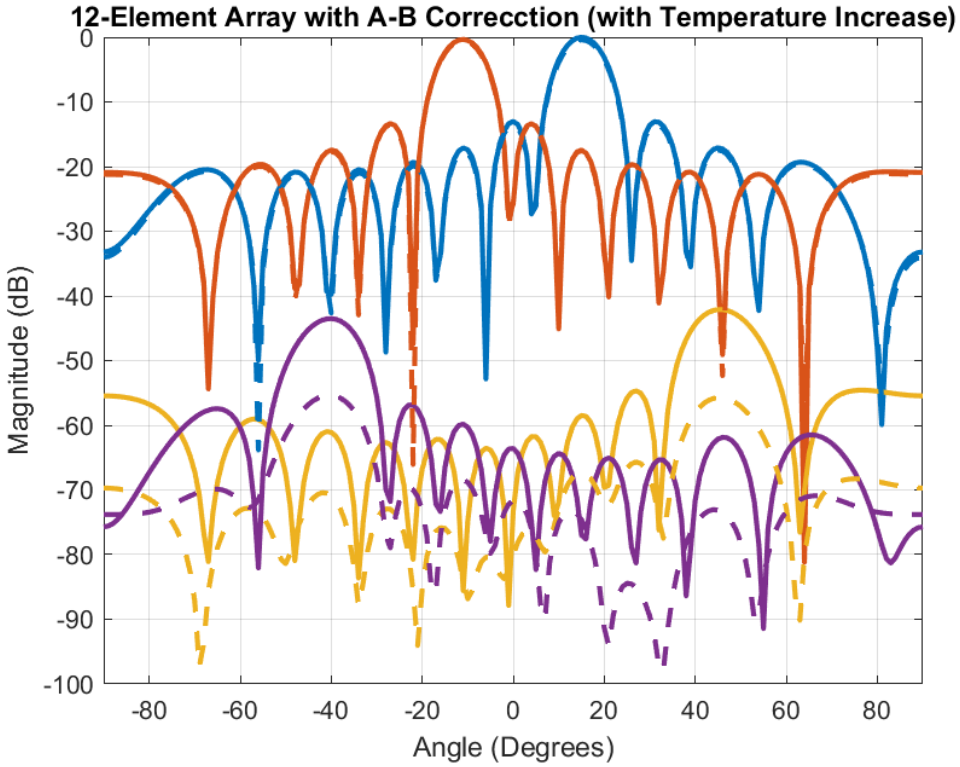


Figure 43: Digital beamforming of a simulated 12-element array with heated nonlinear receive channels, solid lines, and its nonlinear correction using the coefficients of the non-heated channels, dashed lines. The two input tones are at baseband frequencies of 11MHz (blue) and 17MHz (orange) with third-order IMD at 5MHz (yellow) and 23MHz (purple). The correction shows less decorrelation of the third-order spurs.

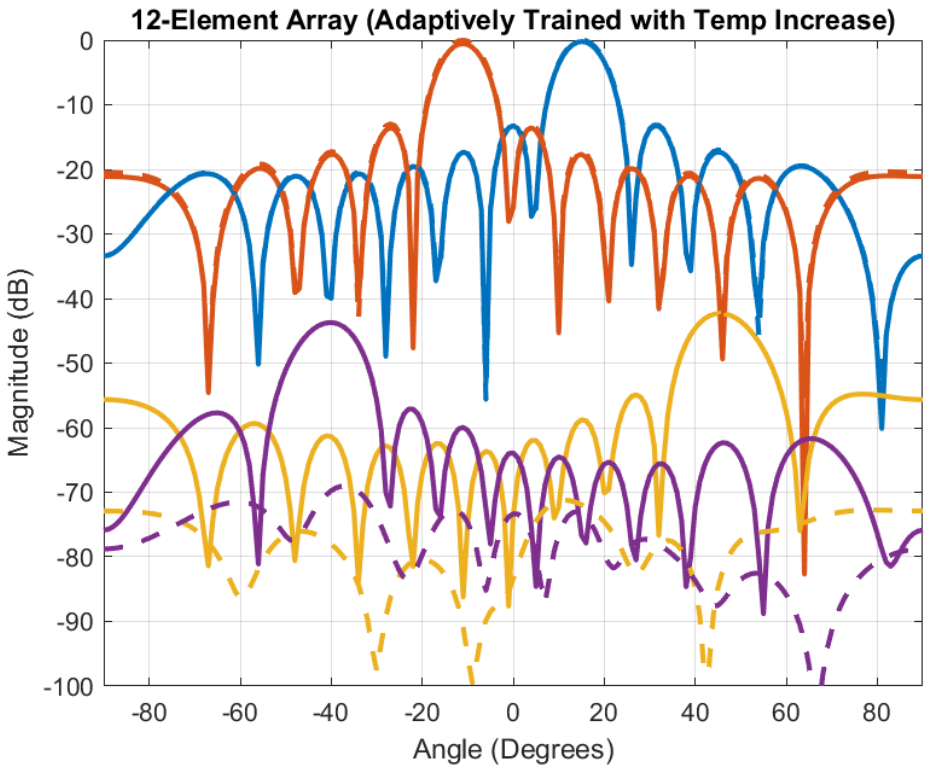


Figure 44: Digital beamforming of a simulated 12-element array with heated nonlinear receive channels, solid lines, and its nonlinear correction using adaptively trained coefficients, dashed lines. The two input tones are at baseband frequencies of 11MHz (blue) and 17MHz (orange) with third-order IMD at 5MHz (yellow) and 23MHz (purple). The correction shows an increase in the decorrelation of the third-order spurs, than when the static correction was used.

more mitigation when compared to the use of static coefficients.

These array simulations show how much more effective an adaptive NLEQ solution can be than using static coefficients. In this particular case, more than 10dB more of decorrelation was achieved by adapting the coefficients to the changes in the array channels.

5.8 Decorrelation of Gaussian Distributed Array Elements

The previous section showed the decorrelation of odd-order spurs at the array level, when the elements are identical, and the importance of an adaptive NLEQ solution to deal with array temperature changes. But, as was discussed in much detail throughout this thesis, the elements of a real array cannot be exactly identical, rather they will have some mean and variance. The distribution of these elements can be described as Gaussian. In this section more arrays are simulated in NASM, similar to the previous section, with Gaussian distributed elements. With this, different coefficient techniques, such as single element coefficients, averaged coefficients, and the root-averaged coefficients described in Chapter 4, are simulated.

Figure 45 shows the performance of NLEQ being applied to each element of an array with Gaussian distributed elements with different standard deviations. Since each element was trained separately for each of the simulated arrays, the performance of NLEQ on decorrelating the odd-order spurs remained the same for different standard deviations. The effect of the different elements can be seen more greatly in the beams of the two main tones. For both the corrected and non-corrected versions, the sidelobes of the array begin to change when the standard deviation of the element's channels is increased.

Figure 46 shows more simulations of arrays with Gaussian distributed element with different standard deviations, but with NLEQ being trained on a single element of the array and then being applied to every element. The performance of this method was good for arrays with identical elements and also the array with elements with a standard deviation of 0.01, but for arrays with elements with greater standard deviations, the performance of NLEQ decreased drastically. For the final simulation, with a standard deviation of 0.2, NLEQ coefficients from a single element lead to increased odd-order spurs instead of the intended mitigation. This

is due the to channel coefficients of each element being uncorrelated enough that the NLEQ coefficients from one element add more nonlinearities to other elements instead of mitigating the spurs.

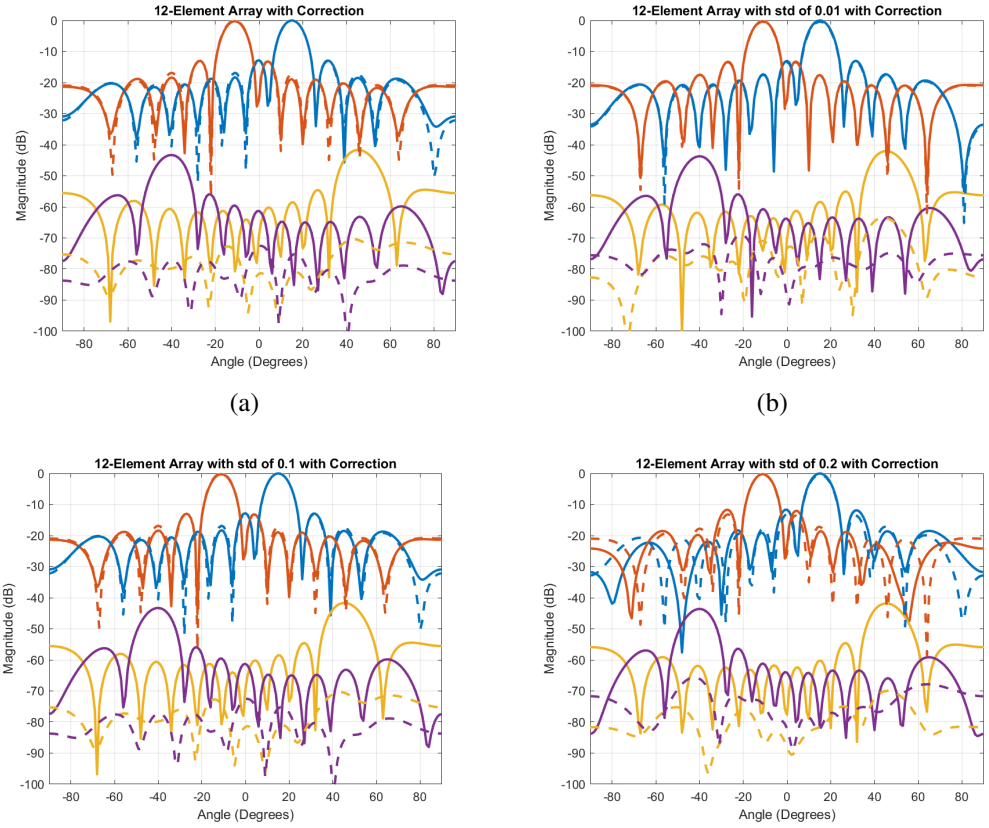
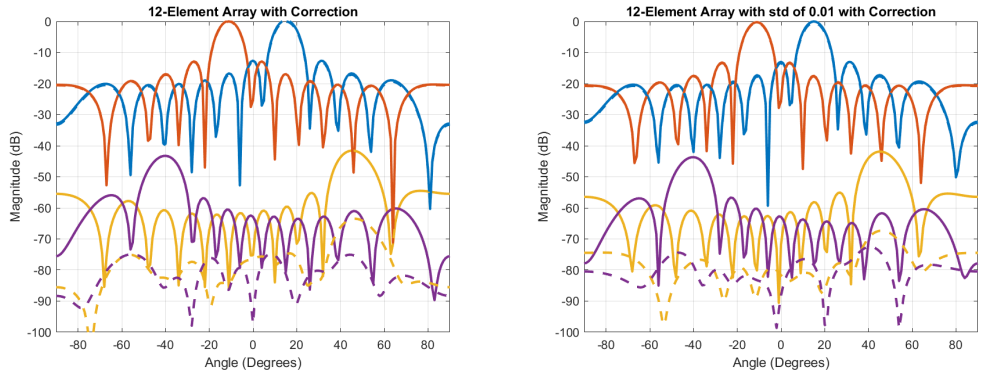


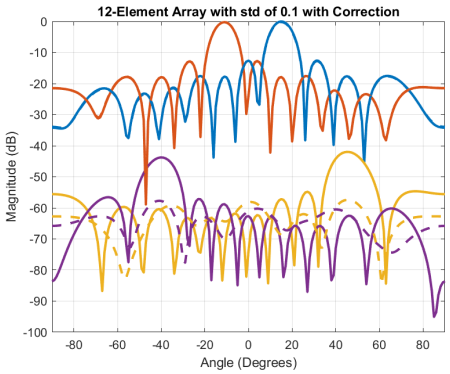
Figure 45: Digital beamforming of a simulated 12-element array with Gaussian distributed channel coefficients with standard deviations of (a) 0, (b) 0.01, (c) 0.1, and (d) 0.2. NLEQ was applied by correcting each element separately.

Lastly, figure 47 shows the simulation of more arrays with Gaussian distributed elements with the NLEQ coefficients being the average coefficients of every element of the array. Though the performance of this method was still less than that of every element being trained individually, it did outperform the single element method in figure 46.

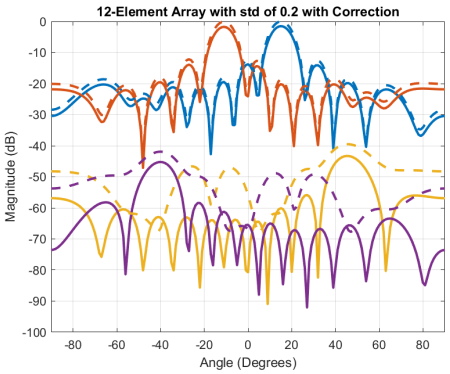


(a)

(b)



(c)



(d)

Figure 46: Digital beamforming of a simulated 12-element array with Gaussian distributed channel coefficients with standard deviations of (a) 0, (b) 0.01, (c) 0.1, and (d) 0.2. NLEQ was applied by correcting one of the elements and then applying those coefficients to all of the elements of the array.

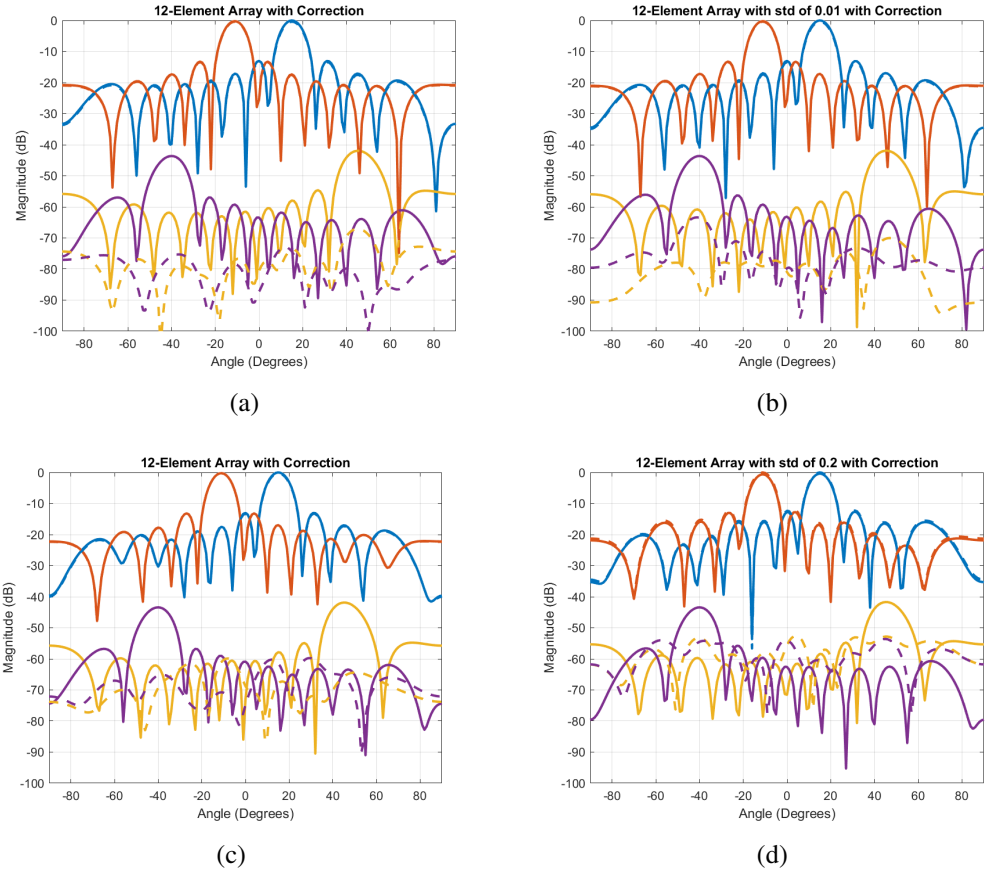
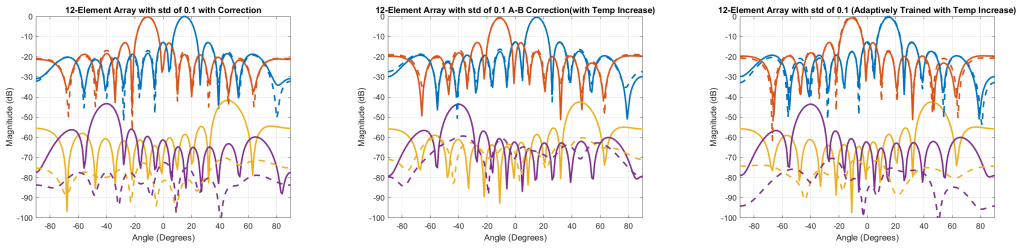


Figure 47: Digital beamforming of a simulated 12-element array with Gaussian distributed channel coefficients with standard deviations of (a) 0, (b) 0.01, (c) 0.1, and (d) 0.2. NLEQ was applied by training each element separately and then averaging those coefficients and applying those averaged coefficients to every element for correction.

5.8.1 Decorrelation of Gaussian Distributed Array Elements with Temperature Change

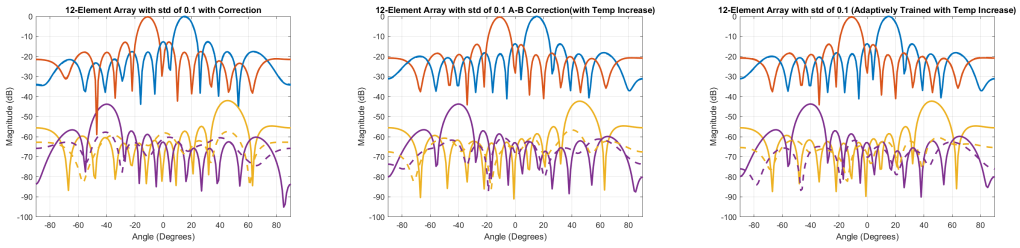
The techniques for decorrelation of Gaussian distributed array elements above show how greatly channel variations can effect the performance of a digital array, and in the previous section, the importance of an adaptive NLEQ solution for dealing with changes in array temperature was also shown. Here we simulate arrays with Gaussian distributed elements and apply NLEQ to them, then we change the temperature of the array and evaluate the continued effectivness of the previously trained coefficients, for different methods of training and standard deviations.

The results of these simulations are shown in Figures 45-46, and were simulated with a channel standard deviation of 0.1. Figure 45 shows the results of applying NLEQ to every individual element of the array, achieving good decorrelation when it was trained and corrected at the same temperature, and still does a good job at attempting to decorrelate the spurs when the coefficients were trained when the channels were at the lower temperature and applied to the array when the channels were at the higher temperature. Figure 46 shows similar results for when the NLEQ coefficients from a single channel were used to correct the spurs in the entire array, but the spurs were not decorrelated when the array was trained and corrected at the lower temperature, possibly due to which element's coefficients were used. Lastly, Figure 47 shows the results from using the average NLEQ coefficients and applying them to the entire array. This shows a slight improvement over using the coefficients from a single element, nearly completely decorrelating the third-order spurs from the array when it was trained and corrected at the same temperature.



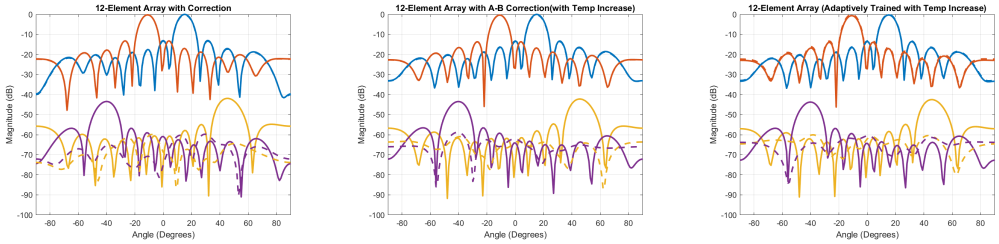
(a) (b) (c)

Figure 48: Digital beamforming of a simulated 12-element array with Gaussian distributed channel coefficients with a standard deviation of 0.1 and NLEQ applied to each individual element. (a) is an array with channels at room temperature, (b) and (c) are arrays with the temperature raised 30°C with the room temperature NLEQ coefficients being applied to (b) and with adaptively trained coefficients being applied to (c).



(a) (b) (c)

Figure 49: Digital beamforming of a simulated 12-element array with Gaussian distributed channel coefficients with a standard deviation of 0.1 and NLEQ being trained on a single element and then applied to every element of the array. (a) is an array with channels at room temperature, (b) and (c) are arrays with the temperature raised 30°C with the room temperature NLEQ coefficients being applied to (b) and with adaptively trained coefficients being applied to (c).



(a) (b) (c)

Figure 50: Digital beamforming of a simulated 12-element array with Gaussian distributed channel coefficients with a standard deviation of 0.1 and the average coefficients from the array elements being used to apply NLEQ to the array. (a) is an array with channels at room temperature, (b) and (c) are arrays with the temperature raised 30°C with the room temperature NLEQ coefficients being applied to (b) and with adaptively trained coefficients being applied to (c).

5.8.2 Techniques for Decorrelation of Gaussian Distributed Array Elements

In this final section, we compare the performance of the different proposed techniques to achieve array-level IMD decorrelation for arrays with Gaussian distributed elements. The correction methods that are compared here are individual element correction, correction using a single element's coefficients, average element coefficient correction, root-average coefficient correction, and hand-tuned root-average coefficient correction. Hand-tuned root-average correction is done by applying hand-tuned coefficients to the root-averaged coefficients. This is done in an attempt to shift the coefficients to the true mean of the array elements, due to the distributions described in Section 4. Finally, due to computational restrictions, these simulations were not run in NASM, but were run with a much more computationally simple simulation as was shown in Section 4.

Figure 51 shows the performance of the different proposed techniques in decorrelating the odd-order spurs of arrays, with a channel standard deviation of 0.1, with different number of elements, up to ten-thousand elements. From this simulation, it can be seen that the root-average method slightly out-performs the av-

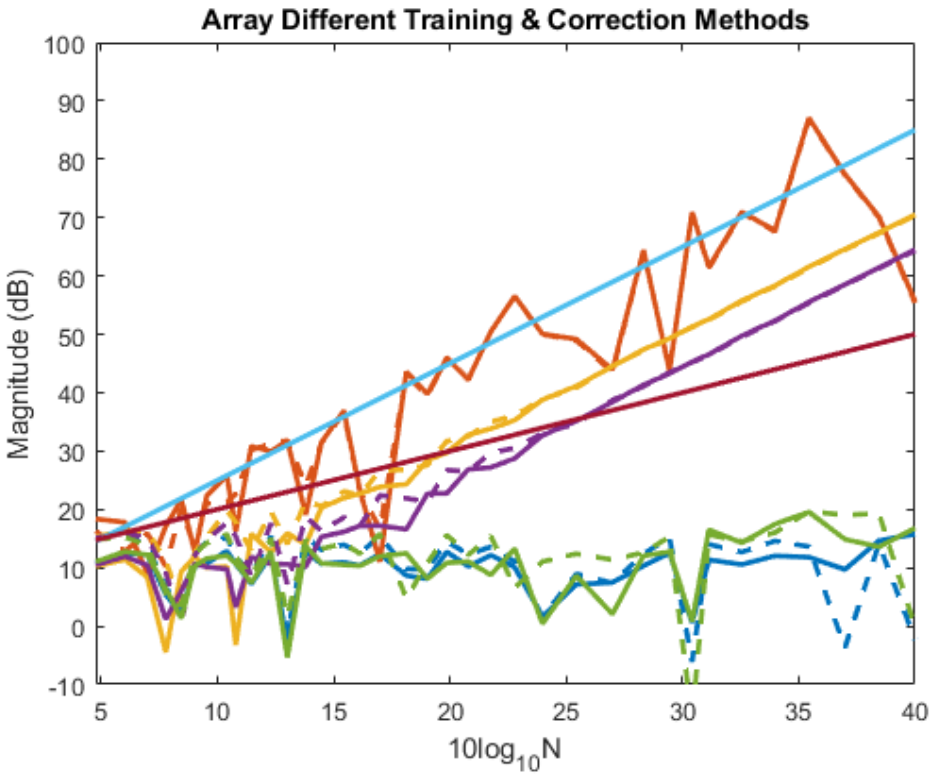


Figure 51: A performance comparison of the different proposed techniques for array-level decorrelation of odd-order spurs for arrays with Gaussian distributed elements, in this simulation the standard deviation was 0.1 and the results were averaged over ten simulations. The cyan and magenta lines are $20log_{10}N$ and $10log_{10}N$, respectively. The remaining solid and dashed lines are the resulting magnitudes of the spurs after NLEQ was applied to the array for the methods of individual element correction (blue), single element coefficient correction (orange), average coefficient correction (yellow), root-average coefficient correction (purple), and hand-tuned root-average coefficient correction (green).

erage coefficient method by 6dB for array sizes larger than 30, achieving nearly perfect decorrelation for array sizes up to about 30 or 50 elements. Using the hand-tuned root-average method greatly outperformed the root-average method, achieving decorrelation of the two spurs for every array size in the simulation, including the ten-thousand element array. Therefore, it can be said that the hand-tuned root-average method outperformed the root-average method by an order of magnitude, but the difficulty lies in determining the best hand-tuned values.

The hand-tuned values that were used for this simulation were found by finding the values that best decorrelated the spurs. This was done by iterating through different combinations of values and selecting the combination that minimized the spurs. It was also considered that, from the cubed Gaussian distributions shown in Section 4, the mean of the coefficients will tend to be higher than the true mean of the elements, therefore, values lower than one were generally used.

6 Conclusion and Future Work

The LMS algorithm provided an elegant solution to NLEQ of a digital array, especially when the array channels experienced variations in both temperature and frequency. This ability to adapt the NLEQ coefficients to system changes is a necessity for applying NLEQ to a real system, especially for digital arrays which will experience greater temperature variations and will often be UBW and tuning to different frequencies.

Mitigation of odd-order spurs on the channel level and adaptability to channel characteristic changes performed well, but some improvements can be made. First, the normalized LMS (NLMS) method should be explored in an attempt to achieve further coefficient convergence, and increase the stability of the algorithm. After this algorithm is explored, a comparison of different basis-polynomials should be explored in-order to determine the necessary terms that should be included in NLEQ, and which terms can be left out, in-order to find the basis that best describes the nonlinear system without over-characterizing it, which can be computationally costly.

Decorrelation of the odd-order spurs in an array was shown to be successful for small arrays, but for arrays larger than one-thousand elements, little decorrelation was achieved. More techniques for finding the coefficients that describe the mean-element of the array will need to be explored. One possible solution is to determine a good way for finding the best hand-tuned coefficients for the root-averaged method for a specific array. Another topic that will need to be explored is to determine how many elements are needed to calculate the average-element of the array, and how are these elements selected. The selection process is extremely important because, as discussed throughout this thesis, the location of the elements on an array greatly influence the temperature of the element and, therefore, its channel

coefficients. NLEQ is an extremely important topic, with increasing interest, for building low-cost digital arrays. Much as been explored and discovered, but there remain many questions, especially for array-level decorrelation techniques.

References

[1] L. Paulsen, T. Hoffmann, C. Fulton, M. Yeary, A. Saunders, D. Thompson, B. Chen, A. Guo, and B. Murmann, "Impact: A low cost, reconfigurable, digital beamforming common module building block for next generation phased arrays", *Proc. SPIE 9479, Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation 2015*, vol. 9479, May 2015.

[2] B. James and C. Fulton, "Decorrelation and mitigation of spurious products in phased arrays with direct conversion transceivers", in *Proc, IEEE MTT-S International Microwave Symposium*, May 2015.

[3] M. Fozooni, M. Matthaiou, E. Bjornson, and T. Q. Duong, "Performance limits of mimo systems with nonlinear power amplifiers", in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Dec. 2015.

[4] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive mimo for next generation wireless systems", *Proc. IEEE Communications Magazine*, vol. 52, 2 Feb. 2014.

[5] S. H. Talisa, K. W. O'Haver, T. M. Comberiate, M. D. Sharp, and O. F. Somerlock, "Benefits of digital phased array radar", *Proc. IEEE*, vol. 104, 3 Mar. 2016.

[6] C. Fulton, M. Yeary, D. Thompson, J. Lake, and A. Mitchell, "Digital phased arrays: Challenges and opportunities", *Proc. IEEE*, vol. 104, 3 Mar. 2016.

[7] J. E. Stailey and K. D. Hondl, "Multifunction phased array radar for aircraft and weather surveillance", *Proc. IEEE*, vol. 104, 3 Mar. 2016.

[8] C. Fulton, P. Clough, V. Pai, and W. Chappell, "A digital array radar with a hierarchical system architechture", in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2009, pp. 89–92.

[9] C. Fulton, T.-Y. Yu, J. Salazar, and R. Palmer, "Updates on next-generation array-based weather radar developments at the university of oklahoma", in *2017 IEEE Conference on Microwaves, Antennas, Communications, and Electronic Systems (COMCAS)*, Nov. 2017.

[10] R. Vansebrouck, C. Jabbour, P. Desgreys, O. Jamin, and V.-T. Nguyen, "Performance study of nonlinearities blind correction in wideband receivers", in

Proc. 21st IEEE International Conference Electronics, Circuits, and Systems (ICECS), Marseille, France, Dec. 2014.

[11] G. L. Cunha, S. Farsi, B. Nauwelaers, and D. Schreurs, "An fpga-based digital predistorter for rf power amplifier linearization using cross-memory polynomial model", *International Workshop on Integrated Nonlinear Microwave and Millimeter-wave Curcuits (INMMiC)*, pp. 1–3, 2014.

[12] Y. H. Lim, Y. S. Cho, I. W. Cha, and D. H. Youn, "An adaptive nonlinear prefilter for compensation of distortion in nonlinear systems", *IEEE Trans. on Signal Processing*, vol. 46, 6 Jun. 1998.

[13] K. Dogancay, "Lms algorithm for blind adaptive nonlinear compensation", in *Proc. IEEE Region 10 Conference, TENCON 2005*, Melbourne, Australia, Nov. 2005.

[14] M. Grimm, R. K. Sharma, M. Hein, R. S. Thoma, and R. Zemmari, "Improved ber performance in gsm by mitigating non-linear distortions in the receiver", in *Proc. European Microwave Conference*, Nuremberg, Germany, Oct. 2013.

[15] M. Grimm, M. Allen, J. Marttila, M. Valkama, and R. Thoma, "Joint mitigation of nonlinear rf and baseband distortions in wideband direct-conversion receivers", *IEEE Trans. Microwave Theory and Techniques*, vol. 62, pp. 166–182, 1 Jan. 2014.

[16] D. Dupleich, M. Grimm, F. Schlembach, and R. S. Thoma, "Practical aspects of a digital feedforward approach for mitigating non-linear distortions in receivers", in *Proc. 11th International Telecommunications in Modern Satellite, Cable and Broadcast Services Conference (TELSIKS)*, Nis, Serbia, Oct. 2013.

[17] Y. Li, J. M. Rabaey, and A. Sangiovanni-Vincentelli, "Analysis of interference effects in mb-ofdm uwb systems", in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2008.

[18] S. Chen and B. Murmann, "An 8-bit 1.25gs/s cmos if-sampling adc with background calibration for dynamic distortion", in *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Nov. 2016.

[19] R. Gangarajaiah, M. Abdulaziz, H. Sjoland, P. Nilsson, and L. Liu, "A digitally assisted nonlinearity mitigation system for tunable channel select filters", *IEEE Trans. Circuits Systems II, Express Breifs*, vol. 63, 1 Jan. 2016.

[20] C. Hemmi, *Pattern characteristics of harmonic intermodulation products in broadband active transmit arrays*, Aug. 2002.

[21] E. Rebeiz and D. Cabric, "How wideband receiver nonlinearities impact spectrum sensing", in *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2014.

[22] S. Mittal, "A survey of architectural techniques for managing process variation", *ACM Computing Surveys*, vol. 48, 4 Feb. 2016.

[23] D. Gomez, M. Sroka, and J. L. G. Jimenez, "Process and temperature compensation for rf low-noise amplifiers and mixers", *IEEE Trans. Circuits and Systems*, vol. 57, 6 Jun. 2010.

[24] H. Chauhan, V. Kvartenko, and M. Onabajo, "A tuning technique for temperature and process variation compensation of power amplifiers with digital predistortion", in *2016 IEEE North Atlantic Test Workshop (NATW)*, May 2016.

[25] C. Berg, "The cube of a normal distribution is inderterminate", *The Annals of Probability*, vol. 16, pp. 910–913, 2 Apr. 1988.