2014

Relaxing the requirements for accurate spectral testing of data converters

Benjamin Thomas Magstadt
Iowa State University

Recommended Citation

Magstadt, Benjamin Thomas, "Relaxing the requirements for accurate spectral testing of data converters" (2014). *Graduate Theses and Dissertations*. 13918.
https://lib.dr.iastate.edu/etd/13918

Relaxing the requirements for accurate spectral testing of data converters

by

Benjamin T Magstadt

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering

Program of Study Committee:
Degang Chen, Major Professor
Randall Geiger
Sumit Chaudhary

Iowa State University

Ames, Iowa

2014

DEDICATION

This work is dedicated to my family who has continually believed in me and supported me throughout the entirety of pursuing my college education.

TABLE OF CONTENTS

Page

DEDICATION ii

ACKNOWLEDGEMENTS v

ABSTRACT vi

CHAPTER 1 INTRODUCTION 1

Motivation 1
Standard Test 2
Challenges of Standard Test 5
Paper Outline 9

CHAPTER 2 ACCURATE SPECTRAL TESTING WITH NON-LINEAR SOURCE AND NON-COHERENT SAMPLING 11

Introduction 11
Non-linear Modeling 13
Proposed Algorithm 14
Simulation Results 23
Robustness 26
Measurement Results 30
Conclusion 34

CHAPTER 3 ACCURATE SPECTRAL TESTING WITH NON-LINEAR SOURCE, NON-COHERENT SAMPLING, AND AMPLITUDE CLIPPED SIGNAL 35

Introduction 35
Proposed Algorithm 37
Simulation Results 52
Conclusion 54

CHAPTER 4 PRE-DISTORTION CODE GENERATION FOR ACCURATE DAC-ADC SPECTRAL TESTING 56

Introduction 56
Proposed Algorithm 58
Simulation Results 64
Robustness 66

Conclusion 68

CHAPTER 5 SUMMARY 69

REFERENCES 71

ACKNOWLEDGEMENTS

I would like to thank my major professor Dr. Degang Chen for his guidance, support, and ideas throughout the course of my work at Iowa State University. I would also like to thank my other committee members Dr. Randall Geiger and Dr. Sumit Chaudhary for all of their input on the research that was conducted.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience.

Finally, I want to say thanks to my family for their continual support and encouragement during my entire stay at Iowa State University.

ABSTRACT

Analog-to-digital converters (ADCs) are becoming increasingly more common to be involved in many systems in integrated circuits. One of the difficulties being faced is to be able to accurately and cost-effectively test the continually higher performance ADCs. Part of this test is being able to assess the dynamic linearity of the ADC through dynamic spectral testing. The standard test method for ADCs can be difficult to implement accurately and cost effectively due to the stringent requirements. Three different algorithms are proposed that can be used to relax the test requirements in order to reduce the cost of the test equipment while still being able to maintain the test accuracy. The accuracy and robustness of these methods will be shown through simulation and measurement results.

The first algorithm developed is relaxing the requirements on the linearity of the test signal and of the need to achieve coherent sampling. The standard test requires that the input signal linearity be at least 20dB more pure than the ADC under test along with always maintaining extremely accurate coherent sampling. The new algorithm will dramatically reduce the purity requirement by allowing the test signal to be much less pure than the ADC under test and at the same time also completely remove the need for coherent sampling.

The second algorithm will further relax the test set up requirements by allowing the test signal to over range the ADC input, in addition to the claims of the first algorithm. The standard test requires the input signal to be very close to the full input range of the ADC under test but to never over range since over ranging causes output clipping. This algorithm allows for the signal to be clipped up to 1% while still ensuring accurate results.

The last algorithm is performing spectral testing of the ADC using a DAC of equal or lower linearity performance. If accurate estimates of the INL/DNL of the ADC using a lower performance DAC can be obtained by other test methods, this new algorithm will generate pre-distortion codes that can be added to the DAC input so that the DAC can produce a high purity sine wave at its output, which can then be used to accurately test the dynamic spectral performance of the ADC when the previous two new algorithms are not already available.

CHAPTER 1

INTRODUCTION

Motivation

The current rapid development of technology is causing a large surge in the semiconductor industry. It is causing the industry to produce higher and higher performance technology packed into smaller areas. This is causing many new technological developments to meet the current demands. This complexity has also caused there to be a larger surge of designers creating an entire system onto one single chip called a System-on-chip or SoC. Along with this trend causing a much larger complexity in the design process of the system, it is also putting a large impact on the ability to be able to accurately and efficiently test the whole system.

In these new large SoC products, the test process can begin to become a significant portion of the cost of the whole end product. This is due to the difficulty being able to test all of the sub-blocks that are involved in the entire SoC. Therefore, it becomes beneficial to be able to keep the testing cost as minimal as possible of each sub-block in order to be able to keep the total cost of the product at a reasonable level.

One solution to this problem of testing large SoCs is to employ Built-in-Self-Test (BIST) circuits. These BIST circuits can then internally be able to test the different sub-blocks of the system more efficiently. In BIST solutions it can be difficult to have access to highly precise instrumentation which can cause some difficulties in achieving accurate testing results.

ADCs are a very popular device in many SoCs as many of today’s applications rely on operating in both the digital and analog domain; the ADC is the necessary bridge between these two. ADCs can also cause a large difficulty in testing. To get accurate test results, the amount of required data could be very large and thus take a long time to acquire and perform calculations on. Also, testing high precision ADCs will require very precise testing instrumentation, which can be difficult to design and costly. Therefore, concentrating on ADC device testing is a very important area.

There are two major areas of the ADCs functionality that need to be tested. The first is the static parameters such as the gain, offset, integral non-linearity (INL), and differential non-linearity (DNL). There are also dynamic parameters such as total harmonic distortion (THD), signal-to-noise ratio (SNR), and spurious free dynamic range (SFDR). The dynamic parameters are especially important in high speed applications such as communication or audio applications. This thesis will be focused on developing different algorithms that are focused for the dynamic spectral testing.

Standard Test

Dynamic spectral testing is achieved by sampling a sine wave input into an ADC and then by looking at the corresponding output frequency spectrum. From this output spectrum, several different calculations can be made that will describe the ADC performance. There are several different IEEE standards, [1-3], that describe the standard way upon which data converters should be tested in order to ensure that accurate results are obtained. In it there are five main criteria that are listed in order to guarantee that the spectral results of the ADC will be characterized accurately:

1. Spectral purity of input signal to be at least 3 to 4 bits more pure than ADC under test
2. The input signal range should be only slightly lower than the ADC input range
3. The signal should be sampled coherently

4. The total number of sampled points should be sufficiently high
5. The sampling clock should have a relatively low level of jitter

The first three specifications on this list can be difficult to obtain in certain testing environments and will be the most heavily targeted throughout the entirety of this thesis. Another commonly practiced method is to make the data length as a power of 2. This is to make the Fast Fourier Transform (FFT) more efficient in calculating the output frequency spectrum. In order to fulfill all of these requirements there are certain ways that are recommended to have the test setup. In this recommendation, it is desired to have the input sine wave generator and the clock that is being input into the ADC synchronized with a common master clock. This will make it possible to be able to achieve coherency.

The equation given by (1.1) demonstrates the discrete representation of an impure sine wave signal that would be achieved after a sine wave is input into an ADC and sampled. There are four components: the fundamental, the higher order harmonics, an added white noise, and a DC component. The higher order harmonics have a frequency component that are integer multiples of the fundamental frequency. These are caused by non-linear distortions from the ADC during the capture of the data. The white noise is caused by added noise to the system as well as quantization error. Lastly, there may be a DC voltage shift of the signal.

$$x[n] = A_1 cos\left(2\pi \frac{f_{signal}}{f_{sample}} n + \phi_1\right) + \sum_{h=2}^{H} A_h cos\left(2\pi \frac{h* f_{signal}}{f_{sample}} n + \phi_h\right) + w[n] + dc \qquad (1.1)$$

Upon collecting the digital data record out of the ADC, the FFT algorithm can be applied in order to get the spectral results. The Discrete Fourier Transform (DFT) that is calculated using the FFT algorithm can be described using (1.2).

$$X\left[k\right]=\sum_{n=0}^{m-1}x\left[n\right]e^{-\frac{2\pi i}{m}nk} \tag{1.2}$$

Accurate spectral results will occur if the coherent sampling condition can be achieved with high precision. This can be described in (1.3), where J is an integer and represents the number of cycles of the sinusoid that is sampled. It is optimal to make J an odd prime integer in order to achieve the most accurate results from the data length as it will ensure no data set is repeated.

$$f_{signal}=J*\frac{f_{sample}}{m} \tag{1.3}$$

It can be shown that when the coherent sampling requirement is met, then the FFT algorithm’s output makes it possible to easily recover the phase and amplitude of the fundamental and harmonic components. These can be easily derived from the respective bins of the FFT output as shown in (1.4) and (1.5).

$$X\left[J\right]=\frac{A_1}{2}e^{j\phi_1} \tag{1.4}$$

$$X\left[J*h\right]=\frac{A_h}{2}e^{j\phi_h} \tag{1.5}$$

Once the amplitudes have been found, it is then easily possible to be able to calculate some of the most widely used dynamic characteristics of the ADC. These calculations are shown in (1.6), (1.7), and (1.8).

$$THD=\frac{\sum_{h=2}^{H}A_h^2}{A_1^2} \tag{1.6}$$

$$SFDR=\frac{\frac{A_1^2}{2}}{(2*max(\left|X_k\right|^2))},k\neq J \tag{1.7}$$

$$SNR = \frac{\frac{A_1^2}{2}}{P_{noise}} \tag{1.8}$$

Figure 1 shows an example of an output spectrum where the ADC was tested accurately according to all of the criteria that were previously mentioned. The fundamental and harmonic components of the signal can be easily identified.

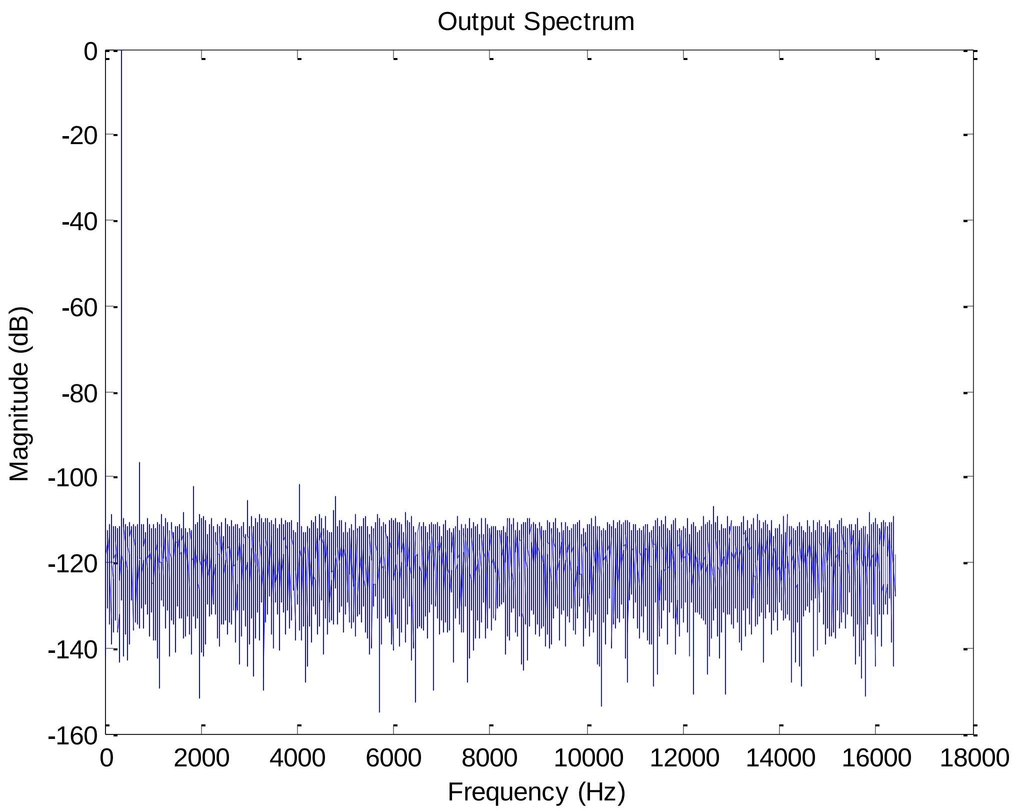


Figure 1: The output spectrum of an ADC that was tested correctly.

Challenges of Standard Test

The previous section described how to correctly test an ADC to get accurate measurements of the desired parameters. This section will show, for the first three requirements, what will occur if the exact specifications are not met.

Linear source

The input signal that is going to be used to test the ADC needs have at least 3 to 4 bits higher level of linear purity than the ADC under test. This means that about an extra 20dB higher purity is required. This becomes difficult when testing very high resolution ADCs to achieve as either the source at this performance is not available or it is costly to bring in equipment from outside of the system to be able to generate the accurate test.

To look at the effect of a non-linear source into the ADC, it is possible to look at the effects of the coefficients of the fundamental and harmonics. In the first case, a pure signal will be modeled in the (1.9) to go through a third-order non-linear system model to represent the ADC as seen in (1.10). From this, it is possible to derive the final signal which will now have accumulated two harmonics as seen calculated in (1.12).

$$x(t) = A_1 cos(wt) \tag{1.9}$$

$$y(t) = \alpha_1(x(t)) + \alpha_2(x(t))^2 + \alpha_3(x(t))^3 \tag{1.10}$$

$$y(t) = \alpha_1 A_1 cos(wt) + \alpha_2 A_1^2 \cos^2(wt) + \alpha_3 A_1^3 \cos^3(wt) \tag{1.11}$$

$$y(t) = \left(\frac{\alpha_2 A_1^2}{2}\right) + \left(\alpha_1 A + \frac{3\alpha_3 A_1^3}{4}\right)\cos(wt) + \left(\frac{\alpha_2 A_1^2}{2}\right)\cos(2wt) + \left(\frac{\alpha_3 A_1^3}{4}\right)\cos(3wt) \tag{1.12}$$

However, if the input signal is non-linear then a different set of equations will result. This example will model the input signal as a fundamental with two harmonics as well as in (1.13). When applying the same non-linear model of the ADC as before, then the new y(t) in (1.14) will result. Along with the y(t) that is shown below, there will also be some higher harmonics that are generated that are not calculated for the below example as it is just showing the effect on the similar harmonics. As it shows, the final result is now much different for the first four terms as the coefficients for each of the harmonics are different.

This will then cause error when looking at the output spectrum of the ADC and will result in the wrong characterization of the ADC.

$$x\left(t\right)=A_1cos\left(wt\right)+A_2cos\left(2wt\right)+A_3cos\left(3wt\right) \tag{1.13}$$

$$\begin{gathered}y\left(t\right)=\left(\alpha_2\left(\frac{A_1^2+A_2^2+A_3^2}{2}\right)+\alpha_3\left(\frac{3A_1^2A_2}{4}+\frac{3A_1A_2A_3}{2}\right)\right)+\\ \left(\alpha_1A_1+\alpha_2(A_1A_2+A_2A_3)+\alpha_3\left(\frac{3A_1^3}{4}+\frac{3A_1^2A_3}{4}+\frac{3A_1A_2^2}{2}+\frac{3A_1A_3^2}{2}+\frac{3A_2^2A_3}{4}\right)\right)\cos\left(\mathrm{wt}\right)+\\ \left(\alpha_1A_2+\alpha_2\left(\frac{A_1^2}{2}+A_1A_3\right)+\alpha_3\left(\frac{3A_1^2A_2}{2}+\frac{3A_1A_2A_3}{2}+\frac{3A_2^3}{4}+\frac{3A_2A_3^2}{2}\right)\right)\cos\left(2wt\right)+\\ \left(\alpha_1A_3+\alpha_2A_1A_2+\alpha_3\left(\frac{A_1^3}{4}+\frac{3A_1^2A_3}{2}+\frac{3A_1A_2^2}{4}+\frac{3A_2^2A_3}{2}+\frac{3A_3^3}{4}\right)\right)\cos\left(3wt\right)+\ldots\end{gathered} \tag{1.14}$$

Coherent sampling

Coherent sampling is a stringent requirement to make sure that the output spectrum has accurate results. The requirement is that there will be exactly an integer number of complete cycles of the sine wave as described by (1.3). If not, a “skirting” effect will be made visible. This effect can be seen in the Figure 2 even where the number of cycles is very close to an integer value. When this happens, the amplitude for the given signal frequency is no longer stored in just one bin, so the equations that were given to calculate the amplitude before will no longer be valid.

Achieving coherent sampling in a system can become very difficult. As the example shows, even a small error in the value of J will result in corrupted data. Achieving this high precision of frequency control can be very difficult. The way that this can be implemented is to have the signal generator and ADC clock synchronized to the same master clock. If both are derived from the same clock, then achieving coherent sampling will be possible.

However, this is not always a possibility and therefore the requirement of coherent sampling will be difficult to achieve.

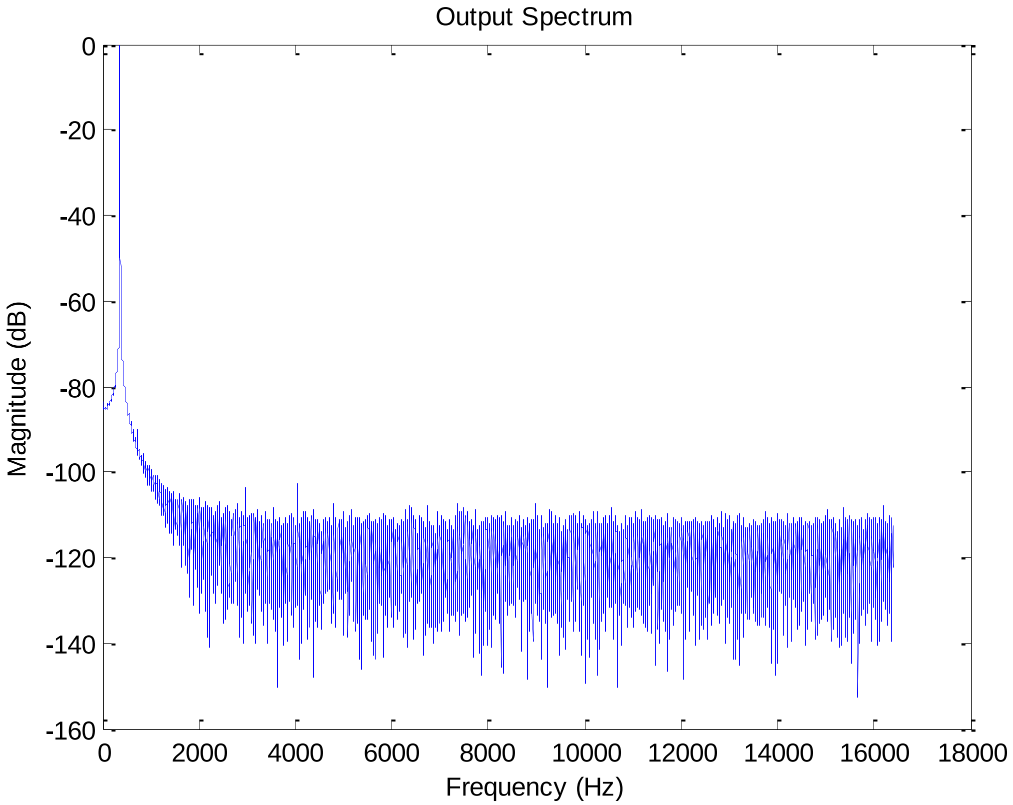


Figure 2: A sinusoid wave that is sampled non-coherently where J=369.01.

Non-clipped signal

The last requirement of interest is that the signal must be near the full-range of the ADC without any clipping. If the signal is much lower than the full-range, the portion of the ADC range that is not tested will not be accounted for in the final measurements. If it goes over the range and clips, many large distortion components due to clipping will appear in the ADC output spectrum, causing severe difficulties in correctly testing the ADC's spectral performance. This can be an issue as it can be very difficult to get a signal as close as possible to the full-range without ever clipping. This clipping can occur because of added noise to the system or some gain of the system that isn't the exact value that was intended.

Therefore, some level of clipping could become possible and then it could result in inaccurate spectral results. In Figure 3, it demonstrates how an output spectrum looks if the signal is over the input range of the ADC and clipped by only 1%. As it shows, there can be no accurate information obtained directly from this spectrum due to the extra spurs caused by the clipping effect.

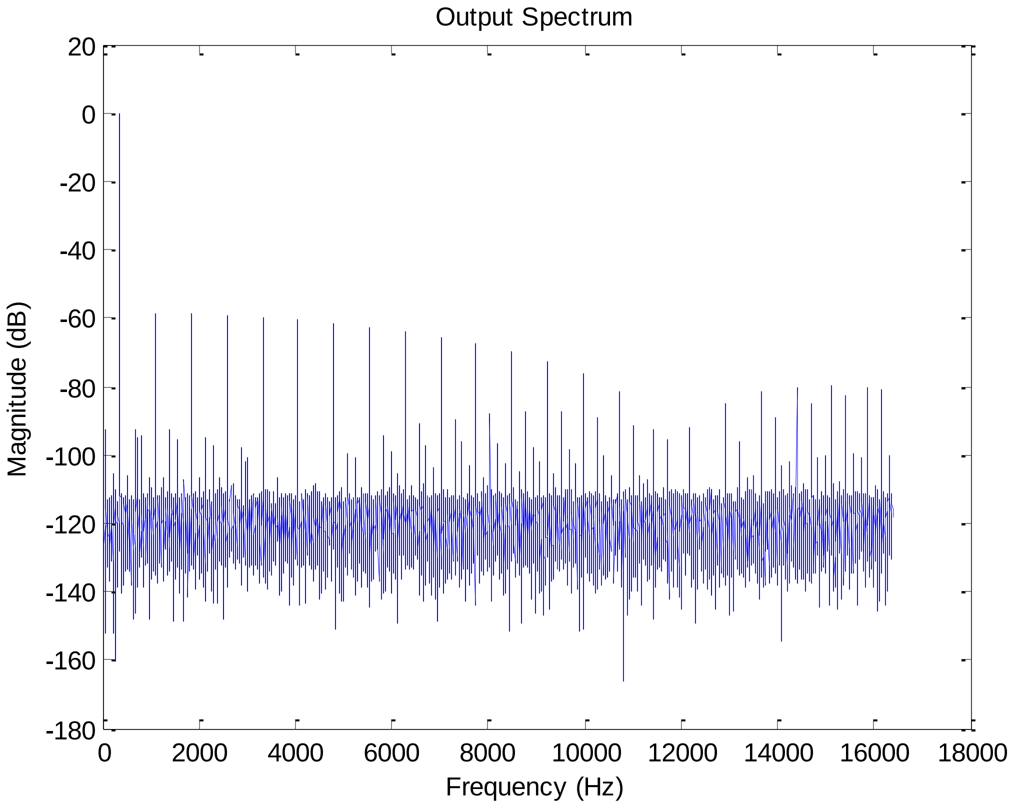


Figure 3: The output spectrum out of an ADC when the input signal is clipped by 1%.

Paper Outline

The rest of this thesis will be discussing alternative algorithms to alleviate the problems that are discussed in the previous section

Chapter 2 will introduce a method that will be able to acquire accurate spectral results without requiring a linear input source or coherent sampling. It will reduce the requirement of the purity of the test signal to be well below the purity of the ADC under test. It will also get rid of the requirement of coherent sampling and still be able to produce accurate spectral test results.

Chapter 3 will also allow clipping of the input source in addition to the previous claims of the algorithm from Chapter 2. Instead of just being near the full input range of the ADC, it allows the signal to be clipped by up to 1% while still maintaining accurate results.

Chapter 4 will discuss a method to be able to acquire accurate spectral results of the ADC by testing with a DAC of similar resolution and similar or worse performance. The requirement is that the approximate INL/DNL of the ADC under test will need to be known. Pre-distortion codes can be calculated and applied to the input of the DAC to make the output signal more pure. When applying this new output signal to the ADC, the ADC can be tested more accurately.

Finally, Chapter 5 will provide a summary of the previous chapters and will elaborate on the overall impact of the research performed.

CHAPTER 2

ACCURATE SPECTRAL TESTING WITH NON-LINEAR SOURCE AND NON-COHERENT SAMPLING

Introduction

As described in Chapter 1, the input signal that is going to be used to test the ADC needs to be at a higher level of linear purity than the ADC under test. Usually at least 20dB more pure is desired, so an input signal with a THD of -115dB or better would be required in order to test a 16-bit ADC with a target THD performance of -95dB. In most systems, a signal generator of this purity is not going to be readily available. Therefore, either extra design effort will have to be made to create an accurate signal generator for test or a signal from off chip will have to be used to be able to provide the test signal needed. Neither of these choices are attractive options as they both increase the cost of test.

Some work has been previously worked on to reduce the difficult condition of the required highly pure input test signal. In [4], an algorithm was described in order to be able to use filters on the input signal in order to get accurate results. Simulation results were shown to validate the method. A similar approach is shown in [5], where again filters are used on the input signal in order to achieve accurate results. Both of these methods only worked on reducing the input signal requirement for the ADC test.

Coherent sampling is another parameter that is required in order to obtain accurate spectral results. If it is not achieved, the skirting effect that was demonstrated in Chapter 1 will introduce errors into calculating the spectral characteristics. Coherent sampling is difficult to achieve without having a common clock source being shared between the ADC under test and the signal generator. This may not be already present in the current SoC; therefore trying to incorporate something that will make this possible could result in a redesign which also increases cost.

Achieving solutions for acquiring accurate spectral results without requiring coherent sampling has long been a focus of research. There are many different strategies for attempting to resolve this problem. Windowing techniques have been widely used in order to achieve good data from non-coherently sampled data as seen in [6] and [7]. However, this can lead to inaccurate results depending on the resolution and type of windows being used on the specific output data. Four parameter sine wave fitting has also been used as seen in [8] and [9]. The time it takes to perform these methods can become a concern, however, when large data sets are used. Another method that has been proposed is the Fundamental Identification and Replacement Method (FIRE) presented in [10]. However, all of these methods are only for fixing the non-coherent sampling issue.

As none of the above described research has dealt with a situation involving a non-pure input signal that is sampled non-coherently, there is currently no good solution for the detailed problem. The method that will be proposed in the next section will be able to reduce the requirement of the input signal purity, while also completely removing the requirement to achieve coherent sampling. The proposed method will be able to test a 16-bit ADC with a target THD of around -95dB by using a test input signal that has a THD as bad as -50dB. A test signal with this level of purity would be very easy to design as well as easily accessible in most systems that are being designed. An added benefit to this proposed method is that the non-pure signal generator will also be able to be characterized at the same time as the ADC under test is being measured. The following sections will show the test setup for the proposed algorithm, the algorithm derivation, and simulation results that prove the accuracy and robustness of the method. Lastly, measurement results from a 16-bit ADC will be discussed for further verification.

Non-linear Modeling

Before entering the discussion on the algorithm, a review of the non-linear modeling of the ADC and input signal will be presented. For simplicity, the ADC will be modeled as a third order non-linear equation as shown in (2.1) as an illustrating example. The value of α_1 will be near one and any deviation is due to the ADC's gain error. The values α_2 and α_3 will be less than 1E-4 as this will be modeling a high performance ADC. For real ADCs, the non-linearities will be more complex but the principle will be the same and the results will be similar. The input signal will consist of a fundamental and two higher order harmonics as shown in (2.2). In this equation, A_1 should have a value near but less than one. This ensures full range testing of the ADC without clipping. The values A_2 and A_3 should be much less than one but depend on how non-linear the input signal is. The constraints on these values will be later determined through simulation results. This signal could also have more harmonics in it as in a more real model, but again the equations will follow the same trend. The result of the input passing through the non-linear model that was derived in Chapter 1 is also showed here again in (2.3).

$$y(t) = \alpha_1\left(x(t)\right) + \alpha_2\left(x(t)\right)^2 + \alpha_3\left(x(t)\right)^3 \tag{2.1}$$

$$x(t) = A_1 cos\left(wt\right) + A_2 cos\left(2wt\right) + A_3 cos\left(3wt\right) \tag{2.2}$$

$$
\begin{aligned}
y(t) = {} & \left(\alpha_2 \left(\frac{A_1^2 + A_2^2 + A_3^2}{2} \right) + \alpha_3 \left(\frac{3A_1^2 A_2}{4} + \frac{3A_1 A_2 A_3}{2} \right) \right) + \\
& \left(\alpha_1 A_1 + \alpha_2 (A_1 A_2 + A_2 A_3) + \alpha_3 \left(\frac{3A_1^3}{4} + \frac{3A_1^2 A_3}{4} + \frac{3A_1 A_2^2}{2} + \frac{3A_1 A_3^2}{2} + \frac{3A_2^2 A_3}{4} \right) \right) \cos(\mathrm{wt}) + \\
& \left(\alpha_1 A_2 + \alpha_2 \left(\frac{A_1^2}{2} + A_1 A_3 \right) + \alpha_3 \left(\frac{3A_1^2 A_2}{2} + \frac{3A_1 A_2 A_3}{2} + \frac{3A_2^3}{4} + \frac{3A_2 A_3^2}{2} \right) \right) \cos(2wt) + \\
& \left(\alpha_1 A_3 + \alpha_2 A_1 A_2 + \alpha_3 \left(\frac{A_1^3}{4} + \frac{3A_1^2 A_3}{2} + \frac{3A_1 A_2^2}{4} + \frac{3A_2^2 A_3}{2} + \frac{3A_3^3}{4} \right) \right) \cos(3wt) + \ldots
\end{aligned}
\tag{2.3}
$$

The DC term should not have any effect on our measurement as it should be a very small change as well as the DC component doesn't have an effect on any of our parameters. The fundamental also should not have a large effect in respect to its original value. The second harmonic value, however, is of concern. The term $\alpha_1 A_2$ is directly passed from the input signal. This portion will be taken care of in the algorithm of this section. The $\frac{\alpha_2 A_1^2}{2}$ term is the non-linear term that is generated by the fundamental from the ADC. This is the term that is desired to characterize the ADC. The other terms, however will cause error in the measurement. Upon examining these terms, it is evident that they should not have a large effect on the measurement as long as the original non-linear harmonics are sufficiently smaller than the fundamental amplitude. This is due to all other terms having the amplitudes of the harmonics included in them. If the original non-linear harmonics from the signal generator become larger however, then large errors could begin to appear in the final output spectrum. A similar analysis can be done on the other harmonic terms.

Proposed Algorithm

The proposed algorithm of this section is going to be able to alleviate the non-linear input signal problem that is discussed in the previous section. It will also eliminate the need to achieve coherent sampling.

Algorithm Flow

The method that the proposed algorithm is going to use can be seen in Figure 4. As it shows, a non-linear source is going to be passed through two separate filters. The filters will be discussed further in the next section. These two output signals will then be sampled by the same ADC separately. After the ADC has finished sampling the input signal, the output data from each run will then be passed into the non-linear algorithm.

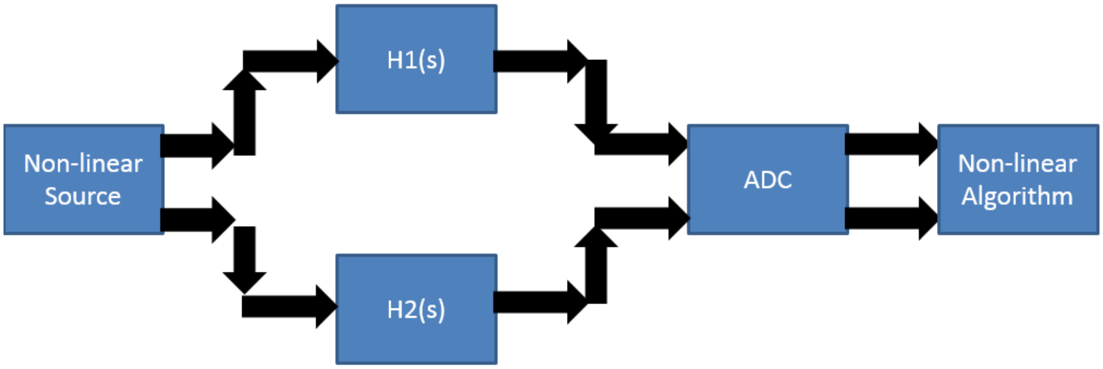


Figure 4: Algorithm flow with coherent sampling.

The above case will only work when the signal is coherently sampled. However, if the signal is not exactly coherently sampled then as previously discussed in Chapter 1, the results of taking the FFT of the ADC output data will result in errors in the output spectrum. Therefore, to correct for this problem, an additional block needs to be added into the algorithm flow as shown in Figure 5.

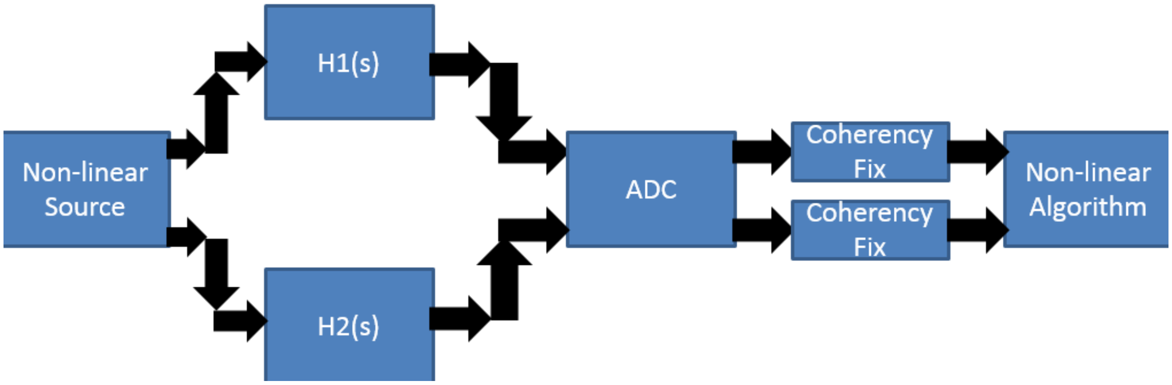


Figure 5: Algorithm flow without coherent sampling.

As it shows, there is only one more addition to the algorithm. This is that there needs to be a coherency fix applied to both of the digital outputs of the ADC before they enter into the non-linear algorithm to ensure accurate spectral results. The coherency fix block will be discussed in one of the next sections.

Filter Characterization/Adjustment

There are no specific criteria of what the two filters' designs must be. However, one of the keys to this algorithm succeeding is to have both sets of signals coming out of the two different filters to have approximately the same amplitude. This will ensure that the non-linearities added to the output from the ADC are going to be the same on both signals. However, in designing the filters, passive components will be used. The values of these devices will most likely be varied from the specification by possibly over 5%. This can then make the amplitudes not matched very accurately.

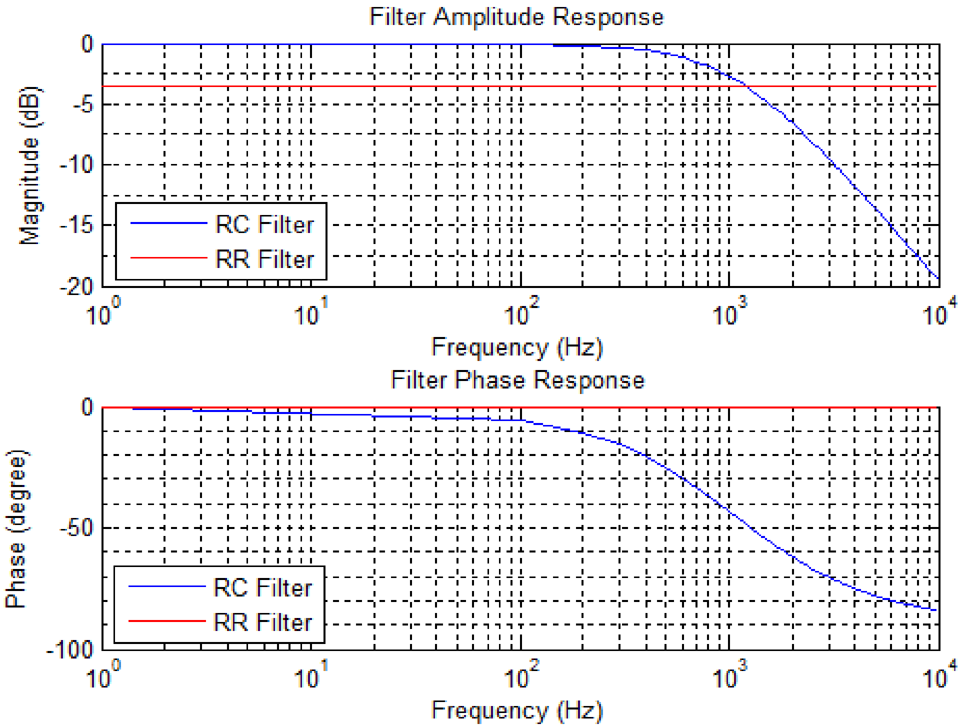


Figure 6: Plots showing the transfer function of two different simple filters.

To alleviate the error caused by this filter uncertainty, several different methods can be implemented. If simple passive components are being used then banks of devices can be tuned using digital components. An example would be having a binary weighted capacitor bank that can then be adjusted by using a digital signal. A different approach would be to change the frequency of the input signal to match the amplitude response of the two different filters. A simple example of filter choices can be seen in Figure 6 as a simple RC low-pass filter and then a simple resistor attenuator filter. With these choices, the frequency could just be adjusted until the RC filter has the same amplitude response as the constant RR filter.

The next problem, after tuning the different filters, is being able to know what the approximate transfer curve of the two different filters. If the filters are just one pole systems as in a standard low-pass RC filter, then a simple two-point characterization can be used. This would be done by measuring the filter at a DC input and at a higher frequency. By using the non-linear signal generator and ADC under test, an accurate enough characterization of the filters can be calculated in order to successfully run the algorithm. More complicated filters past a one-pole system can be used; it just may take extra work in order to be able to achieve a relatively good estimate of the transfer curve of the filter.

There is one last requirement that needs to be put on the two filters. This is that the components used in the filter cannot add any extra non-linearities into the signal. If the two filters add different non-linearities to the input signal that is at a level relative to the ADC’s non-linearities, then the proposed algorithm will not be able to produce accurate results, as the algorithm cannot take this into account.

Non-coherent Correction

As discussed in Chapter 1, when coherent sampling is not achieved the accuracy of the results will be largely affected. Therefore, for this algorithm it needs to be ensured that both sets of data out of the ADC are accurate. Another issue that can come into play with this algorithm is that the source could be very non-linear; therefore there could be large harmonics in the spectrum which could affect the initial estimate of the fundamental. The flow of this coherency fix can be seen in Figure 7.

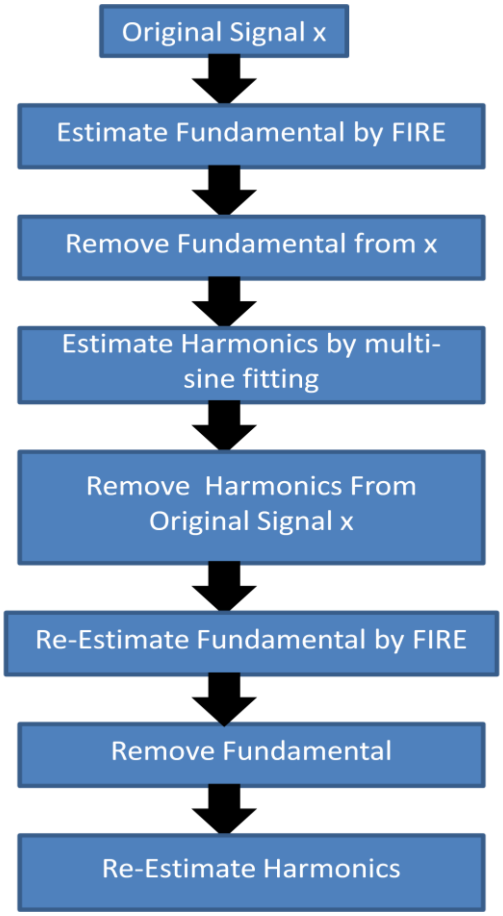

Original Signal x
Estimate Fundamental by FIRE
Remove Fundamental from x
Estimate Harmonics by multi-sine fitting
Remove Harmonics From Original Signal x
Re-Estimate Fundamental by FIRE
Remove Fundamental
Re-Estimate Harmonics

Figure 7: The algorithm flow for correcting a non-coherently sampled output of the ADC.

For this part of the algorithm, it will be using the previously mentioned FIRE method in [10] for its ability to quickly and accurately calculating the fundamental component of a signal. After having estimated the fundamental's amplitude, frequency, and phase, it is possible to remove this component from the data. Then after the fundamental removal, it is possible to estimate the harmonics by using a multiple three parameter sine fitting method to find all of the harmonics components.

Once the multi-sine wave fitting method has been completed to accurately estimate the harmonics, the algorithm may be finished. This will be true if the non-linearities of the signal are not too large. Then the estimates by this method up to this point will be accurate enough for the rest of the algorithm to work. However, if the signals contain very large non-linearities, then the non-coherently sampled harmonics could have an effect on the initial estimate of the fundamental, which could then affect the accuracy of the harmonics as well. The estimates, however, will be good enough to be able to help improve the estimates on the second pass.

The second half of the algorithm is done by first removing the large harmonics with the estimates done in the first pass from the original signal. This will then reduce the power of the harmonics enough that it should make the effect of the harmonic power leakage onto the fundamental negligible for the FIRE algorithm. This will then give a more accurate estimate of the fundamental, including a more accurate frequency estimation of the fundamental. This will then in turn allow for a more accurate multi-sine wave fitting algorithm of the harmonics. Once the fundamental and harmonics have been accurately estimated, the data can then be passed into the non-linear algorithm for final calculations to be performed.

Signal Derivation

Before deriving the individual non-linearities of the ADC & signal generator, it is necessary to derive what the two different signals will look like when they are out of the sampled ADC. For these derivations, all signals are assumed either to be coherent or corrected by the non-coherency correction algorithm of the previous section.

The signal starts by two different signals being originally generated by a non-linear source. The two equations in (2.4) and (2.5) represent the two signals directly out of this non-linear signal generator. The variable D will represent the amplitude of the fundamental and harmonics out of the non-linear source and the variable ϕ will represent the phase of these different components. The variable δ_1 represents the phase difference of the fundamental between the two different signal passes. This is necessary as it may not be possible to start both signal passes at exactly the same time without any added difficulty.

$$Signal_1 = \left|D_1\right| e^{j\omega_0 nT_s} e^{j(\phi_1)} + \sum_{k=2}^{H} \left|D_k\right| e^{jk\omega_0 nT_s} e^{j(\phi_k)} \tag{2.4}$$

$$Signal_2 = \left|D_1\right| e^{j\omega_0 nT_s} e^{j(\phi_1+\delta_1)} + \sum_{k=2}^{H} \left|D_k\right| e^{jk\omega_0 nT_s} e^{j(\phi_k+k*\delta_1)} \tag{2.5}$$

After the signals are generated, they will each separately pass through their respective filters. This will apply the amplitude and phase response of the filters to both signals. This can be seen in (2.6) and (2.7) where ψ is the phase response of the first filter and φ is the phase response of the second filter.

$$Signal_1 = \left|D_1 * H_1\left(j\omega_0\right)\right| e^{j\omega_0 nT_s} e^{j(\phi_1+\psi_1)} + \sum_{k=2}^{H} \left|D_k * H_1\left(jk\omega_0\right)\right| e^{jk\omega_0 nT_s} e^{j(\phi_k+\psi_k)} \tag{2.6}$$

$$Signal_2 = \left|D_1 * H_2\left(j\omega_0\right)\right| e^{j\omega_0 nT_s} e^{j(\phi_1+\delta_1+\varphi_1)} + \sum_{k=2}^{H} \left|D_k * H_2\left(jk\omega_0\right)\right| e^{jk\omega_0 nT_s} e^{j(\phi_k+k*\delta_1+\varphi_k)} \tag{2.7}$$

After the filters, the signals will pass through the ADC. This will then add in the extra non-linearities from the ADC as seen in (2.8) and (2.9). The amplitude of these added non-linearities are given by the variable C and the phase by γ. The initial phase of the added non-linearities will be directly related to the fundamental’s phase. As long as the fundamental’s amplitude of both signals is nearly the same as they enter the ADC, then the non-linearities of the ADC will be the same. The amplitudes can be made this close by just using the ADC as measurement when adjusting the amplitude response of the filter.

$$\begin{aligned}\text{Signal}_1 &= \left|D_1 * H_1\left(j\omega_0\right)\right| e^{j\omega_0 nT_s} e^{j(\phi_1+\psi_1)} + \sum_{k=2}^{H} \left|D_k * H_1\left(jk\omega_0\right)\right| e^{jk\omega_0 nT_s} e^{j(\phi_k+\psi_k)} \\ &+ \sum_{k=2}^{H} \left|C_k\right| e^{jk\omega_0 nT_s} e^{j(k*\phi_1+k*\psi_1+\gamma_k)}\end{aligned} \tag{2.8}$$

$$\begin{aligned}Signal_2 &= \left|D_1 * H_2\left(j\omega_0\right)\right| e^{j\omega_0 nT_s} e^{j(\phi_1+\delta_1+\varphi_1)} + \sum_{k=2}^{H} \left|D_k * H_2\left(jk\omega_0\right)\right| e^{jk\omega_0 nT_s} e^{j\left(\phi_k+k*\delta_1+\varphi_k\right)} \\ &+ \sum_{k=2}^{H} \left|C_k\right| e^{jk\omega_0 nT_s} e^{j(k*\phi_1+k*\delta_1+k*\varphi_1+\gamma_k)}\end{aligned} \tag{2.9}$$

The actual measurement output out of the ADC can be comprised of a fundamental and higher order harmonics. The two different outputs can be represented by (2.10) and (2.11) with the corresponding amplitudes and phases. These values can be accurately acquired given that either the original signal was coherently sampled, or the non-coherent fix algorithm was applied to the raw output codes of the ADC.

$$Output_1 = \left|M_1\right| e^{j\omega_0 nT_s} e^{j(\alpha_1)} + \sum_{k=2}^{\mathrm{H}} \left|M_k\right| e^{jk\omega_0 nT_s} e^{j\alpha_k} \tag{2.10}$$

$$Output_2 = \left|N_1\right| e^{j\omega_0 nT_s} e^{j(\beta_1)} + \sum_{k=2}^{\mathrm{H}} \left|N_k\right| e^{jk\omega_0 nT_s} e^{j\beta_k} \tag{2.11}$$

Non-linear Algorithm Derivation

Now that the signals coming out of the ADC are derived and known, it is possible to begin the derivation of the non-linear algorithm. The first step is going to be able to calculate the initial angles. This can be accomplished by using (2.12) and (2.13).

$$\phi_1 = \alpha_1 - \psi_1 \tag{2.12}$$

$$\delta_1 = \beta_1 - \phi_1 - \varphi_1 \tag{2.13}$$

After these calculations, and an analysis of the previously derived equation, it is desired to cancel out the harmonics caused by the signal generator in order to isolate the harmonics caused by the ADC. To accomplish this, the amplitude ratio of the filters as well as a phase shift needs to be applied to (2.8) and (2.10). Then by subtracting the shifted versions of (2.8) and (2.10) from (2.7) and (2.9), the expressions of (2.14) can be derived, while only looking at the harmonics. These two different equations can then be set equivalent to each other.

$$\begin{gathered}
|M_k| e^{jk\omega_0 nT_s} e^{j\alpha_k} - |N_k| e^{jk\omega_0 nT_s} e^{j\beta_1} * \left|\frac{H_1(jk\omega_0)}{H_2(jk\omega_0)}\right| e^{j(\psi_k - k*\delta_1 - \varphi_k)} = \\
|C_k| e^{jk\omega_0 nT_s} e^{j(k*\phi_1 + k*\psi_1 + \gamma_k)} - |C_k| e^{jk\omega_0 nT_s} e^{j(k*\phi_1 + k*\delta_1 + k*\varphi_1 + \gamma_k)} * \left|\frac{H_1(jk\omega_0)}{H_2(jk\omega_0)}\right| e^{j(\psi_k - k*\delta_1 - \varphi_k)}
\end{gathered} \tag{2.14}$$

Then by solving for the harmonic component due to the ADC in (2.14), (2.15) can be achieved. This equation then gives all of the information about the amplitude and phase of all of the harmonic components that are generated due to the ADC's non-linearities.

$$|C_k| e^{j(\gamma_k)} = \frac{|M_k| e^{j\alpha_k} - |N_k| e^{j\beta_1} * \left|\frac{H_1(jk\omega_0)}{H_2(jk\omega_0)}\right| e^{j(\varphi_1 - \varphi_k - \psi_1 + \psi_k)}}{e^{j(k*\phi_1 + k*\psi_1)} - e^{j(k*\phi_1 + k*\delta_1 + k*\varphi_1)} * \left|\frac{H_1(jk\omega_0)}{H_2(jk\omega_0)}\right| e^{j(\psi_k - k*\delta_1 - \varphi_k)}} \tag{2.15}$$

After the ADC's non-linearities have been solved it is then possible to solve for the signal generator's non-linearities. This can be done in a similar fashion as when solving for the ADC's non-linearities. This time, however, the ADC's non-linearities will need to be cancelled out. This can be done by just providing a phase shift to (2.8) and (2.10) and then subtracting them from (2.7) and (2.9). These two equations can then be set equal to each other. The result of this manipulation can be seen in (2.16).

$$\begin{aligned}&\left|M_k\right|e^{jk\omega_0 nT_s}e^{j\alpha_k}-\left|N_k\right|e^{jk\omega_0 nT_s}e^{j\beta_1}*e^{j\left(k*\psi_1-k*\delta_1-k*\varphi_1\right)}=\\&\left|D_k*H_1\left(jk\omega_0\right)\right|e^{jk\omega_0 nT_s}e^{j(\phi_k+\psi_k)}-\left|D_k*H_2\left(jk\omega_0\right)\right|e^{jk\omega_0 nT_s}e^{j\left(\phi_k+k*\delta_1+\varphi_k\right)}\\&*e^{j\left(k*\psi_1-k*\delta_1-k*\varphi_1\right)}\end{aligned} \tag{2.16}$$

Then by rearranging and solving for the non-linearities of the signal generator, (2.17) can be found. This equation will then be able to quantify all of the different harmonic components that were generated due to the non-linearities of the signal generator that was used for testing the ADC.

$$\left|D_k\right|e^{j(\phi_k)}=\frac{\left|M_k\right|e^{j\alpha_k}-\left|N_k\right|e^{j\beta_1}*e^{j\left(k*\psi_1-k*\delta_1-k*\varphi_1\right)}}{\left|H_1\left(jk\omega_0\right)\right|e^{(j\psi_k)}-\left|H_2\left(jk\omega_0\right)\right|*e^{j\left(k*\delta_1+\varphi_k\right)}*e^{j\left(k*\psi_1-k*\delta_1-k*\varphi_1\right)}} \tag{2.17}$$

After both of the non-linearities have been correctly computed, then all of the parameters of the system have been successfully solved for. This would then allow for a reconstruction of the spectrums of the ADC and signal generator. To get the output spectrum of the ADC, all of the harmonics from the signal generator with the added effect of the used filter would just need to be removed from the output data. Then the resulting spectrum would have components that would be solely caused by the ADC. If the signal generator's spectrum was desired, the same thing could be done by removing the ADC's harmonics and then adding in the effect of the filter's transfer function to the resulting data.

Simulation Results

In this section, simulations results will be shown in order to verify the accuracy of the algorithm proposed in this chapter. This was done by first testing a generated ADC model with a pure input source that is sampled coherently in order to comply by the traditional test methods. Then a non-linear signal is generated and filtered through the two different filters. The same ADC digitizes both of the signals. Then the output signals are passed into the proposed algorithm. The results can then be compared between the two different methods to see the resulting accuracy.

Figures 9 and 10 show an example of one simulation run. In Figure 9, it shows one of the raw output data's spectrums as well as what the real ADC spectrum is from using the traditional test method. As it shows the raw data is skirting due to the non-coherent sampling. Figure 10 shows the comparison of the traditional test method versus the calculated output spectrum that is generated from the non-pure input source that is sampled non-coherently. As it shows, the two different spectrums now line up together accurately. The different harmonic components as well as the fundamental component are estimated accurately.

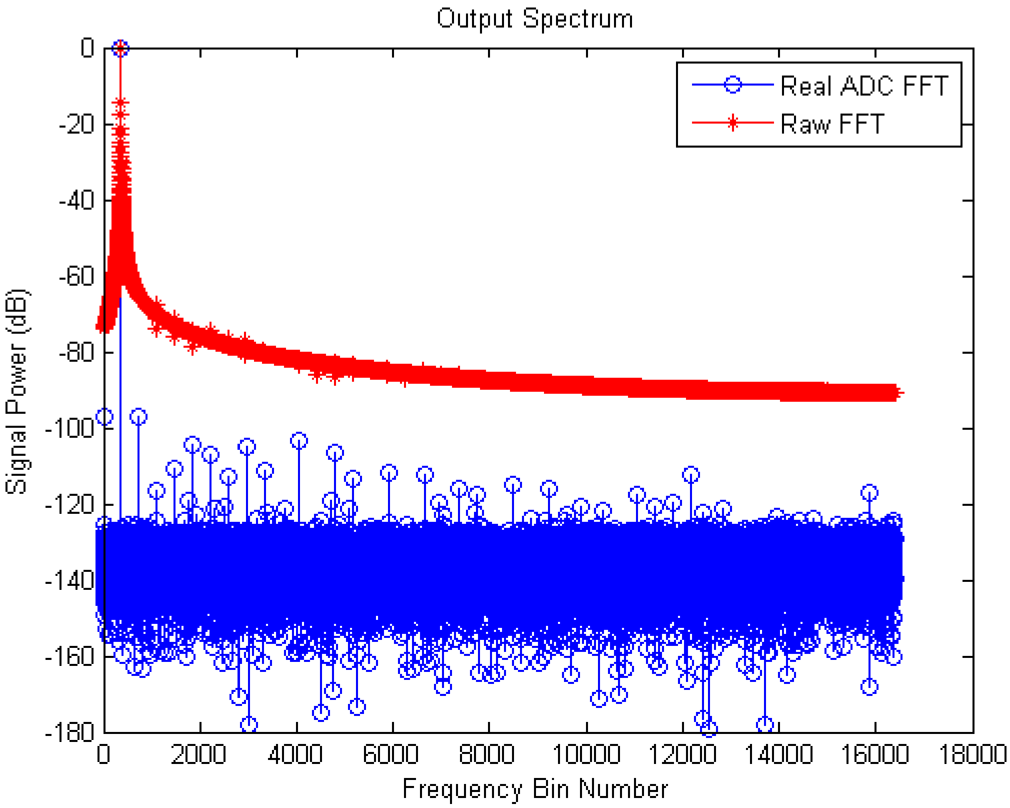


Figure 8: A spectrum of the raw data out of the ADC versus the real spectrum of the ADC using the traditional test.

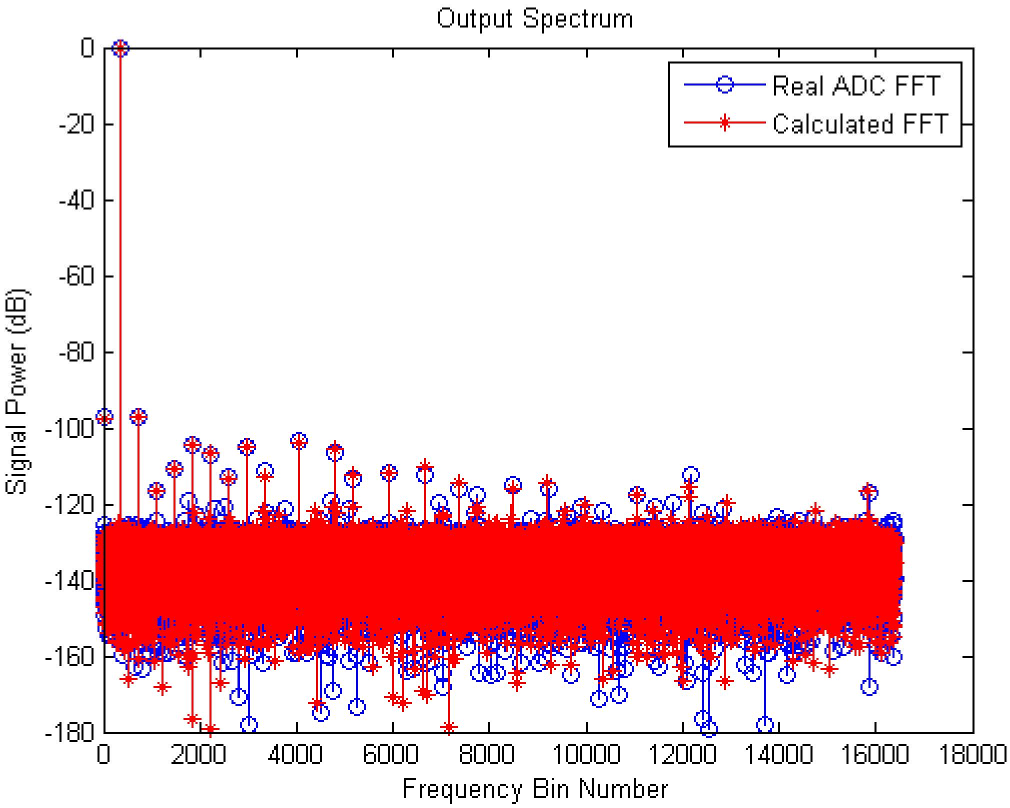


Figure 9: A spectrum of calculated spectrum of the ADC versus the real spectrum of the ADC using the traditional test.

A summary of two different simulations runs can be seen in Tables 1 through 4. The first run is summarized in Table 1 & 2 and the second run is in Table 3 & 4. As it shows, accurate results can be obtained for both the ADC and the signal generator using the proposed algorithm for varying purities of the signal generator. The first run has a signal generator that is about 10dB worse than the ADC under test. The second run still has accurate results, even with a signal generator whose purity is 45dB worse than the ADC under test.

Table 1: ADC characterization results from run 1.

ADC Characterization		
Method	THD (dB)	SFDR (dB)
Pure Coherent Signal	-93.98	97.04
Proposed Algorithm	-93.77	96.9

Table 2: Signal generator characterization results from run 1.

Signal Generator Characterization		
Method	THD (dB)	SFDR (dB)
Pure Coherent Signal	-83.87	91.02
Proposed Algorithm	-83.84	90.88

Table 3: ADC characterization results from run 2.

ADC Characterization		
Method	THD (dB)	SFDR (dB)
Pure Coherent Signal	-93.98	97.04
Proposed Algorithm	-93.88	97.29

Table 4: Signal generator characterization results from run 2.

Signal Generator Characterization		
Method	THD (dB)	SFDR (dB)
Pure Coherent Signal	-57.63	66.36
Proposed Algorithm	-57.64	66.39

Robustness

The robustness of the proposed algorithm is also desired to be verified. There are several different parameters that could affect the performance of the algorithm: input signal purity level, noise, and level of coherency. If the filter is not approximately known this could also cause some error. However, since passive components will normally have a large variation in the value, it is better to use the filter characterization proposed in this chapter in order to have a good estimation, even if large component variation does exist in the filter. This will then lead to no added error.

Input Signal Purity

The first characteristic that will be used in order to test the robustness of the proposed algorithm is the purity level of the input signal. This will be measured by comparing the total power difference of the harmonics that are calculated from the power of the harmonics that are measured using the traditional method. The power difference should be well below the level of the power of the harmonics that are being measured. The same ADC is used as the previous examples, so the real THD is at the -94dB level. The parameter that will be swept then is the THD of the signal generator source. This is done by randomly generating different signal generators with different levels of purity. There are a total of 10,000 randomly generated simulations. The results of this can be seen in Figure 10. As it shows, accurate results are achieved until the THD of the signal generator reaches levels that are worse than the -50dB level. This then shows that the algorithm is robust to the signal generator's purity as bad as the -50dB level for THD. This level of purity in a test signal will be easy to achieve.

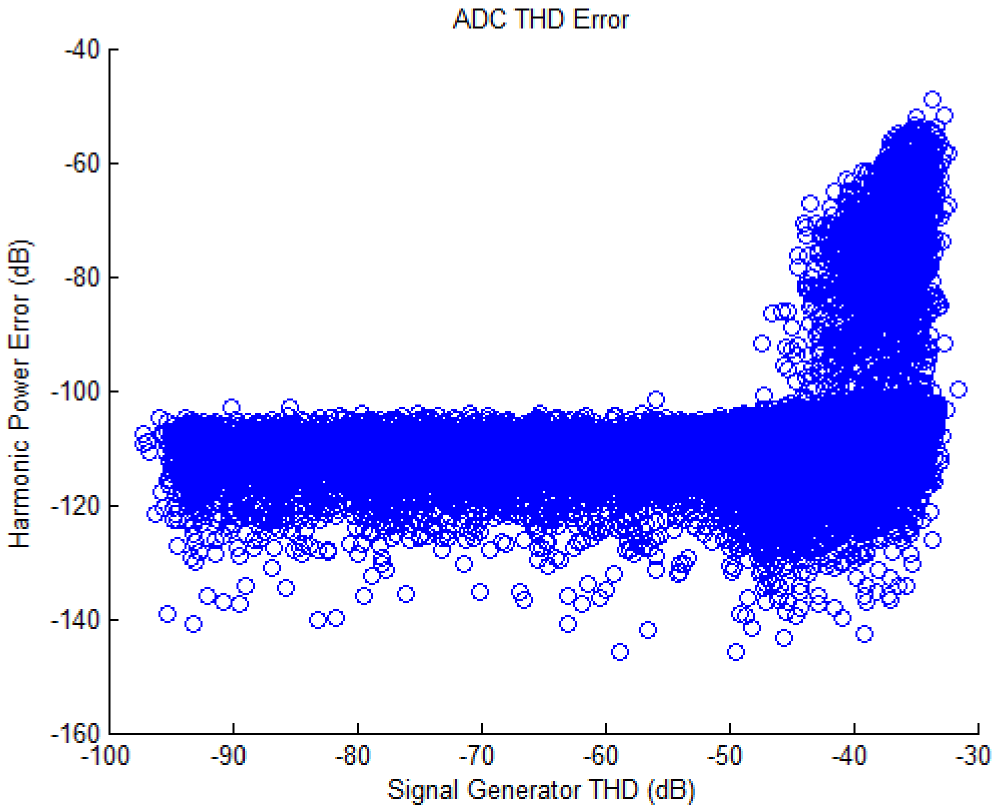


Figure 10: A plot showing the harmonic power estimation error versus the signal generator's THD.

Noise

The amount of noise in a system can directly affect the accuracy of different measurements. This is especially true for high precision ADCs. When the noise level become too high, the noise power can influence the estimation of the harmonics' power. This will lead to incorrect characterization of the spectral performances of the ADC. Therefore, it is important to see how robust a method is to different levels of noise. In this discussion the noise level should be around the same magnitude as the value of the least significant bit (LSB) in the ADC. A simulation of the harmonic power estimation can be seen in Figure 11.

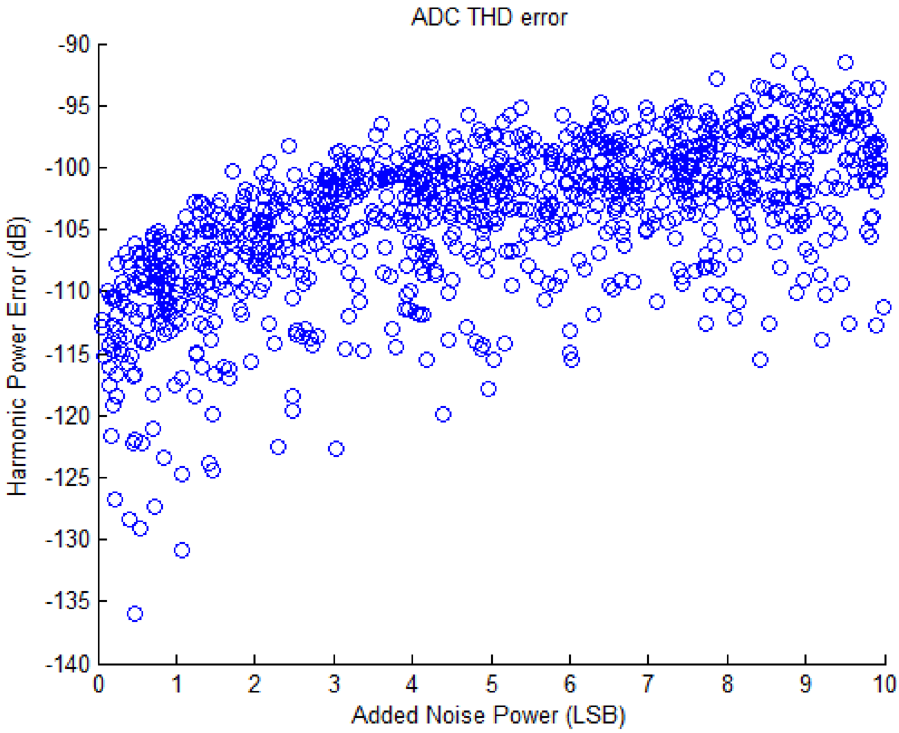


Figure 11: A plot showing the harmonic power estimation error versus the added noise power in least significant bits.

In these simulations the same ADC with a THD of -94dB was used with a sampled data set size of 2^{15}. Therefore, to achieve high accuracy results, it can be shown that only up to around a few LSBs of noise should be present in the measurement system in order to achieve highly accurate results. With more noise results will still be fairly accurate, but at 10

LSBs of noise the error starts to approach the same magnitude as the measurement itself. These low levels of noise to maintain the accuracy is achievable in realistic test setups.

Coherency

One of the goals of this algorithm was to completely remove the requirement of coherent sampling from the test setup. Coherent sampling is achieved by having exactly an integer number of cycles of the sine wave. This integer is given by J_{int}, and the real total number of cycles is given by J. The difference between these two values is then given by δ. This relationship can be seen in (2.18).

$$J = J_{int} + \delta \tag{2.18}$$

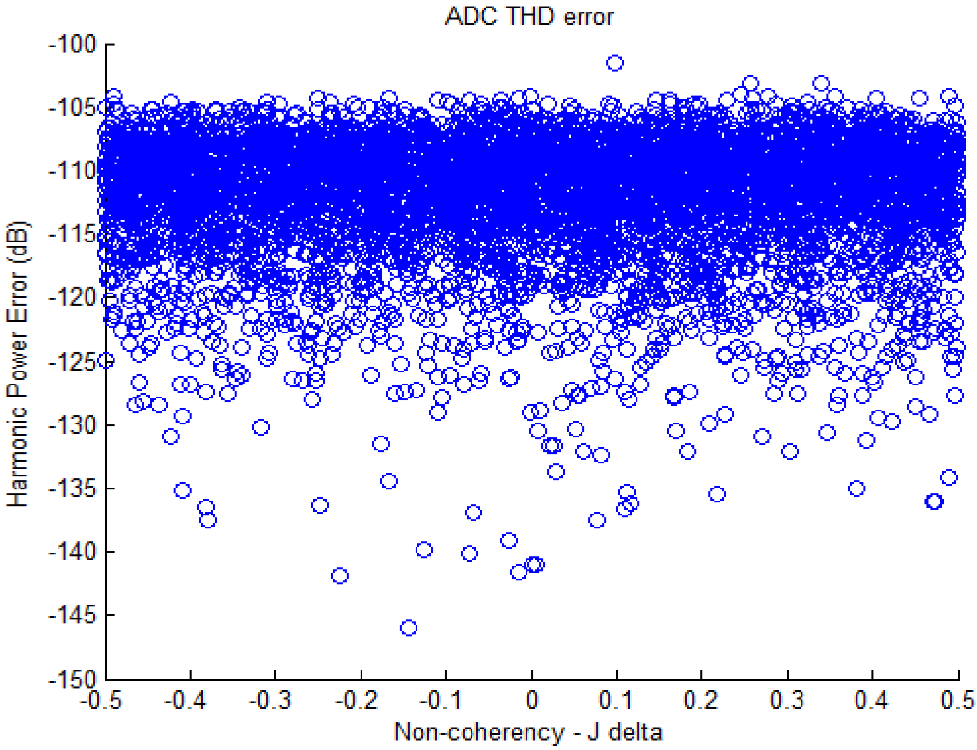


Figure 12: A plot showing the harmonic power estimation error versus the value of δ.

The value of δ has a major effect on the output spectrum. If the value is close to zero, or close to coherent, there will be less of a skirting effect than if this value is farther away from zero. Therefore, it is good to ensure that this δ value does not have a large effect on the measurement results to ensure that the coherent sampling requirement is in fact completely removed. The results of varying this δ value can be seen in Figure 12. As it shows, the harmonic power estimation error is constant across all δ, so therefore it can be said that the coherent sampling requirement is in fact removed.

Measurement Results

To further validate the proposed algorithm, it was necessary to produce measurement results to achieve similar accurate results. A printed circuit board (PCB) test board was designed and created in order to be able to collect the data required in order to verify the algorithm with measurement results. The created PCB test board can be seen in the below Figure 13.

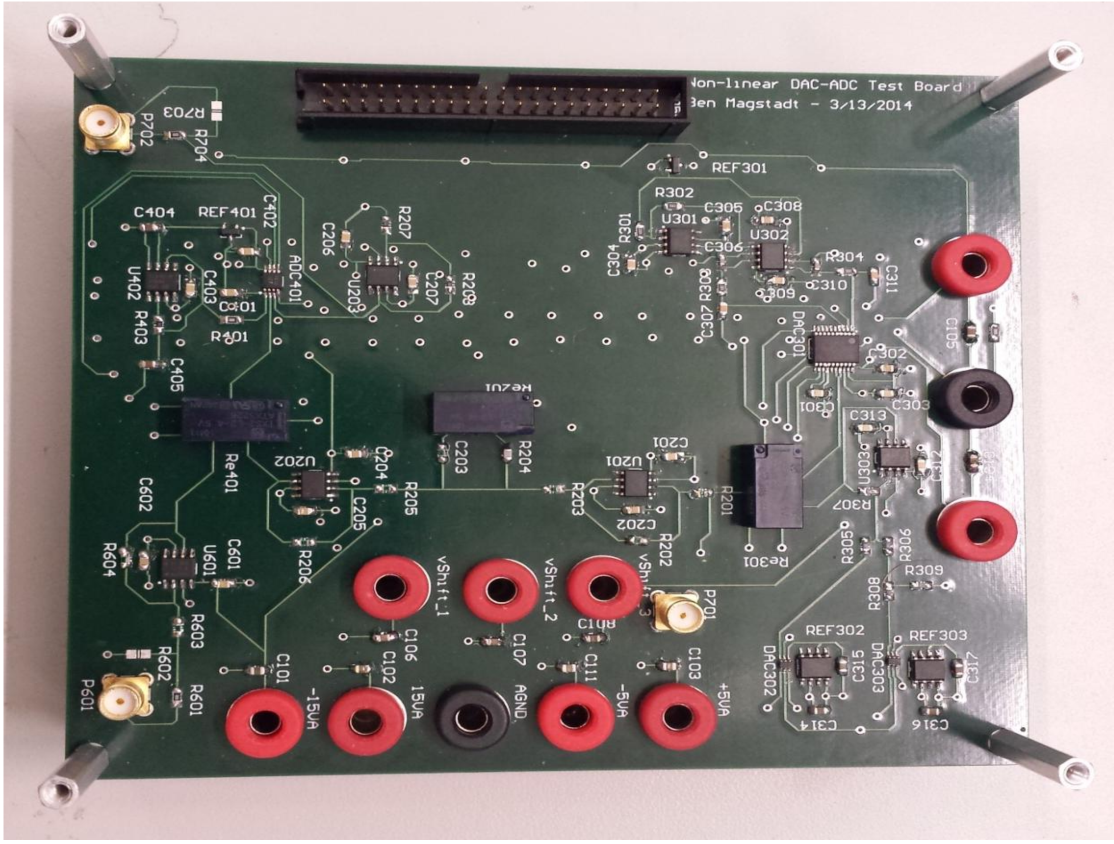


Figure 13: The designed test PCB.

The test board includes a 16-bit SAR ADC which will be the ADC under test. For the signal generator, a DAC will be used in order to generate an impure sine wave. This will then be filtered by two different filters. For this board two different low-pass filters were used. They each have a different DC gain and corner frequency, so the requirement of having the same amplitude response at the testing frequency can still be achieved. On this test board, the output of the DAC is first passed through an inverting operational amplifier configuration, so that it will be able to drive the filters easily. Then after the filters, another inverting operational amplifier configuration is used to drive the signal into the ADC to be digitized. The schematic for this portion of the board can be seen in Figure 14.

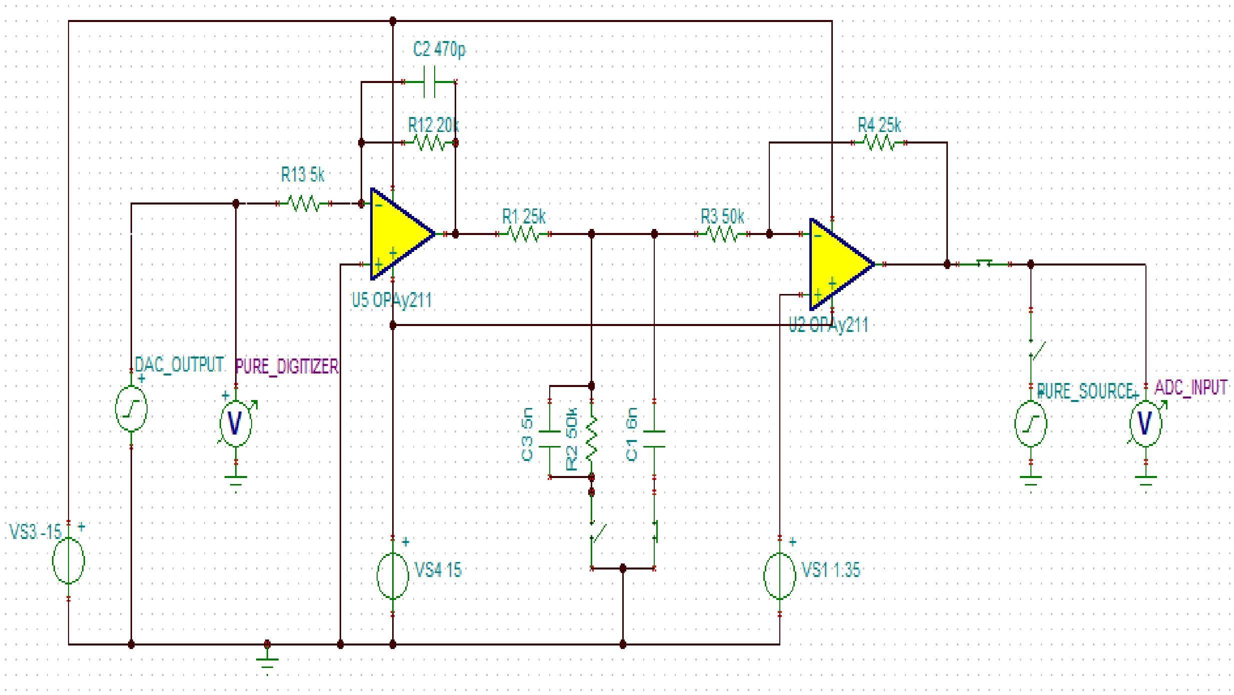


Figure 14: A partial schematic of the test PCB showing the output of the signal generator into the filter and to the input of the ADC.

There are several different parts of this schematic that need to be emphasized. The first is that the filter portion of the test board should not add any additional non-linearities. To accommodate this, high performance ceramic NP0 capacitors and high performance thin film resistors are used for the passive components in the filters. The switches that are used are also important as they could also generate other non-linearities. Therefore, high performance analog relays are used in order to switch between filters and testing signals. For verification purposes, it is also necessary to have an input to the board to supply a pure source to test the ADC in a traditional method.

Upon testing the board, accurate results were achieved. In Figure 15, the raw data output spectrums can be seen. As it shows, they have high harmonics as well as skirting due to the non-coherent sampling. Upon applying the proposed algorithm, the results in Figure 16 are able to be achieved. As it shows, the coherency problem is solved as well as the harmonic powers match up well. The final measurement results can be summarized in Table 5. The signal purity of the test source that was able to achieve these results had a THD of about -79dB.

It can be seen, however, that the noise floor is higher than desired. This is due to the test board voltage sources having too high of a noise level. The higher noise at the lower frequencies in the proposed final spectrum is due to the low-pass filters that are used to filter the two different signals. The larger spurs at the lower frequency are due to the 60Hz wall power supply that makes it through the power supplies. The high noise level of the board will be corrected in future revisions of the board to be able to get a lower noise floor and better results. To get accurate results with this board, a larger set of data needed to be analyzed in order to still be able to achieve accurate results.

Table 5: The measurements of the 16-bit ADC using the traditional method and proposed algorithm.

ADC Characterization		
Method	THD (dB)	SFDR (dB)
Traditional Method	-90.37	91.02
Proposed Algorithm	-90.68	91.27

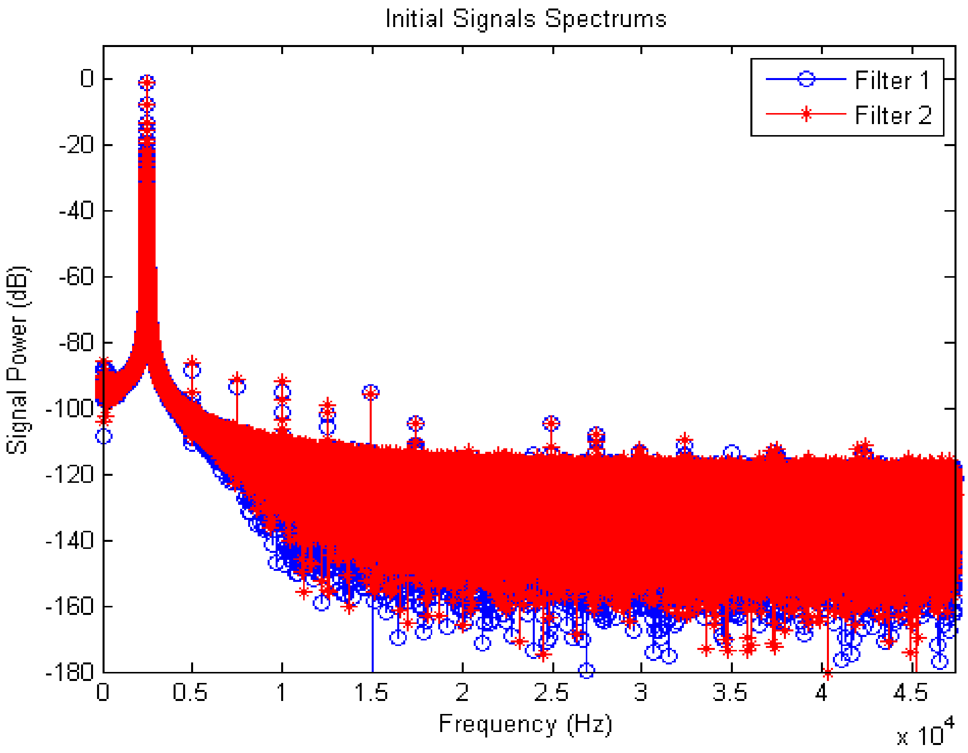


Figure 15: The two initial raw signals that were captured by the ADC.

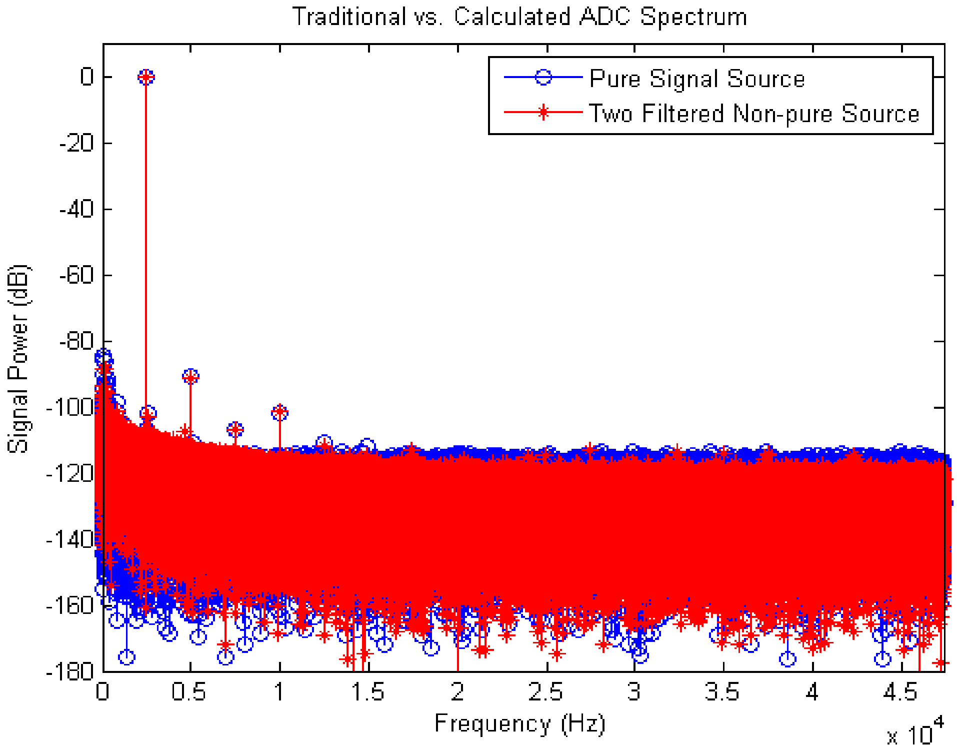


Figure 16: The spectrums found by testing the ADC using a pure source and calculated by the proposed method.

Conclusion

A new accurate spectral testing of an ADC while using a non-linear source and non-coherent sampling was proposed. This method is able to separate the non-linearities of the signal generator from the non-linearities of the ADC by the use of two different filters along with several steps of efficient computations on the output data. Not only can this algorithm accurately estimate the ADC spectral characteristics, but can also characterize the signal generator spectral characteristics as well. This algorithm was verified using different simulations. It was also tested to ensure that the algorithm was robust across different parameters of the algorithm: non-linear source level, noise, and non-coherency. Through these different simulations it was shown that a signal source that had a purity level of -50dB of THD was able to accurately test a 16-bit ADC with a THD of -94dB while being sampled non-coherently. Lastly, a PCB test board was designed and tested to further verify the algorithm's accuracy.

CHAPTER 3

ACCURATE SPECTRAL TESTING WITH NON-LINEAR SOURCE, NON-COHERENT SAMPLING, AND AMPLITUDE CLIPPED SIGNAL

Introduction

It can be very difficult to be able to get an input signal into an ADC very near to the full range without ever clipping. If any gain in the system is slightly off, then measurement errors could exist. If the signal is too small compared to the ADC input range, then possible non-linearities near the rails of the ADC input could be un-tested. If the signal is too large compared to the ADC input range, then there will be very large measurement errors in the output spectrum due to the clipping effect on the output spectrum if standard data processing algorithms are used.

Along with the problem of having a clipped signal is the same exact problems of a non-linear source and non-coherent sampling that were discussed in Chapter 2. Therefore, it is important to develop an algorithm that can take care of all three of these issues together. If all three of these can be combined, then it can be made certain that the dynamic spectral characteristics can be obtained for the whole ADC input range while significantly relaxing the requirements of the whole test. There has been some research in each of these individual three areas.

Some work has been previously worked on to reduce the difficult condition of the highly pure input test signal. In [4], an algorithm was described in order to be able to use filters on the input signal in order to get accurate results. Simulation results were shown to validate the method. A similar approach is shown in [5], where again filters are used on the input signal in order to achieve accurate results. Both of these methods only worked on reducing the input signal purity requirement for ADC test.

Achieving solutions for acquiring accurate spectral results without requiring coherent sampling has long been a source of research. There are many different strategies for attempting to resolve this problem. Windowing techniques have been widely used in order to achieve good results from non-coherently sampled data as seen in [6] and [7]. However, this can lead to inaccurate results depending on the resolution and type of windows being used on the specific output data. Four parameter sine wave fitting has also been used as seen in [8] and [9]. The time it takes to perform these methods can become a concern, however, when large data sets are used. Another method that has been proposed is the Fundamental Identification and Replacement Method (FIRE) presented in [10]. All of these methods are only for fixing the non-coherent sampling issue.

There has also been some research done into how to achieve accurate results when the input signal is amplitude clipped. One way is to use sine wave fitting as shown in [11]. However, this will not always work accurately for high resolution ADCs. Another method can be seen in [12] to reduce the additive spurs generated by the amplitude clipping. Both of these methods are just for testing while only allowing for clipping.

There is one solution that takes into account two of these problems, non-coherent sampling and amplitude clipping, and has been combined into an algorithm in [13]. However, this is just for reducing these two requirements and will not work properly when combined with a non-pure input signal.

It can then be stated that there is no current general solution that can be applied that will fix all three problems: non-pure input source, non-coherent sampling, and amplitude clipping of the input signal. Therefore, the next section will propose a new algorithm that will be able to take all of the conditions and still maintain accurate results. Simulations will show that accurate testing can be obtained with a non-linear source with a THD as bad as -50dB, while having non-coherent sampling, and signal amplitude clipping by as much as 1% when testing an ADC with a target THD of -94dB. Another benefit of this test is that the signal generator's non-linearities will also be obtained in the same process along with the ADC's non-linearities. This can thus have multiple benefits as multiple devices can be characterized with only one algorithm. Combining tests together will ultimately save time as well as resources.

Proposed Algorithm

The proposed algorithm will be able to work with a non-linear input signal that is clipped by the ADC and sampled non-coherently.

Algorithm Flow

Figure 17 shows how the signal flows through the system in order to be able to achieve accurate results using the proposed algorithm. First a non-linear source generates a signal that is going through two different filters. The filters will be discussed further in the next section. Then each of these signals is digitized by the ADC under test separately. After the two signals have been finished being sampled, the two digital signals will be passed into the non-linear, non-coherent, clipping algorithm.

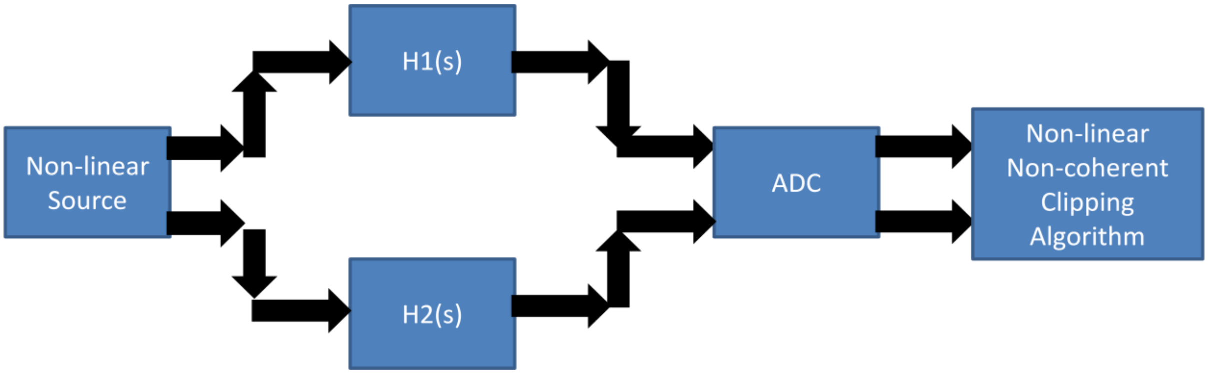


Figure 17: A flow diagram of how the signal progresses through the system.

Once the signals have been successfully captured, the algorithm can begin to calculate based on the two different signals as shown in Figure 18. As it shows, the second step will be to estimate the fundamental’s amplitude, frequency, and phase using sine wave fitting. However, this estimate will only be able to be applied on the portions of the data that are unclipped. Once the fundamental has been estimated, the two signals will be subtracted. By subtracting the signals, the non-linearities added by the ADC will be removed, leaving only a relationship between the harmonics of the two signal generator’s signals. The harmonics can then be estimated by using a multi-sine fitting algorithm. Since there is a known relationship between the remaining signal and the signal generator’s non-linearities, it is now possible to solve for the signal generator’s non-linearities. The signal generator’s non-linearities can then be removed from the original signal. All that is left in the signal’s residue at this point is the errors created by the ADC. From this residue, it will be possible to reconstruct the ADC spectral characteristics using a similar strategy as is used in [13].

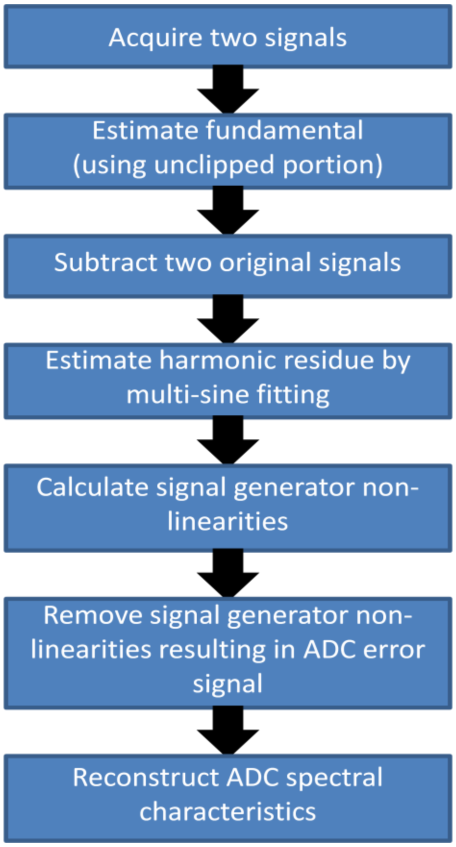

Acquire two signals
Estimate fundamental (using unclipped portion)
Subtract two original signals
Estimate harmonic residue by multi-sine fitting
Calculate signal generator non-linearities
Remove signal generator non-linearities resulting in ADC error signal
Reconstruct ADC spectral characteristics

Figure 18: A flow diagram of the non-linear, non-coherent, and clipping algorithm.

Filter Characterization/Adjustment

There are no specific criteria of what the two filters' architectures must be. Keeping it as simple as possible is advantageous as it is an easier design with a lower probability of extra errors coming into effect. One of the keys to this algorithm succeeding is to have both sets of signals coming out of the two different filters to have approximately the same amplitude. This will ensure that the non-linearities added to the output from the ADC are going to be the same on both signals. However, in designing the filters, passive components will be used. The values of these devices will most likely be varied from the specification by possibly over 5%. This can then make the amplitudes not matched very accurately.

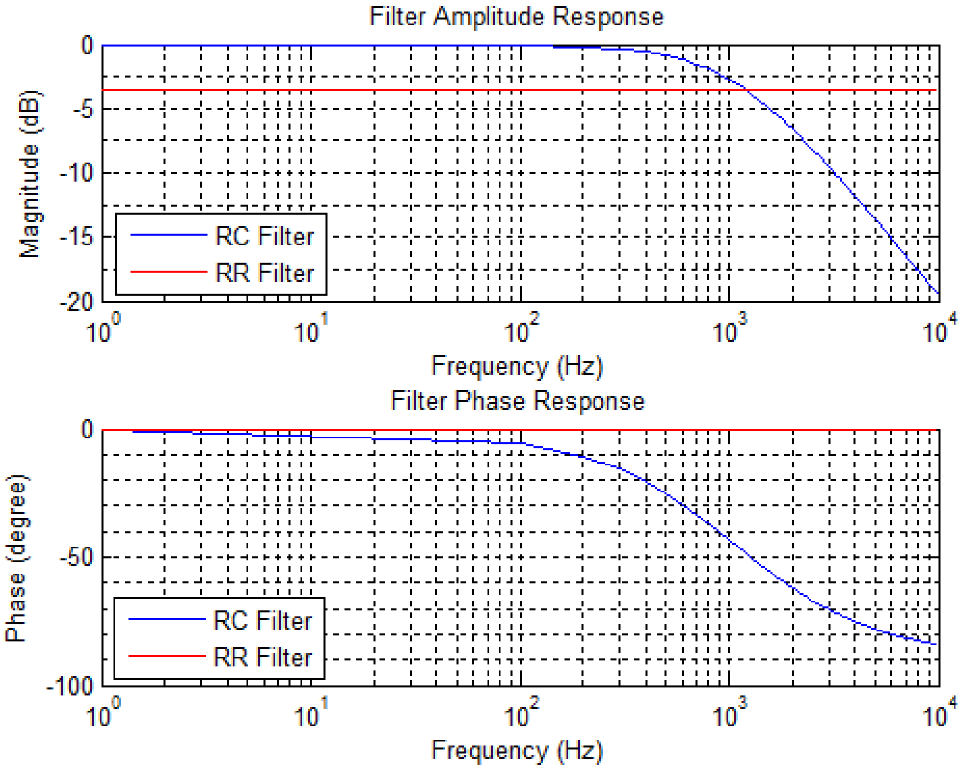


Figure 19: Plots showing the transfer function of two different simple filters.

To alleviate the error caused by this filter uncertainty, several different methods can be implemented. If simple passive components are being used then banks of devices can be tuned using digital components. An example would be having a binary weighted capacitor bank that can then be adjusted by using a digital signal. A different approach would be to change the frequency of the input signal to match the amplitude response of the two different filters. A simple example of filter choices can be seen in Figure 19 as a simple RC low-pass filter and a simple resistor attenuator filter. With these choices, the frequency could just be adjusted until the RC filter has the same amplitude response as the constant RR filter. The only problem with this measurement approach is that the signal that is being used to test the filter response may be amplitude clipped. This will lead to inaccurate measurements. To fix this, the amplitude will need to be adjusted or attenuated to be less than the full scale range of the ADC since for this step the full range of the ADC does not need to be tested.

The next problem, after tuning the different filters, is being able to know what the approximate transfer curve of the two different filters. If the filters are just one pole systems as in a standard low-pass RC filter, then a simple two-point characterization can be used. This would be done by measuring the filter at a DC input and at a higher frequency. By using the non-linear signal generator and ADC under test, an accurate enough characterization of the filters can be calculated in order to successfully run the algorithm. More complicated filters past a one-pole system can be used to still obtain accurate results with the algorithm; it just may take extra work in order to be able to achieve a relatively good estimate of the transfer curve of the filter.

It is also necessary for this algorithm for the two signals after the filter to have approximately the same phase as the signals enter the ADC. To do this, the phase of the initial signal going into the filter will need to be different. By, having an approximation of the filter, the amount of this phase difference will be known as the phase response difference of the two filters.

There is one last requirement that needs to be put on the two filters. This is that the components used in the filter cannot add any extra non-linearities into the signal. If the two filters add different non-linearities to the input signal that is at a level similar to the ADC’s non-linearities or worse, then the proposed algorithm will not be able to produce accurate results, as the algorithm cannot take this into account. The algorithm error introduced by the filter non-linearities can be quantitatively characterized but the topic is outside the scope of this thesis.

Signal Derivation

To begin the derivation of the algorithm it is helpful to understand what the signal is at every point in the system. The signal generator will need to generate two different signals that will have phase difference between them that will result in them having the same phase after the filtering happens and before the ADC. These signals will also have higher order harmonic components as well as a DC component. To achieve this, (3.1) and (3.2) will represent the signals out of the signal generator.

$$Signal_1 = |D_1| e^{j\omega_0 nT_s} e^{j(\phi_1 - \psi_1)} + \sum_{k=2}^{H} |D_k| e^{jk\omega_0 nT_s} e^{j(\phi_k - k*\psi_1)} + dc_0 \tag{3.1}$$

$$Signal_2 = |D_1| e^{j\omega_0 nT_s} e^{j(\phi_1 - \varphi_1)} + \sum_{k=2}^{H} |D_k| e^{jk\omega_0 nT_s} e^{j(\phi_k - k*\varphi_1)} + dc_0 \tag{3.2}$$

These two signals will then go through their respective filters. This will result in the same amplitude and phase of the fundamental frequency component. This can be seen in (3.3) and (3.4). The DC component of the signal may also change at this point due to a difference in the filter. If this difference in the DC component of the signal is large, one of the signals will need to be level shifted so that DC components of the signals are near the same level.

$$Signal_1 = \left|D_1 * H_1\left(j\omega_0\right)\right| e^{j\omega_0 nT_s} e^{j(\phi_1)} + \sum_{k=2}^{H} \left|D_k * H_1\left(jk\omega_0\right)\right| e^{jk\omega_0 nT_s} e^{j(\phi_k + \psi_k - k*\psi_1)} + dc_1 \tag{3.3}$$

$$Signal_2 = \left|D_1 * H_2\left(j\omega_0\right)\right| e^{j\omega_0 nT_s} e^{j(\phi_1)} + \sum_{k=2}^{H} \left|D_k * H_2\left(jk\omega_0\right)\right| e^{jk\omega_0 nT_s} e^{j(\phi_k + \varphi_k - k*\varphi_1)} + dc_1 \tag{3.4}$$

These two signals will then go into ADC and an error signal will be added for each sample to both signals as seen in (3.5) and (3.6). These error signals E_n, from the ADC, will be the same for both signals. This is because the error that will be introduced from the ADC will be dependent on the input fundamental signal. Since the fundamental signal entering the

ADC was the same for both signals, the error will be the same. The small difference in the signals that is due to the harmonics will be small enough to prove negligible. This will be true until the signal generator's harmonics reach too high of a level. At this point, this algorithm will no longer hold true. The harmonics will not have an effect until they become much more comparable to the fundamental amplitude. This will be shown in the later simulation results.

At this point the signal will clip to the range of the ADC. Therefore, the signal and its harmonics in this signal only holds true for the indices of the signal where it is not clipped. These are the indices where the rest of the algorithm will operate. In all of the following equations the time samples of the signal that corresponds to where the signal is clipped will be excluded so that the relationships still hold true.

$$Signal_1 = \left|D_1 * H_1\left(j\omega_0\right)\right| e^{j\omega_0 n T_s} e^{j(\phi_1)} + \sum_{k=2}^{H} \left|D_k * H_1\left(jk\omega_0\right)\right| e^{jk\omega_0 n T_s} e^{j(\phi_k + \psi_k - k*\psi_1)} + dc_1 + E_n \qquad (3.5)$$

$$Signal_2 = \left|D_1 * H_2\left(j\omega_0\right)\right| e^{j\omega_0 n T_s} e^{j(\phi_1)} + \sum_{k=2}^{H} \left|D_k * H_2\left(jk\omega_0\right)\right| e^{jk\omega_0 n T_s} e^{j(\phi_k + \varphi_k - k*\varphi_1)} + dc_1 + E_n \qquad (3.6)$$

The two outputs data sets that are actually collected by the ADC can be represented in the non-clipped region by (3.7) and (3.8). The clipped regions will continue to be ignored as there is no useful data in these regions.

$$Output_1 = \left|M_1\right| e^{j\omega_0 n T_s} e^{j(\alpha_1)} + \sum_{k=2}^{\mathrm{H}} \left|M_k\right| e^{jk\omega_0 n T_s} e^{j\alpha_k} + dc + E_n \qquad (3.7)$$

$$Output_2 = \left|N_1\right| e^{j\omega_0 n T_s} e^{j(\beta_1)} + \sum_{k=2}^{\mathrm{H}} \left|N_k\right| e^{jk\omega_0 n T_s} e^{j\beta_k} + dc + E_n \qquad (3.8)$$

Fundamental Identification

Once the two sets of data have been successfully sampled by the ADC, the fundamental of the signal needs to be identified. One of the signals will be chosen. For this signal, the non-clipped sections of the output will need to be found. The indices of these non-clipped portions of the output will be labeled as NCP. The size of the non-clipped portion of the code will be given as M while the total number of sampled points is given as m. As a first approximation, the output of the non-clipped portions of the signal will then be modeled as only a fundamental sine wave as in (3.9).

$$A_1 * \cos\left(2 * pi * F_{samp} * \frac{J_{int} + \delta}{m} * t\left(NCP\right) + \phi_1\right) + dc \tag{3.9}$$

There will need to be an iterative four-parameter sine-wave fit in order to be able to solve all of the parameters of the equation due to the presences of large higher order harmonics and noise. This will be done by using the Gauss-Newton algorithm. To begin this process, there will need to be initial rough estimates of all of the parameters. These can be found by the following equations that are originally derived in [13] and are now shown in (3.10-3.15). These estimates can be calculated quickly and easily from an FFT output of the ADC output data.

$$X = fft\left(Output\right) \tag{3.10}$$

$$J_{int} = \arg max_{1 \le k \le M/2} \left|X_k\right| \tag{3.11}$$

$$A_{10} = \frac{ADC\_range}{\cos\left(\frac{\left(topHits - 1\right)}{m * \pi}\right) + \cos\left(\frac{\left(bottomHits - 1\right)}{m * \pi}\right)} \tag{3.12}$$

$$dc_0 = \frac{\left(A_{10}\left(\cos\left(\frac{\left(bottomHits - 1\right)}{m * \pi}\right) - \cos\left(\frac{\left(topHits - 1\right)}{m * \pi}\right)\right)\right)}{2} \tag{3.13}$$

$$\delta_{10} = \frac{M}{2\pi} imag\left(\ln\left(\frac{\frac{X_{J_{int}}}{X_{J_{int}+1}} - \frac{X_{J_{int}}}{X_{J_{int}-1}}}{\frac{X_{J_{int}}}{X_{J_{int}+1}} - \frac{X_{J_{int}}}{X_{J_{int}-1}} + e^{\frac{j2\pi}{m}} - e^{\frac{-j2\pi}{m}}} \right) \right) \tag{3.14}$$

$$\phi_{10} = -imag\left(\ln\left(\frac{2mX_{J_{int}}}{A_0} \frac{1-e^{\frac{j2\pi\delta_{10}}{m}}}{1-e^{j2\pi\delta_{10}}} \right) \right) \tag{3.15}$$

Once all of the initial rough estimates of the parameters have been found, it will be possible to begin the approximation iterations to obtain better estimates of all of the parameters. To begin the residue of the estimate can be expressed by (3.16).

$$r = Output(NCP) - dc - A_1 * \cos\left(2 * pi * F_{samp} * \frac{J_{int} + \delta_1}{m} * t(NCP) + \phi_1 \right) \tag{3.16}$$

The iterative algorithm for the four-parameter least-square sine-wave fit can be given by (3.17-3.22). The variable k represents the number of iterations that will be performed. Then all of the desired variables of P will iteratively change by calculating s and then adding them to the previous value of P. The residue can then be further updated and another iteration of the algorithm can be performed. The variable s can be directly added to the previous variable P without worrying about oscillation or divergence since good initial estimates of the variables were previously achieved. This will lead to faster convergence. If this was not the case an attenuation factor would possibly be needed in order to ensure achieving a stable solution.

$$P(k) = \begin{bmatrix} A_1 \\ \delta_1 \\ \phi_1 \\ dc \end{bmatrix} \tag{3.17}$$

$$D=\begin{bmatrix} \frac{\partial r[1]}{\partial A_1} & \frac{\partial r[1]}{\partial \delta_1} & \frac{\partial r[1]}{\partial \phi_1} & \frac{\partial r[1]}{\partial dc} \\ . & . & . & . \\ . & . & . & . \\ \frac{\partial r[M]}{\partial A_1} & \frac{\partial r[M]}{\partial \delta_1} & \frac{\partial r[M]}{\partial \phi_1} & \frac{\partial r[M]}{\partial dc} \end{bmatrix} \tag{3.18}$$

$$x=\begin{bmatrix} r[1] \\ r[2] \\ . \\ . \\ r[M] \end{bmatrix} \tag{3.19}$$

$$s=\begin{bmatrix} \Delta A_1 \\ \Delta \delta_1 \\ \Delta \phi_1 \\ \Delta dc \end{bmatrix} \tag{3.20}$$

$$s=\left(D^T D\right)^{-1} D^T * x \tag{3.21}$$

$$P(k+1)=P(k)+s \tag{3.22}$$

By performing the calculations of s, better estimates of the four parameters can be found by going through multiple iterations of the algorithm.

Multi-Sine Fitting

After the fundamental component of the signal is known, then it is possible to know the accurate signal frequency and thus the harmonic frequencies. As previously discussed, it will be of benefit to do an estimation of the harmonic components of the subtraction of the two signals, as the ADC non-linearities will be removed only leaving the signal generator's non-linearities present. Therefore, a multi-sine wave fitting will need to be done for the non-clipped portions of the subtraction of the two outputs. This residue signal can be given as the signal calculated as in (3.23).

$$r = \begin{bmatrix} r[1] \\ r[2] \\ . \\ . \\ r[M] \end{bmatrix} = Output_2(NCP) - Output_1(NCP) \tag{3.23}$$

Note that the NCP is the portion when neither output is clipped. The function for the remainder of the signal will be a DC component added to the sum of a fundamental and harmonics frequencies in the form of the summation of cosines and sines. This is done in order to remove the non-linear function that would be given if the components were not divided into the sine and cosine components. This function for the non-clipped regions can be given by (3.24).

$$g = dc + \sum_{k=1}^{H} P_k * \cos\left(2 * pi * k * f_{sig} * t(NCP)\right) + Q_k * \sin\left(2 * pi * k * f_{sig} * t(NCP)\right) \tag{3.24}$$

The derivative matrix and variable matrix that will be used in the least-squares estimation can be given by (3.25) and (3.26) respectively.

$$D_0 = \begin{bmatrix} \frac{\partial g[1]}{\partial P_1} & \frac{\partial g[1]}{\partial Q_1} & \cdots & \frac{\partial g[1]}{\partial P_H} & \frac{\partial g[1]}{\partial Q_H} & \frac{\partial g[1]}{\partial dc} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g[M]}{\partial P_1} & \frac{\partial g[M]}{\partial Q_1} & \cdots & \frac{\partial g[M]}{\partial P_H} & \frac{\partial g[M]}{\partial Q_H} & \frac{\partial g[M]}{\partial dc} \end{bmatrix} \tag{3.25}$$

$$s_0 = \begin{bmatrix} P_1 \\ Q_1 \\ P_2 \\ Q_2 \\ . \\ . \\ P_H \\ Q_H \\ dc \end{bmatrix} \tag{3.26}$$

In this case the least square estimation that is desired has some desired qualities that did not exist in the previous section. This is that the equation being estimated is already a linear equation in terms of the variables that are being solved for. This makes the least squares problem much easier to handle. The first thing is that the derivative matrix now becomes much simpler as the derivative in respect to the variable just becomes the coefficient of the respective variable. Also, there won't have to be any iterations in order to converge to the correct solution of all variables. Therefore, only one simple calculation has to be done to solve for the fundamental and harmonic components of the residue signal given by (3.27).

$$s_0 = \left(D_0^T D_0\right)^{-1} \left(D_0^T r\right) \tag{3.27}$$

Algorithm - *D_k*

To solve for the non-linear terms of the signal generator, the equations that were previously derived of the output signal in (3.5-3.8) can be used. When (3.6) and (3.8) are subtracted from (3.5) and (3.7) respectively and set equivalent to each other it will form (3.28). The two outputs of the ADC are now known due to the fundamental identification and multi-sine wave fit methods just performed.

$$\begin{aligned}&\left|M_k\right|e^{jk\omega_0 nT_s}e^{j\alpha_k}-\left|N_k\right|e^{jk\omega_0 nT_s}e^{j\beta_1}=\\&\left|D_k*H_1\left(jk\omega_0\right)\right|e^{jk\omega_0 nT_s}e^{j(\phi_k+\psi_{k-k*\psi_1})}-\left|D_k*H_2\left(jk\omega_0\right)\right|e^{jk\omega_0 nT_s}e^{j(\phi_k+\varphi_k-k*\varphi_1)}\end{aligned} \tag{3.28}$$

The equations can then be reordered to solve for the signal generator's non-linearity terms as shown in (3.29). It is then a simple equation in order to find the solution of the signal generator's components.

$$\left|D_k\right|e^{j(\phi_k)}=\frac{\left|M_k\right|e^{j\alpha_k}-\left|N_k\right|e^{j\beta_1}}{\left|H_1\left(jk\omega_0\right)\right|e^{j\left(\psi_{k-k*\psi_1}\right)}-\left|H_2\left(jk\omega_0\right)\right|*e^{j(\varphi_k-k*\varphi_1)}} \tag{3.29}$$

Algorithm - C_k

The last step is to now solve for the ADC's non-linear terms. This can be done by first finding the error values of the ADC for the input. By re-examining the one of the outputs of the ADC in the non-clipped region, it is found that it is possible to now solve for the error values alone as shown in (3.32).

$$Signal_1=\left|D_1*H_1\left(j\omega_0\right)\right|e^{j\omega_0 nT_s}e^{j(\phi_1)}+\sum_{k=2}^{H}\left|D_k*H_1\left(jk\omega_0\right)\right|e^{jk\omega_0 nT_s}e^{j(\phi_k+\psi_{k-k*\psi_1})}+E_n \tag{3.30}$$

$$Output_1=\left|M_1\right|e^{j\omega_0 nT_s}e^{j(\alpha_1)}+\sum_{k=2}^{\mathrm{H}}\left|M_k\right|e^{jk\omega_0 nT_s}e^{j\alpha_k}+E_n \tag{3.31}$$

$$\begin{aligned}&E_n=Output_1\left(NCP\right)-\\&\left[\left|D_1*H_1\left(j\omega_0\right)\right|e^{j\omega_0 n(NCP)T_s}e^{j(\phi_1)}+\sum_{k=2}^{H}\left|D_k*H_1\left(jk\omega_0\right)\right|e^{jk\omega_0 n(NCP)T_s}e^{j(\phi_k-k*\psi_1)}\right]\end{aligned} \tag{3.32}$$

This error signal, E, now contains the error information of the ADC. However, it only is useful in the area where the original signal is unclipped. During the clipped regions, this error signal contains no information.

The original signal should now be divided up into the portions when it is rising and falling. The time when the data is clipped can be thrown away. Then the error signal at the same index should be kept with the signal for the rising and falling indexed signal. An example of this for a portion of the rising signal can be seen in the Figure 20. As it shows, the signal is only shown when it is rising. The error signal that is associated with that particular index of the signal is then plotted below it.

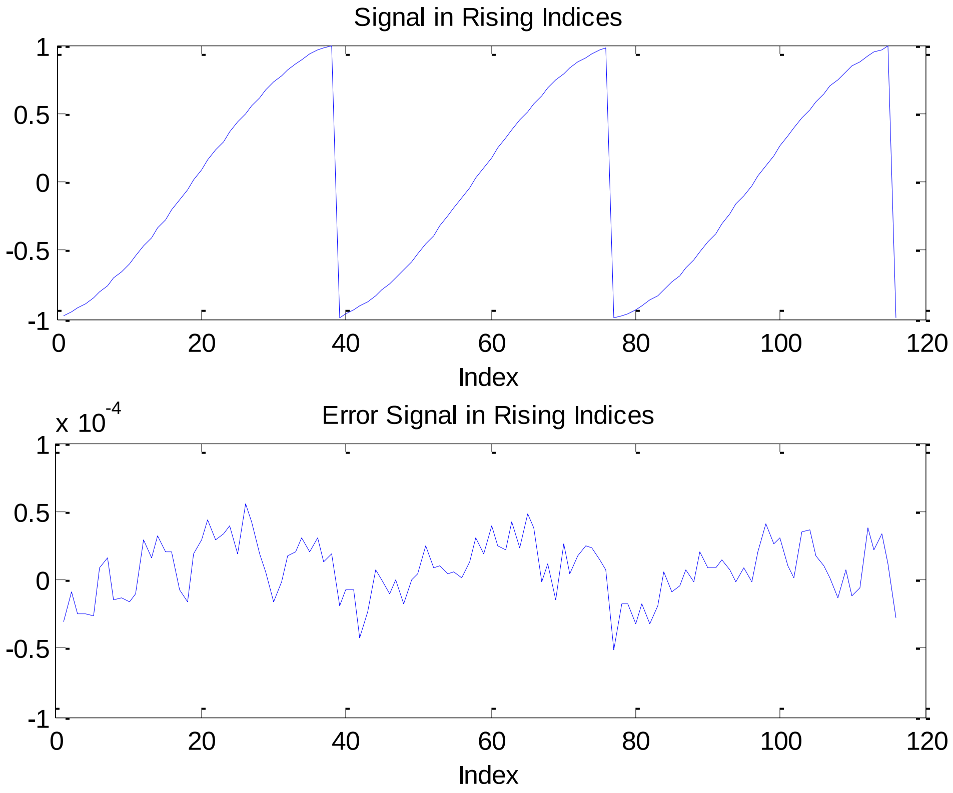


Figure 20: A plot linking the input signal when it is rising to the error signal at the same location.

After this data has been generated, the data needs to be folded upon itself, so that instead of repeating the same cycle, there is now only one rise. This means that all of the data that contains the signal when it is rising, meaning it has a positive slope, needs to be gathered into one data set and then an overall sort of the data will be performed based on the voltage value. All of the error signals are maintained with the original signal voltage index even as it is folded over onto itself. A demonstration of this can be seen in Figure 21. The same technique will be used to generate a similar set of data for the falling input signal data indices.

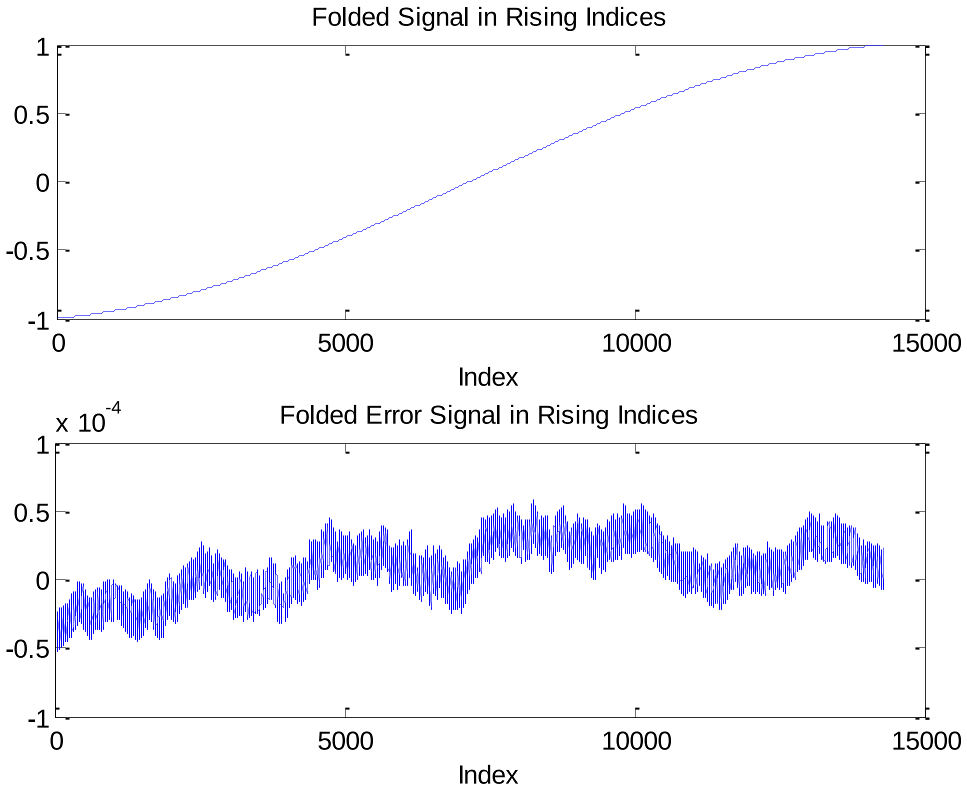


Figure 21: A showing of all of the rising signal points folded onto themselves. The error signal for the particular point is maintained.

After both sets of folded data have been achieved, a new coherent fundamental signal needs to be created that is near the full-range amplitude of the ADC. This signal can be expressed as the signal in (3.33).

$$v_{new} = 0.999 * \cos\left(2\pi * F_{samp} * \frac{J_{int}}{m} * t + \theta\right) \tag{3.33}$$

Now, for every point in the new non-clipped signal an error signal needs to be added to it. This can be done by first determining if the signal is rising or falling at the point under question. This is the same as determining if the slope of the signal at this point is positive or negative respectively. Then a lookup will be done on the appropriate rising or falling data set to find the values that are closest to the new value under question. When the closest points above and below the data point in question are found, a linear interpolation of the error signals that correspond to those two points will be done. A formula to calculate this linear interpolation using the two closest data points (x_0 , y_0) and (x_1 , y_1) can be seen in (3.34).

$$y = y_0 + \left(y_1 - y_0\right)\frac{x - x_0}{x_1 - x_0} \tag{3.34}$$

The result of this interpolation will then be added to this data point on the new data set. This will be repeated for every point in the new signal. Once this is completed, the new signal, v_{new}, should contain a fundamental as well as the equivalent error signal of the ADC from the interpolation which will contain the noise and harmonic information of the ADC. Since the new signal, v_{new}, was created as a coherently sampled and non-clipped signal, then a simple FFT can be taken of the new data and the spectrum will be under the traditional test conditions of Chapter 1. Therefore the simple equations to derive the dynamic spectral characteristics can now be applied.

Simulation Results

Simulations were performed in order to verify the functionality and accuracy of the proposed method. A 16-bit ADC model was randomly generated to be under test. The non-linearities of this ADC came from the randomly generated errors in the transition voltages of the ADC. To perform a good comparison, the ADC model was tested first with a pure, coherently sampled signal that was near the full range of the ADC but never clipped. The output of the ADC could then be tested using the measurement calculations from Chapter 1. This measurement serves as the traditional testing benchmark for comparison. Then a randomly generated non-linear sinusoid signal is generated to go through both of the filters whose outputs are sampled non-coherently by the ADC under test. At the input of the ADC, the amplitude of this signal is 1% over-range causing the amplitude clipping. All of the simulations were done with 2^{15} data points and a half LSB of additive noise. Then when running this signal through the proposed algorithm the ADC characteristics can be estimated as seen in Table 6. As it shows, the characteristics calculated with the proposed method are accurate measurements in comparison to the traditional testing benchmark.

Table 6: The measurements of the 16-bit ADC using the traditional method and proposed algorithm with 1% clipping.

ADC Characterization		
Method	THD (dB)	SFDR (dB)
Pure/coherent/non-clipped	-93.89	96.79
Proposed Algorithm	-93.86	96.83

The algorithm can also provide the measurement characteristics for the signal generator. To be able to provide a good comparison the signal out of the signal generator was digitized with a perfect ADC with very high resolution in order to be able to get an ideal value for the measurements of the signal generator’s spectrum. The comparison of these results can be seen in Table 7. As it shows, the measurement of the signal generator’s spectral characteristics are also accurately estimated.

Table 7: The measurements of the signal generator using the traditional method and proposed algorithm.

Signal Generator Characterization		
Method	THD (dB)	SFDR (dB)
Pure/coherent/non-clipped	-51.04	59.22
Proposed Algorithm	-51.19	59.31

The same simulation was then repeated, except the signal amplitude was increased so now the level of clipping was increased to 5%. The results of this experiment can be seen in Table 8. As it shows, with this higher level of clipping the results with the proposed algorithm are no longer as accurate. This shows that this algorithm works much better when the levels of clipping are at a smaller more reasonable level.

Table 8: The measurements of the 16-bit ADC using the traditional method and proposed algorithm with 5% clipping.

ADC Characterization		
Method	THD (dB)	SFDR (dB)
Pure/coherent/non-clipped	-93.89	96.79
Proposed Algorithm	-90.02	94.03

Another experiment was done to see the effects of the input source's non-linearity level. This time there was a 1% amplitude clipping but now the input source had a worse purity level corresponding to a THD of -40dB. When using this input source, the results that can be seen in Table 9 were achieved. As it shows, with a source this non-linear a larger error was seen by using the proposed method.

Table 9: The measurements of the 16-bit ADC using the traditional method and proposed algorithm with a source with a THD of -40dB.

ADC Characterization		
Method	THD (dB)	SFDR (dB)
Pure/coherent/non-clipped	-93.89	96.79
Proposed Algorithm	-91.37	93.48

Conclusion

In this chapter, an ADC spectral testing algorithm was developed that can be used in order to achieve accurate ADC spectral testing while drastically reducing the requirements on input signal purity, allowing for amplitude clipping, and removing the requirement of coherent sampling. This is achieved by running an input signal through two different filters and then processing the two different output data sets through the proposed algorithm. Once the data is processed, then both the ADC's and signal generator's spectral performances are accurately computed. This results in an accurate test that has significantly less requirements than traditional data converter testing. Having fewer requirements for the test allows for a simpler test setup without the requirement for high precision instrumentation. This will cause the price of testing the part to decrease while also allowing for easier BIST solutions to be possible. Simulation results show the success of the algorithm by testing a 16-bit ADC with a source that has a THD of -51dB, gets amplitude clipped by 1%, and is not sampled coherently. Both the ADC and signal generator were characterized accurately.

CHAPTER 4
PRE-DISTORTION CODE GENERATION FOR ACCURATE DAC-ADC SPECTRAL TESTING

Introduction

There are many cases when there are both a DAC and ADC present on some integrated systems. When there are, it could be of great advantage to be able to use both of these together to be able to test the characteristics of the ADC. In most cases, the DAC and ADC are going to be of similar performance, so it is impossible to use the traditional methods to accurately test the desired parameters of the ADC using the DAC as a signal generator. This is because the DAC's non-linearities will be at the same relative level as the ADC's, so the resulting spectrum of the ADC output will include the extra distortions coming from the DAC as well as those generated by the ADC under test. However, by adding in some extra data processing, it is possible to accurately test the ADC using the DAC as an excitation source when pre-distortion codes are applied at the DAC input. The key to this process is to know how the pre-distortion codes should be generated and applied.

In a new on-chip ADC self-test algorithm currently under industry validation, a non-linear signal source can be used to accurately determine the INL/DNL characteristics of the ADC under test. This is done by applying two non-linear ramp signals to the input of the ADC that are shifted and/or attenuated from each other by an unknown but constant small shift or an unknown but constant small attenuation. Two sets of ADC output data will be fed to the SEIR (stimulus error identification and removal) algorithm which accurately identifies and removes errors present in the non-linear ramp signal. To facilitate this, the on-chip available DAC can be used to generate these two non-linear ramp signals. With these two ramp signals and the SEIR algorithm, the full code INL/DNL measurements of the ADC can be obtained accurately.

Once the ADC is accurately tested, the ADC output code and the accurate INL/DNL measurement for that code can be used to accurately determine the voltage signal present at the ADC input. Since this ADC input voltage signal is actually the DAC output voltage signal, the accurate knowledge of this voltage and the knowledge of the digital code sent to the input of the DAC allows for calculation of the DAC's INL/DNL at that code. Repeating this for all DAC input codes leads to complete characterization of the DAC static transfer function. Upon knowing the DAC's INL/DNL measurements, pre-distortion correction codes can be generated and applied to the DAC input so that DAC output sine wave signal can have significantly better distortion performance. This high purity sine wave, used as input to the ADC, will help to significantly improve the accuracy of the dynamic spectral test results for the ADC under test.

There have been many different research studies where better spectral performance has been achieved by applying pre-distortion codes into a DAC. In [14] pre-distortion codes were applied to an arbitrary waveform generator, including an internal DAC, in order to measure an ADC more accurately. However, the goal of this method is to try to move un-wanted spurs and push them to a higher frequency to be filtered out. This method requires more rigorous calculations as well as the addition of filters at the output of the DAC. Another study in [15] showed how pre-distortion codes were applied to a DAC in order to help correct spectral impurities in a radar system. This method, however, is using several filters as well as trying to correct for other spectral impurities in the entire radar system and not taking into account the spectral impurities of the DAC and ADC that are being used.

Although using pre-distortion has been used in many different applications in the past, no previous studies are available for embedded applications where no accurate instruments are available to measure the DAC distortions and no accurate signal sources are available to test a measurement device either. The proposed algorithm in this chapter is the first attempt at using a non-linear DAC to generate a high purity sine wave signal for accurate ADC spectral testing, by using pre-distortion codes obtained without requiring any accurate measurement devices for generating such codes. Furthermore, the proposed algorithm will make a very simple testing system that will not require extra filtering of higher frequency components in order to achieve accurate ADC spectral test results. It will also present several different options for implementing the proposed method based on trade-offs between the accuracy requirement and the amount of hardware and timing overhead that can be sacrificed in the test setup. Simulation results using a 12-bit DAC and a 12-bit ADC will be presented to demonstrate the algorithm's functionality, accuracy, and robustness. The 12-bit DAC used is a combination of an R-String and an R-2R DAC. The 12-bit ADC is a segmented SAR ADC.

Proposed Algorithm

This algorithm will take advantage of the knowledge of the measured INL/DNL of the ADC to be able to calculate the INL/DNL of the DAC. Then pre-distortion codes can be generated that will allow for a more accurate dynamic spectral test of the ADC. The derivation of this algorithm will not go into detail about how the INL/DNL of the ADC was calculated as that is a separate algorithm under development. Only a brief summary will be given and it will just assume that this information has already been calculated.

ADC’s INL/DNL Acquisition Summary

For the algorithm to work to calculate the ADC’s full code INL/DNL, two different ramp signals were applied to an ADC. The ramp signals were both generated identically by the same DAC to produce the voltage V_{out}. On the first ramp signal the output voltage of the DAC had a gain factor and a voltage level shift applied to it. On the second ramp signal, only a gain factor was applied. These gain factors are small but identical. The two ramp voltages that are sampled by the ADC can be represented by (4.1) and (4.2) where k is the DAC input code.

$$V_1\ (k) = V_{out}(k)*A + DC \tag{4.1}$$

$$V_2\ (k) = V_{out}(k)*A \tag{4.2}$$

These two voltages will serve as the two different inputs to the ADC. From the corresponding two ADC output codes, the ADC INL/DNL were calculated. The final estimation that can be found from using this other algorithm can be seen in Figure 22. As it shows, the error in estimating the INL of the ADC is now small. With only a small error in this estimation, it will be possible to then get a good enough calculation of the DAC’s INL/DNL.

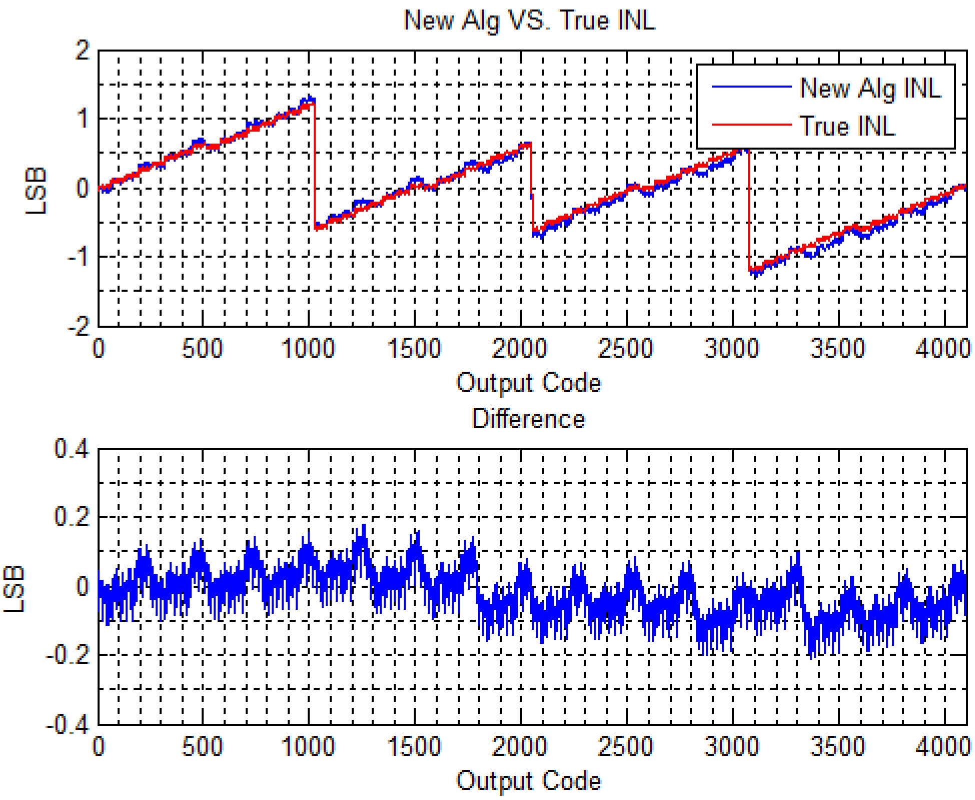


Figure 22: The measure of the ADC's INL and true INL is plotted in the top. In the bottom the error in the estimation of the ADC's INL is plotted.

New Algorithm Derivation

Once the ADC's INL/DNL are known it is then possible to begin calculating the DAC's INL/DNL. From the previous relationship in (4.1) and (4.2), (4.3–4.5) can be put together that relates the input code of the DAC to the output code of the ADC. This is generated as the output voltage of the DAC can be considered to consist of two components. First, the ideal component of this output voltage is due to the input code that is passed to the input of the DAC. Then there will be some error due to component mismatch as well as other possible errors. This will then contribute an error to the output voltage at each input code that will correspond to the INL of each code. These two components will consist of the output voltage that will then have a gain and/or shift applied to it. In these equations, k represents the number of the measurement made. However, since a ramp signal is being generated k will also represent DAC code.

$$m = 2^N \tag{4.3}$$

$$\begin{aligned}&\left(\text{Code}_{\text{DAC}}\left[\text{k}\right] + \text{INL}_{\text{DAC}}\left[\text{Code}_{\text{DAC}}\left[\text{k}\right]\right]\right) * \text{gain} + \text{shift} = \\ &\text{Code1}_{\text{ADC}}\left(\text{k}\right) + \text{INL}_{\text{ADC}}\left[\text{Code1}_{\text{ADC}}\left[\text{k}\right]\right] \qquad \text{for k} = 1:\text{m}\end{aligned} \tag{4.4}$$

$$\begin{aligned}&\left(\text{Code}_{\text{DAC}}\left[\text{k}\right] + \text{INL}_{\text{DAC}}\left[\text{Code}_{\text{DAC}}\left[\text{k}\right]\right]\right) * \text{gain} = \\ &\text{Code2}_{\text{ADC}}\left(\text{k}\right) + \text{INL}_{\text{ADC}}\left[\text{Code2}_{\text{ADC}}\left[\text{k}\right]\right] \qquad \text{for k} = 1:\text{m}\end{aligned} \tag{4.5}$$

From these equations, the DAC code, ADC code, and ADC INL are already known. The gain and the shift that were applied to the ramps can be easily calculated as shown by (4.6) and (4.7).

$$\text{shift} = \text{mean}\left(\text{Code1}_{\text{ADC}} - \text{Code2}_{\text{ADC}}\right) \tag{4.6}$$

$$\text{gain} = \left(\text{Code1}_{\text{ADC}}\left[\text{m}\right] - \text{Code1}_{\text{ADC}}\left[1\right]\right) / \left(\text{m}\right) \tag{4.7}$$

Once the gain and shift are calculated either (4.4) or (4.5) can be used to easily solve for the INL of the DAC for each DAC code. This will lead to two supposedly the same solutions as in (4.8) and (4.9). But in reality, due to noise, the two solutions might be slightly different. To improve precision, the calculated INL of the DAC can be found by averaging the two different INL calculations. An example of the INL estimation can be seen in Figure 23.

$$\text{INL}_{\text{DAC}} = \left(\text{Code1}_{\text{ADC}} + \text{INL}_{\text{ADC}}\left(\text{Code1}_{\text{ADC}}\right) - \text{shift}\right) / \text{ gain} - \left(0:1:2 \wedge \text{N} - 1\right) \tag{4.8}$$

$$\text{INL}_{\text{DAC}} = \left(\text{Code2}_{\text{ADC}} + \text{INL}_{\text{ADC}}\left(\text{Code2}_{\text{ADC}}\right)\right) / \text{ gain} - \left(0:1:2 \wedge \text{N} - 1\right) \tag{4.9}$$

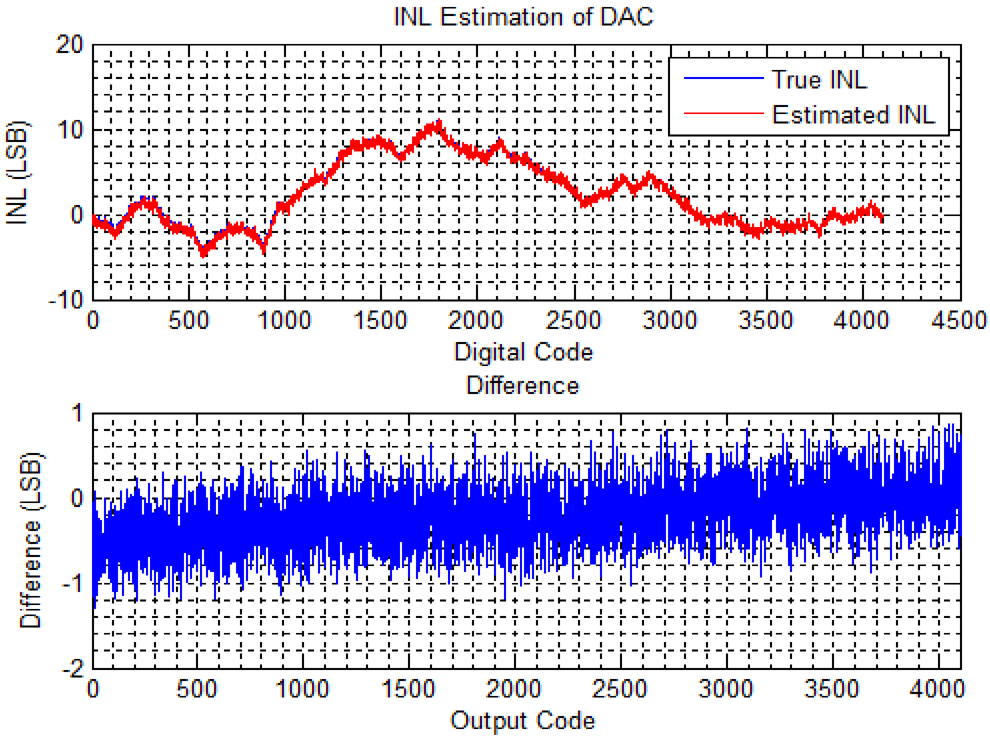


Figure 23: The calculation of the DAC's INL and true INL is plotted in the top. In the bottom the error in the estimation of the DAC's INL is plotted.

Once the INL of the DAC is completed the INL can be added to what the ideal output levels of the DAC were originally thought to be as shown in (4.10). These updated output levels of the DAC will now allow for a better selection of the codes to pass to the DAC in order to get the value closest to the desired output value.

$$\text{OutputLevels}_{\text{calc}} = \text{OutputLevels}_{\text{ideal}} + \text{INL}_{\text{DAC}} \tag{4.10}$$

The knowledge of these more accurate output levels is not enough knowledge by itself. It is also necessary to somehow be able to apply this extra knowledge in order to improve the purity of the DAC output. Therefore, different implementation methods are needed in order to accomplish this.

Implementation Methods

Now that the updated output levels are known, it is possible to select better codes to send to the DAC as pre-distortion codes in order to get better performance. There are many different methods that could be used in order to apply these pre-distortion codes. The general strategy that should be employed when calculating the pre-distortion codes is to minimize the difference between the calculated output levels of the DAC and the desired analog value. When this minimization process is optimized, then the best performance can be achieved. In order to achieve this minimization process there will be some trade-offs between the increase in performance and the extra hardware and timing overhead to implement the method. After some research and experimentation, there were two main strategies that seemed to have the greatest tradeoff potentials.

The first method is to calculate and store all of the calculated output levels of the DAC. Then, when a certain output voltage is desired, a search algorithm will find the closest calculated output level to the desired level and then send that code to the DAC. This will have the greatest performance as it will achieve the best implementation of the minimization of the difference of the calculated output level and the desired analog output. However, it can cause some extra overhead, such as area overhead for storing all of the output voltage levels in memory as well as implementing the search method.

The second method is to just simply add the rounded DAC's INL at the particular code that would have been passed to the DAC before any pre-distortion codes. This method can be described by (4.11) for every k^{th} code that is desired to be seen at the DAC output.

$$Code_{DAC}\left[k\right] = OriginalCode_{DAC}[k] + INL_{DAC}\left[OriginalCode_{DAC}[k]\right] \qquad (4.11)$$

The reason for the effectiveness of this method is that the DAC INL curve should be approximately continuous locally. By adding the INL at the originally desired code, this should move the new code very near the most optimal code that was found in the above method. If the INL was constant across this small range, these methods would be identical. However, since this will not be exactly true, there will be some small deviations from the optimal method. This method will still have increased performance, but not as good as the first method. It will remove a majority of the harmonic distortions as the systematic error is removed, but there will still be some small errors. This will lead to a little more distortion and noise than the previous method one. It does have some advantages in implementation however. If the number of data points for the FFT data is less than the number of codes of the DAC, then the number of INL codes that need to be calculated will be the number of data points. Also, no additional memory would be needed to store the values as they can just be directly added to the previously desired code while sending the data to the DAC, instead of storing and then reading in a search algorithm later.

Simulation Results

Simulation results were performed in order to verify the proposed algorithms using a 12-bit ADC and a 12-bit DAC. The ADC is modeled by a segmented capacitive architecture. The DAC is modeled by a hybrid of an R-String and an R-2R DAC. In all of the simulations, a size of 2^{12} data points are used with an additive noise of 0.3 LSB. First the two ramp signals are generated with the DAC and sampled by the ADC. The other algorithm then calculates the ADC's INL/DNL. From this information, the DAC's INL/DNL is calculated with the formulas presented in the previous section. The DAC's spectral performance is then tested using a perfect ADC with high resolution for the cases when there are no pre-distortion correction codes applied and when the two different strategies are imposed. The results can be seen in Table 8. As it shows, the THD & SFDR can be greatly improved by using either of the correction methods. Correction 1 is going to be better as previously predicted. The SNR also improved with both of the correction methods compared to no correction method being applied. Again correction strategy 1 will result in the best noise results as well due to the optimal DAC pre-distortion codes being selected. A portrayal of the spectrums with and without correction can be seen in Figure 24.

Table 10: This table summarizes the measurements of the signal generator using the traditional method and proposed algorithms.

DAC Characterization			
Measurement	THD (dB)	SFDR (dB)	SNR (dB)
No Correction	-58.7	65.35	65.77
Correction 1	-88.47	93.08	69.26
Correction 2	-84.01	90.01	67.17

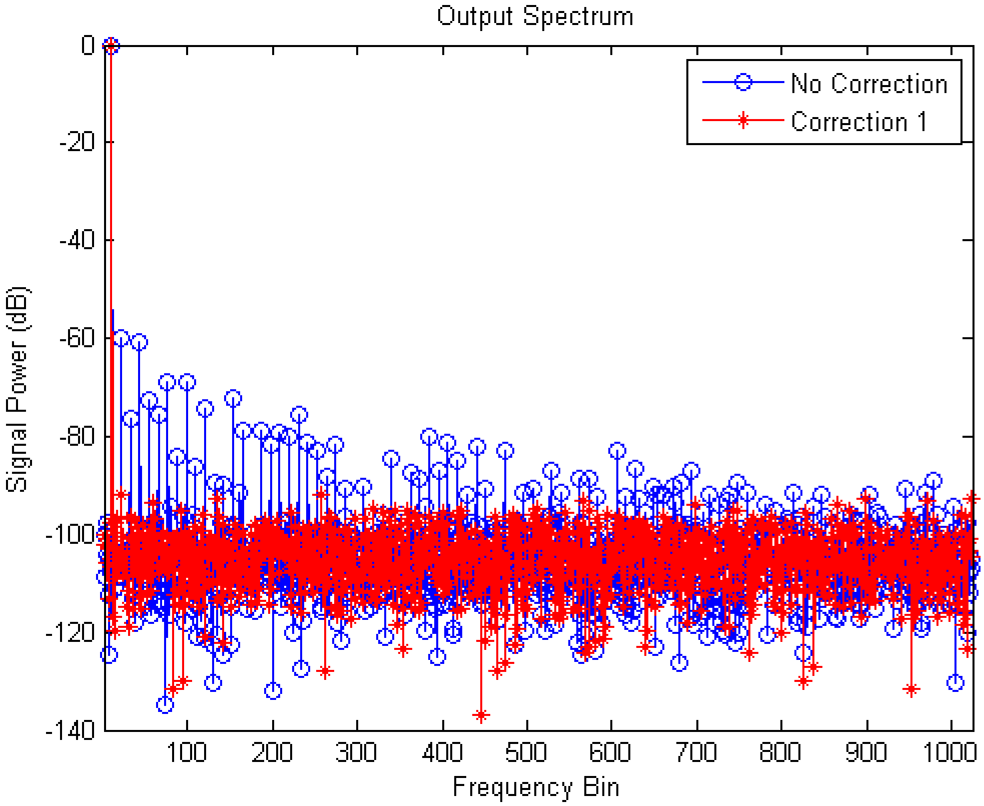
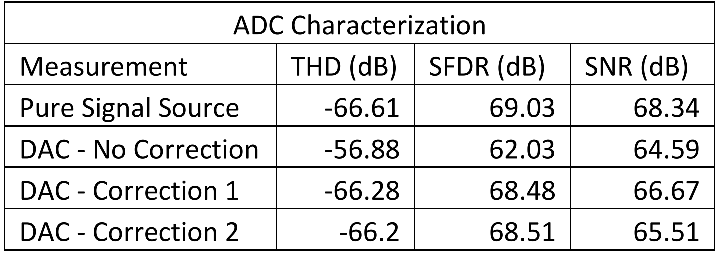


Figure 24: The output spectrum of the DAC with and without pre-distortion correction codes being applied.

The ADC can also be tested multiple ways. First it will be tested with a pure sine wave to give the ideal characteristics of the ADC for comparison. Then it can also be tested by using the DAC with no correction or correction strategy 1 and 2. Table 9 shows the results of these different tests of the ADC. As this table shows, without any pre-distortion correction codes applied to the DAC, the results are dominated by the DAC’s non-linearities and inaccurate results are achieved. With either of the correction codes, accurate estimates of the THD and SFDR are achieved. The measured SNR, however, will have some error in it. This is due to the DAC having a similar level of quantization noise as the ADC as well as some additional noise. Therefore, this noise will be reflected into the ADC’s SNR measurement. To achieve better results for SNR, a higher resolution test source would have to be used.

Table 11: This table summarizes measurements of the ADC using the traditional method and proposed algorithms.



ADC Characterization			
Measurement	THD (dB)	SFDR (dB)	SNR (dB)
Pure Signal Source	-66.61	69.03	68.34
DAC - No Correction	-56.88	62.03	64.59
DAC - Correction 1	-66.28	68.48	66.67
DAC - Correction 2	-66.2	68.51	65.51

Robustness

For further verification purposes, it is also necessary to look at the robustness and repeatability of the proposed algorithm. To do this, the ADC and DAC models were randomly generated and different errors were examined to see if any trends existed. The first characterization examined is the harmonic power estimation error for the ADC’s THD measurement. This error while sweeping the real THD of the ADC can be seen in Figure 25. As it shows, having a correction method provides a large decrease in the amount of harmonic estimation error. As it shows, correction strategies 1 & 2 are fairly close to each other with strategy 2 only having a small advantage. The error power does get slightly larger as the real THD of the ADC gets worse. However, this small trend is not a large issue, as the increase in power error is small compared to the increase in distortion power that causes it.

The other trend that was analyzed was the SNR performance of the ADC measurement. The SNR power measurement error versus the true SNR of the ADC can be seen in Figure 26. As it shows, correction by pre-distortion codes using strategy 1 will always have the smallest error, followed by strategy 2. There are no large trends that are visible in this relationship.

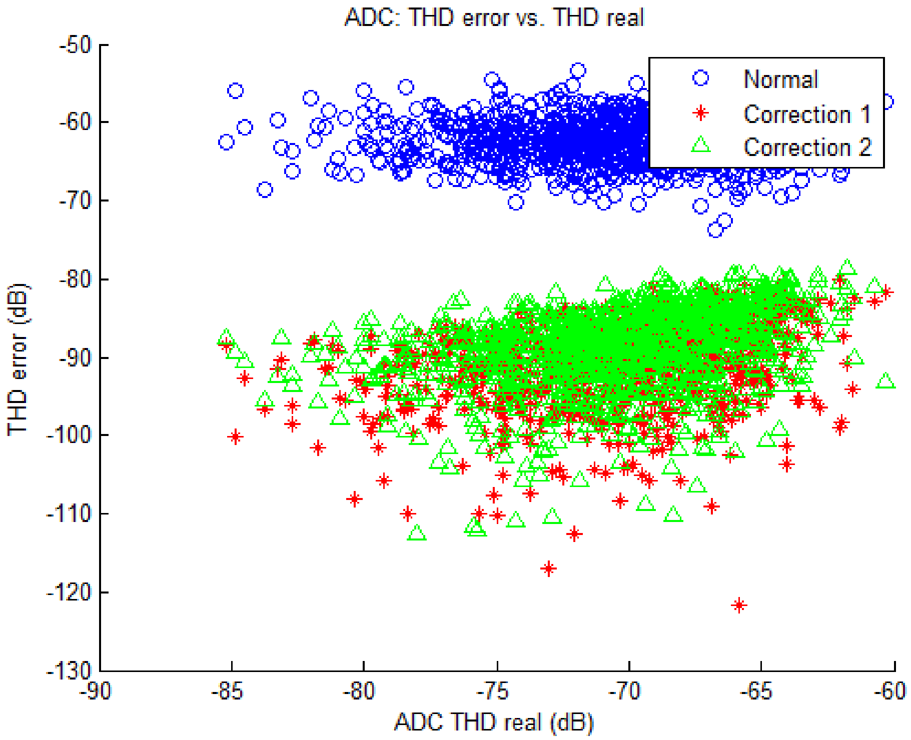


Figure 25: A scatter plot showing the error of harmonic power estimation versus the real THD measurement.

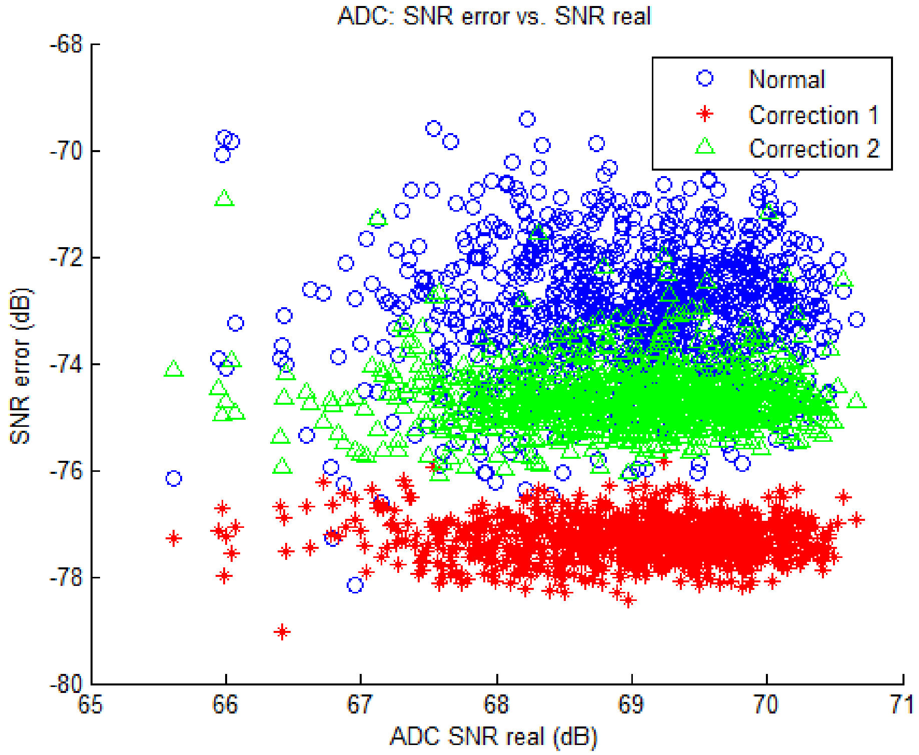


Figure 26: A scatter plot showing the error of noise power estimation versus the real SNR measurement.

Conclusion

The methods that were proposed in this chapter were developed so that a DAC of similar or worse performance could be use in order to get accurate spectral testing of an ADC. This is made possible by another algorithm under development that is able to identify the INL/DNL of an ADC using a non-linear source, or DAC, to generate two ramp signals. Using this information, the INL/DNL of the DAC is solved and then pre-distortion codes can be sent to the DAC in order to get a much higher purity level on the output. This higher purity signal is then able to achieve accurate results upon testing the ADC. This method was simulated using 12-bit resolution data converters and shown that the accuracy of the test is good. Robustness tests were also shown to be able to further show that the algorithm is repeatable across a large variety of performances of ADCs and DACs.

CHAPTER 5

SUMMARY

In this thesis, the spectral testing of high performance ADCs was thoroughly discussed. First, the traditional method to obtain accurate spectral results of ADCS defined by IEEE standards was discussed in Chapter 1. It then showed how the traditional methods caused difficulties and constraints on testing of ADCs. Due to these difficulties that exist in ADC spectral testing, three different algorithms were proposed in Chapters 2-4.

In Chapter 2, an algorithm was proposed to eliminate the need for coherent sampling while also dramatically reducing the requirement for the input signal purity level. The accuracy and robustness of this method was verified through different simulations. Finally, the algorithm was again verified by creating and testing a test PCB board.

In Chapter 3, an algorithm was proposed to again eliminate the need for coherent sampling and reducing the requirement for the input signal purity level, but it also allows for the input signal to be amplitude clipped. This method was proved to be accurate through different simulations results.

In Chapter 4, a special case of ADC spectral testing was discussed. This was when a similar resolution DAC would be being used to generate the input test signal. Then if the INL/DNL measurements of the ADC are known, it would be possible to generate pre-distortion codes to the DAC that would make it possible to get more accurate measurement results of the ADC under test. Simulation results showed the different implementations methods of this algorithm and the resulting accuracy.

In all of the algorithms that were proposed it was the goal to maintain accurate results while attempting to make the requirements of the test to be less stringent. By making the test setup less stringent, it can reduce the costly highly accurate testing equipment. It will also make it easier to be able to perform BIST solutions where there might not be as high of accurate signals available on the system under test. Therefore, this will allow for cheaper and easier ADC spectral testing to be possible. In the future, more work can be done on the proposed methods of this thesis in order to further prove the robustness of the algorithms across all realistic testing scenarios. Also, more measurement results will need to be verified before implementing the methods in a commercialized setting.

REFERENCES

[1] IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters, IEEE Standard. 1241-2010.

[2] IEEE Standard for Terminology and Test Methods of Digital-to-Analog Converter Devices, IEEE Standard. 1658-2011.

[3] IEEE Standard for Digitizing Waveform Recorders, IEEE Standard. 1057-2007.

[4] Yu, Chen, and Geiger. “Accurate Testing of ADC’s Spectral Performance Using Imprecise Sinusoidal Excitation.” IEEE Int. Symposium on Circuits and Systems, 1 (2004): pp 645-648.

[5] Slepicka. “The estimation of test signal quality by means of two simple filters in ADC Testing.” Computer Standards and Interfaces, 34:1 (2012): pp 156-161.

[6] Harris. “On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform.” Proceedings of the IEEE, 66:1 (1978).

[7] Carbone, Nunzi, and Petri. “Windows for ADC Dynamic Testing via Frequency-Domain Analysis.” IEEE Trans. Instr. & Measurement, 50:6 (2001): pp 1571-1576.

[8] Bilau, Megyeri, Sarhegyi, Markus, and Kollar. “Four Parameter Fitting of Sine Wave Testing Result: Iteration and Convergence.” IEEE Trans Instr. & Meas., 45:2 (1996): Pp 588-593.

[9] Simon, Pintelon, Sujbert, and Schoukens. “An Efficient Nonlinear Least Square Multisine Fitting Algorithm.” IEEE Trans. Instr. & Meas., 51:4 (2002): pp 750-755.

[10] Sudani. “FIRE: A Fundamental Identification and Replacement Method for Accurate Spectral Test Without Requiring Coherency.” Instrumentation and Measurement, 62.11 (2013): pp 3015-3025.

[11] Giaquinto and Trotta. “Fast and accurate ADC testing via an enhanced sine wave fitting algorithm.” IEEE Trans. Instr. & Meas., 46 (1997): pp 1020-1025.

[12] Tomioka et al. “A/D Converter Clipping Noise Suppression for High-Sensitivity Carrier-Sensing of Cognitive Radio Transceiver.” Glabal Telecommunications Conference, (2007): pp 4190-4174.

[13] Sudani and Chen. “Accurate Full Spectrum Test Robust to Simultaneous Non-Coherent Sampling and Amplitude Clipping.” IEEE International Test Conference (ITC), Sept. 2013.

[14] Wakabayashi et al. “Low-Distortion Sinewave Generation Method Using Arbitrary Waveform Generator.” Journal of Electronic Testing, 28:5 (2012): pp 641-651.

[15] Netterstrom. “Using digital pre-distortion to compensate for analog signal processing errors.” IEEE Internation Radar Conference, May 1990.