A Digital Method for Phase Noise Measurement

Allan C. Ecker

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2014

Reading Committee:

Mani Soma, Chair

Howard J. Chizeck

Joshua R. Smith

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

A Digital Method for Phase Noise Measurement

Allan C. Ecker

Chair of the Supervisory Committee:
Professor Mani Soma
Electrical Engineering

An all-digital method is used to detect sinusoidal phase noise components while reducing the need for computation intensive post-processing. We use the frequency-selective behavior of a narrow bandwidth infinite impulse response (IIR) digital filter to replace the frequency-domain evaluation of an FFT. This form of digital post-processing reduces the calculation overhead of making frequency-domain observations. We provide a model for the inherent nonidealities of the relevant quantization processes which will need to be managed in a practical implementation of this technique for manufacturing test.

We also present a method for capturing phase noise amplitude from arbitrary bit stream data. This extension allows the technique to function on any set of binary data that has sufficient edges to provide information on the phase noise under evaluation.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

JITTER: a measurement of timing uncertainty, evaluated in a variety of situations as various measurements of time differences between digital edges.

PHASE NOISE: another measurement of timing uncertainty, generally evaluated in the frequency domain rather than the time domain.

ATE: Automated Test Equipment, a system of monitoring devices used to evaluate functionality and quality of integrated circuits during their production.

TDC: a Time to Digital Converter, any mechanism for evaluating the position in the time domain of a crossing point of a voltage waveform.

USRP: the USRP signal generator is a general purpose RF signal generator, which in particular can be used to create a high frequency clock signal with an accompanying spur.

FPGA: Field Programmable Gate Arrays are reconfigurable circuits capable of being programmed to perform different operations.

BIST: Built-In Self-Test refers to the many subcircuits and other mechanisms built into modern ASICs for testing purposes. Many modern ASICs will have significant area devoted to self-test.

PRBS: Pseudo-Random Bit Sequences are lists of numbers generated by a computer to bear a desired resemblance to true random numbers.

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation:

To the University of Washington, a great place to find collaborators when deadlines loom,

To Ken Blakkan, for providing a quick wit and a deep knowledge of the field, and as solid a sounding board as can realistically be hoped,

To Doctor Mani Soma, for years of encouragement and support,

To his committee for making time to evaluate this work,

To Ryan and Caytlin, for being an island of calm in a sea of chaos. (Oh shut up, Ryan, you know it's true.)

To Karla, for lots of sympathy in general and one really important mix-tape in particular,

To his Mother Emily, who taught him to be curious and excited about the world,

and to his Sister Cathy, who never for a split second stopped believing in him.

DEDICATION

to my Niece Catarina, who might be four now,
but who will probably end up writing one of these herself one day.

Chapter 1

INTRODUCTION

1.1 Motivation

Modern communications systems such as cell phones and wireless routers operate in an environment with many other narrow bandwidth communication signals and as such are frequently subject to sinusoidal disturbances. Simple measurements such as peak jitter or single-cycle jitter may mis-characterize or underestimate the impact of such interference. It is important to evaluate frequency-domain characteristics of phase noise. To do this, the frequency domain may be monitored directly (with a spectrum analyzer) or indirectly by taking time-domain data and computing the Fast Fourier Transform (FFT). With sufficient record length and measurement bandwidth, it is possible to accurately measure phase noise at any frequency between the bandwidth and the inverse of the record length.

While these tests are part of industry practice when test time and complexity are not budget-constrained, test for modern RF integrated circuits is frequently limited by available Automated Test Equipment (ATE) resources, which may prevent the use of analog spectrum analysis due to resource constraints, or the use of complex calculations such as FFT. Analog tests of any kind may be outside the scope of a tester chosen to minimize cost per second of run time.

Recent focus in industry has been on spot-testing of individual frequencies in the output spectra of signals. Because full spectrum testing is expensive, and because a particular frequency mask is likely to be well characterized by a few spot frequencies, high-volume test has begun to focus more and more on testing only one or two frequencies, and then making a judgment on whether a circuit or system is passing or not. This particular use case has been of interest to integrated circuits manufacturers.

Standard specification of devices and components has relied on spot test of the magnitude of spectral content at one or two offset frequencies from a carrier rather than on complete

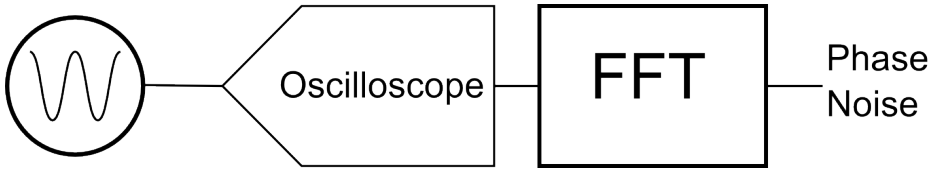

Figure 1.1: Conventional analog spectral analysis

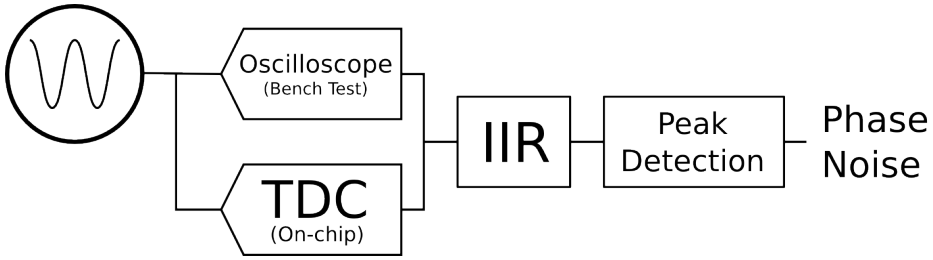

Figure 1.2: IIR phase noise extraction technique

descriptions of the spectral content of an output signal. This has been done for reasons of test cost, and to simplify the description of test to something which will be comparable across multiple technologies. A technique which solves this problem will be of particular value to industry, especially if it is possible to integrate this solution to low-cost test boards or on integrated circuits as part of the built-in self-test budget.

The focus of much of the work of the past ten years has been on measuring jitter of high frequency systems rather than phase noise. In this work we will focus on the extraction of phase noise from time-domain data, which will allow us to take advantage of the work done in jitter measurements. Current phase noise analysis techniques include all-digital methods for acquiring time-domain data such as waveform crossing points. In this we present analysis which shows that an all-digital detection of sinusoidal phase noise at a particular frequency is possible from this time-domain data. In our technique a time to digital converter (TDC) replaces analog sampling, and a single infinite-impulse response (IIR) digital filter is used as a replacement for conventional FFT, eliminating the need for storing the entire waveform before calculations can begin, and the need for making calculations to evaluate frequencies which are not relevant.

1.2 Organization

The following chapters describe a method for phase noise measurement. In the next chapter, background and recent research on this topic are presented. The principles of analog to digital conversion and digital filtering are discussed. Recent work describing strategies for monitoring phase noise and jitter are listed and described. Comparisons between this technique and others and design issues are introduced. Tradeoffs in this design space are considered.

In chapter 3 on the theory of this technique begins with a discussion of crossing point detection, in general and how the particular case of test data in this work is handled.

Crossing point detection, as it is handled in this work, is described. A section on the relationship between time and voltage in a clock-like signal is presented as a mnemonic for relating the time-domain signal of a TDC to the voltage domain more commonly used in analog and digital signal processing. Sampling theory is employed to show the relationship between the high-frequency input and the observed low-frequency signal as extracted by a TDC. This relationship is then used to describe the appearance of near-carrier spurs as low-frequency tones via undersampling. The use of an IIR to evaluate this low-frequency signal is described, along with the details of design for this filter. The final step of peak detection is also covered.

Additionally, this technique is extended to show its usefulness in evaluating phase noise in non-clock digital wave forms or data signals. In this discussion, it is observed that the Fourier domain description of the phase noise is dependent on the total number of periods in a measurement, and that where near-carrier interference is involved, the changes to the sampling position created by "missing" edges will not cause large errors in the captured phase noise. A unified algorithm for handling both clock-like signals and data signals is then presented.

In chapter 4, a number of simulated test cases are described. We conduct a number of demonstrations of the same basic system concept of a TDC and IIR filter for measurement and frequency selectivity. In the basic software demonstration, we show that a sinusoidal interferer can be measured using a digital filter acting only on time domain capture of

edges. This basic demonstration lays the groundwork for operating in a more complex noise environment, which could contain multiple interferers and other sources of noise. We then present considerations for the nonidealities of the TDC and IIR, in particular quantization error introduced by both circuits.

A simulated test case with a data signal is also presented in this section, illustrating the technique operating with a real data signal. This case is presented along with analysis of issues related to handling a data signal with this technique.

In Chapter 5, experimental results showing this technique successfully extracting information from real data signals are presented, along with details concerning the handling of quantization and other real-world issues the technique may need to take into account. In the first experimental test case, we examine sinusoidal interferers both near to and far from the carrier signal to evaluate whether the technique can operate in more complex noisy environments, reacting gracefully to multiple sources of noise.

In the near-carrier demonstration, we design a specific system for extracting injected phase noise from physical test data of a 1GHz clock with an injected 1MHz offset phase noise. The test data for this experiment is a time sequence from a Digital Sampling Oscilloscope (DSO) at a sampling rate of 20GHz for 500μs for a total of 10M points, which will be converted into time domain data via crossing point detection. This setup is comparable to a system where ATE is taking time-domain measurements of a clock signal. This experiment measures the 1MHz offset phase noise, and can do so with only crossing-time information.

Using this system as a starting point, the results of simulating the impact of quantization noise in the TDC and IIR are examined in detail, with particular attention paid to the impact of different sources of error caused by the many sources of discretization found in digital systems. While these sources of error definitely decrease the precision of measurement, we show that these are manageable sources of error and that tradeoffs can be made to retain a viable test under nominal conditions. This analysis makes a strong case that the technique is compatible with implementation on field-programmable gate array (FPGA) or built-in self-test (BIST) section of an application-specific integrated circuit (ASIC).

Finally, we show that the merits of this system translate well to the case of a data signal, where instead of a clock waveform or sinusoidal carrier, the source of edges is a

pseudorandom bit sequence (PRBS), which will have many "missing" edges. This evaluation uses an analysis of sampling theory to show that removing many edges from the clock (as occurs in a PRBS) will not prevent the technique from functioning properly, but rather introduces a manageable source of noise.

In Chapter 6, issues relating to these real world issues and further analysis of the test cases are presented, along with some discussion on the relative merits of the technique. Analysis of errors is presented, along with a comparison to other measurement techniques. The practice of normalization in this technique is considered.

The critical path of the IIR circuitry is described in detail. The choice to replace Root Mean Square (RMS) detection with peak detection for final evaluation is explained, and the matter of prior knowledge of the data signal for some simulations is acknowledged and considered in more detail.

Chapter 7 presents the conclusion that this technique is potentially viable as a measurement technique for phase noise.

An appendix describing the use of fixed point math is included. The limitations of fixed point math are described to present an accurate picture of what can be achieved with simple (non-floating point) multiplication hardware.

1.3 Criteria of Success

To successfully address the problem of measuring phase noise at low cost, a measurement system must be capable of monitoring frequencies at the native speed of the process, meaning that a signal at the clock frequency of the logic used must be observable to the system. It must also limit or eliminate analog circuitry to minimize the cost of adding circuitry to a device under test. It should be able to selectively evaluate if a specific frequency is experiencing phase noise. If possible, it should also eliminate the need for large amounts of digital processing, which may be available on a test system, but which would be hard to integrate into a small area.

Chapter 2

REVIEW OF PREVIOUS WORK

2.1 Background

The topic of this work is a specific class of measurements of a signal for unwanted sinusoidal interference. Before approaching this specific technique, a general overview of the approach to the signals of interest will be helpful in presenting the technique clearly. Here, we will review analog to digital conversion and filter design.

First, a look at analog to digital conversion. Virtually all modern forms of measurement employ analog to digital conversion at some point. The analog nature of most real signals in the physical world makes some form of these devices a requirement on most hardware, from cell phones to automobile controls. For the purposes of this discussion, we will look at the way this impacts the information contained in a signal more closely than a particular type of hardware for this task.

The act of analog to digital conversion causes two major reductions in the information content of a signal. (Whether that information is desirable or not is another topic: it's often possible to capture all or nearly all *desired* information in analog to digital conversion, while losing only information which is unnecessary.) These two reductions are the conversion to discrete time and to discrete level.

While the conversion to discrete level is not without consequences, it is possible without any major alteration of signal content: quantization results in some error in the vertical level of the signal at every point, but the precision of the conversion sets a hard limit on how great this error can be, and the relationship between a signal and its quantization error is usually such that much of the error appears as broadband noise. As a result, a well-conditioned and sufficiently small quantization step is innocuous to a signal, regardless of its spectral content.

Conversion to discrete time can be done without any error on the signal, if the sampling

rate is high enough. If all spectral content in a sampled signal is below the Nyquist rate (one half the frequency of the sampling rate), it will be possible to perfectly reproduce all captured content. However, in the real world, many signals exist outside this special zone of frequency, and so sampling becomes a major point of entry for potentially unwanted signals.

Analog to Digital conversion performs both of these tasks, operating in a variety of media, sampling rates, resolutions, and domains. For an instrument to be of value in detecting unwanted signals, it must be able to evaluate only the entry point of such a signal, or *a priori* knowledge of the signal must allow the user to be certain there is only one explanation for a given readout.

The evaluation of an undesired sinusoidal signal may make use of the property of aliasing to evaluate a signal which would otherwise fall outside the Nyquist rate. This technique is called undersampling, and in cases where a stable time base is available, it can make it possible to evaluate signals tens or hundreds of times the frequency of operation of equipment.

On the topic of filter design, we will consider the limitations and advantages of two types of digital filters (infinite impulse response and finite impulse response) by comparing them to analog filters. Analog filters have the advantage of a large body of work concerning their design, but their disadvantages, particularly where power use, size, and noise vulnerability are concerned, disqualify them from most high frequency test procedures which are cost-constrained.

All digital filters must simulate the behavior of an analog filter in a discrete time environment. For this reason, no digital filter can perform operations above the sampling rate, or below frequencies not contained in the finite time of evaluation. Both of these constraints are inherent to digital filtering, and cannot be relieved completely.

The Finite Impulse Response (FIR) filters are the simpler of the two types of filters, and may be thought of as convolving a fixed pattern with an incoming signal. This fixed pattern is of finite length, which is unlike analog filters, whose frequency responses may drop to nearly zero nearly immediately, but which usually have some nonzero value at all values of time greater than zero. As a consequence, an FIR filter is unconditionally bounded-input bounded-output stable, and will always have a linear phase. A key drawback of FIR filters is that, as the frequency of interest of a filter decreases (relative to the sampling rate), the

part of its impulse response of appreciable value grows longer, requiring a fixed pattern of greater length to accurately filter the incoming signal.

The Infinite Impulse Response (IIR) filter, by contrast, employs feedback in its design. As a result, it does not suffer the limitation of the FIR filter for the purposes of observing low-frequency signals, since feedback allows for difference equations which describe resonant behavior. However, this resonant behavior also introduces the possibility of instability. To be unconditionally stable, an IIR filter must be designed with care that its impulse response is decaying rather than growing. When computational resources are plentiful, the FIR is often preferred over the IIR for its unconditional stability, but where computational resources are scarce, such as in test systems, the relatively compact design of the IIR may reduce complexity significantly.

2.2 Recent Work

Current phase noise measurement techniques predominantly focus on the classification of jitter as a time-domain phenomenon. This approach proceeds naturally both from digital analysis and from the availability of TDCs for evaluating jitter [2] [3]. Phase noise is the product of multiple frequency-dependent processes, and as such is often best represented in the frequency domain, more commonly evaluated off-chip with the use of spectrum analyzers or time interval analyzers [4].

For the extraction of phase noise, work has been done in using the FFT to compute the spectral components of a signal from its time-domain behavior as shown in figure 1.1. Spectral analyzers have been applied to this problem directly as a common means of detection, and for this reason some works have attempted to move this analog apparatus into the testing environment or even on-chip. [5]

In [6], digital filtering was used to separate phase noise into two bands, allowing unwanted sources of phase noise to be considered separately from timing variation introduced by spread-spectrum clocking. This technique is targeted to the specific application of spread-spectrum clocking, which includes a certain amount of intentional clock variation at a certain frequency. The phase locked loop of the clock generator in this system is used as a way to evaluate jitter. This is a commonly used technique because the PLL must adjust itself

continuously to maintain phase lock, which requires it to generate its own measurement of the phase of each clock pulse. The jitter sequence extracted from the PLL, however, includes a large source of variability which has nothing to do with noise: the spread-spectrum clocking's modulation profile. The authors select a frequency cutoff and use a high-pass filter on the data, which causes the loss of some jitter information below a certain offset frequency, but they are also able to show that in many circumstances this loss is not severe. They also use a low-pass filter to extract a lower band, which is dominated by the spread-spectrum clocking profile. This grants the added benefit of allowing diagnostics to be run on the clocking mechanism, independent of high-frequency noise. This isolation of phase noise into two bands with digital filters is very similar to our use of a narrow band digital filter.

This system successfully meets several of the criteria outlined in the previous chapter: the phase locked loop allows a stream of data to be taken from a very fast clock. This technique operates only on digital information once it leaves the PLL, and so does not add to analog complexity. Once the data are read into a computer, FFT analysis allows individual frequencies to be examined in detail. However, all processing beyond the initial frequency separation must be carried out off-system and offline. The proposed method will eliminate the need for complex off-system processing.

Similarly, in [7], frequency-domain transforms were used to construct classifiers for types of jitter. In [8], a method is presented for measuring phase noise as a function of time-domain jitter obtained from available TDC structures. This method relies on the relationship between the voltage of an interfering signal and the crossing time of the monitored waveform. It has also been shown that 1/f phase noise can be evaluated with phase measurements [9]. A direct relationship between the crossing points of a waveform and a voltage sampling has also been demonstrated in [10].

While frequency-domain techniques have been applied in a number of cases to BIST-gathered data, the bandwidth requirements of performing FFT have been an impediment. Previously, this has been solved with a decimation filter [11], which throws away information to keep bandwidth manageable. It has also been shown that single-shot time measurements such as those made in [4] can be combined with DSP to extract quantitative information about sinusoidal jitter [12]. Another technique for managing FFT bandwidth, swept digital

filters, has been used to measure signals [13].

In [12], for example, the authors create a period-tracking circuit to extract a period jitter sequence. The period-tracking circuit operates by observing a period comparator and increasing or decreasing its delay. The delay adjustments are done with a variable delay line, a circuit which can be built with switched capacitors or by multiplexing variable numbers of buffers (depending on the amount of delay and the precision desired).

This causes the total delay to approximate the period of the clock, plus noise, including phase noise. This process is analogous to the phase comparator in a phase-locked loop. The final result of this process is a measurement of the period, plus any noise. In the event of low-offset noise such as sinusoidal jitter, ordinary DSP of any kind can detect it. This detection scheme is therefore compatible with our technique's key aspect of filtering and evaluating. The authors describe a process of using the FFT on the finished data, which in cases where the amount of data or the frequency of the signal is not excessive can save significant time on processing.

In [16], a mixer is used to shift the frequency of an incoming signal into the range of a fixed-frequency detector. While it is always desirable to limit the use of complex RF circuitry in test and measurement, the mixer is a broadly-used and simple way to transform a difficult RF problem into a simple analog or mixed signal problem, and phase noise measurements are no exception. The authors also use a fixed-frequency linear receiver and sweep the frequency of measurement, another trick made possible with the mixer, to take narrow band measurements. A primary claim of this paper is that it makes phase noise measurements traceable, meaning that they can be reliably compared with some accuracy to the "ground truth" of a spectrum analyzer or other high-quality measurement. The availability of the carrier itself as a measurement point, along with the possibility of robust calibration of the mixer and detector hardware, make this possible.

While our technique eliminates the need of a mixer by using the TDC itself to accomplish the transition to baseband, there are clear similarities in the use of a narrowband filter to simplify evaluation and reduce the need for complex evaluation techniques such as the FFT. The concession of the mixer allows the monitor to evaluate signals at the native speed of whatever technology it is built on, and while this technique does not eliminate analog

circuitry entirely, it does limit it to a mixer and a detector at a single frequency. This frequency selective system also avoids off-system processing. The analog circuitry may be too expensive for many systems however.

It has been demonstrated that phase noise can be measured with a spectrum analyzer under a wide variety of circumstances, including on-wafer [17]. Where an orthogonal mixing process (such as those used to demodulate phase-shift keyed signals) is available, a phase noise component can be extracted with both its phase and magnitude intact. [18]

Previous techniques have relied on analog measurements to evaluate phase noise or have focused on evaluating jitter. The work presented below is a digital method to measure phase noise directly from digitized waveform zero-crossings. It extends our previous work [19] with more details concerning error analysis of this technique, particularly with respect to quantization error in the digital and mixed signal operations involved. A measurement of the accuracy of this technique expressed in terms of Signal to Noise Ratio (SNR) has been added to more clearly illustrate the tradeoffs in precision where appropriate. The SNR metric is now applied to show the relationship of this technique to the full FFT and to illustrate the effects of quantization error. Peak detection has replaced RMS amplitude detection as a simpler and less error-prone way of evaluating signal scale. Other improvements include the addition of a comparison of memory usage to FFT, an illustration of the effect of aliasing in this technique, and an analysis of the tradeoff between measurement duration and SNR. Also included is our more recent work [20] describing the use of this technique for evaluating phase noise in data signals.

Because this technique is functionally equivalent to applying a digital filter to an incoming group of samples, it can run at the native speed of the system. It uses no analog circuitry beyond the TDC common to all previous techniques. The IIR makes this technique frequency-selective, and also significantly reduces the overall scale of the digital circuitry required to take phase noise measurements.

2.3 Test Cost Comparison

Beyond the obvious benefit of making costly measurements with a spectrum analyzer unnecessary, this test also has some big savings against performing a conventional FFT on

measured data.

First, pushing the entire sequence of measurements back to the test computer where FFT can be done would require a link that carried one measurement every clock cycle which could be a problem on an inexpensive tester. With this technique, the digital circuitry required is potentially simple enough to implement on a testing board near the IC, without involving the tester until the result is ready for readout.

Second, the number of calculations is smaller, but more importantly the FFT operation requires simultaneous access to the entire sequence where the IIR method only requires access to one piece of data at a time. Thus the greatest benefit of this technique is that the data does not need to be completely captured before calculations can begin, and in fact they can be made synchronous to the test, provided the hardware that performs the operations is able to finish one evaluation of the IIR in one cycle of the measured waveform.

2.4 Design Tradeoffs

In this work, interpolation is used to detect the crossing points of the data. In the case of a TDC implementation, the accuracy and precision of the TDC will be a primary driver of performance of this technique. While this work makes the assumption of available time-domain data for use with the crossing point method described, applying this technique with more limited resources (on-board or on-chip testing with a direct TDC) will introduce quantization noise from the converter. If the TDC makes use of a reference signal, any phase noise introduced by it will also impact performance. In order for this technique to perform under such conditions, both sources of error will need to be minimized.

As with conventional measurements of low-offset phase noise, a long measurement time is necessary to detect a low-offset phase noise component. This requirement is not relaxed by this technique, although the post-processing is reduced or eliminated depending on the implementation in hardware and software.

Because this is a sampling technique, phase noise outside the band from the inverse of the measurement time to one half the clock frequency will “alias” down into this band, and if these spectral components overlap the IIR bandwidth, contribute to the error in the final measurement.

Chapter 3

THEORY

As discussed in chapter 1, we use a TDC to measure the crossing points of the signal and then use these measurements to extract phase noise. We will describe our method for crossing point detection, which in our test bench measurements replaces the TDC. We will show that raw TDC output has a linear relationship to a voltage series of the signal, and thus to its Discrete Fourier Transform (DFT). The relationship to sampling theory is exploited by the IIR filter to extract phase noise, and a peak detection measurement is used to compute scale.

We also describe a technique for extending this method to the monitoring of serial data signals for the same phase noise components detected in clock signals. To accomplish this, we employ a simple algorithm which exploits the properties of the signal under test to monitor the signal with an acceptable level of precision despite the many missing edges which occur in a data signal.

Both methods will be described in this section in detail.

3.1 Crossing Point Detection

In a practical implementation, the TDC operation will be carried out by a digital edge detection circuit such as a Vernier Delay Line (VDL). In a VDL, a chain of detectors (usually flip-flops) with varying delays are triggered together, thus converting the edge transition to a digital code. In such a circuit, the edge of a fast signal can be accurately determined even in a system where gate delays are of similar magnitude to the signal under test.

To simulate the result of using a TDC to detect crossing points, time domain measurements are used to capture the positive zero-crossings of the waveform, which are extracted by interpolating points on either side of each crossing. Starting from the original discrete time domain measurements $V(n_t)$, we can label the samples immediately after positive edges

as the series m. A simple evaluation of the crossing points would set them at the points m_n/f_s, where n is the index of the original voltage series' crossing point and f_s is the original sampling rate. If we include interpolation to pick each crossing point more accurately, the set of crossing points in time can then be written as:

$$T(n) = \frac{1}{f_s}\left(m_n + \frac{V(m_n - 1)}{V(m_n) - V(m_n - 1)}\right) \tag{3.1}$$

(The effects of this sampling method vs. actual TDC will be discussed in chapter 6.) This gives us the time series $T(n)$. It is important to note that this crossing point detection method can be replaced with any TDC. Like any data converter, the TDC will experience quantization error in the final output. This source of error is dependent on the resolution of the TDC used, and will be a limiting factor in precision of phase noise measurements with this technique.

3.2 Time/Voltage Relationship

From the sequence of time-domain measurements above, we next produce a voltage-domain sampling of the waveform under test. This analysis assumes the signal under test is a predominantly sinusoidal waveform. Here, a "predominantly sinusoidal" waveform must meet the following criteria: Its zero-crossings must occur exactly once per period of the fundamental, and the waveform near the crossing points must be measurable by a TDC. The signal under examination for this analysis may have both sinusoidal and white noise as seen in figure 3.1.

If our sequence of measurements represents the positive transition points of a periodic waveform which meets the above criteria, it is a time-sampled time waveform, and we can represent $T(n)$ above in terms of a constant, linear increase and $\Delta T(n)$:

$$T(n) = \Delta T(n) + n \cdot T_0, \tag{3.2}$$

Where T_0 is the period of the fundamental, as demonstrated in [10]. The intervals $\Delta T(n)$ represent the full "phase series" of the signal under test. The values of common jitter metrics can be extracted from $\Delta T(n)$. This series is also the starting point for this

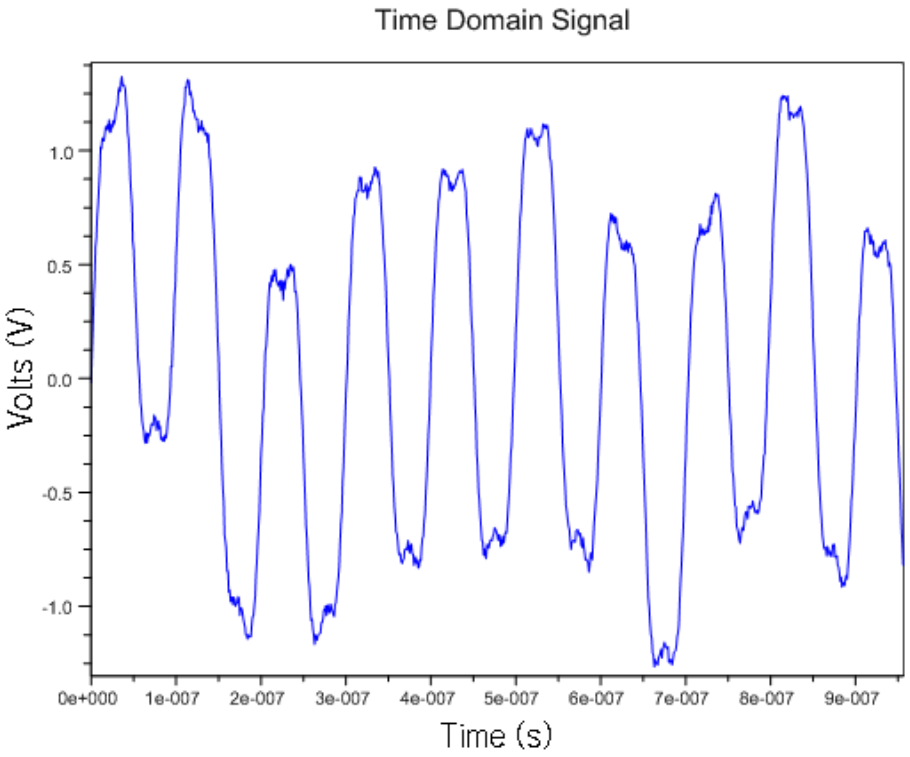

Figure 3.1: A signal compatible with this technique $V(n_t)$, including sinusoidal phase noise and white noise displayed in the time domain.

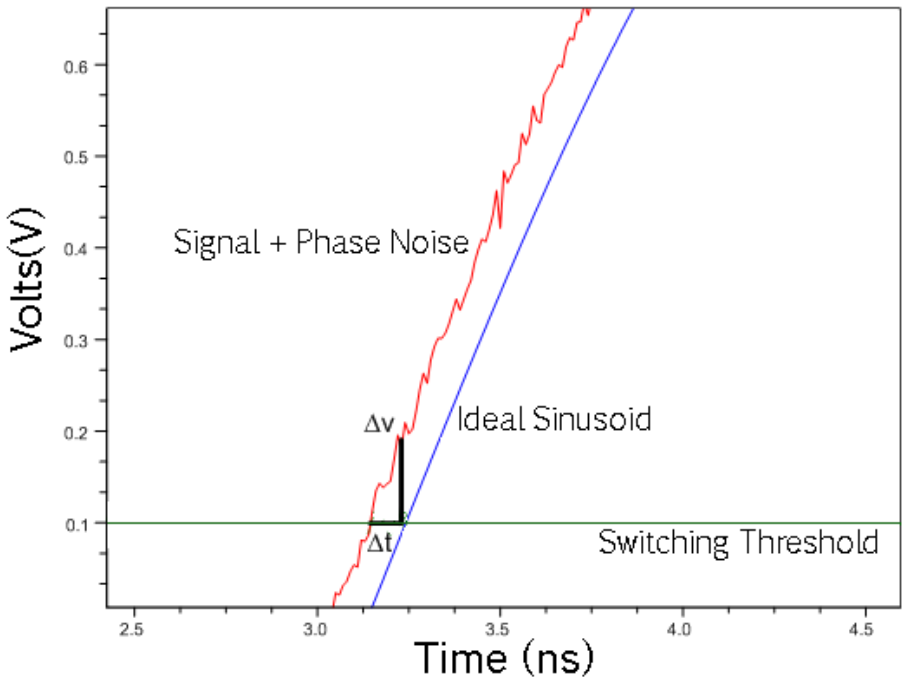

Figure 3.2: Δt of a waveform from the ideal will be proportional to ΔV

technique's measurement of phase noise. As shown in figure 3.2, from the definition of the derivative, we can say that for small values of $\Delta T(n)$,

$$\frac{\Delta V(n)}{\Delta T(n)} = \frac{dv}{dt} \tag{3.3}$$

Where $\frac{dv}{dt}$ is the derivative of the time-domain signal, at the points where $\Delta T(n)$ is measured. For a sinusoidal signal with small perturbations, this derivative is a constant:

$$\frac{dv}{dt} = A\omega_0 \tag{3.4}$$

Where A is the amplitude and ω_0 is the frequency of the fundamental. For any sufficiently small perturbation of the signal, $\Delta T(n)$ becomes a predictor of $\Delta V(n)$ by the formula:

$$\Delta V(n) = \Delta T(n) \cdot A\omega_0 \tag{3.5}$$

This relationship was previously demonstrated by Keller [10]. It is important to note that any errors in this assumption will turn into errors in the measured values of ΔV. For example, large noise sources which cause the value of ΔT to be less linearly related to ΔV will introduce mixing products, degrading the quality of the measurement.

We now have a time-domain voltage sampling of the original signal. This voltage series is not sampled at the crossing points (that set would be all zeros) but at varying times slightly away from the crossing points, at what would be the crossing points in a pure sinusoid. (In a practical implementation, this is really the crossing points of the reference signal.) The Fourier transform of this series is an evaluation of the frequency-domain of the signal. So if a DFT of the signal from these voltage samples was:

$$F(\omega) = \sum_{n=0}^{N-1} \Delta V(n) e^{-j\frac{2\pi n\omega}{N}} \tag{3.6}$$

The relationship between $\Delta V(n)$ and $\Delta T(n)$ can be used to construct this function from only the time series $T(n)$:

$$\begin{aligned} F(\omega) &= \sum_{n=0}^{N-1} A\omega_0 \Delta T(n) e^{-j\frac{2\pi n\omega}{N}} \\ F(\omega) &= A\omega_0 \sum_{n=0}^{N-1} \Delta T(n) e^{-j\frac{2\pi n\omega}{N}} \end{aligned} \tag{3.7}$$

With this relation, we can compute the frequency domain behavior of a signal from a time-domain crossing-point sampling technique. Since the computed relationship between ΔV and ΔT is linear, the frequency domain behavior may be observed directly from the time series $\Delta T(n)$, with only a scaling factor to correct for any differences between the two results.

3.3 Sampling Theory and this Technique

From now on we may treat the time series $\Delta T(n)$ as a sampling of $V(n)$ from the original waveform, so long as the original signal $v(t)$ meets the criteria above. This sampling has the frequency $\frac{1}{T_0}$, so aliasing will be a major factor in the analysis of this technique for

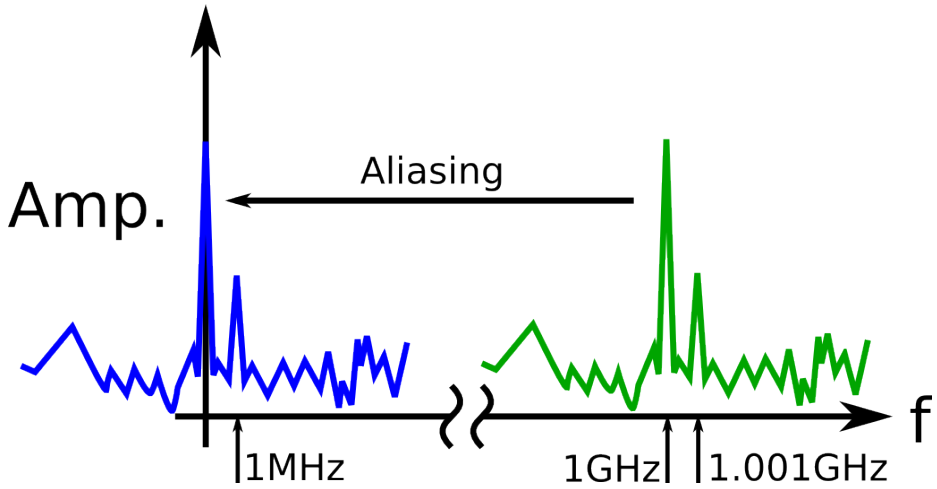

Figure 3.3: With a clock at 1GHz, a phase noise component at 1.001GHz aliases down to 1MHz.

the impacts of noise and other non-ideal signal properties. As the sampling rate and the fundamental of this signal are the same, the fundamental will alias to DC, along with all of its harmonics.

The sampling rate of the derived voltage waveform is the same frequency as the excitation. We cannot obtain any oversampling information by speeding up the capture, because the capture rate is fixed at the frequency of the signal under test. With more samples we can increase the frequency resolution of $F(\omega)$, because the spacing between its frequency bins is $\frac{1}{T_{total}}$ This technique will not measure the harmonics of a signal since they will be mixed down to the same frequency as the signal (DC). This technique's value is in the detection of phase noise on the signal at some offset from the harmonic.

3.4 Undersampling

While this technique is useful for observing low-frequency phenomena such as ground noise and baseband interference, a much more interesting application is using undersampling to detect phase noise very close to the clock frequency. Because we are sampling at the frequency of the signal under test, phase noise at the frequency $\omega_o - \Delta\omega$ will alias to $\Delta\omega$, so for example phase noise at 1.001GHz will map to 1MHz (as shown in figure 3.3), where

our technique can capture it.

As mentioned in chapter 2, this use of aliasing is only valid if there are no large noise sources at very low frequency. Fortunately, this is often the case with RF systems, which have good DC blocking characteristics.

3.5 IIR Extraction Method

We have shown how the output from a TDC may be directly converted into a voltage-domain sampling, and how all-digital techniques such as the FFT may be used to extract phase noise.

Although the use of all-digital evaluation eliminates the need for costly analog circuitry, using FFT to evaluate the phase noise of the signal under test may still be time consuming and costly, particularly if local computational resources are not up to the task of storing and processing the many data points necessary to evaluate low-offset phase noise. To alleviate this pressure with minimal loss of accuracy, we propose an IIR filter whose storage and computation requirements can be fixed while providing very long impulse response. A long impulse response allows very low frequency evaluations without requiring large amounts of storage or processing power. This allows a particular instance of sinusoidal phase noise to be isolated from the other noise in the signal as shown in figure 3.4.

For this technique to work the IIR should be fairly narrow band, so a narrowband filter of the form:

$$H(s) = \frac{\frac{s}{\omega_n}Q}{\frac{s^2}{\omega_n^2} + \frac{s}{Q\omega_n} + 1} \tag{3.8}$$

is used, where ω_n is the frequency of interest and Q is a quality factor. A two-pole filter with a high Q as in figure 3.5 is desirable, since this translates into an IIR with only two taps. This analog filter is converted to a digital IIR filter, which results in a digital IIR which will extract only the behavior of the signal under test at a given frequency.

The impulse-invariance transform has been chosen from the different ways a filter may be converted, both because of its simplicity and its faithful translation of the center frequency in the presence of sampling. More sophisticated techniques such as the matched-z transform

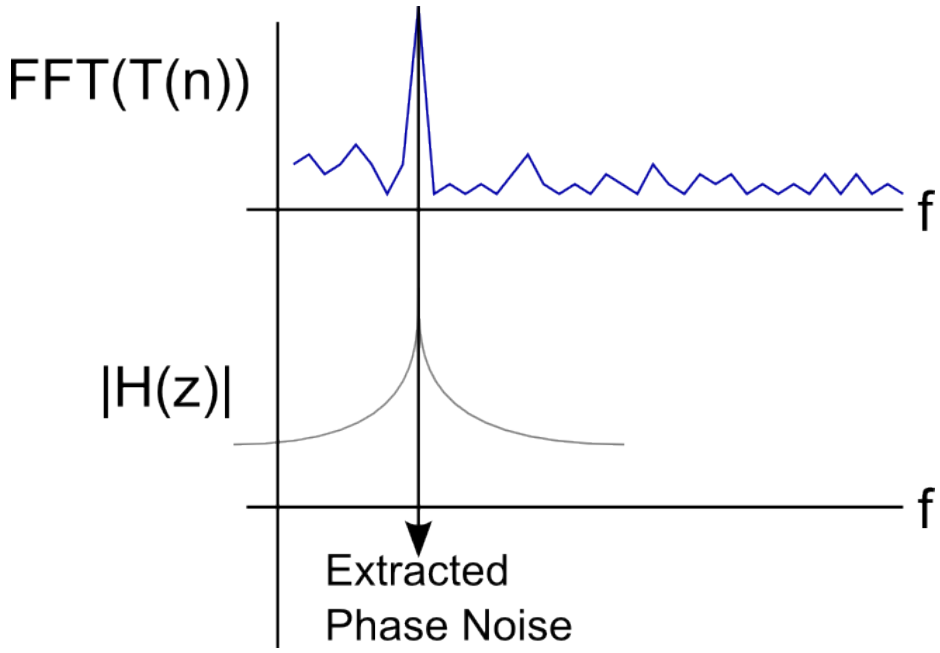

Figure 3.4: Filtering the signal at a given frequency allows the extraction of its phase noise at that frequency.

may present a more robust guarantee of stability in the presence of large quantization in filter coefficients, as well as placing filter zeros accurately, but these qualities are not necessary to this particular filter application.

The impulse-invariance transform has one very important advantage as well: its faithful representation of the impulse response at sampling times. In the case of a low frequency narrow band filter, it is important that the impulse response of the filter match well to an underdamped oscillation as close to precisely the center frequency as possible. The precise frequency of a pole may not be terribly important in the case of a filter constructed to achieve rolloff with many poles, but in the case of a narrow band filter achieved by precise tuning of a single pair of complex poles, this becomes the single most important design parameter.

To perform the impulse-invariance transform, the analog filter must be put into a sum of first order fractions by partial fraction expansion of this form:

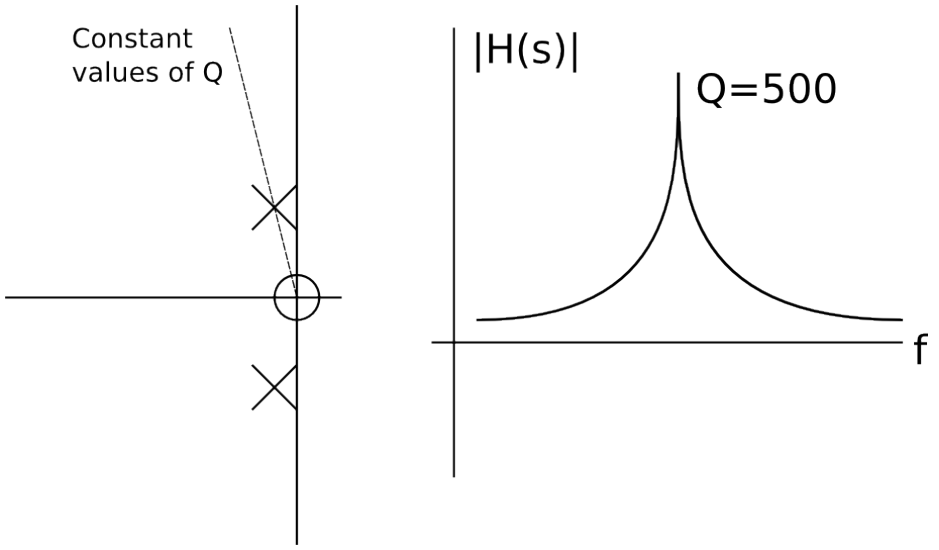

Figure 3.5: The two-pole band pass filter design keeps the poles near the imaginary axis for a constant Q of 500.

$$H(s) = \frac{R_0}{s - p_0} + \frac{R_1}{s - p_1} \tag{3.9}$$

Which in the case of this filter function yields a digital filter equivalent to $H(s)$ in the form above is a filter of this form:

$$H(z) = \frac{TR_0}{1 - e^{p_0 T} z^{-1}} + \frac{TR_1}{1 - e^{p_1 T} z^{-1}} \tag{3.10}$$

Where T is the sampling rate. Plugging in the values for R_0 and the other filter values yields the form:

$$H(z) = \frac{\omega_n T_s}{Q} \frac{1 - e^{-\frac{\omega_n T_s}{2Q}} cos(\omega_n T_s) z^{-1}}{1 - 2cos(\omega_n T_s) e^{-\frac{\omega_n T_s}{Q}} z^{-1} + e^{-\frac{\omega_n T_s}{Q}} z^{-2}} \tag{3.11}$$

This form can be used to compute coefficients for an IIR filter. For our equivalent sampling rate of T=1ns and center frequency of 10MHz and a Q of 500, equation 3.10 evaluates to:

$$H(z) = A_{IN} \frac{1 - 0.9979 z^{-1}}{1 - 1.9958 z^{-1} + 0.99987 z^{-2}} \tag{3.12}$$

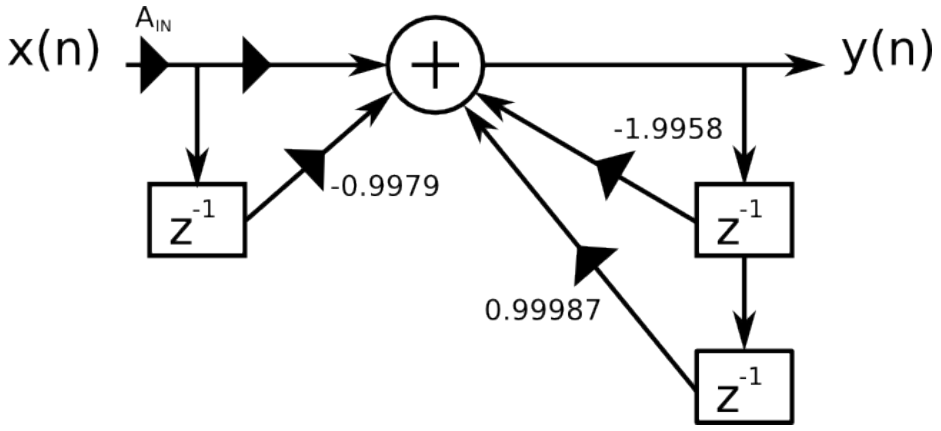

Figure 3.6: The resulting filter design with Q=500, T=1ns, $\omega_0 = 10 * 2\pi$ Mrad/s.

The input gain has been separated out to make it easy to perform calibration. In a hardware implementation, these coefficients could be calculated ahead of time for a single frequency, or put into a lookup table for specific frequencies of interest. Note that for each frequency of interest, only three numbers need be calculated or stored.

Because the filter is transformed using T, the time spacing of samples, each filter will be specific to a given frequency of operation. The total number of calculations required is four multiplications (one each for the three delayed values of the output and one for the input) and three additions. The storage requirement of this technique gets no larger as the number of samples increases: it is always three numbers. The computational requirement of this technique scales linearly with the number of samples rather than super-linearly as is the case with the FFT. Fixed-point calculations may be used (See Appendix A for details) in place of floating point calculations to further speed up this process.

Because of this, calculations for this technique may be carried out online, and memory requirements are small, giving this technique excellent prospects for moving it on board or even on chip.

Table 1: Memory Usage (Number of Registers)

	100 Samples	1k Samples	1M Samples
FFT [22]	128	1,024	1,048,576
IIR	3	3	3

As shown in figure 3.6, this technique requires four multiplications and three additions per cycle. The FFT's total computation requirements grow as O(N log N) with the number of samples, where as this techniques requirements grow as O(N). More importantly, since the number of calculations per cycle is fixed regardless of how long the measurement is taken, it is possible to implement this technique in fixed hardware which does not need to be able to allocate more resources as the measurement time grows.

Table 2: Calculation Requirements (Real Multiplications)

	100 Samples	1k Samples	1M Samples
FFT [22]	1792	20,480	41,943,040
IIR	400	4000	4,000,000

This method may save a few thousand calculations in small numbers of samples, but given the superior utility of the FFT this may be worth it. In cases where a clear picture of phase noise requires very long observations, such as near-carrier phase noise, this method presents a viable alternative to calculating very long FFTs.

3.6 Peak Detection

Once the IIR filter has isolated the phase noise in a band of interest, it can be extracted by measuring the amplitude of the output signal of the filter. In this algorithm we use a peak detection of the filter's output, or

$$A_{pk} = max(y(n)) \tag{3.13}$$

Where $y(n)$ is the output of the IIR filter. This operation requires one digital comparison operation and the storage of one number, and like the IIR it may be implemented with very minimal circuitry and may function online. The Root Mean Squared of the signal can also be used, although it is more computationally intensive.

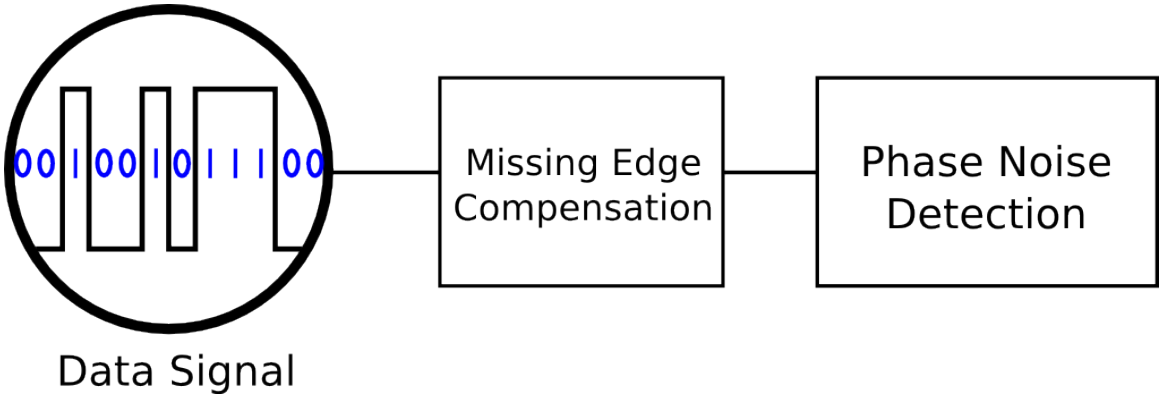

Figure 3.7: This technique compensates for missing edges in a data waveform.

3.7 Application to Data Signals

While it may often be possible to evaluate a system's sensitivity to phase noise by observing a clock, this requirement excludes analysis of bit sequences, which may be the most direct measurement of phase noise. We will show that this method can be extended to include measuring a data signal, and demonstrate that the sources of error introduced by this are manageable and can be characterized. This extension can measure sinusoidal phase noise (spurs) in a data signal by compensating for "gaps" left in a transition series by sets of consecutive high or low values.

The proportional relationship between the delay series and the time series in a sinusoid remains in the case of a slew-rate limited clock. The delay series $\Delta T(n)$ can be related to a voltage sampling of the waveform by:

$$\Delta V(n) = \Delta T(n) \cdot SR \tag{3.14}$$

The equivalence between time and voltage holds as long as the slope of the input waveform during the sampling is constant. The Fourier transform of this series is an evaluation of the frequency-domain of the signal. The relationship between $\Delta V(n)$ and $\Delta T(n)$ as in [19] can be used to construct the Fourier transform from only the time series $T(n)$:

$$F(\omega) = SR \sum_{n=0}^{N-1} \Delta T(n) e^{-j\frac{2\pi n\omega}{N}} \tag{3.15}$$

As with the sinusoidal signal, in the case of a square signal with some fixed slew rate (see figure 3.2), the transitions may be used as a voltage sampling at the frequency of the signal. The signal itself and its harmonics will alias to the DC bin of the FFT.

3.8 Data Signal and Compensating for Missing Edges

A data signal will have "missing" edges if monitored by a TDC which reports the time difference from a reference. In a real data signal this will drastically shorten the final vector of values. To successfully extract phase noise information from this signal, the changes to it must be considered carefully to predict the success of a measurement technique. This section will show that a useful subset of phase noise can be extracted even from this reduced data set.

In the particular edge detection scheme used in this work, the TDC is modeled such that each new edge time is reported including the time accumulated since the last data edge, resulting in many long delays reported between edges in the data signal. To unwind the added unit intervals and restore the data to a set of crossing points where noise is fixed around one unit interval, we employ the transformation:

$$\Delta T'(n) = \frac{\Delta T(n)}{floor(\frac{\Delta T(n)+UI/2}{UI})} \tag{3.16}$$

This transformation finds the number of samples that have been skipped using the floor function. To avoid misplacing samples whenever noise has pushed a value higher or lower than the expected unit interval delay, this function uses an offset before dividing by the number of intervals.

Although the technique interprets this new dataset as though it were a uniform sampling, which it is not, under the circumstances of low-offset phase noise measurement, we will show that this error may be safely treated as a manageable degrading of the detection threshold of the measured phase noise. There are two criteria that must be met for this to be true:

first, the projected uniform sampling must not introduce a large vertical (magnitude) error in the individual samples, and second, it must not change the frequency of measurement. Note that phase is not included in this evaluation, since the technique makes a magnitude evaluation only. In our analysis we will deal with these problems separately, solving the first problem by analyzing the magnitude error introduced by missing samples, and the second by computing the discrete frequency of the final observed spur.

To answer the question of magnitude, this technique operates under the condition that there are no large out of band spurs, and that the phase noise of interest is close to the carrier frequency. This constrains the problem to phase noise which will appear in the sampled edge series as having a low frequency. Each sample in the edge series which has undergone the transformation in equation 3.16 may also be represented as:

$$\Delta T'(n) = \frac{1}{s} \sum_{m=n-s}^{n} \Delta T(m) \tag{3.17}$$

Where s is the number of periods skipped before an edge, and $\Delta T'(n)$ is the set of time durations summed together by the increased delay to a sampled edge. This has the effect of averaging the skipped samples. This averaging will act upon the signal as a low pass filter, having a more significant impact on signal components which change rapidly, but leaving slow-moving signals relatively unaffected. (See figure 3.8.) If the phase noise to be observed is of a low offset, it will pass through the transformation with very little error, resulting in a final output close to the result that would be found in the event of a uniform resampling.

The problem of frequency is answered by calculating the destination of the spur in the discrete frequency domain. In the discrete time domain, the sampling rate of a waveform is not as important as the relationship a signal inside it has to the discrete frequency variable, z. From this perspective, all that matters is the total time of the measurement, and the original frequency of the phase noise under test. We can show this by comparing discrete frequency bins: the discrete frequency at which a spur will appear is dependent only on its original frequency and the total measurement duration:

$$z_{spur} = f_{spur} \cdot T_{total} \tag{3.18}$$

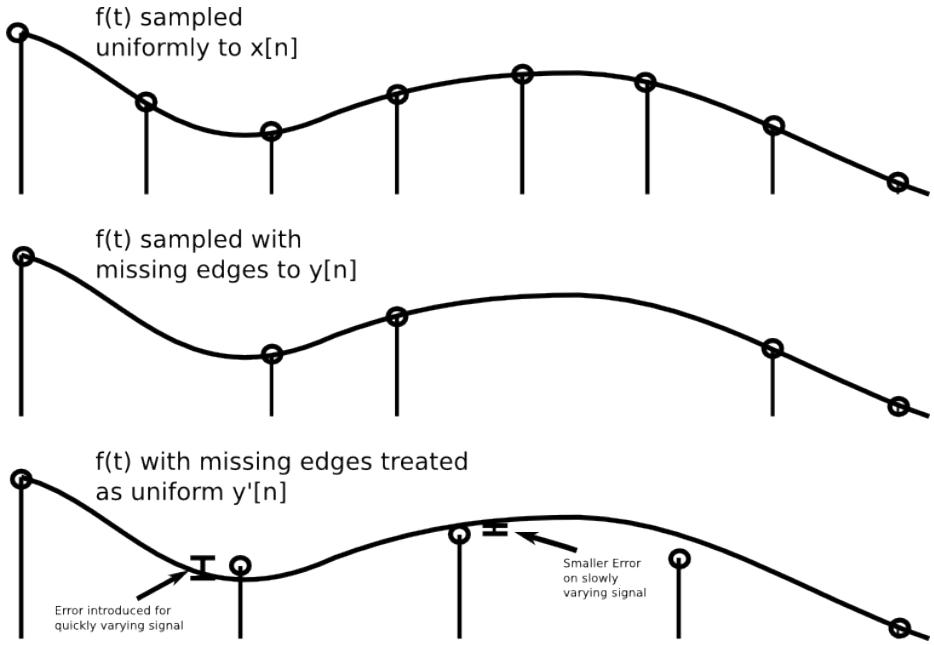

Figure 3.8: Treating a non-uniform sampling as uniform introduces error which is small for low-frequency information.

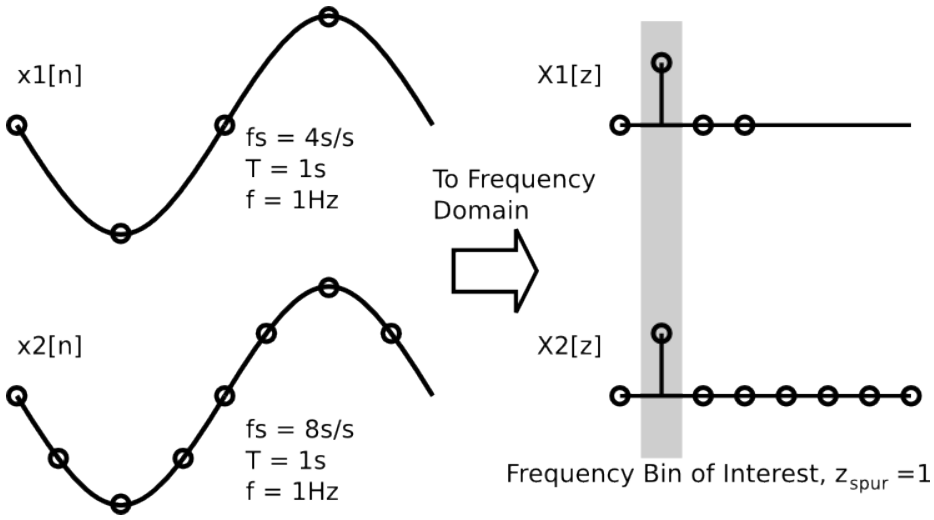

Figure 3.9: Regardless of sampling rate, sampling the same waveform for the same amount of time places it in the same frequency bins in the descrete frequency domain.

Where f_{spur} is the frequency offset of the phase noise spur and T_{total} is the complete measurement time. (See figure 3.9.) The important observation is that neither of these quantities are changed by reducing the number of data points. If a frequency to be observed remains inside the available frequency bins, reducing the number of samples within the same time interval has no effect on where that frequency will appear in the final frequency series $x[z]$.

The Filter coefficients in the prior technique [19] also conform to this observation: the pole frequencies of the filter are a function of the product of the original frequency of observation and the total measurement time, neither of which is dependent on the final sampling rate.

Empirically, we observe that a low offset spur can be readily detected in data with frequent edges, such as a random bit sequence with a mean of 0.5, but data sets with very infrequent edges begin to make detecting spurs more difficult.

3.9 Algorithm

The algorithm for extracting phase noise, then, is as follows:

Step 1: From the specification of phase noise for a specific communication or clock waveform, which includes the specific noise frequency and the required noise amplitude at that frequency, determine the duration of the measurement. If the signal is a data signal, find the average number of edges which will occur during the measurement.

Step 2: Design an IIR filter to detect the frequency of interest using the above design. If the signal is a data signal, scale the frequency to match the expected number of edges.

Step 3: Extract crossing points from the clock waveform and use these values as the input to the IIR.

Step 4: Measure the magnitude of the output of the IIR.

Step 5: Calculate the phase noise at the desired offset frequency.

All of the steps in our algorithm can be accomplished with digital hardware, and due to the use of IIR instead of FFT the bandwidth and computation requirements are greatly reduced. The IIR may run at the same time as the test, reducing or eliminating the need for post-processing time.

Chapter 4

SIMULATED RESULTS

Simulations conducted with MatLab show that an IIR filter can extract phase noise from a variety of signals, including sinusoidal signals and data signals. This chapter presents simulations of this technique operating on sinusoidal signals with white noise and injected phase noise, both under simple conditions and with nonidealities such as TDC quantization error and simulated quantization error in the IIR coefficients. Additionally, it is demonstrated that this technique may be applied to pseudorandom bit signals in the presence of white noise and data-dependent jitter.

4.1 Simple Case

Initially, the algorithm described above was tested under simple conditions where all computations were done at the available resolution of MatLab. This simplifying practice made it possible to present a basic, clear case for the effectiveness of this technique.

Our initial tests used a simple model of the TDC which suffers extremely low quantization noise due to its floating-point resolution. There are sources of error in this version of the TDC, since it is based on interpolation to detect crossing points, but this case requires no consideration for the quantization behavior of a TDC. The IIR also is modeled at the resolution available to MatLab, allowing us to ignore quantization error and the potential for instability in a very high-Q filter.

Using the same sampling rate (20GHz), carrier frequency (1GHz), low-frequency injected noise (250kHz) and measurement duration (500μs) as will be used in the final data, two simulations are run, one with injected noise, one where it is absent. The carrier amplitude is 1V peak and the sinusoidal noise injected is 50mV peak. In figure 4.1, the measurement shows an absence of any low-frequency noise. In figure 4.2, both monitoring methods (whole-signal FFT and peak output of IIR filter) show the injected low-frequency noise clearly. At

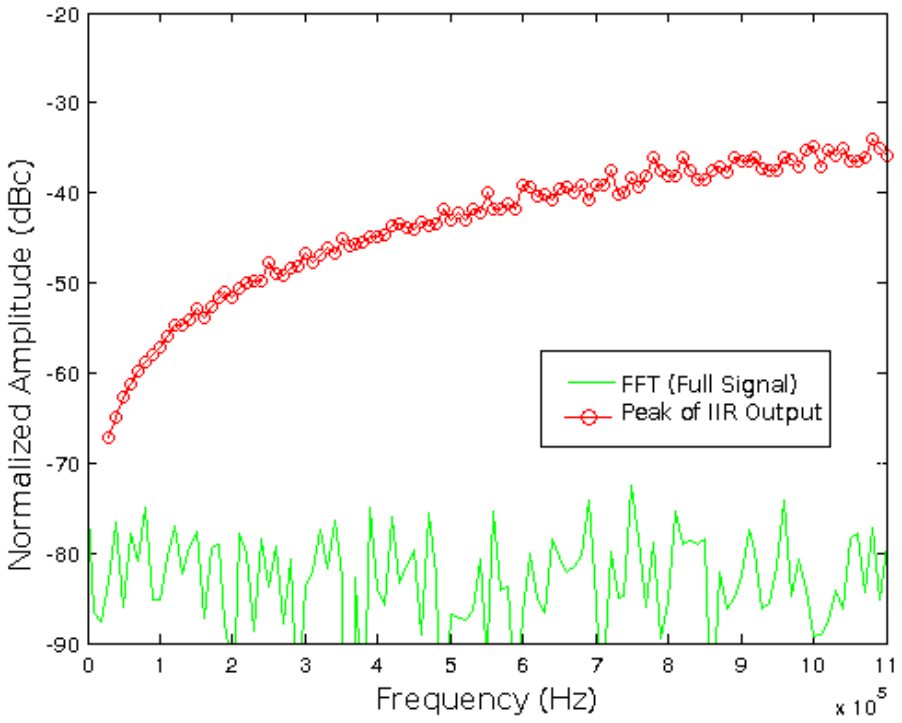

Figure 4.1: The filter output reflects the lack of injected noise.

250kHz, the peak output for the case where there is no low frequency noise is roughly -50dBc and -22dBc for the case where the noise is present.

The IIR's narrow bandwidth and high Q allow it to isolate the signal contained in the time series $t(n)$ from the surrounding frequency bins. By repeating this measurement with a new filter generated for each frequency, a frequency spectrum of the time series can be estimated. Although this is not necessary for practical implementations of this method, it is useful for comparisons against FFT of the original data. We have also simulated the case where this technique successfully captures phase noise near the fundamental of the measured signal as seen in figure 4.3.

In the figure, the undersampled phase noise is converted into low-frequency sinusoidal noise by the crossing point detection, which acts like an undersampled ADC, capturing a mixed down version of the carrier's spectrum. An IIR filter at a low frequency can extract a phase noise component at a small offset.

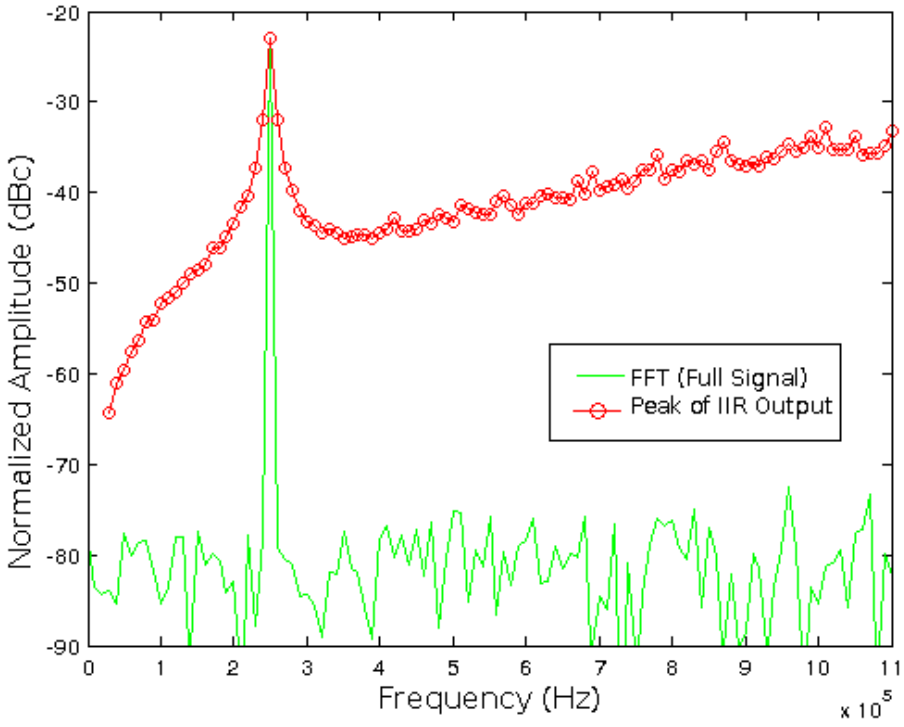

Figure 4.2: The filter output reflects the low frequency noise when present.

4.2 TDC Considerations

Like all data converters, TDCs exibit quantization noise. In practical applications, a high resolution may not be attainable. In MatLab we created a TDC with finite quantization levels and used it in place of our previous ideal TDC model. This model operates by taking each crossing point and subjecting it to quantization consistent with a chosen resolution.

While it is well established that quantization noise is not always spectrally "white," [21] it does have the property of being spectrally distributed over the available samples. Thus, while noise power is directly proportional to the size of the TDC's smallest measurable interval, it will also be inversely proportional to the total number of samples. This relationship means we can trade off between quantization interval and evaluation time.

To conduct an investigation of this relationship, we chose a metric for SNR based on two different frequency measurements. The first is the scale of the evaluated signal with

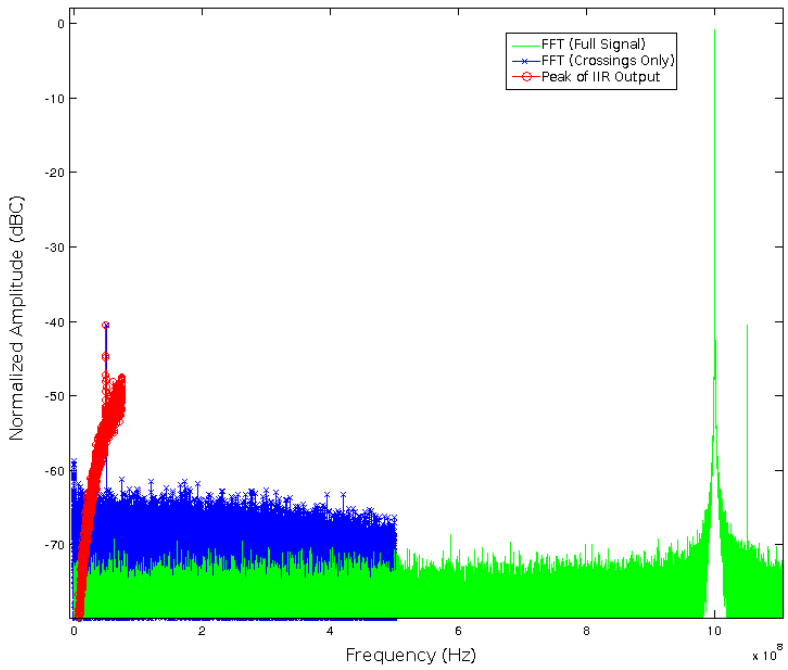

Figure 4.3: The final spectrum's phase noise aliases down to the measurable base band in this simulation.

the IIR tuned to the frequency of the monitored phase noise component, and the second is the scale of "nearby" spectral noise, found by tuning the IIR to 1.5 times the frequency of the monitored phase noise component. This frequency was chosen to place any harmonic distortion products outside the band of the filter. The ratio between these is used as a simple but illustrative measurement of the quality of the technique as it is impacted by quantization in the TDC.

Plotting the SNR against number of bits yields figure 4.4, which also shows different lengths of evaluation time. It is important to note that long evaluation time does not necessarily translate into large record length: if implemented with a hardware TDC, longer evaluation times require only more actual time, not more record length.

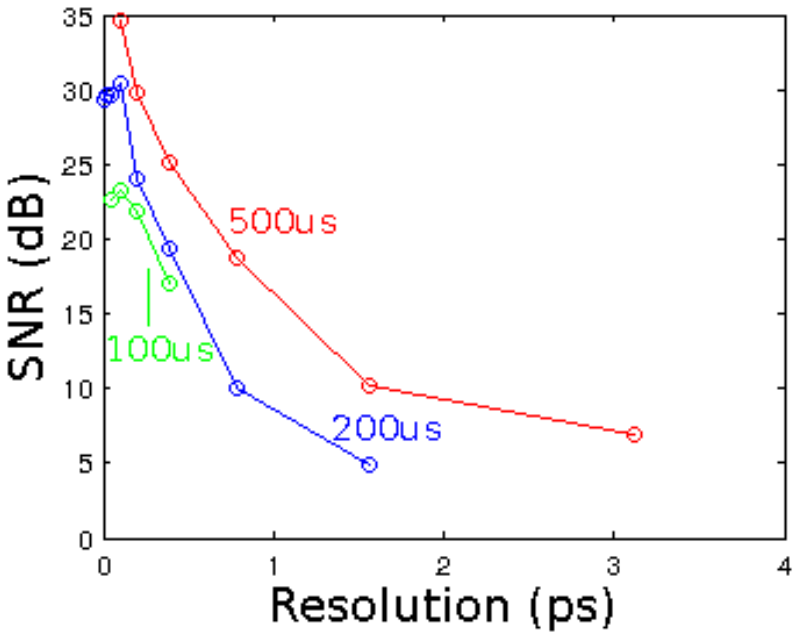

Figure 4.4: Impacts of TDC quantization noise at measurement durations of 100, 200, and 500μs.

4.3 IIR Considerations

When executed in MatLab, floating point computations are performed at each tap of the filter using tap values which are also floating point. This results in high precision, but will not be available to a simple hardware implementation, especially if that hardware is expected to perform synchronously to a high-speed clock. A more accurate way to represent the activity in a hardware IIR is to account for quantization effects.

First, there is the effect of quantization at each calculation. This results in quantization noise similar to that caused by the TDC. As with the TDC, these effects can be traded off against measurement duration to improve the quality of measurement.

Next, there is the hazard of instability that is introduced by quantization of the tap values of the IIR. Because high Q filters are used in this technique, this hazard is particularly serious. Quantization of the position of a pole can move the value of the "true" pole from a stable configuration to an unstable configuration if the error is large enough. By definition,

a high Q filter will have poles very near the edge of the unit circle, so a relatively small error can result in instability.

We experimentally selected differing quantization levels as demonstrated in [21], finding that, for Q=500, our 2-pole filters became essentially immune to instability above 12 bits of resolution. Since the digital filter is a compact structure which need only store a few values, this is not a difficult number of bits to achieve in a compact area.

4.4 Simulated Case Study With Data Signal

We next applied the same technique to a data signal. Using the method for compensating for missing edges outlined in chapter 3, we monitored phase noise in a simulated Pseudo-Random Bit Sequence (PRBS) to demonstrate this technique's ability to function on non-sinusoidal data.

To demonstrate the technique, a sampling rate of 40Gs/s is chosen to simulate capture with an oscilloscope and a data rate of 5Gb/s (for an equivalent center frequency of 2.5GHz is constructed with 1V peak and phase noise of 1mV peak at an offset of 5MHz or 2.505GHz. For this demonstration 25kbits of a PRBS for a total record length of 5μs are created.

To validate that this technique can function under real world conditions, white noise and Data-Dependent Jitter (DDJ) representative of a bandwidth limitation caused by a pole at 2GHz are added to the sequence. Positive edges in this data stream are logged and then combined into a single series of transitions. Transitions whose delays are above a threshold have an integer number of intervals subtracted from them to correctly position them within the data series as though the data were a periodic sampling.

Once the transitions have been logged, our technique for compensating for missing edges is applied and combined with [19] to extract the phase noise from this sequence. Overlaying the extracted frequency domain plot with the FFT of the entire 40Gs/s signal (figure 4.5), the undersampled phase noise is clearly visible as a distinct signal in the captured data. In figure 4.6, the frequency domain plot verifies that a false positive is unlikely, since an observation taken when there is no phase noise will result in a nearly 20dBc difference in the scale of the numerical result.

For a single-frequency test at this offset, the technique's threshold of detection can be

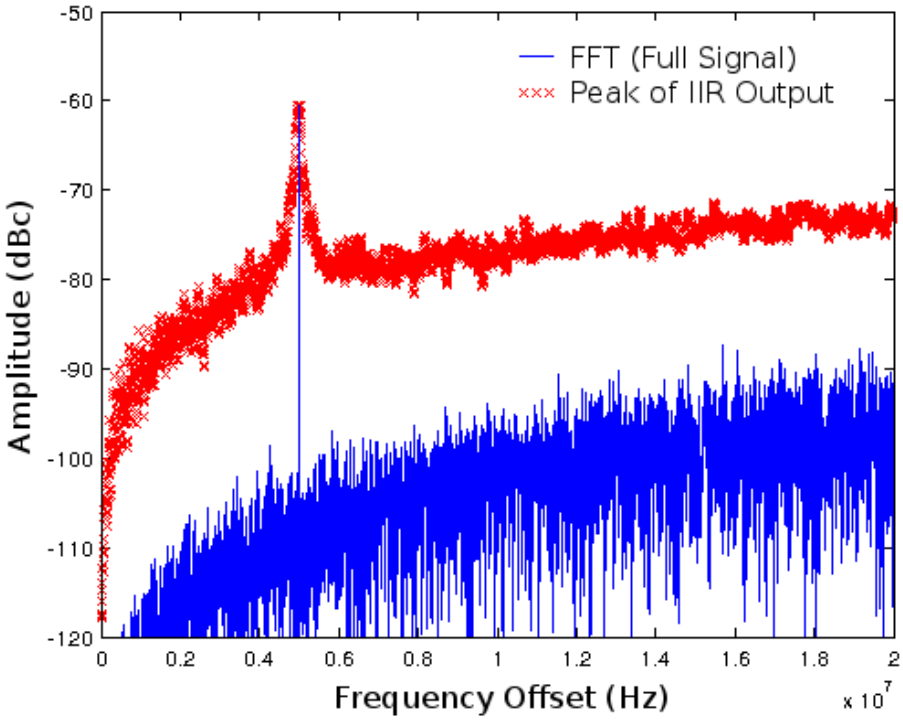

Figure 4.5: Phase noise at an offest of 5MHz from the clock of the data is plainly visible in the FFT of the data, and in the detected amplitude.

estimated to be -70dBc. Spurs smaller than this are on a similar scale to the broad band noise of the technique and should not be considered measurable with this technique under these circumstances.

4.5 Impact of Real Data Conditions

In the above case, a PRBS was used with an average of exactly 0.5. This may not be representative of all data on all protocols, as some data may have fewer edges than others. To simulate protocols and conditions with comparatively sparse edges, we operated the technique with a variety of PRBS with varying means. As can be predicted from the discussion in chapter 3 concerning noise and missed transitions, sparse edges degrade effective detection threshold. Because the technique can only sample when an edge occurs, PRBS sequences with means very different from 0.5 will receive fewer measurements and

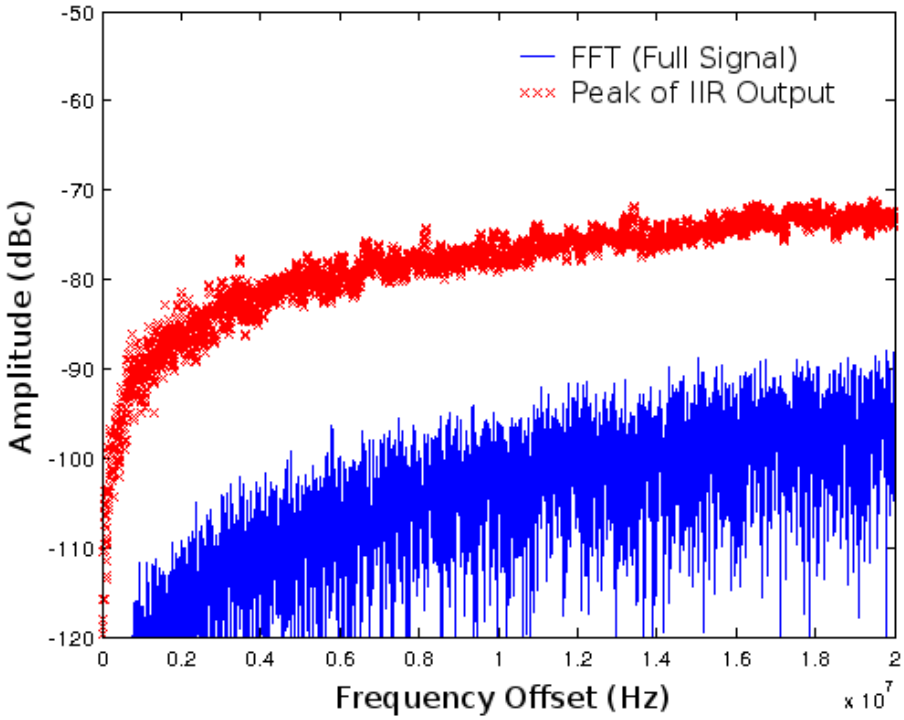

Figure 4.6: With the phase noise component removed, the tests associated with the 5MHz offset report values on average 20dB lower.

thus result in a higher detection threshold. In figure 4.7, we display the results of applying the technique at varying average data values. Note that near a mean of 0.5, the technique performs well, but as edges become more infrequent, the quality of the data degrades.

Since a major potential contributor to error in this technique is DDJ, particular attention in simulation has been paid to verifying that the technique can adequately characterize phase noise in the presence of large amounts of DDJ, as might be found in a very dispersive channel. As in [1] (pages 76-78), we model DDJ as the impact of an RC response. We conduct the simulated test at a variety of effective channel bandwidths ($f_c = 1/RC$) to determine the cutoff at which the technique begins to fail.

As shown in figure 4.8, the technique performs reasonably well while the cutoff frequency f_c is at or above the data rate, but begins to experience significant problems with noise as the bandwidth decreases further. While this technique can detect phase noise in a healthy

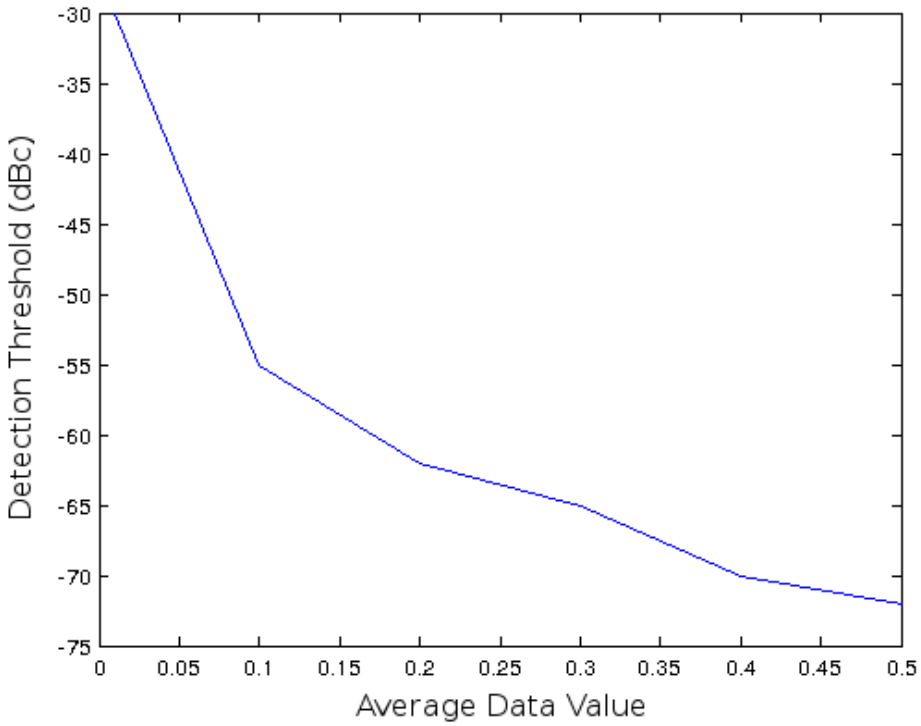

Figure 4.7: The minimum scale in dBc of a detectable spur is lowest where the average value of bits in the PRBS is 0.5.

channel, a pathologically dispersive connection will reduce the detection threshold.

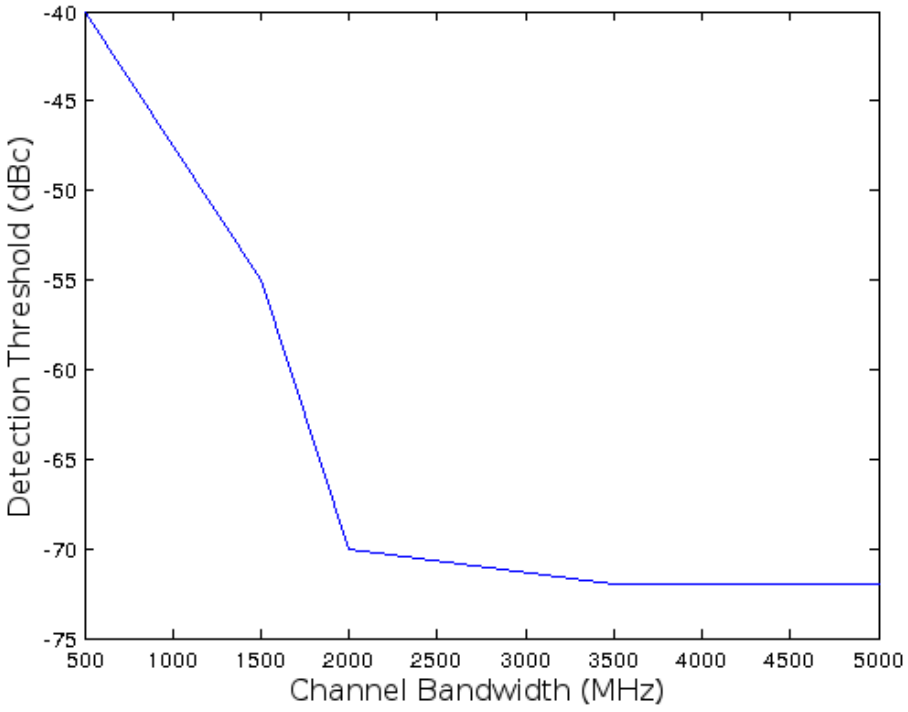

Figure 4.8: As channel bandwidth increases, the minimum scale in dBc of a detectable spur decreases.

Chapter 5

EXPERIMENTAL RESULTS

As with the simulated results, we will first demonstrate the function of the technique where the TDC and IIR are not impeded by finite quantization. For this simple case study we will compute crossing points with floating-point precision and perform the IIR calculations the same way. We will look at two physical test cases, first with a low-frequency injected tone, and second with a tone injected at an offset of 1MHz. These systems prove the technique's theoretical underpinnings. With this demonstration completed, we will revisit the data from the perspective of all components being vulnerable to quantization, with design choices equivalent to those that would need to be made to construct a hardware implementation of the algorithm.

5.1 Simple Case Study: 250kHz Tone

To validate the theoretical work and simulation, laboratory test bench equipment was used to construct a real-world signal with a large, single-tone phase noise component at 250kHz. To accomplish this, two sinusoidal signals, one at a frequency of 1GHz and 1V peak generated by an HP 8663A Synthesized Signal Generator and one at 250kHz and 50mV peak from an Agilent 33250A Arbitrary Waveform Generator, were merged and measured with a LeCroy DSO operating at a sampling rate of 20GHz for 500μs as shown in figure 5.1. The crossing points were extracted as described in chapter 3 and with our technique the spectrum was extracted with the same normalization scale as the simulated data.

Among the non-ideal characteristics of this signal are: white noise, a number of nearby sources of phase noise in other frequency bands, and the oscillator's intrinsic phase noise. All of these are limited in their impact on the final phase noise extraction by the IIR, which band-limits the measurement to the frequency of interest.

The plot in figure 5.2 shows that in the presence of the low frequency sinusoidal noise

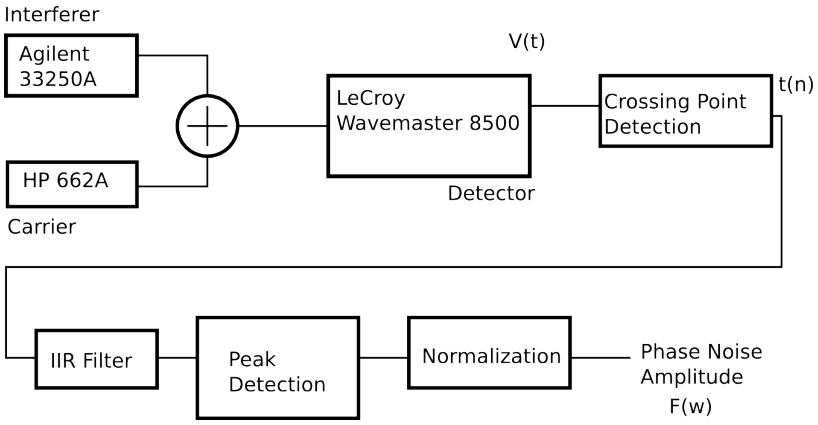

Figure 5.1: Two sinusoidal signals are combined at a summing node in this test case.

in the original signal, the algorithm extracts a clear difference between the injected low frequency noise at 250kHz and the nearby background, dropping more than 10dB within 50kHz bandwidth. This low frequency noise obscures any mixed down phase noise, however.

5.2 Simple Case Study: 1MHz Offset Tone

To measure low-offset phase noise, an Ettus Research USRP N210 signal generator is monitored with a 20GS/s DSO as shown in figure 5.3. Figure 5.4 shows the measured phase noise with conventional FFT and with our technique. The center frequency is 1GHz and the offset of the phase noise is 1MHz, or an absolute frequency of 1.001GHz. As with the initial test at 1GHz, a clear separation is made between nearby background and the measured phase noise. The frequency domain plot is shown as calibrated against the FFT data. Repeated measurements yield an average difference from this calibrated value of 1.003%.

The finite bandwidth of the filter integrates noise from adjacent FFT bins, so this technique has a higher noise floor than the original TDC output by around 10dB near the frequency of interest. Other sources of phase noise include unblocked radio transmissions and the intrinsic phase noise of the signal generators used. This technique's value lies in the dramatic potential reduction in overall digital complexity required to make sophisticated phase noise measurements. There is a trade-off between precision and test cost.

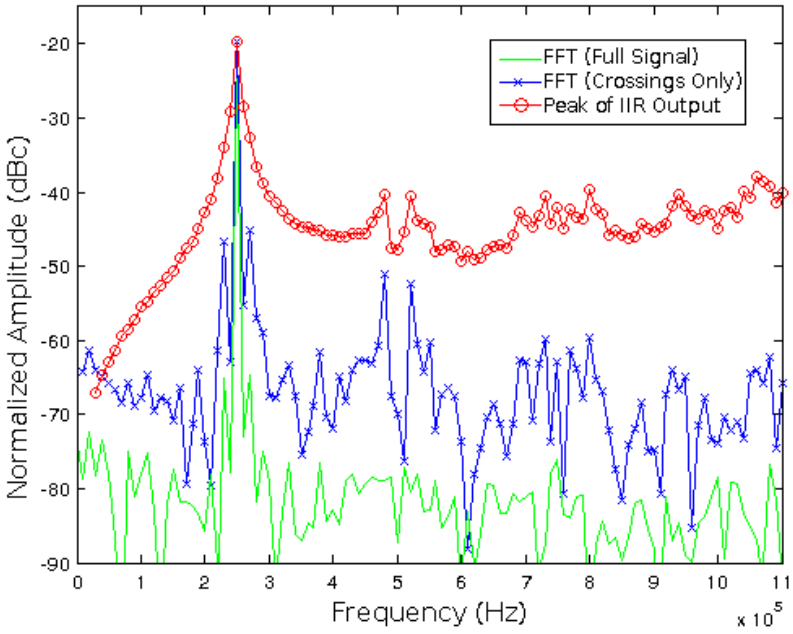

Figure 5.2: The injected noise at 250kHz is visible in the final IIR output.

5.3 Case Study with Quantized TDC and Quantized IIR

For our final test case, we use the existing data from the 1MHz offset case with forced quantization noise at the relevant points in the system. The TDC is modeled as having finite quantization steps, while the IIR is modeled both as having discrete available values for its coefficients and as quantizing the data passing through it. In this way, we simulate possible sources of injected quantization error.

Because the data captured with the oscilloscope is of limited length (not a problem for a hardware implementation with a TDC), the tradeoff between measurement time and TDC precision is fixed at only 500μs, so the modeled TDC is chosen to have a resolution of 3ps. This may not be available on some hardware, but the tradeoff in measurement time can be made more favorable in a hardware system than in this case.

The IIR filter is chosen to have a quantization of 16 bits, as many adders, multipliers and other drop-in core elements will likely operate at this precision. The magnitude is measured by a digital peak detect, which will be a trivial circuit to implement compared with the IIR.

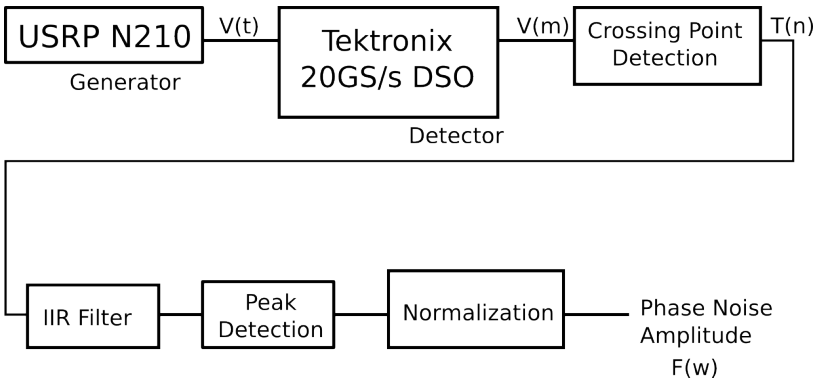

Figure 5.3: Two sinusoidal signals are merged in software to create a signal with injected phase noise, output by the Ettus Research USRP.

The impact of the additional error in the case of a Quantized TDC and IIR is visible figure 5.5, where the overall out-of-band noise increases with the introduction of quantization error in the TDC and IIR. In the case of the 3ps resolution TDC, the phase noise is still detected. In a practical case, it may be desirable to make use of the tradeoff illustrated in figure 4.4 to improve SNR.

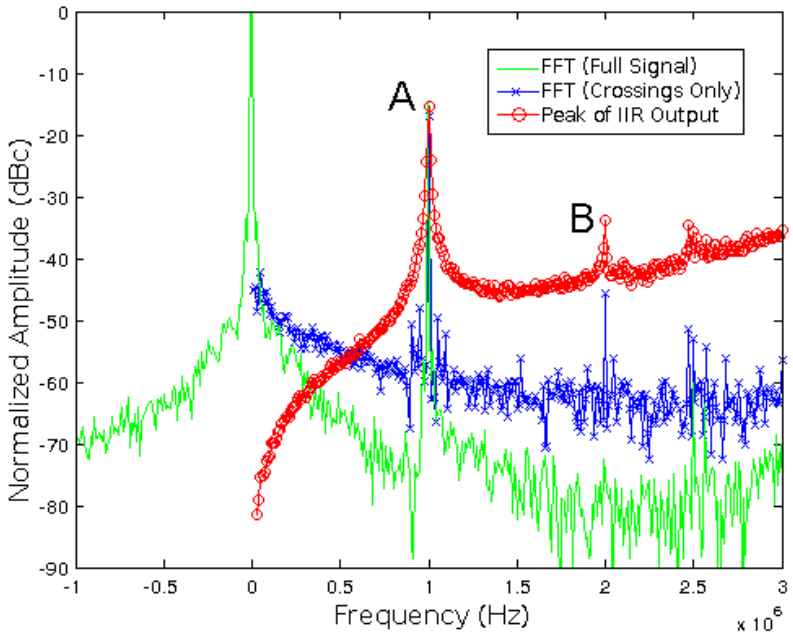

Figure 5.4: Injected phase noise (A) on a 1GHz signal detected at an offset of 1MHz. At (B) there is a harmonic of 1MHz introduced by crossing point detection.

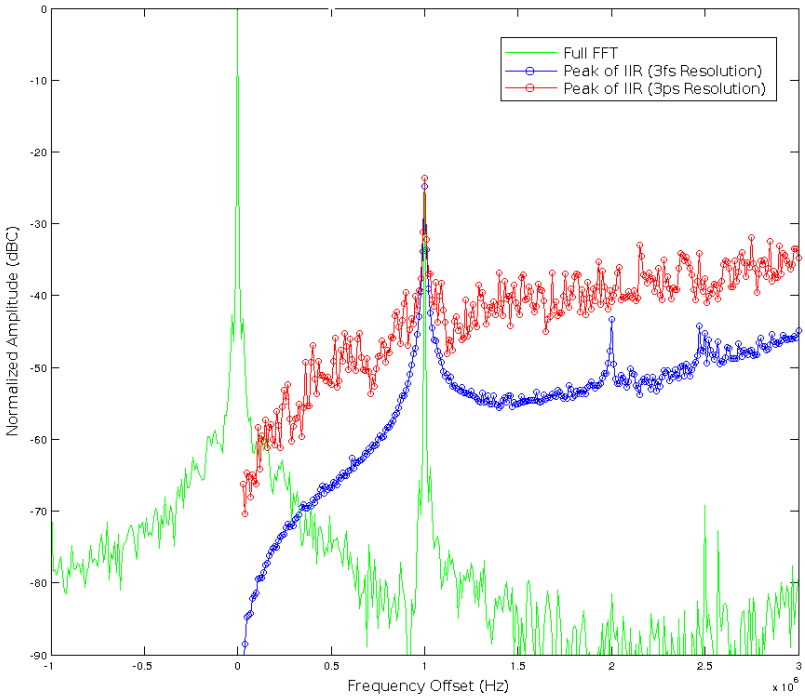

Figure 5.5: The impact of quantization noise on SNR.

Chapter 6

DISCUSSION

6.1 Error Analysis

There is a frequency dependence in the magnitude at which a phase noise component is detected. This dependence varies both in the frequency of the measured phase noise and in the relationship of the carrier to the sampling rate of the original waveform. To the extent that the final evaluation of a phase noise component remains above the noise floor, this can be canceled out by frequency-selective amplification of the translated signal.

Because this technique relies on monitoring a signal for a long time (in the case of the 1MHz offset measurement, 500μs), noise from the entire duration of the observation will alias into the band of measurement. This will tend to limit both the precision of the magnitude measurement as noise makes further precision difficult to attain. In particular, low-frequency sources of noise such as 1/f noise will become increasingly significant at low offset frequencies. Additionally, all the common sources of near-in phase noise in a PLL will contribute to noise in the measurement, since phase lock to the reference is a requirement of this technique.

Since the sampling rate in this technique is the same as the frequency of the fundamental, desired low-offset phase noise will appear in the same band as any low-frequency noise components, as well as any noise sources near the harmonics of the fundamental. Any noise whose final frequency after aliasing is in the band of interest will contribute to the calculated value of that noise.

The frequency dependence observed in simulation may reach a point where a measured phase noise component is below the level of the surrounding instrument noise, resulting in data which cannot be distinguished from this instrument noise. Even with pre-distortion this will limit the precision of this technique where the dependence results in a very small output value.

As described, this technique does not yet accurately compute relative scales of phase noise components. Scales must be computed ahead of time for a particular frequency, based on simulation or previously measured data.

6.2 Comparison to Other Measurement Techniques

Because this is a calibrated technique, most sources of systemic error are eliminated. However, there are significant sources of noise, so these have been used to compute a statement of the noise level of this measurement for comparison to other techniques.

For the case as depicted in figure 5.4, the sampling rate is 20GHz and the carrier (and thus, the sampling rate of our technique) is 1GHz. Measurements with this technique are undersampled by a factor of $\frac{f_s}{f_c}$, where f_s is the sampling rate and f_c is the carrier frequency. For this reason any broad band noise sources (white noise, 1/f noise, quantization noise) will increase by some amount. Additionally, the finite bandwidth of the IIR will introduce noise by integrating nearby noise sources into each measurement.

Table 3: Comparison to Other Techniques

	Full FFT	IIR Out
SNR(dB)(3fs)	59	31
SNR(dB)(3ps)	59	13
Time	O(N log N)	O(N)
Post-Processing	Yes	no
Bandwidth	f_s	f_c

Table 4: Comparison with Data Signal Measurement

	Full FFT	Peak IIR Out
Detection Threshold (clk)	-120 dBc	-85 dBc
Detection Threshold (data)	-90 dBc	-65 dBc
Time	O(N log N)	O(N)
Post-Processing	Yes	No

6.3 Linearity of Phase Noise Measurements

Since our technique is based on the model of a sinusoidal signal's slope being constant at the crossing points, it is susceptible to an induced nonlinearity. For example, a sinusoidal phase noise component at a 250kHz offset may produce images at 500kHz, 750kHz, and so on due to the deviation of the sinusoid from the expected slope as crossing points appear further from the expected value of the slope. An example of this can be seen in figure 5.4. Some caution is recommended with very large sources of phase noise when executing this technique.

6.4 Normalization

The normalization requirements of this technique could be eliminated with further work on the filter design. Optimizing the filter for more tightly-controlled bandwidth at the frequency of interest could net a constant rather than varying noise contribution and provide knowledge of the necessary coefficients for each measurement per frequency. The current filter's bandwidth is defined as

$$BW = \frac{\omega_0}{Q}, \tag{6.1}$$

which creates a frequency dependence in the resolution capability of this technique. Normalization also will need to be carried out on the sampling technique itself, which introduces its own frequency dependence. Ideally, the design of the IIR filter will simultaneously address both of these dependencies by including models of the sources of frequency dependencies. Because the gain of a digital filter is programmable, having it adjust to a known frequency dependence can be done at little or no cost.

6.5 IIR Critical Path and Decimation

We may note that in figure 3.6, the following operation is shown as being carried out once per clock cycle:

$$H(z) = A_{IN}\frac{1 - 0.9979z^{-1}}{1 - 1.9958z^{-1} + 0.99987z^{-2}} \tag{6.2}$$

This calculation means that each clock cycle, the sum of four numbers, each of which is the product of a multiplication, must be calculated. While the gain term can be calculated ahead of time at the cost of one clock cycle of latency, the critical path is still through one multiplier and two summing operations, assuming each calculation is given its own multiplier.

While some optimization of this filter to reduce delay is certainly possible, it is also likely that the operations will not resolve within one clock cycle, and in a test application the clock should not be slowed down to allow test circuitry to operate. In circumstances where computation must take longer than one clock cycle, decimation is acceptable.

As discussed in chapter 3, a data signal is already missing many edges. These missing edges must be considered in the filter design, but their absence does not, in the case of a low-offset spur, prevent successful measurement. For the same reason, edges may be intentionally removed to allow the computation to be completed.

If we have a signal with interference and, instead of sampling at the clock rate, we sample at one half or one fourth, this will be enough time for the computation to resolve. To filter correctly, however, the filter must be constructed to expect one half or one quarter the sampling rate. The filter will now be undersampling by a greater ratio than before, aliasing the phase noise into a new position.

Fewer samples mean that it will take longer to achieve a good result in a way similar to the effect of lost data points in the data signal measurement variation on this technique.

6.6 RMS Amplitude Detection and Peak Amplitude Detection

In this work, RMS (Root Mean Square) amplitude detection was initially chosen as a method of evaluating the scale of the detected noise. This choice was made to ensure an accurate report of the energy in the signal, regardless of start-up behavior or changes to the character of the final output signal (strongly noise-dominated or mostly sinusoidal). Investigation of signal properties in simulation, however, revealed that such caution was not necessary, and peak detection replaced RMS as an evaluation method, which actually proved to be more robust.

First, the primary fear concerning peak detection, that start up behavior would create

many false positives in the case of peak detection, failed to materialize. If an IIR is observing a signal with a strong component inside its bandwidth, the transient behavior at the output will gradually accumulate energy from the continued repetition of the signal until it reaches steady state, drowning out any noise. In fact, peak detection actually proves to be more resistant to transient behavior, since it retains no memory of a startup condition.

Second, peak measurement produces results much more efficiently. Computing RMS requires an additional multiplication and sum for each clock cycle, and this calculation, like all others, must be guarded against overflow. Peak measurement can be accomplished with a single comparison operation per cycle.

6.7 Prior Knowledge Use in Filter Design for Data Signals

The design of filters used in evaluating phase noise in data signals makes use of prior knowledge of the total number of samples in a data signal. In a real data signal, an exact figure for this number of samples may not be available. A fundamental assumption of the impulse invariance transform is that the frequency in samples per cycle of the frequency of interest is known, thus giving a perfectly reliable value for the **average** sampling rate. In this section we will examine the impact of an imperfect evaluation of the average sampling rate.

First, we will examine the creation of a filter with a known error in its center frequency. The filter frequency in the z domain is linearly dependent on the sampling rate, so the error of the frequency of the filter, relative to its separation from the carrier, will be:

$$\Delta f_0 = \Delta f_s \tag{6.3}$$

where Δf_s is the error in the average sampling rate. This relationship follows from the definition of the impulse invariance transform of the filter. The scale is relative to carrier separation rather than the carrier frequency because the aliasing operation of the TDC occurs before the filter is ever involved. Thus a small error in the average sampling rate will be tolerable. The finite Q of the filter will limit the loss in gain caused by this error, to a point.

Next we must consider the size of the error. As with most other error sources in this measurement, it is improved by longer measurement time. If the bit stream's mean is 0.5, the number of transitions will approach 0.5 times the total number of bits transmitted over time. The average sampling rate approaches one half what we would have with a clock signal. For other means, the average sampling rate must be calculated ahead of time.

The probability that a given bit will follow an edge can be written as $P\left(d(n) =!d(n-1)\right)$ or:

$$P_{edge} = (p(d(n) = 0) \cap p(d(n-1) = 1)) \cup (p(d(n) = 1) \cap p(d(n-1) = 0)) \tag{6.4}$$

In other words, the probability of an edge is equal to the probability that the last value was a zero and the current value is a one, plus the probability that the last value was a one and the current value is a zero. If we designate the mean of the bitstream, which is also the probability of a given bit being a one, μ, we can write this expression as:

$$P_{edge} = \mu * (1-\mu) + (1-\mu) * \mu = 2(\mu - \mu^2) \tag{6.5}$$

As the number of total bits gets quite large, the number of samples will approach the number of bits times this number, improving the quality of the final evaluation. The impulse invariance transform must be modified slightly to accommodate the new effective sampling rate, of course. The impulse invariance transform can be adjusted thus:

$$H(z) = \frac{T}{P_{edge}} \sum_{k=1}^{N} \frac{R_k}{1 - e^{p_k T/P_{edge}} z^{-1}} \tag{6.6}$$

The new impulse invariance transform adjusts the center frequency of the filter to accommodate the new average sampling rate, although of course it does not prevent the frequency of observation from having any error given a random data stream. We can trace this change forward to the full filter equation thus:

$$H(z) = \frac{\omega_0 T}{QP_{edge}} \frac{1 - e^{-\frac{\omega_0 T}{2QP_{edge}}} cos(\omega_0 T/P_{edge}) z^{-1}}{1 - 2cos(\omega_0 T/P_{edge}) e^{-\frac{\omega_0 T}{QP_{edge}}} z^{-1} + e^{-\frac{\omega_0 T}{QP_{edge}}} z^{-2}} \tag{6.7}$$

The resulting filter function will be appropriate to any long bitstream for which the mean is known. In the case of a bitstream with an unknown but stationary mean, a "monitoring phase" could be used to acquire a value for the mean by use of an accumulator set to evaluate the number of hits for the function $P\left(d(n) =!d(n-1)\right)$.

Chapter 7

CONCLUSION

The measurement of phase noise frequently requires the use of sophisticated analog equipment. In particular, phase noise near the carrier either requires the use of instrumentation with long record length to achieve sufficient frequency resolution, or narrow band discrimination via a mixer. While large PCBs which can support mixers may be available for some volume test, there are many circumstances where a test that can be performed using only digital circuits is desirable.

In this thesis, it is demonstrated that a mapping between simple time-domain measurements and a true voltage sampling can be used to analyze phase noise. In particular, the TDC element is shown to be suitable for these measurements despite its limitations. By taking advantage of the relative simplicity of digital filtering as compared with the full FFT, this technique makes detailed frequency-domain measurements possible with the relatively simple hardware of digital filters with feedback (IIRs). This hardware can operate at frequencies comparable to those of other calculations being done by a given hardware system, as the number of calculations necessary per clock cycle is small. No analog components are used, and the digital logic employed is highly compact. It is demonstrated that this method can be applied to digital signals as well as sinusoids or clock waveforms.

This thesis presents results which make use of real-world examples of phase noise to show that the technique works. These demonstrations test the ability of a simple IIR to evaluate phase noise, given a TDC with finite precision and under noisy conditions. The particular impact of finite precision of digital computation is examined to verify that its effects can be managed as sources of noise in the measurement. Another demonstration using digital signals shows that the same technique can be used with some consideration for the number of available samples. The impact of "missing" edges in these signals is examined in detail for pseudorandom digital signals.

In a world of ubiquitous radio devices, device and component manufacturers are actively searching for ways to create tests that verify system functionality in the presence of phase noise. The particular case of monitoring data and clocks for signal integrity is a primary concern.

A system including this design for built-in self-test would be able to achieve the desired measurements of recent industrial interest: small numbers of narrow bandwidth measurements of the magnitude of phase noise can be taken from existing test circuits such as the time to digital converter at a low cost to system scale. This meets the particular need of industry to achieve simple mask tests without bringing in large amounts of extra circuitry.

Additionally, spot test may have value in the field of channel hopping, particularly if the test can be compacted to very small scales or moved on chip. A system that could allow an online IC to examine nearby frequencies for space in the spectrum not currently occupied by other radio signals would be of particular value. If a solution can be made small enough and inexpensive enough, many radio systems on mobile devices could benefit. A highly-integrated, low power evaluation tool for physical detection of nearby signals could allow mobile radio devices to intelligently avoid crowded parts of the spectrum at run time, potentially improving the quality of data transmission in consumer electronics.

The contribution to knowledge of this thesis is the demonstration that low-cost digital filtering (IIRs) can be used in conjunction with available crossing point measurement techniques to evaluate phase noise on clock and data signals.

Appendix A

FIXED-POINT COMPUTATION CONSIDERATIONS

Performing digital filtering using fixed-point mathematics is possible by essentially choosing a fixed precision to represent fractions, by redefining the multiplication operation to mean "multiply and then shift" on two numbers. The area- and timing-efficient construction of an IIR requires this, trading some dynamic range for significant reductions in area, power, and delay. To describe the limitations of such a digital calculation framework for avoiding floating point calculations, we will find measurements for important signal-related quantities such as dynamic range and quantization noise in the context of the digital calculations inside an IIR.

First, a brief description of fixed-point calculations for fractional values. An IIR must be able to multiply two numbers with the result having the correct magnitude. To accomplish this, fractional values must be symbolically represented in the binary numbers used, and correctly propagated into the product. While this is often accomplished with floating-point calculations in non-realtime software such as MatLab, a high-speed, low transistor count implementation should use a fixed-point representation of fractional numbers.

The number of "fraction bits" in a fixed-point number is fixed, so multiplications between any two numbers in this representation may incur quantization error, even though all quantities are represented digitally. For example, the multiplication of 1.5 x 1.5, represented in binary fraction notation as 1.1 x 1.1, is 2.25 in decimal and 10.01 in binary. If only one fraction bit is being used, the last digit of that number will be lost, resulting in an answer of 10.0 or 2. Thus any discussion of fixed-point calculations should consider the impact of quantization error in these calculations. (It is also worth noting that a multiplier with fraction bits can be simplified, since all that is needed of the truncated bits is their final carry out.)

If a large number of fraction bits is used, the impact of these errors can be reduced,

lowering the "noise floor" of calculations to a manageable error. The greatest magnitude of a single error for a system with B fraction bits is 2^{-B}, which can be represented in decibels as $-6.02 \cdot B\, dB$. (For example, 10 fraction bits will yield maximum noise at close to -60dB, or $\frac{1}{1024}$.)

On the other end of the scale, a limited number of integer bits will reduce the total available "signal swing" for calculations. Any calculation resulting in a larger value will fail with an overflow error. In the above example with 10 fraction bits, if the fixed number of bits used is 16, this leaves 6 integer bits (5 if a sign bit is used) for describing the full scale of a waveform. Available peak scale and quantization error experience a direct tradeoff in the design of such a system. For IIR design, where the impulse response is typically bounded by +/- 1, the tradeoff can be made quite favorable to noise with little risk of encountering overflow errors, but care must be taken that the total energy added by the signal does not result in any output of the filter which exceeds the available range.

If an IIR making use of these fixed-point calculations encounters an overflow error, the results will be corrupted and unusable, so the maximum allowable signal must be carefully considered ahead of time.

BIBLIOGRAPHY

[1] M. Li, "Jitter, Noise, and signal Integrity at High-Speed," 2007, Prentice Hall

[2] L. Vercesi, A. Liscidini, R. Castello, "Two-Dimensions Vernier Time-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 45, pp. 1504-1512, August 2010

[3] K. Ichiyama, M. Ishida, T. Yamaguchi, M. Soma, "Noval CMOS Circuits to Measure Data-Dependent Jitter, Random Jitter, and Sinusiodal Jitter in Real Time," *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, May 2008

[4] S. Tabatabaei, F. Ben-Zeev, T. Farahmand, "Jitter Spectrum Analysis Using Continuous Time Interval Analyzer (CITA)," *IEEE 2005 International Test Conference*, pp. 207, November 2005

[5] M. Mendez-Rivera, J. Silva-Martinez, E. Sanchez-Sinencio, "On-Chip Spectrum Analyzer for Built-In Testinng Analog ICs" *International Symposium on Circuits and Systems, 2002, IEEE*, vol. 5, pp. 61-64, August 2002

[6] J. Hsu, C. Su, "Timing Jitter and Modulation Profile Extraction for Spread-Spectrum Clocks," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, pp. 847-856, April 2010

[7] W. Huang, H. Peng, "HHT Algorithm in Decomposition of High-Frequency Period Jitter," *The 2nd IEEE International Conference on Information Management and Engineering 2010*, pp. 226-230, April 2010

[8] K. Blakkan, M. Soma, "A Time Domain Method to Measure Oscillator Phase Noise," *2009 27th IEEE VLSI Test Symposium*, pp. 297-302, May 2009

[9] L. Cosart, L. Perigrino, A. Tambe, "Time Domain Analysis and its Practical Application to the Measurement of Phase Noise and Jitter," *IEEE Transactions on Instrumentation and Measurement*, vol. 46, pp. 1016-1019, August 1997

[10] Keller, Mark W. Kos, A. B. Silva, T. J. Rippard, W. H. Pufall, M. R., "Time Domain Measurement of Phase Noise in a Spin Torque Oscillator," *Applied Physics Letters*, vol. 94, pp. 193105 - 193105-3, May 2009

[11] M. Takamiya, H. Inohara, M. Mizuno, "On-Chip Jitter-Spectrum-Analyzer for High-Speed Digital Designs," *2004 IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, vol. 1, pp. 250-532, February 2004

[12] C.-Y. Kuo and J.-L. Huang, "A Period Tracking Based On-Chip Sinusoidal Jitter Extraction Technique," *2006 24th IEEE VLSI Test Symposium*, pp. 405, May 2006

[13] M. Lin, G. Tsai, T. Chang, S. Chu, "An Approach to FPGA-based Time-Frequency Spectrogram by Real-Time Sweep Spectral Extraction Algorithm," *IEEE Region 10 Tencon 2005*, pp. 1-4, November 2005

[14] M. Soderstrand, H. Loomis, R. Gnanasekaran, "Pipelining Techniques for IIR Digital Filters," *IEEE International Symposium on Circuits and Systems*, pp. 121 - 124, May 1990

[15] W. Ulbrich, T. Noll and B. Zehner, "MOS-VLSI Pipelined Digital Filters for Video Applications," *1984 IEEE International Conference on ICASSP*, vol. 9, pp. 386-389, March 1984

[16] D. Adamson, "A possible alternative method of making traceable phase noise measurements," in *IEE Colloq. Microw. Measur.*, pp. 7/1-7/6, February 1999

[17] J.M. Yang, D.C. Yang, P. G. Cheng, and J. M. Dickson, "Automated phase noise measurement of Ku-band MMIC VCO on-wafer," in *Proc. Int. Microw. Symp. Dig.*, pp. 1763-1766, June 1999

[18] L. Angrisani, M. DApuzzo, and M. DArco, "A digital signal-processing approach for phase noise measurement," *IEEE Trans. Instrum. Measur.,* vol. 50, pp. 930935, August 2001.

[19] A. Ecker, K. Blakkan, M. Soma, "A Digital Method for Phase Noise Measurement", *International Test Conference* 2012

[20] A. Ecker, M. Soma, "A Method for Phase Noise Extraction from Data Communication", *VLSI Test Conference* 2014

[21] V. Ingle, J. Proakis, "Digital Signal Processing using MATLAB," second ed, 2007, Cengage Learning

[22] C. Phillips, J. Parr, "Signals, Systems and Transforms" second ed, 1999, Prentice Hall

[23] A. Dandapat, S. Ghosal, P. Sarkar, D. Mukhopadhyay, "A 1.2-ns16x16-Bit Binary Multiplier Using High Speed Compressors" *International Journal of Electrical and Electronics Engineering* pp. 4:3, 2010